



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Διδακτορική Διατριβή

Μία Τεχνολογικά Υποστηριζόμενη Διδακτική Προσέγγιση για τον Αντικειμενοστρεφή Προγραμματισμό

ΣΤΑΥΡΟΥΛΑ Κ. ΓΕΩΡΓΑΝΤΑΚΗ

Τριμελής Συμβουλευτική Επιτροπή

Επιβλέπων καθηγητής

Συμεών Ρετάλης, Επίκουρος Καθηγητής Παν. Πειραιώς

Μέλη Τριμελούς Συμβουλευτικής Επιτροπής

Νικήτας-Μαρίνος Σγούρος, Καθηγητής Παν. Πειραιώς
Δημήτριος Σάμψων, Επίκουρος Καθηγητής Παν. Πειραιώς

Πειραιάς, Ιούνιος 2008

Περίληψη

Το ερευνητικό πρόβλημα το οποίο πραγματεύεται η παρούσα διδακτορική διατριβή, είναι η δημιουργία μιας ολοκληρωμένης διδακτικής προσέγγισης για τη σχεδίαση και ανάπτυξη εφαρμογών Αντικειμενοστρεφούς φιλοσοφίας Προγραμματισμού, με στόχο να βοηθήσει ουσιαστικά: (α) στην ανάπτυξη Αντικειμενοστρεφούς σκέψης επίλυσης προβλημάτων (development of object oriented (OO) problem solving skills), (β) στην εμπέδωση βασικών εννοιών-αρχών του παραδείγματος αυτού, (γ) στην ανάπτυξη ικανότητας σχεδίασης, υλοποίησης, ελέγχου, αποσφαλμάτωσης απλών προγραμμάτων σε Java με τη χρήση Computer Aided Software Engineering (CASE) εργαλείων και προγραμματιστικών περιβαλλόντων και (δ) στην αντιμετώπιση καταγεγραμμένων μαθησιακών προβλημάτων.

Η διδακτική προσέγγιση που προτείνεται στη διατριβή (α) αξιοποιεί στη φιλοσοφία της προηγούμενα ερευνητικά ευρήματα, (β) περιλαμβάνει υποστηρικτικό μαθησιακό υλικό με ποικιλία στη μορφή του το οποίο δίνει έμφαση στην αντιμετώπιση μαθησιακών προβλημάτων, (γ) ενσωματώνει συστάσεις διδασκαλίας (teaching recommendations) και καλές διδακτικές-παιδαγωγικές πρακτικές που έχουν διατυπωθεί με τη μορφή οδηγιών (guidelines) ή σχεδιαστικών προτύπων (design patterns) σε προηγούμενες ερευνητικές μελέτες προκειμένου να αντιμετωπιστούν ήδη καταγεγραμμένα μαθησιακά προβλήματα και (δ) αξιοποιεί στοχευμένα, εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα. Τα θέματα που η προσέγγιση περιλαμβάνει συνάδουν με τις οδηγίες του Οδηγού Σπουδών για θέματα Πληροφορικής των IEEE/ACM (Computing Curricula 2001) και η εφαρμογή της στη διδακτική πράξη κάνει χρήση των τεχνολογιών του Παγκόσμιου Ιστού και της συνδυασμένης μαθησιακής στρατηγικής (blended learning strategy). Επιπλέον διέπεται από συγκεκριμένη παιδαγωγική αρχή που είναι η εκμάθηση του γνωστικού αντικείμενου κοντά σε ειδικούς (apprenticeship) με πειραματισμό και με έμφαση στη μοντελοποίηση (modeling approach) και προτείνει ένα σύνολο μαθησιακών δραστηριοτήτων για τους εμπλεκόμενους στη μαθησιακή διαδικασία, που λαμβάνουν χώρα με συγκεκριμένη σειρά και τρόπο για την επίτευξη του στόχου της. Οι μαθησιακές αυτές δραστηριότητες μπορούν να χρησιμοποιηθούν ως οδηγός υλοποίησης της προτεινόμενης προσέγγισης σε πραγματικές εκπαιδευτικές συνθήκες.

Η προτεινόμενη διδακτική προσέγγιση εφαρμόστηκε επί δύο ακαδημαϊκά έτη σε τρεις μελέτες περίπτωσης και σε αυθεντικά εκπαιδευτικά περιβάλλοντα στην Τριτοβάθμια εκπαίδευση, τόσο σε προπτυχιακό όσο και σε μεταπτυχιακό επίπεδο. Για τις εφαρμογές αυτές πραγματοποιήθηκαν συστηματικές αξιολογήσεις, με τις οποίες διερευνήθηκε η μαθησιακή αποτελεσματικότητα της προσέγγισης και από τα συμπεράσματα που προέκυπταν σε κάθε εφαρμογή επέρχονταν βελτιώσεις με αποτέλεσμα η διδακτική προσέγγιση να αναπτυχθεί εξελικτικά και να πάρει την τελική μορφή της.

Σημαντικά πρωτότυπα στοιχεία της διατριβής αποτελούν:

- Η αξιοποίηση προηγούμενων ερευνητικών ευρημάτων και η ενσωμάτωση παρεμβάσεων τόσο στη δημιουργία πρωτόλειου μαθησιακού υλικού όσο και στη σχεδίαση-εφαρμογή της διδακτικής προσέγγισης στην κατεύθυνση να “θεραπευτούν” μαθησιακές δυσκολίες-παρανοήσεις που έχουν διαπιστωθεί.
- Η κάλυψη από το διδακτικό περιεχόμενο της προσέγγισης της θεματικής ανάπτυξης που προτείνεται από τις IEEE και ACM στο Computing Curricula 2001.
- Το πλήθος των εφαρμογών της προσέγγισης, η διάρκειά τους και το πλήθος των συμμετεχόντων σε αυτές.
- Η διαφορετικότητα μεταξύ των εφαρμογών της προσέγγισης όσον αφορά στα περιβάλλοντα και στις συνθήκες εφαρμογής.
- Η συστηματική μελέτη αξιολόγησης της κάθε εφαρμογής σε χρονοσειρές (επαναλαμβανόμενες στην κάθε εφαρμογή) και η βελτίωση της μεθόδου κατόπιν των επανειλημμένων αυτών εφαρμογών και αξιολογήσεων, για να πάρει την τελική προτεινόμενη μορφή της.

Επίσης οι εφαρμογές της προσέγγισης, απέφεραν εμπειρικά δεδομένα για τη διδασκαλία του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού, κάτι για το οποίο υπάρχει μικρή σχετικά καταγεγραμμένη εμπειρία –συγκρινόμενη με την αντίστοιχη για το διαδικασιακό (procedural) Προγραμματισμό– και κατά συνέπεια προήγαγαν το πεδίο έρευνας αυτού του τομέα και ανέδειξαν προβλήματα και δυσκολίες-παρανοήσεις των σπουδαστών για το γνωστικό αυτό αντικείμενο που δεν είχαν καταγραφεί.

Abstract

This thesis deals with the research problem of creating a complete teaching approach for designing and developing object oriented (OO) programming philosophy applications, with the aim to help essentially: (a) the development of OO problem solving skills by the learners, (b) the consolidation of main concepts-principles for this programming paradigm, (c) the development of skills for design, implementing (coding), checking, debugging of simple Java programs using Computer Aided Software Engineering tools and programming environments and (d) the treatment of recorded learning problems.

The teaching approach, which this thesis proposes, (a) exploits in its philosophy previous research findings in the area, (b) includes supporting teaching material with a variety of resources, which emphasises on the treatment of learning problems, (c) incorporates teaching recommendations and selected instructional practices recorded either in the form of guidelines or design patterns in previous research studies and (d) utilises educational tools and programming environments in the context of specific targets. The subjects included in that approach are in accordance with topics recommended by the IEEE/ACM Computing Curricula 2001, and the approach's implementation in educational settings uses WEB technologies and the blended learning strategy. Moreover, the approach adopts the pedagogical principle of learning by experts (apprenticeship) with experimentation while emphasising on modelling, and also proposes a set of learning activities that take place with specific order and method, in order to reach the respective targets. It is possible to use these learning activities, as a guide for implementing the approach in actual educational settings.

The proposed teaching approach was tested during two academic years, in three case studies and in authentic educational environments in Higher education for undergraduate and postgraduate students. During these studies, and with the aim to investigate the learning effectiveness of the approach, systematic evaluations have been conducted. The result of each evaluation study was used as source of improvement, so that the teaching approach was developed evolutionally until its final form.

Essential novel components of this thesis are:

- The exploitation of previous research findings and the incorporation of mediations towards the direction of treatment of recorded learning difficulties-misconceptions, both in learning material and in the design-application of the approach.
- The coverage of sections recommended by the IEEE/ACM Computing Curricula 2001, throughout the teaching content.
- The number of case studies that were implemented, their duration and the number of participated students.
- The diversity among the case studies, concerning environments and each application's conditions.
- The systematic evaluation study performed in each application and method's improvement upon completion of the repeated applications and evaluations, until its final form.

Additionally, the application of the approach in educational settings, has added significant empirical data about teaching OO, a subject with less recorded experience –compared to the experience accumulated about procedural programming– and thus, these applications advanced this subject's research area and revealed learning problems and students' difficulties-misconceptions for the subject, unrecorded till now.

Ευχαριστίες

Σε αυτή την ενότητα, καλούμαι να αποτυπώσω όχι μόνον αυτό που ο τίτλος της δηλώνει, αλλά συναισθήματα και σκέψεις στη διαδρομή μιας πορείας τεσσάρων περίπου χρόνων που περιελάμβανε πολλά γεγονότα και πολλή προσπάθεια. Να αποδώσω με λέξεις τις σχέσεις με ανθρώπους που με στήριζαν άοκνα και με άλλους με τους οποίους συνεργάστηκα λιγότερο στενά. Με λίγα λόγια νιώθω την ανάγκη να φανεί με αυτό το κείμενο, όλο το ταξίδι μου από την αρχή μέχρι το τέλος. Ελπίζω στο τέλος της πορείας αυτής, να μην παραλείψω σημαντικά και κυρίως σημαντικούς, που δε θα ήθελα να μην έχουν αναφερθεί.

Ευχαριστώ θερμά τη μητέρα μου, που η ύπαρξη της με στηρίζει και με ενισχύει πάντα.

Στη συνέχεια, πάνω από όλους εκφράζω τις θερμές μου από καρδιάς ευχαριστίες προς τον επιβλέποντα καθηγητή μου κ. Συμεών Ρετάλη που ήταν πάντα δίπλα μου και υπήρξε συνεχώς σύμβουλος, καθοδηγητής, αρωγός και υποστηρικτής μου καθ' όλη τη διάρκεια αυτών των ετών. Η υποστήριξη του και η καθοδήγησή του ήταν πολύτιμες για μένα και χωρίς αυτές δε θα προέκυπτε το τελικό αποτέλεσμα που παρουσιάζεται εδώ. Τον ευχαριστώ για τη θερμή συμπαράσταση του και την ενθάρρυνση που μου προσέφερε τις στιγμές ακριβώς που χρειαζόταν, με τον τρόπο που χρειαζόταν. Μετά από κάθε συνάντηση μαζί του (και ήταν συχνές και μεγάλης συνήθως διάρκειας) πάντα είχε ανοίξει κάποιος δρόμος, κάποια προοπτική να ξεπεραστούν εμπόδια και να δοθούν λύσεις σε προβλήματα. Τη συνεργασία μας και τη διαρκή βοήθεια που μου προσέφερε θα τη θυμάμαι πάντα.

Ευχαριστώ επίσης πολύ τα μέλη της τριμελούς συμβουλευτικής επιτροπής καθηγητές κκ. Νικήτα-Μαρίνο Σγούρο και Δημήτριο Σάμψων για τη βοήθεια και τη συμπαράστασή τους κατά τη διάρκεια εκπόνησης της διατριβής.

Ακόμα ευχαριστώ θερμά τον καθηγητή κ. Γιάννη Ψαρομήλιγκο για την πολύ σημαντική του βοήθεια, υποστήριξη και συμπαράσταση σε κάθε επίπεδο για ό,τι απαιτούσε η διεξαγωγή μιας εφαρμογής στην πράξη της διδακτικής προσέγγισης που προτείνει η διατριβή. Επίσης ευχαριστώ τον κ. Διονύση Αδαμόπουλο, διδάσκοντα του Τμήματος Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων, για τη συνεργασία μας και τη βοήθειά του στη διεξαγωγή μιας άλλης εφαρμογής της προσέγγισης.

Η συνδρομή των συνυποψήφιων συναδέλφων διδασκόντων, από τη συναναστροφή μαζί τους και την ανταλλαγή απόψεων πάνω στη δουλειά που ο καθένας μας εκπονούσε είναι επίσης πολύ σημαντική. Οι συχνές συναντήσεις που ο επιβλέπων όλων μας καθηγητής κ. Συμεών Ρετάλης είχε την πολύ καλή ιδέα να έχουμε, αποτελούσαν πηγή πολύτιμης ανατροφοδότησης για μένα. Θέλω στο σημείο αυτό να αναφερθώ ιδιαίτερα στην ουσιαστική βοήθεια που μου προσέφεραν για τη διατριβή μου, δύο πολύ αγαπητοί συνάδελφοι, ο Πέτρος Λάλος και ο Φώτης Λαζαρίνης.

Ευχαριστώ θερμά την εξαδέλφη μου Εβίτα, για όλες τις μακράς διάρκειας από τηλεφώνου συζητήσεις μας σχετικά με τη διατριβή και τη βοήθεια που αυτές μου προσέφεραν.

Στην πορεία εκπόνησης της διδακτορικής διατριβής είχα την τύχη να έχω δίπλα μου σημαντικούς φίλους που με βοήθησαν, στάθηκαν κοντά μου και με επηρέασαν καθ' οιονδήποτε τρόπο. Η υποστήριξή τους όλες τις στιγμές είναι ανεκτίμητη. Παρακολουθούσαν με πραγματική έγνοια, ήξεραν την πορεία μου, τις ευχάριστες στιγμές της και τις δυσκολίες της και μου συμπαραστέκονταν με σημαντικό και καθοριστικό για μένα τρόπο. Η αναφορά των ονομάτων τους σε αυτό το σημείο θα κούραζε και δε θα αναδείκνυε τη σημασία του καθενός χωριστά, που για μένα υπάρχει και θα παραμείνει για πάντα. Όλοι τους ξέρουν και καταλαβαίνουν ότι περιέχονται σε αυτές τις πολύ λίγες λέξεις...

Θερμότερες είναι οι ευχαριστίες μου προς όλους τους φοιτητές χάρη στους οποίους δοκιμάστηκε και αξιολογήθηκε η προτεινόμενη στη διατριβή διδακτική προσέγγιση. Η κριτική τους και τα σχόλιά τους, ήταν πολύτιμα για τη βελτίωση του αποτελέσματος της ερευνητικής μου προσπάθειας.

Τέλος, δεν πρέπει να ξεχάσω να εκφράσω τις ευχαριστίες μου για την εκπαιδευτική άδεια που η υπηρεσία μου μου χορήγησε, πράγμα που μου έδωσε τη δυνατότητα να αφοσιωθώ στην εκπόνηση της διατριβής μου και να έχω ουσιαστικά αποτελέσματα και πρόοδο στη δουλειά μου.

Τελειώνοντας αυτές τις λίγες σελίδες των ευχαριστιών, δεν θα ήθελα να έχω παραλείψει να αναφερθώ σε όποιον κατά τη διάρκεια της πορείας μου, συνέδραμε με οποιοδήποτε τρόπο στην προσπάθειά μου. Δεν ξέρω αν τα κατάφερα, γι' αυτό ζητώ ταπεινά συγγνώμη από όσους άθελά μου δε μνημονεύθηκαν με κανέναν τρόπο και υπόσχομαι ότι πάντα θα υπάρχουν στιγμές που θα τους θυμάμαι, παρότι δε συνέβη τη στιγμή που γράφω αυτές τις γραμμές. Είναι άχαρο και τελικά ανέφικτο να προσπαθείς

να χωρέσεις σε λέξεις που γράφεις, όσα έχουν καταγραφεί στην καρδιά σου και στην ψυχή σου. Τους ευχαριστώ θερμά όλους.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

РАНЕЕЗНАМО ПЕРПАА

Αντί Προλόγου

Η διδακτορική μου διατριβή, από τον τίτλο της ξεκάθαρα δηλώνει τη σχέση της με την εκπαίδευση, τη διδασκαλία και τη μαθησιακή διαδικασία γενικότερα.

Από την εμπειρία που έχω αποκομίσει από τη θέση μου ως δασκάλας στη Δευτεροβάθμια εκπαίδευση και την όλη σχέση μου με την εκπαιδευτική διαδικασία, θέλησα σε αυτό το σημείο να αποτυπώσω κάποιες σκέψεις μου.

Καταρχήν θεωρώ το αποτέλεσμα της έρευνας μου, ένα λιθαράκι σε αυτό που λέγεται διδασκαλία, μεταφορά γνώσης και πρόσκτηση γνώσης, με τη χαρά και την απόλαυση που αυτό συνεπάγεται για το διδάσκοντα και για τους διδασκόμενους, όταν επιτύχουν να έχουν εκείνη την “απογείωση” που σημαίνει η σχέση τους.

Πέρα από τα εργαλεία, το υλικό, τις υποδομές, ό,τι ο όρος Τεχνολογία σημαίνει, το σημαντικότερο που πάντα θα παραμένει στη διδασκαλία αποτελεί αυτή η σχέση και η αμοιβαία ικανοποίηση, όταν τα δύο μέρη (δάσκαλος και μαθητής) “συναντηθούν” αληθινά.

Ως δάσκαλοι ας μην ξεχνάμε ότι έχουμε να μάθουμε πολλά από τους μαθητές μας, αν “σκύψουμε” να τους ακούσουμε με προσοχή, με “ανοιχτά” αυτιά και “γλυκιά” διάθεση, σε συνθήκες που θα νιώθουν και οι ίδιοι άνετα και όχι ως “απειλούμενοι υποψήφιοι εξεταζόμενοι”. Δανείζομαι εδώ, αυτό που από έρευνα συναδέλφου μου στα Τεχνολογικά Επαγγελματικά Εκπαιδευτήρια (Τ.Ε.Ε.) –στο πλαίσιο της διδακτορικής της διατριβής– έχω διαβάσει και μοιραστεί μαζί της. Ένας ερωτώμενος στο πλαίσιο της έρευνάς της μαθητής Τ.Ε.Ε. είπε: “Καθηγητές, ακούτε ακόμα και τους πιο αδιάφορους μαθητές...Θα μάθετε πολλά! ”.

Το τοπίο στην εκπαίδευση, έχει σίγουρα αλλάξει με την είσοδο της Τεχνολογίας και την αξιοποίησή της σε αυτό τον τομέα. Η τάξη, το μάθημα, οι σημειώσεις και το βιβλίο έχουν διαφοροποιηθεί σε σχέση με τις προηγούμενες δεκαετίες. Η μορφή όλων αυτών –και άλλων που δεν υποπτευόμαστε– θα αλλάξουν σίγουρα ξανά και ξανά στο μέλλον, αν δεν εκλείψουν και δεν αντικατασταθούν εντελώς με κάτι που δεν μπορούμε ίσως να φανταστούμε αυτή τη στιγμή. Παρ’ όλα αυτά, θα ήθελα να επισημάνω εδώ πως ως εκπαιδευτικός γνωρίζω καλά πόσο απέχει η εκπαιδευτική πραγματικότητα από όσα τεκταίνονται στο χώρο της έρευνας. Αυτή συχνά απέχει μακράν των δυνατοτήτων που η ένταξη της Τεχνολογίας στην εκπαίδευση προσφέρει.

Και αυτό είναι ένα σοβαρό θέμα που θα πρέπει να απασχολήσει όσους σχεδιάζουν, αποφασίζουν και καθορίζουν την πραγματικότητα αυτή.

Όλοι όσοι έχουμε άμεση σχέση και εμπλοκή με την εκπαίδευση, ξέρουμε καλά πόσο καθοριστικός είναι ο ρόλος της συμβατικής (παραδοσιακής) “από-καθέδρας” διδασκαλίας και η προσωπική επαφή δάσκαλου-μαθητή, που περιλαμβάνει όχι μόνο μετάδοση γνώσεων αλλά παιδεία ψυχική και νοητική καθώς και μετάδοση αξιών και τρόπων συμπεριφοράς. Όσο ακόμα υφίσταται η τάξη, το αμφιθέατρο και η απ’ ευθείας επικοινωνία διδασκόντων και διδασκόμενων, όσοι εμπλεκόμαστε σε αυτές τις διαδικασίες επικοινωνίας ας έχουμε υπόψη τη σημασία της δημιουργίας ζεστού και ενθαρρυντικού κλίματος που θα παροτρύνει και θα παρακινεί όσους εκπαιδεύονται και ιδιαίτερα τους έχοντες “χαμηλό ενδιαφέρον” και ακόμη περισσότερο τους όχι και τόσο “καλούς”, όπως λέγονται, μαθητές-φοιτητές. Και ακόμα ας μας απασχολήσει, ποια είναι αυτά τα στοιχεία που επιβάλλουν το διαχωρισμό μεταξύ “καλών” και “κακών” μαθητών-φοιτητών μέχρι να συναντηθούμε μαζί τους στην εκπαιδευτική διαδικασία.

Σε αυτό το σημείο, ευχαριστώ όλους συνολικά τους μαθητές τους οποίους έχω συναντήσει, για όσα με έκαναν να σκεφτώ και να μάθω.

Ακόμα, ευχαριστώ θερμά όλους όσοι μπήκαν στον κόπο να διαβάσουν αυτές μου τις σκέψεις...

Ρούλα Γεωργαντάκη

Περιεχόμενα

Περίληψη	i
Abstract.....	iii
Ευχαριστίες	v
Αντί Προλόγου.....	ix
Περιεχόμενα.....	xi
Κατάλογος Πινάκων	xv
Κατάλογος Σχημάτων.....	xix
Συνοτομογραφίες.....	xxiii
ΚΕΦΑΛΑΙΟ 1 Εισαγωγή.....	1
1.1. Το γνωστικό αντικείμενο του Αντικειμενοστρεφούς Προγραμματισμού	1
1.2. Προσεγγίσεις στη διδασκαλία του ΑΠ και σημεία που επιδέχονται επέμβαση	5
1.3. Αντικείμενο της διατριβής.....	8
1.4. Οργάνωση της διατριβής	12
ΚΕΦΑΛΑΙΟ 2 Η Διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού	17
2.1. Εισαγωγή	17
2.2. Δυσκολίες και παρανοήσεις των εκπαιδευομένων	18
2.3. Μακροχρόνιες ερευνητικές μελέτες για τη διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού	29
2.4. Διδακτικές προσεγγίσεις.....	42
2.4.1. Κύριες διδακτικές προσεγγίσεις	42
2.4.1.1. Διδασκαλία ΑΠ με τη χρήση της γλώσσας προγραμματισμού Java και ενός επαγγελματικού ολοκληρωμένου περιβάλλοντος ανάπτυξης (IDE) – Κλασική διδακτική προσέγγιση.....	42
2.4.1.2. Η “imperatives first” ή “procedurals first” προσέγγιση	43
2.4.1.3. Η “objects first” προσέγγιση.....	44
2.4.2. Στρατηγικές της “objects first” προσέγγισης.....	45
2.4.2.1. Η “concepts first” στρατηγική	45
2.4.2.2. Η “design driven” στρατηγική.....	46
2.4.2.3. Η “model first” στρατηγική	48
2.5. Εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα για τον Αντικειμενοστρεφή Προγραμματισμό.....	51

2.5.1. Προγραμματιστικοί μικρόκοσμοι	52
2.5.2. Εκπαιδευτικά προγραμματιστικά περιβάλλοντα	57
2.5.3. Εργαλεία αναβάθμισης των δυνατοτήτων προγραμματιστικών περιβαλλόντων	70
2.5.4. Εργαλεία λογισμικού για την υποστήριξη της “objects first” προσέγγισης	71
2.5.5. Περιβάλλοντα παιχνιδιών	72
2.5.6. Εκπαιδευτικές γλώσσες προγραμματισμού	73
2.5.7. Διδακτικοί πόροι για την υποστήριξη της χρήσης της γλώσσας Java σε εισαγωγικά μαθήματα	74
2.6. Συστάσεις διδασκαλίας	82
2.7. Σύνοψη	88
ΚΕΦΑΛΑΙΟ 3 Η Προτεινόμενη Διδακτική Προσέγγιση για τον ΑΠ	91
3.1. Εισαγωγή	91
3.2. Μαθησιακές δυσκολίες στον ΑΠ (ή προβλήματα προς λύση)	92
3.3. Το διδακτικό περιεχόμενο (ή διδακτέα ύλη)	94
3.4. Η εκπαιδευτική φιλοσοφία της προτεινόμενης διδακτικής προσέγγισης	96
3.5. Ροή δραστηριοτήτων	103
3.6. Μαθησιακό περιβάλλον με αξιοποίηση των Διαδικτυακών τεχνολογιών	113
3.7. Μαθησιακό υλικό	116
3.7.1. Πόροι μαθησιακού υλικού	117
3.7.2. Σχεδιασμός μαθησιακού υλικού και μαθησιακές δυσκολίες	125
3.8. Μαθησιακά εργαλεία	129
3.9. Ρόλος εμπλεκόμενων στη μαθησιακή διαδικασία	132
3.10. Σύνοψη	135
ΚΕΦΑΛΑΙΟ 4 Εφαρμογή και Αξιολόγηση της Προτεινόμενης Διδακτικής Προσέγγισης – Μελέτες Περίπτωσης (Case Studies)	137
4.1. Εισαγωγή	137
4.2. Η εξελικτική ανάπτυξη της προσέγγισης μέσω επανειλημμένων εφαρμογών και αξιολογήσεων	138
4.3. Εννοιολογικό πλαίσιο αξιολόγησης	140
4.4. Μελέτη περίπτωσης 1 (case study-1)	143
4.4.1. Χώρος εφαρμογής	143
4.4.2. Συμμετέχοντες	143

4.4.3. Υλοποίηση της εφαρμογής	145
4.4.4. Αξιολόγηση της εφαρμογής.....	145
4.4.4.1. Επεξεργασία ερωτηματολογίων.....	145
4.4.4.2. Επεξεργασία εργασιών.....	159
4.4.5. Συμπεράσματα – Βελτιώσεις	164
4.5. Μελέτη περίπτωσης 2 (case study-2).....	166
4.5.1. Χώρος εφαρμογής.....	166
4.5.2. Συμμετέχοντες	167
4.5.3. Υλοποίηση της εφαρμογής	167
4.5.4. Αξιολόγηση της εφαρμογής.....	168
4.5.4.1. Επεξεργασία ερωτηματολογίων.....	169
4.5.4.2. Επεξεργασία εργασιών.....	193
4.5.4.3. Επεξεργασία αποτελεσμάτων εξετάσεων	197
4.5.4.4. Στατιστική επεξεργασία ερωτηματολογίων για τη μέτρηση της μαθησιακής αποτελεσματικότητας της προτεινόμενης διδακτικής προσέγγισης και τον προσδιορισμό των παραγόντων που συμβάλλουν σε αυτήν	199
4.5.5. Συμπεράσματα – Βελτιώσεις	218
4.6. Μελέτη περίπτωσης 3 (case study-3).....	219
4.6.1. Χώρος εφαρμογής.....	220
4.6.2. Συμμετέχοντες	220
4.6.3. Υλοποίηση της εφαρμογής	221
4.6.4. Αξιολόγηση της εφαρμογής.....	222
4.6.4.1. Επεξεργασία ερωτηματολογίων.....	222
4.6.4.2. Επεξεργασία εργασιών.....	240
4.6.4.3. Ανάπτυξη τελικού project για το μάθημα Web Development.....	241
4.6.5. Συμπεράσματα	241
4.7. Σύνοψη.....	242
ΚΕΦΑΛΑΙΟ 5 Συμπεράσματα	245
5.1. Επισκόπηση της ερευνητικής πορείας	245
5.2. Αποτελέσματα της ερευνητικής πορείας και καινοτομικά στοιχεία.....	246
5.3. Όροι και συνθήκες εφαρμογής της διδακτικής προσέγγισης όπως προκύπτουν από τις μελέτες περίπτωσης	251
5.4. Μελλοντικές ερευνητικές κατευθύνσεις.....	254
5.5. Επιλογικά σχόλια	255

Βιβλιογραφικές αναφορές	259
ΠΑΡΑΡΤΗΜΑ Α Θεματική Ανάπτυξη του Διδακτικού Περιεχομένου	273
ΠΑΡΑΡΤΗΜΑ Β Ερωτηματολόγιο προ-αξιολόγησης μελέτης περίπτωσης 1 ..	281
ΠΑΡΑΡΤΗΜΑ Γ Ερωτηματολόγιο μετά-αξιολόγησης μελέτης περίπτωσης 1 ..	289
ΠΑΡΑΡΤΗΜΑ Δ Ερωτηματολόγιο προ-αξιολόγησης μελέτης περίπτωσης 2...	301
ΠΑΡΑΡΤΗΜΑ Ε Ερωτηματολόγιο μετά-αξιολόγησης μελέτης περίπτωσης 2 ..	309
ΠΑΡΑΡΤΗΜΑ ΣΤ Ερωτηματολόγιο προ-αξιολόγησης μελέτης περίπτωσης 3 ..	321
ΠΑΡΑΡΤΗΜΑ Ζ Ερωτηματολόγιο μετά-αξιολόγησης μελέτης περίπτωσης 3 ..	329

Κατάλογος Πινάκων

Πίνακας 2.1. Συγκριτική παρουσίαση Προγραμματιστικών Μικρόκοσμων-Εκπαιδευτικών Προγραμματιστικών περιβαλλόντων για τη διδασκαλία του ΑΠ.....	78
Πίνακας 2.2. Συγκριτική παρουσίαση άλλων εργαλείων και γλωσσών Προγραμματισμού για τη διδασκαλία του ΑΠ.....	80
Πίνακας 3.1. Ενδεικτικοί ρόλοι του ανθρώπινου δυναμικού.....	133
Πίνακας 4.1. Κίνητρα παρακολούθησης (case study-1).....	144
Πίνακας 4.2. Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στο γνωστικό αντικείμενο του ΑΠ (case study-1).....	146
Πίνακας 4.3. Προτιμήσεις και στάσεις των φοιτητών σχετικά με το αντικείμενο του ΑΠ (case study-1).....	146
Πίνακας 4.4. Συμβολή παραγόντων της διδακτικής προσέγγισης στη βελτίωση γνώσεων και δεξιοτήτων (case study-1).....	148
Πίνακας 4.5. Αξιολόγηση των μαθησιακών πόρων του υλικού μεμονωμένα (case study-1).....	150
Πίνακας 4.6. Αξιολόγηση του μαθησιακού υλικού στο σύνολό του και της διδακτικής προσέγγισης (case study-1).....	151
Πίνακας 4.7. Αξιολόγηση της ευχρηστίας του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ (case study-1).....	153
Πίνακας 4.8. Αξιολόγηση της ευχρηστίας του προγραμματιστικού μικρόκοσμου Jeroo (case study-1).....	155
Πίνακας 4.9. Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στο γνωστικό αντικείμενο του ΑΠ (case study-2).....	169
Πίνακας 4.10. Προτιμήσεις και στάσεις των φοιτητών σχετικά με το αντικείμενο του ΑΠ (case study-2).....	170
Πίνακας 4.11. Συμβολή παραγόντων της διδακτικής προσέγγισης στη βελτίωση γνώσεων και δεξιοτήτων (case study-2).....	175

Πίνακας 4.12. Αξιολόγηση των μαθησιακών πόρων του υλικού μεμονωμένα (case study-2)	177
Πίνακας 4.13. Αξιολόγηση του μαθησιακού υλικού στο σύνολό του και της διδακτικής προσέγγισης (case study-2)	178
Πίνακας 4.14. Αξιολόγηση της ευχρηστίας του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ (case study-2)	179
Πίνακας 4.15. Αξιολόγηση της δυσκολίας αντικειμενοστρεφών εννοιών – αρχών (case study-2)	181
Πίνακας 4.16. Reliability Analysis – Scale (Alpha) (a) (case study-2)	201
Πίνακας 4.17. Reliability Analysis – Scale (Alpha) (b) (case study-2)	202
Πίνακας 4.18. Έλεγχος πολυσυγγραμμικότητας – a (case study-2)	211
Πίνακας 4.19. Έλεγχος πολυσυγγραμμικότητας – b (case study-2)	213
Πίνακας 4.20. Regression Analysis – SPSS output (case study-2)	214
Πίνακας 4.21. Regression Analysis – Model Summary (case study-2, φοιτητές με εμπειρία στον Προγραμματισμό)	217
Πίνακας 4.22. Κίνητρα παρακολούθησης (case study-3)	221
Πίνακας 4.23. Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στο γνωστικό αντικείμενο του ΑΠ (case study-3)	223
Πίνακας 4.24. Προτιμήσεις και στάσεις των φοιτητών σχετικά με το αντικείμενο του ΑΠ (case study-3)	223
Πίνακας 4.25. Συμβολή παραγόντων της διδακτικής προσέγγισης στη βελτίωση γνώσεων και δεξιοτήτων (case study-3)	228
Πίνακας 4.26. Αξιολόγηση των μαθησιακών πόρων του υλικού μεμονωμένα (case study-3)	229
Πίνακας 4.27. Αξιολόγηση του μαθησιακού υλικού στο σύνολό του και της διδακτικής προσέγγισης (case study-3)	231
Πίνακας 4.28. Αξιολόγηση της ευχρηστίας του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ (case study-3)	232

Πίνακας 4.29. Αξιολόγηση της δυσκολίας αντικειμενοστρεφών εννοιών – αρχών
(case study-3).....233

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

РАНЕЕЗНАМО ПЕРПАА

Κατάλογος Σχημάτων

Σχήμα 2.1. Περιβάλλον KarelJ.Robot	53
Σχήμα 2.2. JKarelRobot World.....	54
Σχήμα 2.3. JKarelRobot Flow Chart.....	54
Σχήμα 2.4. Η διαπροσωπεία χρήστη του περιβάλλοντος Jeroo.....	55
Σχήμα 2.5. Παράδειγμα προγράμματος στο περιβάλλον Jeroo.....	55
Σχήμα 2.6. Η διαπροσωπεία χρήστη του περιβάλλοντος Alice.....	56
Σχήμα 2.7. Περιβάλλον προγραμματισμού objectKarel.....	57
Σχήμα 2.8. Οπτικοποίηση εφαρμογής στο περιβάλλον BlueJ.....	58
Σχήμα 2.9. Object Workbench και μέθοδοι αντικειμένου (BlueJ)	59
Σχήμα 2.10. Αποτέλεσμα κλήσης μεθόδου (BlueJ).....	59
Σχήμα 2.11. Παρουσίαση interface κλάσης (BlueJ).....	59
Σχήμα 2.12. Implementation κλάσης - Επισήμανση λάθους κώδικα (BlueJ)	60
Σχήμα 2.13. Περιβάλλον DrJava	61
Σχήμα 2.14. Περιβάλλον DrScheme – ProfessorJ	62
Σχήμα 2.15. Περιβάλλον GiniPad	64
Σχήμα 2.16. Ginipad - Source browser.....	65
Σχήμα 2.17. jGrasp – UML.....	66
Σχήμα 2.18. jGrasp - Workbench	66
Σχήμα 2.19. Version 3 του D-ChK εργαλείου.....	67
Σχήμα 2.20. Class structure και οπτικοποίηση κατάστασης-συμπεριφοράς αντικειμένων (greenfoot)	68
Σχήμα 2.21. Δημιουργία αντικειμένου (greenfoot)	69
Σχήμα 2.22. Μέθοδοι αντικειμένου (greenfoot).....	69
Σχήμα 2.23. Περιβάλλον Robocode	73

Σχήμα 3.1. Παρουσίαση των συστατικών της προτεινόμενης διδακτικής προσέγγισης	92
Σχήμα 3.2. Ροή δραστηριοτήτων του μαθησιακού σεναρίου	104
Σχήμα 3.3. Φάση παρατήρησης	105
Σχήμα 3.4. Μοντέλο κλάσεων με ιδιότητες, λειτουργίες και σχέσεις (ArgoUML) ..	106
Σχήμα 3.5. “Σκελετικός” κώδικας Java (ArgoUML)	107
Σχήμα 3.6. Λειτουργίες μεθόδων (BlueJ)	108
Σχήμα 3.7. Φάση II - Επίλυση προβλημάτων με λεπτομερείς συμβουλευτικές - κατευθυντήριες οδηγίες	112
Σχήμα 3.8. Δικτυακός τόπος εφαρμογής της προτεινόμενης προσέγγισης στο Moodle LMS	114
Σχήμα 3.9. Ασύγχρονο forum συζητήσεων	114
Σχήμα 3.10. Υποβολή ερώτησης από εκπαιδευόμενο, στο forum συζητήσεων.....	114
Σχήμα 3.11. Απάντηση εκπαιδευόμενου σε θέμα συζήτησης, που έχει τεθεί στο forum συζητήσεων.....	115
Σχήμα 3.12. Δικτυακός τόπος στο περιβάλλον της ηλεκτρονικής πύλης μαθημάτων e- class του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά.....	115
Σχήμα 3.13. Σφαιρική παρουσίαση του Διαδικτυακά υποστηριζόμενου μαθησιακού περιβάλλοντος.....	116
Σχήμα 3.14. Διαπροσωπεία-υλοποίηση κλάσης	117
Σχήμα 3.15. Υπερίσχυση μεθόδων	117
Σχήμα 3.16. Μορφές πολυμορφισμού	118
Σχήμα 3.17. Overriding versus Overloading	118
Σχήμα 3.18. Στατική μεταβλητή κλάσης	119
Σχήμα 3.19. Dynamic binding	119
Σχήμα 3.20. Reference types	120
Σχήμα 3.21. Αντιδιαστολή overloaded-overridden μεθόδων	120

Σχήμα 3.22. Δημιουργία αντικειμένων	121
Σχήμα 3.23. Δραστηριότητα στο BlueJ	122
Σχήμα 3.24. Δραστηριότητα στο ArgoUML	122
Σχήμα 3.25. Ερώτηση για τη διάκριση κλάσης-αντικειμένου	122
Σχήμα 3.26. Δημιουργία αντικειμένων	122
Σχήμα 3.27. Υλοποίηση composed κλάσης.....	123
Σχήμα 3.28. Ερώτηση για τη διάκριση σχέσεων	123
Σχήμα 3.29. Περιπτώσεις δοκιμής (test cases) της μελέτης περίπτωσης	124
Σχήμα 3.30. Δυναμική οπτικοποίηση κατάστασης μνήμης.....	127
Σχήμα 3.31. Δυναμική οπτικοποίηση κατάστασης μνήμης και ροή προγράμματος .	127
Σχήμα 3.32. Reference types – composed κλάσεις.....	128
Σχήμα 3.33. Κατανομή μνήμης – primitive types	128
Σχήμα 3.34. Ένα στιγμιότυπο των διαγραμμάτων κλάσεων του αυθεντικού παραδείγματος της “λαϊκής αγοράς” (ArgoUML).....	130
Σχήμα 3.35. Ο παραγόμενος “σκελετικός” κώδικας (ArgoUML).....	130
Σχήμα 3.36. Μεταφορά του αυθεντικού παραδείγματος στο BlueJ	131
Σχήμα 3.37. Στιγμιότυπο από το περιβάλλον SUN One Studio, μετά την εκτέλεση του προγράμματος με main μέθοδο που υλοποιεί το αυθεντικό παράδειγμα	132
Σχήμα 4.1. Βαθμολογία των εργασιών του e-learning μαθήματος (case study-2)	197
Σχήμα 4.2. Βαθμολογία στις τελικές εξετάσεις, των φοιτητών που παρακολούθησαν το e-learning μάθημα (case study-2).....	198
Σχήμα 4.3. Βαθμολογία στις τελικές εξετάσεις, των φοιτητών που δεν παρακολούθησαν το e-learning μάθημα (case study-2)	198
Σχήμα 4.4. Έλεγχος κανονικότητας (case study–2).....	206
Σχήμα 4.5. Έλεγχος ομοσκεδαστικότητας (case study–2).....	208
Σχήμα 4.6. Έλεγχος γραμμικότητας (case study–2)	209
Σχήμα 4.7. Βαθμολογία των εργασιών του e-learning μαθήματος (case study-3)	240

РАНЕЕЗНАМО ПЕРПАА

Συντομογραφίες

Λατινικές

ACM	Association of Computing Machinery
CASE	Computer Aided Software Engineering
COOL project	Ερευνητικό έργο “Comprehensive Object Oriented Learning”
CS	Computer Science
ICTs	Information and Communication Technologies
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronic Engineers
IF	Imperatives First
LMS	Learning Management System
OF	Objects First
OO	Object Oriented
OOP	Object Oriented Programming
PF	Programming First
UML	Unified Modeling Language

Ελληνικές

A.E.I.	Ανώτατο Εκπαιδευτικό Ίδρυμα
ΑΠ	Αντικειμενοστρεφής Προγραμματισμός
ΔΤΨΣ	Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων
Η/Υ	Ηλεκτρονικός Υπολογιστής
Ι.Ε.Κ.	Ινστιτούτο Επαγγελματικής Κατάρτισης
Μ.Ο.	Μέσος όρος
ΠΜΣ	Πρόγραμμα Μεταπτυχιακών Σπουδών

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1. Το γνωστικό αντικείμενο του Αντικειμενοστρεφούς Προγραμματισμού

Στην εκπαίδευση στον ευρύτερο χώρο της Επιστήμης των Υπολογιστών (Computer Science, CS), είναι εδραιωμένη η σημασία του ρόλου του Προγραμματισμού και της απόκτησης προγραμματιστικών δεξιοτήτων (programming skills) από τους εκπαιδευόμενους. Οι ενώσεις “Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS)” και “the Association for Computing Machinery (ACM)” ανέλαβαν από κοινού την αποστολή να αναπτύξουν οδηγίες για το Πρόγραμμα Σπουδών προπτυχιακών προγραμμάτων αυτής της περιοχής Επιστημών. Αποτέλεσμα αυτής της προσπάθειας αποτελεί η τελική αναφορά (final report) του 2001 (CC2001, 2001) για τη διδακτέα ύλη στους Υπολογιστές (Computing Curricula). Σε αυτήν αναφέρεται χαρακτηριστικά ότι “Καθ’ όλη τη διάρκεια της ιστορίας του επιστημονικού αυτού τομέα, τα περισσότερα εισαγωγικά μαθήματα έχουν εστιάσει κατά κύριο λόγο στην ανάπτυξη προγραμματιστικών δεξιοτήτων και αυτό απορρέει από έναν αριθμό πρακτικών και ιστορικών παραγόντων...” (CC2001, 2001, p.22) και ότι “Τα μέλη του CC2001 Task Force πιστεύουν ότι το programming-first μοντέλο είναι πιθανό να παραμείνει το επικρατέστερο για το προβλέψιμο μέλλον.” (CC2001, 2001, p.24).

Στον παραπάνω Οδηγό Σπουδών των IEEE/ACM παρέχονται τρεις διαφορετικοί τρόποι υλοποίησης του *programming-first* μοντέλου (CC2001, 2001, section 7.6), δηλαδή αυτού που υποστηρίζει τη μεγάλη σημασία του γνωστικού αντικείμενου του Προγραμματισμού για την εκπαίδευση στην Επιστήμη των Υπολογιστών. Αυτοί είναι: η *imperative-first* προσέγγιση που χρησιμοποιεί το παραδοσιακό διαδικασικό (procedural) παράδειγμα Προγραμματισμού, η *objects-first* προσέγγιση που δίνει έμφαση στις αρχές του Αντικειμενοστρεφούς Προγραμματισμού και της Αντικειμενοστρεφούς σχεδίασης από την αρχή και η *functional-first* προσέγγιση που παρουσιάστηκε από το MIT τη δεκαετία του 80 και εισάγει αλγοριθμικές έννοιες με

τη χρήση μιας γλώσσας προγραμματισμού με απλή, μινιμαλιστική functional σύνταξη, όπως η Scheme.

Στη δεκαετία του 1960, ο Kristen Nygaard (από τη θέση του Director of Research), μαζί με τον Ole-Johan Dahl ανέπτυξαν τις πρώτες αντικειμενοστρεφείς γλώσσες προγραμματισμού, SIMULA I και SIMULA 67, στο Νορβηγικό Κέντρο Υπολογιστών (Norwegian Computing Center, NCC). Αυτό διαφοροποίησε εντελώς τον τρόπο σκέψης των ανθρώπων για τον Προγραμματισμό σε παγκόσμια κλίμακα (Fjuk et al., Chapter1, p.2). Η Αντικειμενοστρεφής προσέγγιση Προγραμματισμού που διαμορφώθηκε μέσα από την έρευνα του Kristen Nygaard, αποτέλεσε έναν τρόπο προσέγγισης και χειρισμού της πολυπλοκότητας των υπολογιστικών συστημάτων τα οποία αποτελούν τα θεμέλια της σύγχρονης “κοινωνίας της πληροφορίας”. Με αυτή την επινόηση, ο Kristen Nygaard έκανε την προεργασία για το μοντέρνο τρόπο Προγραμματισμού Υπολογιστών, με μια γλώσσα προγραμματισμού σχεδιασμένη για να προσομοιώνει πολύπλοκα συστήματα πραγματικών συνθηκών (real-world systems). Πολύ αργότερα, το 2001, ιδρύθηκε το ερευνητικό εργαστήριο Simula (Simula Research Laboratory) με απόφαση του Νορβηγικού Κοινοβουλίου, το οποίο φέρει το όνομα της αντικειμενοστρεφούς γλώσσας που οι δύο ερευνητές δημιούργησαν. Την ίδια χρονιά τους απονεμήθηκε το μετάλλιο John von Neumann από την IEEE και την επομένη, το 2002, τους απονεμήθηκε το βραβείο A.M. Turing από την ACM “για τις θεμελιώδεις για την εμφάνιση του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού ιδέες, με τη σχεδίαση των προγραμματιστικών γλωσσών Simula I and Simula 67.” (Fjuk et al., Chapter1, p.6). Η ACM, στη νεκρολογία για τους δύο αποδέκτες του βραβείου (ACM, 2002), αναφέρει ότι το βραβείο A.M. Turing θεωρείται το Nobel των Υπολογιστών (Nobel prize of Computing). Επίσης, αναφέρεται στις εκτεταμένες συνέπειες που είχε η εφεύρεση του Αντικειμενοστρεφούς Προγραμματισμού (ΑΠ) (object-oriented programming, OOP), που σήμερα είναι το ευρύτερα χρησιμοποιούμενο προγραμματιστικό μοντέλο, ως εξής: “Η δουλειά τους, οδήγησε σε μια θεμελιακή αλλαγή του τρόπου σχεδιασμού και προγραμματισμού των συστημάτων λογισμικού, με αποτέλεσμα επαναχρησιμοποιήσιμες, αξιόπιστες και επιδεκτικές αλλαγών εφαρμογές που έχουν εκσυγχρονίσει τη διαδικασία συγγραφής του κώδικα λογισμικού και έχουν διευκολύνει τον προγραμματισμό.” (Fjuk et al., Chapter1, p.6-7).

Στο πολύ αξιόλογο βιβλίο του Grady Booch που αφορά στην Αντικειμενοστρεφή (object-oriented, OO) σχεδίαση (Booch, 1991), παρουσιάζεται πολύ εύστοχα η

εγγενής πολυπλοκότητα των συστημάτων λογισμικού. Επίσης περιγράφονται οι συνέπειες αυτής της πολυπλοκότητας στην ανάπτυξη των συστημάτων, ο Αλγοριθμικός ή Διαδικασιακός τρόπος αποσύνθεσης ενός συστήματος (top-down structure ή procedure-oriented decomposition) προκειμένου να αντιμετωπιστεί η πολυπλοκότητά του και παρουσιάζεται και προτείνεται ένας εναλλακτικός τρόπος αποσύνθεσης, ο Αντικειμενοστρεφής (object-oriented decomposition) με τα ουσιώδη χαρακτηριστικά του που τον διαφοροποιούν από το Διαδικασιακό (procedural) (Booch, 1991, chapter 1). Στα κεφάλαια 2 και 7 του παραπάνω βιβλίου, παρουσιάζονται εκτενώς τα μεγάλης σημασίας πλεονεκτήματα της Αντικειμενοστρεφούς σχεδίασης σε σχέση με την Αλγοριθμική ή Διαδικασιακή, όπως για παράδειγμα τα μικρότερου μεγέθους συστήματα που παράγει (μέσω της επαναχρησιμοποίησης κοινών μηχανισμών) καθώς και η μεγαλύτερη ελαστικότητα σε αλλαγές των παραγομένων με αυτό τον τρόπο συστημάτων, άρα και η καλύτερη δυνατότητα εξέλιξης τους.

Έχει πλέον επικρατήσει η άποψη ότι η ανάπτυξη λογισμικού με Αντικειμενοστρεφή φιλοσοφία, είναι αποδοτικότερη από την παραδοσιακή διαδικασιακή φιλοσοφία και ότι επίσης η Αντικειμενοστρεφής τεχνολογία παρέχει πολύ καλύτερα θεμέλια για το σχεδιασμό λογισμικού. Η σύγχρονη επομένως τάση στις απαιτήσεις της ανάπτυξης λογισμικού έχει κάνει προφανή την ανάγκη για τη διδασκαλία του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού και κατά συνέπεια το αντικείμενο αυτό έχει εδραιώσει τη θέση του στο Πρόγραμμα Σπουδών των Πανεπιστημιακών ιδρυμάτων.

Από όλα τα παραπάνω, αλλά και από πληθώρα άρθρων που δημοσιεύονται σε διεθνή περιοδικά και επιστημονικά συνέδρια και αφορούν στην έρευνα τη σχετική με τη διδασκαλία του ΑΠ και από τις απαιτήσεις που η αγορά εργασίας στο χώρο της ανάπτυξης λογισμικού προτάσσει, προκύπτει η μετάβαση που έχει γίνει από το παραδοσιακό διαδικασιακό παράδειγμα Προγραμματισμού (procedural programming paradigm) που παλαιότερα επικρατούσε, στο Αντικειμενοστρεφές (object-oriented programming paradigm) και η ανάγκη της αποτελεσματικής διδασκαλίας του τελευταίου.

Χαρακτηριστικά αποσπάσματα μερικών μόνον άρθρων από τη βιβλιογραφία, που δηλώνουν την υπερίσχυση του Αντικειμενοστρεφούς παραδείγματος προγραμματισμού έναντι του παραδοσιακού διαδικασιακού, παρατίθενται στη συνέχεια:

- “Το Αντικειμενοστρεφές παράδειγμα Προγραμματισμού έχει κερδίσει κατά πολύ το ενδιαφέρον την τελευταία δεκαετία...” (Bennedsen and Caspersen, 2004a).
- “Η διδασκαλία προγραμματισμού και μοντελοποίησης με τη χρήση αντικειμενοστρεφών μεθόδων αποτελεί κοινό τρόπο στην εισαγωγική εκπαίδευση στην Επιστήμη των Υπολογιστών, κατά τη διάρκεια των τελευταίων δέκα χρόνων” (Berge et al., 2003a).
- “Η διδασκαλία της Αντικειμενοστρέφειας κατά το πρώτο έτος σπουδών, ιδιαίτερα με Java, είναι όλο και πιο συχνή” (Kölling and Rosenberg, 2001b).
- “Η Αντικειμενοστρέφεια συνεχίζει να εδραιώνει τη θέση της στα Προγράμματα Σπουδών, ιδιαίτερα στα πρώτα μαθήματα” (Lewis, 2000).
- “Η εισαγωγή στον Προγραμματισμό με αντικειμενοστρεφείς γλώσσες μετατίθεται πιο νωρίς στα Προγράμματα Σπουδών με αργό ρυθμό, ενώ εισάγεται συχνά τελευταία σε επίπεδο Λυκείου” (Henriksen and Kölling, 2004).
- “Η objects-first στρατηγική για τη διδασκαλία του Προγραμματισμού έχει επικρατήσει των imperative-first και functional-first στρατηγικών κατά την τελευταία δεκαετία” (Xinogalos et al., 2006).
- “Πρόσφατα, πολλοί επαγγελματίες μηχανικοί λογισμικού χρησιμοποιούν το Αντικειμενοστρεφές παράδειγμα Προγραμματισμού και κατά συνέπεια, η Αντικειμενοστρεφής ανάλυση, σχεδίαση και ο προγραμματισμός έχουν βρει έδαφος στην εκπαίδευση της Επιστήμης των Υπολογιστών” (Ragonis and Ben-Ari, 2005b)
- “Το Αντικειμενοστρεφές παράδειγμα αντικατέστησε το προηγούμενο διαδικασιακό, το οποίο αποτέλεσε το κύριο προγραμματιστικό παράδειγμα τόσο στον ακαδημαϊκό χώρο όσο και στην αγορά λογισμικού επί χρόνια” (Teif and Hazzan, 2004).
- “Η γενική τάση αλλαγής που παρατηρείται, από τις διαδικασιακές στις αντικειμενοστρεφείς γλώσσες στα πρώτα μαθήματα Προγραμματισμού, αντιστοιχεί στη μετακίνηση της αγοράς λογισμικού από μικρότερα, ατομικά έργα κωδικοποίησης σε μεγάλης κλίμακας ομαδικά έργα” (Barr et al., 1999).

1.2. Προσεγγίσεις στη διδασκαλία του ΑΠ και σημεία που επιδέχονται επέμβαση

Για τη διδασκαλία του ΑΠ έχουν αναπτυχθεί, προταθεί και ακολουθηθεί τρόποι, με διαφορές στις αρχές και τα χαρακτηριστικά τους, το διάστημα που αυτός έχει εμφανιστεί ως ιδιαίτερο παράδειγμα Προγραμματισμού και έχει συμπεριληφθεί ως γνωστικό αντικείμενο στα Προγράμματα Σπουδών.

Παρότι το παράδειγμα αυτό Προγραμματισμού, συνιστά ουσιαστικά μια εντελώς διαφορετική φιλοσοφία και ένα διαφορετικό τρόπο αντιμετώπισης των προβλημάτων προς επίλυση, έχει διδαχθεί χωρίς ουσιαστικές διαφοροποιήσεις στον τρόπο διδασκαλίας από το διαδικασιακό Προγραμματισμό, κυρίως το πρώτο διάστημα από την εμφάνισή του και την ένταξή του στα Προγράμματα Σπουδών, λόγω έλλειψης εμπειρίας των διδασκόντων, κατάλληλων συγγραμμάτων, εκπαιδευτικών εργαλείων και κατάλληλου υποστηρικτικού διδακτικού υλικού (Kölling and Rosenberg, 2001a; Kölling et al., 2003). Αυτό που συνέβαινε ήταν η αλλαγή στη γλώσσα Προγραμματισμού που διδασκόταν (για παράδειγμα από C σε C++), χωρίς να επιχειρείται μια σε βάθος αναζήτηση νέων προσεγγίσεων στον τρόπο διδασκαλίας, με αποτέλεσμα να διδάσκεται μια γλώσσα προγραμματισμού με αντικειμενοστρεφή προσανατολισμό και σαφώς άλλες δυνατότητες, αλλά με τη φιλοσοφία του διαδικασιακού Προγραμματισμού (Bennedsen and Caspersen, 2004b; Groven et al., 2003). Επίσης αρκετά συγγράμματα έχουν διατηρήσει έναν διαδικασιακό τρόπο αντίληψης για τον Προγραμματισμό (Kölling, 2003). Ο παραπάνω τρόπος αναφέρονται στη βιβλιογραφία ως *“imperatives first”* ή *“procedurals first”* προσέγγιση στη διδασκαλία του ΑΠ. Επιπλέον, συχνά η ακολουθούμενη προσέγγιση ήταν γλωσσο-οδηγούμενη (ή οδηγούμενη από τη σύνταξη, syntax-driven) δηλαδή επίκεντρό της αποτελούσε η σύνταξη της γλώσσας προγραμματισμού και όχι οι προγραμματιστικές τεχνικές (Bennedsen and Caspersen, 2004b), μέθοδος που αποτελεί την κλασική διδακτική προσέγγιση του Προγραμματισμού γενικά (με τη χρήση μιας γλώσσας προγραμματισμού και ενός επαγγελματικού ολοκληρωμένου περιβάλλοντος ανάπτυξης (Integrated Development Environment, IDE)) (Xinogalos and Satratzemi, 2002).

Είναι ευρέως αποδεκτό, ότι ο ΑΠ απαιτεί άλλο τρόπο σκέψης και συνεπώς άλλες μεθόδους προσέγγισης στη διδασκαλία του (Kölling and Rosenberg, 2000). Ο τρόπος

όπου συγκλίνουν οι απόψεις, είναι ο “objects early” δηλαδή η παρουσίαση των θεμελιωδών αντικειμενοστρεφών εννοιών (αντικείμενο, κλάση) πολύ πρώιμα στη διδασκαλία (Henriksen and Kölling, 2004; Kölling and Rosenberg, 2001a) και ουσιαστικά αυτό συνιστά την “objects first” (OF) προσέγγιση στη διδασκαλία του ΑΠ που έχει επικρατήσει τα τελευταία χρόνια. Γύρω από τους τρόπους υλοποίησης της προσέγγισης αυτής διεξάγεται σημαντική έρευνα (Kölling et al., 2003; Bergin et al., 2003; Sanders and Dorn, 2003a; Buck and Stucki, 2001; Xinogalos et al., 2006; Proulx et al., 2002; Cooper et al., 2003; Bruce et al., 2001; Alphonse and Ventura, 2003; 2002; Goldman, 2004).

Στο Lewis (2000), παρουσιάζεται η άποψη πως αποτελεί μύθο ότι “η φράση “objects first” είναι καλά ορισμένη”, πράγμα που υποδηλώνει τη διαφορετικότητα με την οποία μπορεί να υλοποιηθεί η OF προσέγγιση.

Μια υλοποίηση της με έμφαση στη μοντελοποίηση (model first ή model driven) είναι αυτή η οποία ουσιαστικά αποτελεί στρατηγική που ακολουθεί τη Σκανδιναβική παράδοση (“Scandinavian tradition”), έχει δε χαρακτηριστεί από τον Kristen Nygaard ως Σκανδιναβική προσέγγιση στην αντικειμενοστρέφεια (Scandinavian approach to object-orientation) (Madsen, 1995; Smørdal, 1999). Αυτή η υλοποίηση δίνει έμφαση στην *εννοιολογική μοντελοποίηση (conceptual modeling)* ως καθοριστικό χαρακτηριστικό της αντικειμενοστρέφειας και επίσης στις *modeling “πλευρές”* της (Bennedsen and Caspersen, 2004a).

Η βασική ιδέα είναι η θεώρηση ενός αντικειμενοστρεφούς προγράμματος ως ένα μοντέλο (model), το οποίο προσομοιώνει τη συμπεριφορά ενός πραγματικού ή ενός φανταστικού τμήματος του κόσμου (Bennedsen and Caspersen, 2004b) και το οποίο μοντέλο εξετάζεται σε διαφορετικά επίπεδα, τα οποία είναι: αυτό του εννοιολογικού μοντέλου που περιγράφει τα βασικά στοιχεία ενός προβλήματος και τις μεταξύ τους σχέσεις (problem domain → model), αυτό ενός πιο λεπτομερούς μοντέλου κλάσεων (class model) που παρέχει μια σφαιρική θεώρηση της λύσης, καθώς και αυτό της πραγματικής υλοποίησης της λύσης σε μια αντικειμενοστρεφή γλώσσα προγραμματισμού (programming code). Διατυπώνει την άποψη, ότι ο ΑΠ θα αντιμετωπίζεται ως μοντελοποίηση (modeling) και ότι οι βασικές έννοιες του αντικειμενοστρεφούς παραδείγματος προγραμματισμού θα εμπεδώνονται από νωρίς (Groven et al., 2003; Berge et al., 2003b) και ότι μια τέτοια στρατηγική υλοποίησης απαιτείται να υποστηριχθεί από κατάλληλα εργαλεία σε συνδυασμό με τη χρήση κατάλληλα επιλεγμένων παραδειγμάτων (Groven et al., 2003).

Με την “model first” στρατηγική επιτυγχάνεται η εξισορρόπηση ανάμεσα στα δύο άκρα που είναι από τη μια η μοντελοποίηση με τη χρήση, για παράδειγμα, ενός CASE εργαλείου που παράγει πλήρως τον κώδικα και από την άλλη η εστίαση στις λεπτομέρειες της γλώσσας προγραμματισμού που συνεπάγεται τη μη εκμάθηση του προγραμματιστικού παραδείγματος. Η στρατηγική αυτή πραγματεύεται επομένως την *ενοποίηση της εννοιολογικής μοντελοποίησης και της κωδικοποίησης*. Στοχεύει στην απόκτηση δεξιοτήτων για την επίλυση προβλημάτων, εφαρμόζοντας την αντικειμενοστρεφή φιλοσοφία μέσω σχεδίασης μοντέλων και επίσης δεξιοτήτων ανάπτυξης κώδικα που βασίζεται σε αυτά τα μοντέλα (Bennedsen and Caspersen, 2004b).

Εκτός από τις προσεγγίσεις στη διδασκαλία του ΑΠ, στη βιβλιογραφία εμφανίζονται ερευνητικές μελέτες που περιγράφουν μαθησιακές δυσκολίες των εκπαιδευομένων (Teif and Hazzan, 2004; Ragonis and Ben-Ari, 2002; Ragonis and Ben-Ari, 2005a; Ben-Ari et al., 2002; Wiedenbeck and Ramalingam, 1999; Wiedenbeck et al., 1999; Ragonis and Ben-Ari, 2005b; Carter and Fowler, 1998; Holland et al., 1997; Fleury, 2000; Fleury, 2001; Milne and Rowe, 2002; Lahtinen et al., 2005). Σε κάποιες από τις έρευνες διατυπώνονται και συστάσεις διδασκαλίας για την αποφυγή των δυσκολιών και παρανοήσεων (Fleury, 2000; Fleury, 2001; Holland et al., 1997; Ala-Mutka) ή οδηγίες (Kölling and Rosenberg, 2001b). Επίσης έχουν διατυπωθεί μια σειρά από παιδαγωγικά σχεδιαστικά πρότυπα (pedagogical patterns) που αφορούν στη διδασκαλία (Bergin, 2006a και Bergin, 2006b), στην εκπαίδευση στην τάξη (Classroom Education) (Anthony, 1996) καθώς και παιδαγωγικά σχεδιαστικά πρότυπα για την αποδοτική διδασκαλία σεμιναρίων (A Pedagogical Pattern Language About Teaching Seminars Effectively) (Fricke and Voelter, 2000).

Από όλα τα παραπάνω, συνάγεται ότι η διδασκαλία του ΑΠ, χρήζει ανάπτυξης συγκεκριμένων διδακτικών προσεγγίσεων με εκπαιδευτικές αρχές και φιλοσοφία προσανατολισμένες στις ευρέως επικρατούσες τάσεις. Επίσης χρήζει προσεγγίσεων που θα αξιοποιούν την προηγούμενη ερευνητική προσπάθεια που αφορά στην επισήμανση δυσκολιών-παρανοήσεων, στη διατύπωση συστάσεων για τη μαθησιακά αποτελεσματικότερη διδασκαλία του αντικείμενου και στην ανάπτυξη εκπαιδευτικών εργαλείων για το σκοπό αυτό.

Συνεπώς αποτελεί ένα ερευνητικά ενδιαφέρον θέμα και θα συνεισέφερε στην αποτελεσματική διδασκαλία του ΑΠ, η ανάπτυξη μιας ολοκληρωμένης διδακτικής προσέγγισης γι' αυτό το γνωστικό αντικείμενο, η οποία θα αποτελεί ένα πλήρες

εκπαιδευτικό σενάριο που θα ακολουθεί την “objects first” προσέγγιση και θα την υλοποιεί με “model first” στρατηγική με ό,τι αυτό συνεπάγεται (ενσωμάτωση κατάλληλων εργαλείων, δημιουργία κατάλληλου μαθησιακού περιβάλλοντος για την ένταξη της προσέγγισης, σχεδιασμός κατάλληλων παραδειγμάτων και μαθησιακού υλικού γενικότερα). Επίσης, η προσέγγιση αυτή θα καλύπτει όλα εκείνα τα θέματα που θεωρούνται απαραίτητα για την εισαγωγή στην Αντικειμενοστρεφή φιλοσοφία Προγραμματισμού και συγχρόνως θα εφαρμόζει στην πράξη συστάσεις διδασκαλίας που έχουν διατυπωθεί με διάφορους τρόπους (οδηγίες - σχεδιαστικά πρότυπα) και θα επιχειρεί συγκεκριμένες παρεμβάσεις με στόχο την αντιμετώπιση καταγεγραμμένων μαθησιακών δυσκολιών και παρανοήσεων.

Βασικοί στόχοι της διδασκαλίας ενός εισαγωγικού μαθήματος στον Αντικειμενοστρεφή Προγραμματισμό θα πρέπει να είναι:

- Η ανάπτυξη Αντικειμενοστρεφούς σκέψης επίλυσης προβλημάτων.
- Η εμπέδωση βασικών εννοιών-αρχών του παραδείγματος αυτού.
- Η ανάπτυξη ικανότητας σχεδίασης, υλοποίησης, ελέγχου, αποσφαλμάτωσης απλών προγραμμάτων σε μια αντικειμενοστρεφή γλώσσα Προγραμματισμού με τη χρήση CASE εργαλείων και προγραμματιστικών περιβαλλόντων.
- Η αντιμετώπιση καταγεγραμμένων διδακτικών προβλημάτων.

1.3. Αντικείμενο της διατριβής

Όλες οι παραπάνω διαπιστώσεις αποτέλεσαν το κίνητρο για την εκπόνηση της παρούσας διδακτορικής διατριβής. Το ερευνητικό πρόβλημα με το οποίο ασχολείται είναι η σχεδίαση, ανάπτυξη, εφαρμογή και αξιολόγηση μιας ολοκληρωμένης διδακτικής προσέγγισης για την εισαγωγή στην Αντικειμενοστρεφή φιλοσοφία Προγραμματισμού που θα έχει τα εξής χαρακτηριστικά:

- Θα **περιλαμβάνει** εκείνο το **διδακτικό περιεχόμενο που θεωρείται απαραίτητο** για να εισάγει τους εκπαιδευόμενους στη φιλοσοφία της Αντικειμενοστρέφειας.
- Θα **επιχειρεί με συγκεκριμένες παρεμβάσεις να αντιμετωπίσει μαθησιακές δυσκολίες**.
- Θα στηρίζεται στην **“objects first”** διδακτική προσέγγιση για τη διδασκαλία του ΑΠ με **“model first”** στρατηγική υλοποίησης.

- Θα συμπεριλαμβάνει για την επίτευξη του στόχου της, τη **σχεδίαση και χρήση κατάλληλων παραδειγμάτων και μαθησιακού υλικού**.
- Επίσης θα συμπεριλαμβάνει τη **χρήση κατάλληλων εκπαιδευτικών εργαλείων-προγραμματιστικών περιβαλλόντων** καθώς και τη **δημιουργία κατάλληλου μαθησιακού περιβάλλοντος με αξιοποίηση των Διαδικτυακών τεχνολογιών**.
- Θα προτείνει συγκεκριμένες μαθησιακές δραστηριότητες που θα λαμβάνουν χώρα με συγκεκριμένη σειρά και τρόπο –δηλαδή θα αποτελούν ένα **μαθησιακό σενάριο, (learning script)**– για την επίτευξη του γενικού στόχου της που αποτελεί η εισαγωγή στην Αντικειμενοστρεφή φιλοσοφία Προγραμματισμού. Θα μπορούν αυτές οι δραστηριότητες να χρησιμοποιηθούν ως οδηγός υλοποίησης της προτεινόμενης προσέγγισης σε πραγματικές εκπαιδευτικές συνθήκες.

Αξίζει να σημειωθεί σε αυτό το σημείο, ότι ο πλήρης κατάλογος των χαρακτηριστικών που διαθέτει η διδακτική προσέγγιση που η παρούσα διατριβή πραγματεύεται, συμπληρωνόταν κατά τη διάρκεια της εκπόνησής της, μετά από εκτενή βιβλιογραφική έρευνα, δοκιμές ρύθμισης και εφαρμογές της προσέγγισης που σχεδιάστηκε κατά φάσεις, μέχρι την τελική μορφή της που παρουσιάζεται εδώ.

Η διδακτική προσέγγιση που προτείνεται στη διατριβή, αξιοποιεί στη φιλοσοφία της προηγούμενα ερευνητικά ευρήματα. Περιλαμβάνει υποστηρικτικό μαθησιακό υλικό με ποικιλία στη μορφή του, το οποίο δίνει έμφαση στην αντιμετώπιση μαθησιακών προβλημάτων, ενσωματώνει συστάσεις διδασκαλίας (teaching recommendations) και καλές διδακτικές-παιδαγωγικές πρακτικές που έχουν διατυπωθεί με τη μορφή οδηγιών (guidelines) ή σχεδιαστικών προτύπων (design patterns) σε προηγούμενες ερευνητικές μελέτες, προκειμένου να αντιμετωπιστούν ήδη καταγεγραμμένα μαθησιακά προβλήματα και αξιοποιεί στοχευμένα, εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα. Τα θέματα που η προσέγγιση περιλαμβάνει συνάδουν με τις οδηγίες του Οδηγού Σπουδών για θέματα Πληροφορικής των IEEE/ACM (CC2001, 2001) και η εφαρμογή της προσέγγισης κάνει χρήση των τεχνολογιών του Παγκόσμιου Ιστού. Διέπεται από συγκεκριμένη παιδαγωγική αρχή που είναι η εκμάθηση του γνωστικού αντικείμενου κοντά σε ειδικούς (apprenticeship) με πειραματισμό και με έμφαση στη μοντελοποίηση (modeling approach).

Προκειμένου να επιτευχθεί ο απώτερος στόχος της διδακτορικής διατριβής, κρίθηκε απαραίτητο να ακολουθηθεί η παρακάτω πορεία:

- Μελετήθηκαν εκτενώς προηγούμενες έρευνες, που αφορούν στη διαπίστωση και καταγραφή δυσκολιών και παρανοήσεων σχετικά με το γνωστικό αντικείμενο

(έγινε εκμετάλλευση των στοιχείων που υπάρχουν διάσπαρτα στη βιβλιογραφία και εντοπίστηκαν μέσα από αυτά βασικές αρχές που πρέπει να διέπουν μια διδακτική προσέγγιση για την εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό).

- Προσδιορίστηκαν τα θέματα τα οποία περιλαμβάνει η προσέγγιση, ώστε να ικανοποιούνται οι οδηγίες που προβλέπονται από τον οδηγό σπουδών σε θέματα πληροφορικής των ΙΕΕΕ/ACM και συγχρόνως οι έννοιες και οι αρχές να καλύπτονται ικανοποιητικά, έτσι ώστε να παρουσιάζονται τα ουσιώδη θέματα για την εισαγωγή των σπουδαστών στη συγκεκριμένη φιλοσοφία Προγραμματισμού.
- Μελετήθηκε βιβλιογραφία σχετική με τη σχεδίαση και πρόταση διδακτικών προσεγγίσεων για τη διδασκαλία του ΑΠ, προκειμένου να αναδειχθούν αυτές με τη μεγαλύτερη καταγεγραμμένη βιβλιογραφικά μαθησιακή αποτελεσματικότητα.
- Μελετήθηκαν προηγούμενες ερευνητικές προσπάθειες σχετικές με την ανάπτυξη και παρουσίαση εκπαιδευτικών και άλλων περιβαλλόντων και εργαλείων όπως προγραμματιστικά περιβάλλοντα και εργαλεία CASE, για τη διδασκαλία αυτού του παραδείγματος Προγραμματισμού καθώς και εφαρμογές τους στη διδακτική πράξη. Από τη μελέτη αυτή αναδείχθηκαν τα κατάλληλα εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα που εντάχθηκαν και αξιοποιήθηκαν στην προσέγγιση που σχεδιάστηκε.
- Αναπτύχθηκε μαθησιακό υλικό που συνοδεύει και υποστηρίζει τη διδακτική προσέγγιση.
- Σχεδιάστηκε η διδακτική προσέγγιση με βάση τους παραπάνω άξονες (εκπαιδευτική φιλοσοφία, αρχές, στρατηγικές, υλικό, εργαλεία, περιβάλλον) καθώς και το πλαίσιο εφαρμογής της.
- Δημιουργήθηκε πλαίσιο αξιολόγησης της προσέγγισης και εργαλεία συλλογής δεδομένων (ερωτηματολόγια, ασκήσεις-ερωτήσεις κλπ), που στηρίζονται στη μέθοδο αξιολόγησης CADMOS-E (Retalis et al., 1998; Psaromiligkos and Retalis, 2003).
- Για να διερευνηθεί η επίτευξη των στόχων της διδακτικής προσέγγισης, διενεργήθηκαν, στο πλαίσιο της διατριβής, τρεις εφαρμογές της στη διδακτική πράξη και ισάριθμες μελέτες αξιολόγησης (evaluation studies) τους. Από αυτές έχουν εξαχθεί συμπεράσματα για τη βελτίωση της προσέγγισης μεταξύ των εφαρμογών της και την αξιολόγηση του τελικού ερευνητικού προϊόντος που

προέκυψε μετά τις επαναλαμβανόμενες εφαρμογές και αξιολογήσεις του σε χρονοσειρές.

Η αξιολόγηση εστιάζει:

- Στην καταγραφή της γνώμης-αποδοχής των σπουδαστών με την εξαγωγή ποσοτικών και ποιοτικών στοιχείων.
- Στην αποτίμηση των μαθησιακών αποτελεσμάτων της προσέγγισης.
- Τεκμηριώθηκε η διδακτική προσέγγιση με τη δημοσίευση άρθρων σε διεθνή περιοδικά και την ανακοίνωση τμημάτων της σε ελληνικά επιστημονικά συνέδρια.

Βασικές συνιστώσες της συνεισφοράς της έρευνας που διεξήχθη στο πλαίσιο της διδακτορικής διατριβής αποτελούν τα εξής:

1. Η εκτενής βιβλιογραφική επισκόπηση η οποία παρουσιάζει συγκεντρωμένα στοιχεία διάσπαρτα στη βιβλιογραφία και καλύπτει ένα κενό στη βιβλιογραφία ως προς την καταγραφή των δυσκολιών-παρανοήσεων, εργαλείων, προσεγγίσεων για τη διδασκαλία του ΑΠ και τη συνολική παρουσίασή τους.
2. Η συγκεκριμένη προσέγγιση διδασκαλίας που αναπτύχθηκε για το σύνολο των βασικών εννοιών-θεμάτων-αρχών του Αντικειμενοστρεφούς Προγραμματισμού, με εστίαση στη σφαιρική αντιμετώπιση προβλημάτων στη διδασκαλία του. Η προσέγγιση αυτή, αφορά στη συγκεκριμένη φιλοσοφία Προγραμματισμού, χωρίς να εστιάζει στις σημασιολογικές και συντακτικές λεπτομέρειες της γλώσσας Προγραμματισμού που χρησιμοποιείται (Java).
3. Εφαρμόστηκαν για τη δημιουργία πρωτόλειου μαθησιακού υλικού και τη σχεδίαση-εφαρμογή της διδακτικής προσέγγισης, συστάσεις διδασκαλίας (teaching recommendations) και καλές διδακτικές-παιδαγωγικές πρακτικές που έχουν διατυπωθεί με τη μορφή οδηγιών (guidelines) ή σχεδιαστικών προτύπων (design patterns). Αξιοποιήθηκαν συνεπώς με ουσιαστικό τρόπο, ευρήματα προηγούμενης έρευνας προς την κατεύθυνση να αντιμετωπιστούν διαπιστωμένες δυσκολίες-παρανοήσεις των σπουδαστών και επιχειρήθηκαν συγκεκριμένες παρεμβάσεις για τα μαθησιακά προβλήματα που είχαν διαπιστωθεί.
4. Έγιναν αξιολογήσεις της εφαρμογής της διδακτικής προσέγγισης και παρουσιάστηκαν αποτελέσματα τους σε χρονοσειρά (επαναλαμβανόμενα σε κάθε εφαρμογή). Αυτό αποτελεί πρωτοτυπία στη μέθοδο αξιολόγησης, που συνήθως πραγματοποιείται μια φορά χωρίς επανάληψη σε επόμενο χρόνο. Συγχρόνως η

μέθοδος βελτιωνόταν ενδιάμεσα στις επανειλημμένες αυτές εφαρμογές και με βάση τα αποτελέσματα των αξιολογήσεων, για να πάρει την τελική προτεινόμενη μορφή της.

5. Αποδείχθηκε η αποτελεσματικότητα της προσέγγισης με τις επανειλημμένες εφαρμογές και αξιολογήσεις της.
6. Οι εφαρμογές της διδακτικής προσέγγισης, απέφεραν εμπειρικά δεδομένα για τη διδασκαλία του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού, κάτι για το οποίο υπάρχει μικρή σχετικά καταγεγραμμένη εμπειρία (συγκρινόμενη με την αντίστοιχη για το διαδικασιακό Προγραμματισμό) και κατά συνέπεια προήγαγαν το πεδίο έρευνας αυτού του τομέα.
7. Από τις μελέτες αξιολόγησης των εφαρμογών της προσέγγισης προέκυψαν σημαντικά συμπεράσματα για το βαθμό συμβολής διαφόρων παραγόντων στη μαθησιακή διαδικασία και συνεπώς τη μαθησιακή αποτελεσματικότητά τους. Οι παράγοντες αυτοί αφορούν στο μαθησιακό υλικό και στους μαθησιακούς πόρους που το αποτελούν, στον τρόπο διεξαγωγής της εφαρμογής, στη χρήση τεχνολογιών του Παγκόσμιου Ιστού, στα εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα που χρησιμοποιήθηκαν.
8. Αναδείχθηκαν μαθησιακά προβλήματα και δυσκολίες-παρανοήσεις των σπουδαστών για το γνωστικό αντικείμενο του Αντικειμενοστρεφούς Προγραμματισμού που δεν είχαν καταγραφεί.

1.4. Οργάνωση της διατριβής

Η παρουσίαση της διατριβής που ακολουθεί γίνεται με την παρακάτω οργάνωση:

Στο **Κεφάλαιο 2 – “Η Διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού”**, παρουσιάζονται τα βιβλιογραφικά ευρήματα που προέκυψαν από την εκτενή έρευνα που πραγματοποιήθηκε καθ’ όλη τη διάρκεια εκπόνησης της διατριβής προς αυτήν την κατεύθυνση. Σε αυτά περιλαμβάνονται: οι καταγεγραμμένες δυσκολίες-παρανοήσεις των εκπαιδευομένων από έρευνες βραχείας και μακράς διάρκειας, άλλα ευρήματα από αυτές τις έρευνες για τη διδασκαλία του ΑΠ, οι διδακτικές προσεγγίσεις που έχουν αναπτυχθεί και προταθεί για τη διδασκαλία του ΑΠ και τα κύρια χαρακτηριστικά τους, η παρουσίαση των εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων για την υποστήριξη της διδασκαλίας του και

τέλος συστάσεις για τη διδασκαλία του, που προκύπτουν από όλη τη βιβλιογραφία που διερευνήθηκε και μελετήθηκε.

Στο **Κεφάλαιο 3 – “Η προτεινόμενη διδακτική προσέγγιση για τον ΑΠ”**, παρουσιάζεται η διδακτική προσέγγιση που σχεδιάστηκε και προτείνεται στη διατριβή για τη διδασκαλία του ΑΠ. Παρατίθενται τα θεμέλιά της και τα δομικά στοιχεία από τα οποία αποτελείται.

Αναλυτικότερα, στην παρουσίαση της προτεινόμενης διδακτικής προσέγγισης περιλαμβάνονται:

- τα προβλήματα (μαθησιακές δυσκολίες) προς επίλυση,
- το διδακτικό περιεχόμενο ή διδακτέα ύλη,
- η εκπαιδευτική φιλοσοφία της με τις αρχές και τις μαθησιακές στρατηγικές υλοποίησής της, βάσει των οποίων έχουν προταθεί συγκεκριμένες δραστηριότητες που λαμβάνουν χώρα με συγκεκριμένη σειρά,
- το μαθησιακό περιβάλλον μέσα στο οποίο αυτή εντάσσεται,
- το μαθησιακό υλικό που σχεδιάστηκε, συνοδεύει και υποστηρίζει την προσέγγιση και η σχεδίαση των πόρων του,
- τα μαθησιακά εργαλεία που χρησιμοποιούνται και οι λόγοι επιλογής τους και
- ο ρόλος όσων εμπλέκονται στη μαθησιακή διαδικασία.

Τα θεμέλια για τη σχεδίαση της προτεινόμενης προσέγγισης, αποτέλεσαν το διδακτικό περιεχόμενο (διδακτέα ύλη), οι μαθησιακές δυσκολίες του γνωστικού αντικειμένου και η εκπαιδευτική φιλοσοφία που αποφασίστηκε να τη διέπει. Δομικές συνιστώσες της αποτελούν το μαθησιακό υλικό, η τεχνολογική της υποδομή και το ανθρώπινο δυναμικό που εμπλέκεται στην εφαρμογή της στη διδακτική πράξη, με τους ρόλους που ο καθένας επιτελεί.

Στο **Κεφάλαιο 4 – “Εφαρμογή και αξιολόγηση της προτεινόμενης διδακτικής προσέγγισης – Μελέτες περίπτωσης (case studies)”**, παρουσιάζονται αναλυτικά τρεις μελέτες περίπτωσης που πραγματοποιήθηκαν για την εφαρμογή της προσέγγισης σε αυθεντικά εκπαιδευτικά περιβάλλοντα στην Τριτοβάθμια εκπαίδευση και αναδεικνύεται η εξελικτική ανάπτυξη της μέσα από κύκλους αξιολόγησης και βελτιστοποίησης. Οι μελέτες αυτές πραγματοποιήθηκαν σε προπτυχιακό επίπεδο στο Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων (ΔΤΨΣ) του Πανεπιστημίου Πειραιά επί δύο συνεχόμενα ακαδημαϊκά έτη και σε μεταπτυχιακό επίπεδο στο Τεχνολογικό Εκπαιδευτικό Ίδρυμα (Τ.Ε.Ι.) Πειραιά. Συγκεκριμένα, αναφέρονται αναλυτικά ο χώρος και ο χρόνος εφαρμογής της κάθε μελέτης

περίπτωσης, οι συνθήκες υλοποίησης της, οι μελέτες αξιολόγησης (evaluation studies) που διεξήχθησαν, τα ευρήματα της κάθε μιας και τα συμπεράσματα που προέκυπταν από την κάθε εφαρμογή και τα οποία αποτελούσαν πολύτιμα στοιχεία για τη βελτίωση και την ενίσχυση της επόμενης, με κατάληξη την τελική μορφή της προτεινόμενης διδακτικής προσέγγισης.

Οι μελέτες περίπτωσης πραγματοποιήθηκαν:

- Σε μακράς διάρκειας σεμινάριο κατά το εαρινό εξάμηνο του 2005 από το Μάρτιο του 2005 έως το Μάιο του 2005, με τη μορφή φροντιστηριακού μαθήματος για τους προπτυχιακούς φοιτητές του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά.
- Με τη μορφή e-learning μαθήματος, ως μέρος του υποχρεωτικού μαθήματος “Αντικειμενοστρεφής Προγραμματισμός” στο Τμήμα ΔΤΨΣ του Πανεπιστημίου Πειραιά, αλλά ανεξάρτητο αυτού και υποστηρικτικό αυτού. Η εφαρμογή αυτή έγινε κατά το εαρινό εξάμηνο του 2006 (Φεβρουάριος 2006 – Ιούνιος 2006).
- Σε σεμιναριακό e-learning μάθημα από το Μάρτιο του 2006 έως το Μάιο του 2006, στο Τ.Ε.Ι. Πειραιά, στο πλαίσιο του Προγράμματος Μεταπτυχιακών Σπουδών (ΠΜΣ) “Information Technology”, που διεξάγεται σε συνεργασία με το Paisley University of Scotland.

Στο κεφάλαιο αυτό, περιλαμβάνονται και στοιχεία για το εννοιολογικό πλαίσιο των μελετών αξιολόγησης που πραγματοποιήθηκαν για κάθε μελέτη περίπτωσης.

Στο **Κεφάλαιο 5 – “Συμπεράσματα”** γίνεται μια επισκόπηση των βημάτων που ακολουθήθηκαν κατά τη διάρκεια της ερευνητικής πορείας, παρατίθεται συνοπτικά το τελικό αποτέλεσμα της πορείας αυτής και τα συμπεράσματα που εξήχθησαν. Αξιολογείται η έρευνα και τα επιτεύγματά της σε σχέση με προηγούμενες έρευνες και αναφέρονται καινοτομικά στοιχεία της. Επισημαίνονται θέματα που χρήζουν περαιτέρω διερεύνησης και μελλοντικά ερευνητικά σχέδια που αφορούν στην ενίσχυση, επέκταση, εφαρμογή και αξιοποίηση της προτεινόμενης διδακτικής προσέγγισης.

Τέλος παρατίθενται οι Βιβλιογραφικές αναφορές, στο Παράρτημα Α η θεματική ανάπτυξη του διδακτικού περιεχομένου της προτεινόμενης διδακτικής προσέγγισης και στα Παραρτήματα Β έως Ζ αντίστοιχα, τα ερωτηματολόγια προ-αξιολόγησης και μετά-αξιολόγησης της κάθε μιας από τις εφαρμογές της προσέγγισης στη διδακτική πράξη. Στο τέλος της διατριβής επισυνάπτεται ψηφιακός δίσκος (CD) που περιλαμβάνει το κυρίως μαθησιακό υλικό που αναπτύχθηκε και το οποίο έχει

ενοποιηθεί σε “πακέτο” σύμφωνα με την προδιαγραφή IMS Content Packaging specification (IMS, 2003).

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑΣ

РАНЕЕ НЕ ПЕРПА

ΚΕΦΑΛΑΙΟ 2

Η Διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού

2.1. Εισαγωγή

Στο κεφάλαιο αυτό γίνεται μια εκτενής βιβλιογραφική επισκόπηση σχετικά με τη διδασκαλία του γνωστικού αντικειμένου του ΑΠ. Βιβλιογραφικά δεδομένα υπάρχουν τόσο από μικρής διάρκειας έρευνες όσο και από μακροχρόνιες και πιο εκτενείς. Από όλες μπορούμε να συνάγουμε συμπεράσματα για τις μαθησιακές δυσκολίες στη διδασκαλία του ΑΠ (αυτές οι δυσκολίες αναφέρονται στις ενότητες 2.2. και 2.3.).

Τα μαθησιακά προβλήματα καθώς και συστάσεις για βελτίωση της διδασκαλίας του ΑΠ προκύπτουν από μικρής διάρκειας έρευνες (ενότητα 2.2.) που σχετίζονται κατά κύριο λόγο με τη διεξαγωγή ενός μαθήματος ΑΠ ή διδασκαλίας της γλώσσας Java του προγράμματος σπουδών κάποιου εκπαιδευτικού ιδρύματος διάρκειας ενός εξαμήνου ή μικρότερης, από έρευνες με τη χρήση ερωτηματολογίου για τη διερεύνηση συγκεκριμένων θεμάτων και από μελέτες της αντίληψης προγραμμάτων από τη μεριά των εκπαιδευομένων.

Οι μακροχρόνιες έρευνες (ενότητα 2.3.) έχουν διεξαχθεί στο πλαίσιο διδακτορικών διατριβών και ερευνητικών έργων (projects) με θέμα τη διδασκαλία του ΑΠ. Σε αυτές περιγράφονται στοιχεία για τη διδακτική προσέγγιση που ακολουθήθηκε, την ερευνητική τους μεθοδολογία για τη μελέτη θεμάτων και τα συμπεράσματά τους. Η παρούσα διατριβή αποτελεί μια συστηματική και μακροχρόνια έρευνα στη διδασκαλία του ΑΠ και μπορεί να συγκριθεί με αυτές τόσο σε επίπεδο διδακτικής όσο και σε επίπεδο ερευνητικής μεθοδολογίας.

Συνεπώς, στο παρόν κεφάλαιο, παρουσιάζονται αρχικά οι δυσκολίες και οι παρανοήσεις των εκπαιδευομένων στον ΑΠ που έχουν καταγραφεί από έρευνες σε διαφορετικές βαθμίδες της εκπαίδευσης. Ακολουθεί η περιγραφή μακροχρόνιων ερευνών που έχουν διεξαχθεί και αφορούν στη διδασκαλία του ΑΠ. Αναφέρονται ο στόχος, η μεθοδολογία και τα ευρήματα τους. Η βιβλιογραφική επισκόπηση συνεχίζει με την παράθεση των κύριων χαρακτηριστικών των διδακτικών προσεγγίσεων που έχουν αναπτυχθεί και προταθεί για τη διδασκαλία του γνωστικού αυτού αντικειμένου.

Στη συνέχεια παρουσιάζονται εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα τα οποία έχουν αναπτυχθεί για την υποστήριξη της διδασκαλίας του ΑΠ. Τέλος αναφέρονται οδηγίες και συστάσεις διδασκαλίας που έχουν διατυπωθεί για τη μαθησιακά αποτελεσματικότερη διδασκαλία του αντικειμένου του ΑΠ, όπως έχουν προκύψει από διάφορες έρευνες και διδακτικά πειράματα. Οι οδηγίες και οι συστάσεις, βοήθησαν στο να σχεδιαστεί πιο συστηματικά η προτεινόμενη διδακτική προσέγγιση.

Αρκετά από τα δεδομένα που παρουσιάζονται σε αυτό το κεφάλαιο έχουν προκύψει και δημοσιευθεί μετά τη σχεδίαση της διδακτικής προσέγγισης της διατριβής και του μαθησιακού υλικού που τη συνοδεύει, καθώς και τις εφαρμογές της στη διδακτική πράξη, ή κατά τη διάρκεια αυτών των φάσεων εκπόνησης της διατριβής.

2.2. Δυσκολίες και παρανοήσεις των εκπαιδευομένων

Στη διεθνή βιβλιογραφία είναι καταγεγραμμένες δυσκολίες και παρανοήσεις που παρουσιάζουν οι εκπαιδευόμενοι κατά την εκμάθηση του ΑΠ, οι οποίες έχουν προκύψει από έρευνες σε μαθητές Γυμνασίου, Λυκείου, προπτυχιακούς και μεταπτυχιακούς σπουδαστές. Στη συνέχεια παρουσιάζονται αυτές οι έρευνες και τα ευρήματά τους. Συγκεκριμένα τα ευρήματα προέρχονται:

1. Από έρευνα που διεξήχθη από το τμήμα “*Department of Education in Technology and Science*” του πανεπιστημίου “*Technion – Israel Institute of Technology*” της πόλης Χαΐφα (*Haifa*) του Ισραήλ. (Teif and Hazzan, 2004), σε δύο ομάδες μαθητών Γυμνασίου, 22 και 25 ατόμων αντίστοιχα, χωρίς προηγούμενη γνώση προγραμματισμού και υπολογιστών.

Οι μαθητές παρακολούθησαν μαθήματα εισαγωγής σε βασικές αντικειμενοστρεφείς έννοιες (κλάσεις, αντικείμενα, ιδιοχαρακτηριστικά, μέθοδοι και αμοιβαίες σχέσεις μεταξύ αυτών των εννοιών) για ένα εξάμηνο, με σύντομες διαλέξεις και εργαστηριακές ασκήσεις με τη χρήση ενός γραφικού-οπτικού περιβάλλοντος “*χελώνας*” (μικρόκοσμος) που σχεδιάστηκε και αναπτύχθηκε ειδικά για τις ανάγκες της έρευνας. Η συλλογή εμπειρικών δεδομένων έγινε μέσω ερωτηματολογίων, φύλλων εργασίας (*worksheets*), συνεντεύξεων και

παρατήρησης στην τάξη. Τέσσερα κύρια αποτελέσματα έχουν προκύψει από την επεξεργασία αυτών των στοιχείων:

- Οι μαθητές συγγέουν τα ιδιοχαρακτηριστικά (attributes) ενός αντικειμένου με τα μέρη (parts) από τα οποία αυτό αποτελείται
- και επίσης τα συγγέουν με τις ενέργειες που μπορεί να εκτελέσει (actions it performs).
- Θεωρούν ότι ένα σύνολο από αντικείμενα (set of objects) π.χ. μια συγκεκριμένη ομάδα ποδοσφαιρικού είναι κλάση και όχι αντικείμενο και
- θεωρούν τις σχέσεις ιεραρχίας (hierarchy) και όλου – μέρους (whole – part), σχέσεις κλάσης – αντικειμένου (class – object).

2. Από έρευνα η οποία διεξήχθη επί δύο έτη στο τμήμα Διδασκαλίας των Επιστημών (Department of Science Teaching) στο Επιστημονικό Ινστιτούτο Βάιζμαν (Weizmann Institute of Science) της πόλης Ρεχόβοτ του Ισραήλ, (Ragonis and Ben-Ari, 2002). Η έρευνα αφορούσε σε μαθητές Λυκείου, ηλικίας 15-16 ετών (18 μαθητές την πρώτη χρονιά (2000-01) και 29 τη δεύτερη (2001-02)), χωρίς προηγούμενη γνώση υπολογιστών που παρακολούθησαν εισαγωγικά μαθήματα ΑΠ με τη χρήση της γλώσσας Java και του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ (το περιβάλλον αυτό περιγράφεται στην ενότητα 2.5.) για τις εργαστηριακές ασκήσεις.

Αναδείχθηκαν δυσκολίες στην αντίληψη της δημιουργίας στιγμιότυπων (instantiation) με τη μέθοδο του κατασκευαστή (constructor) και τους διαφορετικούς τρόπους αρχικοποίησης στιγμιότυπων (initialization). Η συλλογή εμπειρικών δεδομένων για την κατανόηση από τη μεριά των μαθητών, των δυσκολιών και των παρανοήσεών τους, έγινε μέσω φύλλων εργασιών (worksheets) και εργασιών που παρέδιδαν (homework), συνεντεύξεων των μαθητών που παρέδωσαν τις εργασίες και παρατήρησης στην τάξη κατά τη διάρκεια όλων των μαθημάτων. Συγκεκριμένα προέκυψαν τα εξής:

- Η δυσκολία των μαθητών στην κατανόηση της μεθόδου του κατασκευαστή γενικά και της θέσης του μέσα στην κλάση, αλλά και ιδιαίτερα στις περιπτώσεις,
 - ◆ δημιουργίας αντικειμένων με κλήση του εξ' ορισμού κατασκευαστή,
 - ◆ αρχικοποίησης των τιμών των ιδιοχαρακτηριστικών με τη χρήση τιμών κατά τη δήλωσή τους (έτσι όλα τα αντικείμενα έχουν ίδιες τιμές στα ιδιοχαρακτηριστικά τους και αυτό είναι περίπτωση που δε συνάδει με την

πραγματικότητα και κάνει τους μαθητές να διερωτώνται “όλα τα αντικείμενα έχουν ίδιες τιμές στα ιδιοχαρακτηριστικά τους;”),

- ◆ αρχικοποίησης των τιμών των ιδιοχαρακτηριστικών με τη χρήση σταθερών τιμών στον κατασκευαστή (έτσι όλα τα αντικείμενα έχουν ίδιες τιμές στα ιδιοχαρακτηριστικά τους και αυτό είναι ίδια περίπτωση με την παραπάνω που δυσκολεύει για τους ίδιους λόγους τους μαθητές).
 - Επίσης σε κάποιο βαθμό οι μαθητές θεωρούν πως μόνον και μόνον η δήλωση του κατασκευαστή μέσα στην κλάση, αρκεί για να δημιουργηθεί ένα αντικείμενο και δυσκολεύονται να κατανοήσουν τη χρήση του τελεστή new και τις λειτουργίες που αυτή συνεπάγεται (δηλαδή την κατανομή μνήμης και την κλήση του κατασκευαστή).
 - Ένα άλλο πρόβλημα είναι ότι όταν έχουν αρχικοποιηθεί οι τιμές των ιδιοχαρακτηριστικών με τη χρήση τιμών κατά τη δήλωση τους μέσα στην κλάση, νομίζουν ότι δε χρειάζεται να δημιουργηθεί αντικείμενο, αλλά ότι ήδη υπάρχει.
 - Ο μηχανισμός χρήσης παραμέτρων στη μέθοδο του κατασκευαστή δυσκολεύει την κατανόηση από τους αρχάριους μαθητές.
 - Στην περίπτωση κλάσης που διαθέτει ιδιοχαρακτηριστικό/α του τύπου μιας άλλης κλάσης (συντεθειμένη κλάση, composed class), γενικά δυσκολεύονται να κατανοήσουν ότι ο τύπος ενός ιδιοχαρακτηριστικού μπορεί να είναι μια άλλη κλάση.
 - Στην περίπτωση των συντεθειμένων κλάσεων, θεωρούν ότι αν έχουν δημιουργηθεί τα συστατικά αντικείμενα, δε χρειάζεται να δημιουργηθεί το αντικείμενο της συντεθειμένης κλάσης.
3. Από την ίδια έρευνα, η οποία ανέδειξε προβλήματα στην κατανόηση της συνολικής (σφαιρικής) ροής – εκτέλεσης προγραμμάτων (program flow) (τη συνολική εικόνα της εκτέλεσης του προγράμματος) (Ragonis and Ben-Ari, 2005a), όταν δοθεί ιδιαίτερη έμφαση μόνο στις αντικειμενοστρεφείς έννοιες και το περιβάλλον BlueJ χρησιμοποιηθεί αποκλειστικά τονίζοντας και αξιοποιώντας τα εκπαιδευτικά χαρακτηριστικά του (τη δυνατότητα δημιουργίας αντικειμένων αλληλεπιδραστικά από το εικονίδιο μιας κλάσης και κλήσης μεθόδου αλληλεπιδραστικά από το εικονίδιο ενός αντικειμένου). Με αυτό τον τρόπο βέβαια, επιτυγχάνεται η κατανόηση των αντικειμενοστρεφών εννοιών, αλλά δε γίνεται εκμετάλλευση της ιδιότητας του προγραμματιστικού περιβάλλοντος να

αποτελεί ένα πλήρες περιβάλλον ανάπτυξης εφαρμογών σε Java, γεγονός που οδηγεί στα προβλήματα κατανόησης της συνολικής εικόνας της εκτέλεσης του προγράμματος. Στην έρευνα, διατυπώνεται η άποψη ότι η διαπίστωση αυτή θα πρέπει να παρατηρείται και σε άλλες περιπτώσεις και δεν αποτελεί χαρακτηριστικό του περιβάλλοντος BlueJ. Οι μαθητές αδυνατούν να καταλάβουν τη σχέση μεταξύ του κώδικα Java, του περιβάλλοντος BlueJ και της ροής του προγράμματος.

Βασική αιτία για τα παραπάνω προβλήματα είναι η έλλειψη δυναμικής οπτικοποίησης της ροής των προγραμμάτων που στον ΑΠ είναι αρκετά περίπλοκη (το BlueJ δίνει έμφαση στην οπτικοποίηση αντικειμενοστρεφών εννοιών και την οπτικοποίηση της στατικής δομής των διαγραμμάτων κλάσεων, αλλά δεν υπάρχει δυναμική οπτικοποίηση της εκτέλεσης του προγράμματος) (Ben-Ari et al., 2002).

Ακόμα, η δυσκολία που δημιουργεί η διαμοιρασμένη φύση της ροής ελέγχου σε ένα αντικειμενοστρεφές πρόγραμμα (υπάρχουν παραπάνω από μια κλάσεις και επίσης κλάσεις-προγράμματα με μέθοδο `main` απαραίτητα για την εκτέλεση) αναφέρεται και στα συμπεράσματα έρευνας των Wiedenbeck και Ramalingam (Wiedenbeck and Ramalingam, 1999; Wiedenbeck et al., 1999), η οποία αφορούσε στην κατανόηση από τη μεριά των σπουδαστών μικρής έκτασης διαδικασιακών (procedural) και αντικειμενοστρεφών προγραμμάτων.

4. Από μια πρόσφατη δημοσίευση, που αφορά επίσης στην ίδια έρευνα, στην οποία αναφέρονται αρκετές διαπιστωμένες δυσκολίες και παρανοήσεις που έχουν καταταχθεί σε κατηγορίες και υποκατηγορίες που αφορούν στις αντικειμενοστρεφείς έννοιες αυτές καθαυτές, σε θέματα που επηρεάζουν τη συνολική κατανόηση και αντίληψη του αντικειμένου (τομέα) και σε θέματα που αφορούν στη συνολική αντίληψη της ροής αντικειμενοστρεφών προγραμμάτων (Ragonis and Ben-Ari, 2005b). Εκτενέστερη αναφορά σε αυτή τη δημοσίευση γίνεται σε επόμενη ενότητα αυτού του κεφαλαίου (ενότητα 2.3.).
5. Από ημι-δομημένες συνεντεύξεις πρωτοετών προπτυχιακών σπουδαστών στο Πανεπιστήμιο της κομητείας Κέντ στην πόλη Καντέρμπουρι της Μ. Βρετανίας (*University of Kent at Canterbury*), που διδάσκονταν ΑΠ με τη χρήση της γλώσσας Java (Carter and Fowler, 1998).

Στην παραπάνω δημοσίευση, αναφέρονται δυσκολίες από το πρώτο τμήμα του μαθήματος μόνον, οι οποίες είχαν τη μεγαλύτερη επίδραση στους σπουδαστές και αφορούν:

- Στη σύγχυση που προκαλούσε η χρήση πολλαπλών κατασκευαστών (multiple constructors) και την ανάγκη να υποστηριχθεί αυτή η χρήση αναφέροντας συγκεκριμένα παραδείγματα από τον πραγματικό κόσμο (real world examples).
- Στη συγγραφή προγραμμάτων με περισσότερες από μια κλάσεις. Συγκεκριμένα οι σπουδαστές συνέχεαν εάν πρέπει να τοποθετήσουν όλες τις κλάσεις στο ίδιο αρχείο, ή σε διαφορετικά.

Στην έρευνα αυτή αναφέρεται ότι δεν υπήρξε πρόβλημα στη διάκριση των εννοιών κλάση και αντικείμενο, κάτι που παρατηρήθηκε στην αμέσως παρακάτω έρευνα.

6. Από παρατήρηση των σπουδαστών στο πλαίσιο του εξ' αποστάσεως μεταπτυχιακού μαθήματος "Object-oriented Software Technology (M868)" και κατά τη διάρκεια του επίσης εξ' αποστάσεως εισαγωγικού προπτυχιακού μαθήματος "Computing: An Object-oriented approach (M206)" και τα δύο με τη χρήση της γλώσσας Smalltalk ως πρώτη γλώσσα προγραμματισμού. Τα μαθήματα διεξάγονταν στο τμήμα "Faculty of Mathematics and Computing" του Ανοικτού Πανεπιστημίου (The Open University) της Μ. Βρετανίας (Holland et al., 1997).

Από την έρευνα αυτή αναδείχθηκαν παρανοήσεις που οι σπουδαστές αναπτύσσουν, όπως:

- Ταύτιση (συνδυασμός/συγκερασμός) αντικειμένου/μεταβλητής και θεώρηση του αντικειμένου σαν ένα "περιτύλιγμα" μιας μεταβλητής, αν στα παραδείγματα που χρησιμοποιούνται, αναφέρονται συχνά κλάσεις με μια μόνο μεταβλητή στιγμιοτύπου (instance variable) π.χ. κλάση Account με μεταβλητή στιγμιοτύπου balance. Στα παραδείγματα των κλάσεων θα πρέπει για την αποφυγή αυτής της παρανόησης να υπάρχουν τουλάχιστον δύο μεταβλητές στιγμιοτύπου.
- Λανθασμένη θεώρηση ότι όλες οι μεταβλητές στιγμιοτύπου μιας κλάσης είναι του ίδιου τύπου (objects of the same class) π.χ. αριθμοί, αν χρησιμοποιούνται συνεχώς τέτοια παραδείγματα π.χ. κλάση Account με μεταβλητές στιγμιοτύπου τις balance και limit που και οι δύο είναι αριθμητικού τύπου αντικείμενα. Στα παραδείγματα των κλάσεων θα πρέπει για την αποφυγή

αυτής της παρανόησης να υπάρχουν τουλάχιστον δύο μεταβλητές στιγμιοτύπου που θα αποτελούν αντικείμενα διαφορετικών κλάσεων δηλαδή θα είναι διαφορετικού τύπου π.χ. κλάση Account με μεταβλητές στιγμιοτύπου τις balance και name.

- Οι σπουδαστές σχηματίζουν τη λανθασμένη αντίληψη ότι τα αντικείμενα είναι απλές εγγραφές (όπως οι εγγραφές μιας βάσης δεδομένων (database records)), χωρίς συμπεριφορά, όταν υπερτονίζεται το κομμάτι των μεταβλητών στιγμιοτύπου μιας κλάσης εις βάρος της συμπεριφοράς των αντικειμένων της (methods) π.χ. μια κλάση CDs που τα αντικείμενα της αναπαριστούν μουσικά CD και περιέχουν πληροφορίες για τον τίτλο, τον καλλιτέχνη, τα τραγούδια που περιέχονται σε αυτά κλπ. Αυτό μπορεί να αποφευχθεί αν χρησιμοποιηθούν παραδείγματα στα οποία η συμπεριφορά των αντικειμένων μιας κλάσης μεταβάλλεται σημαντικά, ανάλογα με την κατάστασή τους π.χ. ένα αντικείμενο Account το οποίο “αρνείται” να κάνει μια ανάληψη αν δεν υπάρχει επαρκές υπόλοιπο, ενώ η ανάληψη πραγματοποιείται κανονικά αν δεν υπάρχει πρόβλημα υπολοίπου.
- Ταύτιση (συνδυασμός/συγκερασμός) κλάσης/αντικειμένου, ιδίως όταν χρησιμοποιούνται συχνά παραδείγματα κλάσεων από τις οποίες δημιουργείται μόνο ένα αντικείμενο. Γι’ αυτό θα πρέπει πάντα στα παραδείγματα που δίνονται να δουλεύουμε με αρκετά αντικείμενα από κάθε κλάση.
- Σύγχυση ταυτότητας/ιδιοχαρακτηριστικού που καλείται name ή κάπως παρόμοια. Π.χ. για το παράδειγμα της κλάσης Account, ιδιοχαρακτηριστικά name και balance. Το name συγγέεται με την ταυτότητα του αντικειμένου ή τη μεταβλητή αναφοράς (reference type) (π.χ. myAccount) η οποία “συνδέεται” με το αντικείμενο και αντικείμενα που έχουν την ίδια τιμή σε αυτή τη μεταβλητή θεωρούνται από τους σπουδαστές ως το ίδιο αντικείμενο. Αυτές οι συγχύσεις οδηγούν και σε άλλες παρανοήσεις που έχουν να κάνουν με τη σύνδεση περισσότερων της μιας μεταβλητής αναφοράς στο ίδιο αντικείμενο, τη δυνατότητα αποσύνδεσης μιας μεταβλητής αναφοράς από ένα αντικείμενο και σύνδεσης της με άλλο και τη διάκριση αντικειμένων με ακριβώς ίδια κατάσταση (state). Συγκεκριμένα δημιουργούνται οι εξής παρανοήσεις:
 - ◆ Μόνο μια μεταβλητή αναφοράς μπορεί να “συνδέεται” με ένα συγκεκριμένο αντικείμενο μια συγκεκριμένη στιγμή.
 - ◆ Αν μια μεταβλητή αναφοράς “συνδεθεί” με ένα αντικείμενο, θα

παραμένει για πάντα “συνδεδεμένη” με αυτό.

- ◆ Μια μεταβλητή αναφοράς “συνδεδεμένη” με ένα αντικείμενο “μοναδικά”, το καθορίζει για πάντα.
- ◆ Δύο διαφορετικές μεταβλητές αναφοράς πρέπει να “συνδέονται” σε δύο διαφορετικά αντικείμενα.
- ◆ Μπορείς να “μάθεις” από το αντικείμενο, ποιες μεταβλητές αναφοράς “δείχνουν σε” αυτό;
- ◆ Δύο αντικείμενα της ίδιας κλάσης με την ίδια κατάσταση είναι το ίδιο αντικείμενο.
- ◆ Δύο αντικείμενα που έχουν την ίδια τιμή στο ιδιοχαρακτηριστικό name είναι το ίδιο αντικείμενο.

Για την αποφυγή των παραπάνω παρανοήσεων, *συστήνεται* οι σπουδαστές να πειραματίζονται με ένα σύνολο παραδειγμάτων όπως:

- ◆ *Πολλαπλές αναθέσεις* μεταβλητών αναφοράς στο ίδιο αντικείμενο, αλλαγή της κατάστασης του αντικειμένου μέσω μιας από αυτές και έλεγχος της αλλαγμένης κατάστασης μέσω των άλλων.
- ◆ *Ανάθεση εκ νέου* άλλου αντικειμένου σε μία από τις μεταβλητές αναφοράς και έλεγχος της αλλαγής της κατάστασης του αντικειμένου μετά από αποστολή μηνύματος και συγχρόνως έλεγχος της κατάστασης του αντικειμένου στο οποίο “συνδέονται” οι άλλες μεταβλητές αναφοράς.
- ◆ *Αντιμετάθεση* των μεταβλητών αναφοράς που “συνδέονται” σε δύο αντικείμενα με τη χρήση ενδιάμεσης μεταβλητής.
- ◆ *Μεταβλητές στιγμιότυπου με την ίδια τιμή* σε δύο φανερώς διαφορετικά στιγμιότυπα.
- ◆ *Παραδείγματα που αποδεικνύουν ότι δύο στιγμιότυπα με ίδια κατάσταση* δεν είναι το ίδιο αντικείμενο, αποστέλλοντας μηνύματα που κάνουν διαφορετικές μεταξύ τους τις καταστάσεις τους.
- Θεωρούν ότι μια μέθοδος μεταβάλλει την κατάσταση ενός αντικειμένου απλά και μόνο με εντολές ανάθεσης και όχι με αποστολή μηνυμάτων. Δηλαδή, δημιουργείται η λανθασμένη εντύπωση ότι στις μεθόδους όλη η “δουλειά” γίνεται με εντολές ανάθεσης και όχι με αποστολή μηνυμάτων. Αυτή η παρανόηση δημιουργείται όταν χρησιμοποιούνται παραδείγματα όπως π.χ. το αντικείμενο Account που διαθέτει μεταβλητές στιγμιότυπου π.χ. balance αριθμητικού τύπου και έτσι οι μέθοδοί του που χειρίζονται αυτές τις

μεταβλητές χρησιμοποιούν ανάθεση παρά αποστολή μηνυμάτων (κλήση μεθόδων). Σχεδόν τα περισσότερα αρχικά παραδείγματα στη διδασκαλία επιλέγονται ώστε όλη η κατάσταση των αντικειμένων να παριστάνεται από αμετάβλητα αντικείμενα όπως οι αριθμοί. Για να αποφευχθεί η παρανόηση, θα πρέπει να γίνεται χρήση παραδειγμάτων στα οποία οι τιμές των μεταβλητών στιγμιοτύπου δεν είναι αμετάβλητα αντικείμενα.

7. Από ηχογραφημένες συνεντεύξεις αρχαρίων σπουδαστών που παρακολουθούσαν εισαγωγικά μαθήματα ΑΠ με τη χρήση της γλώσσας Java, στο Πανεπιστήμιο *Aurora* (*Aurora University*) στο Κολοράντο. Συμμετείχαν 28 εθελοντές σπουδαστές χωρίς προηγούμενη εμπειρία στον ΑΠ (Fleury, 2000; Fleury, 2001). Από την πρώτη εκ των δύο αυτών ερευνών (Fleury, 2000), αναδείχθηκαν κανόνες που οι σπουδαστές λανθασμένα “οικοδομούν” όταν την αποκτημένη γνώση την επεκτείνουν και πέραν της ορθής περιοχής εφαρμογής της, όπως:
 - Ο compiler της Java μπορεί να διαχωρίσει μεθόδους με ίδιο όνομα, μόνον αν αυτές έχουν διαφορές στη λίστα των παραμέτρων τους. Κάτι απολύτως σωστό για μεθόδους που ανήκουν στη ίδια κλάση. Αλλά οι φοιτητές το επέκτειναν και για μεθόδους διαφορετικών κλάσεων και έτσι θεωρούν ότι: μέθοδοι διαφορετικών κλάσεων μπορούν να έχουν το ίδιο όνομα μόνο αν έχουν διαφορετικές υπογραφές.
 - Θεωρούν τη χρήση του κατασκευαστή προαιρετική και νομίζουν ότι ο μόνος λόγος χρήσης του είναι η αρχικοποίηση των μεταβλητών του αντικειμένου. Αν καταργήσεις τον κατασκευαστή μιας κλάσης και τον αντικαταστήσεις με μια μέθοδο που αρχικοποιεί τις μεταβλητές, θεωρούν ότι δεν υπάρχει πρόβλημα. Δεν αντιλαμβάνονται δηλαδή την κατανομή μνήμης (memory space allocation) για τα αντικείμενα, που συμβαίνει με την κλήση του κατασκευαστή, αλλά μόνον το κομμάτι της αρχικοποίησης που επίσης συμβαίνει με την ίδια κλήση. Δηλαδή οι σπουδαστές αντιλαμβάνονται αυτό που εύκολα είναι παρατηρήσιμο και όχι το δύσκολο παρατηρήσιμο.
 - Αντιλαμβάνονται τη χρήση αριθμητικών μεταβλητών και αριθμητικών σταθερών σαν actual παραμέτρους μεθόδων με ακέραιες formal παραμέτρους, αλλά δεν αντιλαμβάνονται τη χρήση formal παραμέτρων σαν actual παραμέτρους, γιατί είναι πολύ εξοικειωμένοι με τους αριθμούς επί χρόνια.
 - Θεωρούν ότι η χρήση του τελεστή “τελεία” (dot operator) είναι έγκυρη μόνο για την κλήση μεθόδων και όχι για την αναφορά σε μεταβλητές στιγμιοτύπου,

επειδή αυτές είναι συνήθως ιδιωτικές (private) και ο τελεστής “τελεία” δε χρησιμοποιείται σε αυτές.

8. Από τη δεύτερη εκ των δύο προηγούμενων έρευνα (Fleury, 2001), η οποία ανέδειξε προβλήματα των σπουδαστών σχετικά με την κατανόηση της αρχής της κελυφοποίησης (encapsulation) και το σχεδιασμό κώδικα που μπορεί να επαναχρησιμοποιηθεί (code re-use), όπως:

- Ο σχεδιασμός μιας κλάσης με δημόσια (public) ιδιοχαρακτηριστικά αντί ιδιωτικά (private) (κάτι που έρχεται σε αντίθεση με την αρχή της κελυφοποίησης και που είχε σε συγκεκριμένο παράδειγμα κλάσης ως αποτέλεσμα να απομακρυνθεί μια μέθοδος που δε χρειαζόταν πια), θεωρήθηκε προτιμητέα από μερικούς σπουδαστές λόγω της μικρότερης έκτασης του κώδικα και της εξάλειψης της κλήσης μεθόδων, παρότι είχαν κατακτήσει τη γνώση ότι είναι ανεπιθύμητο να έχεις δημόσια (public) ιδιοχαρακτηριστικά σε μια κλάση.
- Η κατάργηση μιας κλάσης σε ένα πρόγραμμα και η μεταφορά όλης της “δουλειάς της” στη μέθοδο main (που μειώνει την κελυφοποίηση και την πιθανότητα επαναχρησιμοποίησης κώδικα σε άλλες εφαρμογές), θεωρήθηκε από κάποιους σπουδαστές λόγος για ευκολότερη κατανόησή του, αφού δε χρειάζεται να “μεταπηδάς” από κλάση σε κλάση όταν το διαβάζεις και αφού έχεις να κάνεις με λιγότερες κλάσεις, άρα έχεις να “καταπιαστείς” και να αντεπεξέλθεις με λιγότερα.
- Η προσθήκη παραμέτρων σε μεθόδους κατασκευαστών που πριν δεν είχαν, πράγμα που παρείχε αυξημένη ευελιξία για να χρησιμοποιηθούν οι κλάσεις σε άλλες εφαρμογές, θεωρήθηκε από κάποιους φοιτητές χειρότερη του προγράμματος γιατί τους είναι δυσκολότερο να το διαβάσουν και επομένως να το διορθώσουν και να το ελέγξουν (debug and test). Επιπλέον εκτίμησαν ότι οι αλλαγές έκαναν το πρόγραμμα δυσκολότερο στη συντήρηση και τις κλάσεις του δυσκολότερο να επαναχρησιμοποιηθούν γιατί θεωρούν ότι συνδέθηκαν στενότερα μεταξύ τους.
- Η δημιουργία δύο παρόμοιων κλάσεων μέσα σε ένα πρόγραμμα και η χρήση ενός αντικειμένου από την κάθε μια, θεωρήθηκε προτιμότερη από την πιο γενική περίπτωση της ύπαρξης μιας κλάσης και τη δημιουργία δύο αντικειμένων από αυτή, γιατί λανθασμένα εκτιμούν ότι ο κώδικας είναι σαφέστερος, πιο τμηματοποιημένος (modularity), πιο εύκολος στη συντήρηση

και στην ανίχνευση λαθών. Επίσης είπαν ότι η συντήρηση και η μετατροπή του θα είναι πιο εύκολες, γιατί θα πρέπει να αλλάξει μόνο μια κλάση και επίσης ότι η επαναχρησιμοποίηση θα είναι ευκολότερη γιατί μπορεί να χρησιμοποιηθεί μόνο η μια ή η άλλη κλάση. Δηλαδή προτίμησαν την ιδιαιτερότητα των διπλών κλάσεων από τη γενικότητα της επαναχρησιμοποίησης κλάσεων (class re-use).

Συνολικά από τη δεύτερη έρευνα (Fleury, 2001) μπορεί να συναχθεί ότι:

- ◆ Η μείωση των γραμμών κώδικα και του αριθμού των κλάσεων θεωρήθηκαν μεγαλύτερης σημασίας από την κελυφοποίηση.
- ◆ Κάποιοι ενοχλούνται από τη “μεταπήδηση” από κλάση σε κλάση όταν διαβάζουν προγράμματα με πολλές κλάσεις.
- ◆ Οι μέθοδοι που διαθέτουν παραμέτρους είναι δυσκολότερο να κατανοηθούν από αυτές χωρίς παραμέτρους.
- ◆ Η επανάληψη κώδικα είναι ευκολότερο να κατανοηθεί από ότι η επαναχρησιμοποίηση κλάσεων.
- ◆ Συνολικά οι σπουδαστές είναι πιο ευαίσθητοι στο τι κατανοούν άμεσα ευκολότερα παρά στα μακροπρόθεσμα θέματα της συντήρησης και της επαναχρησιμοποίησης.

9. Από έρευνα που διεξήχθη με τη χρήση ηλεκτρονικού ερωτηματολογίου που συμπλήρωσαν και υπέβαλλαν δευτεροετείς φοιτητές του πανεπιστημίου της πόλης *Dunbee της Μ. Βρετανίας (United Kingdom)* και μέλη του δικτύου *τριτοβάθμιας εκπαίδευσης LTSN (Learning and Teaching Support Network, United Kingdom higher-education network)* (δηλαδή διδάσκοντες πανεπιστημίων από όλη τη Μεγάλη Βρετανία). Η έρευνα αφορούσε στη δυσκολία αντιμετώπισης διαφόρων εννοιών και θεμάτων του ΑΠ σε εισαγωγικά μαθήματα κυρίως με τη χρήση της γλώσσας προγραμματισμού C++ και συμμετείχαν σε αυτή συνολικά 66 σπουδαστές και διδάσκοντες. Δυσκολότερα αναδείχθηκαν τα ζητήματα που σχετίζονται με το τι συμβαίνει στη μνήμη (memory-related) καθώς εκτελούνται τα προγράμματα (Milne and Rowe, 2002). Αυτό γιατί οι σπουδαστές δεν είναι ικανοί να κατανοήσουν τι συμβαίνει στη μνήμη, αφού δεν μπορούν να δημιουργήσουν ένα σαφές νοητικό μοντέλο (mental model) της εκτέλεσης του προγράμματός τους. Η διαπίστωση αυτή αναδεικνύει την ανάγκη οπτικοποίησης του τι συμβαίνει στη μνήμη –τι αποθηκεύεται, πώς τα αντικείμενα σχετίζονται το ένα με το άλλο– καθώς εκτελείται ένα πρόγραμμα, με το σχεδιασμό ενός

εργαλείου οπτικοποίησης του προγράμματος. Θα πρέπει να αναφερθεί ότι η παραπάνω έρευνα εστιάζει κυρίως σε έννοιες της συγκεκριμένης γλώσσας προγραμματισμού (C++).

Παρόμοια ευρήματα παρουσιάζονται και σε έρευνα που διεξήχθη με τη χρήση ηλεκτρονικού ερωτηματολογίου στην οποία συμμετείχαν 559 σπουδαστές και 34 διδάσκοντες από 6 πανεπιστήμια 5 Ευρωπαϊκών χωρών (Γερμανία, Ισλανδία, Φινλανδία, Ρουμανία, Λετονία) (Lahtinen et al., 2005). Η έρευνα αυτή αφορούσε σε δυσκολίες στην εκμάθηση και τη διδασκαλία του Προγραμματισμού γενικά. Συγκεκριμένα, ανάμεσα στις δυσκολότερες έννοιες αναδείχθηκαν οι “δείκτες” (pointers), οι αναφορές (references), οι αφηρημένοι τύποι δεδομένων (abstract data types). Δηλαδή έννοιες αφηρημένες και σχετιζόμενες με την κατάσταση στη μνήμη του υπολογιστή.

Παρόμοιο αποτέλεσμα, αναφέρεται και στα συμπεράσματα έρευνας των Wiedenbeck και Ramalingam (Wiedenbeck and Ramalingam, 1999; Wiedenbeck et al., 1999), η οποία αφορούσε στην κατανόηση από τη μεριά των σπουδαστών μικρής έκτασης διαδικασιακών (procedural) και αντικειμενοστρεφών προγραμμάτων. Συγκεκριμένα οι “κρυφές” λειτουργίες –όπως η κλήση του κατασκευαστή και του καταστροφέα (destructor)– δημιούργησε μεγαλύτερη δυσκολία στους αρχάριους στο να δημιουργήσουν μια νοητική αναπαράσταση (mental representation) ενός αντικειμενοστρεφούς προγράμματος παρά ενός αντίστοιχου διαδικασιακού.

Συνοψίζοντας τα ευρήματα των παραπάνω ερευνών, προκύπτουν οι εξής κατηγορίες δυσκολιών και παρανοήσεων που έχουν διαπιστωθεί κατά τη διδασκαλία του ΑΠ:

- Ορθός προσδιορισμός των ιδιοχαρακτηριστικών των κλάσεων.
- Συγχύσεις σχέσεων.
- Συγχύσεις/ταυτίσεις εννοιών.
- Λανθασμένες θεωρήσεις για τον τύπο των ιδιοχαρακτηριστικών μιας κλάσης, τη φύση των αντικειμένων, για τις μεθόδους και πώς μεταβάλλουν την κατάσταση των αντικειμένων.
- Σύγχυση ενός ιδιοχαρακτηριστικού με ιδιαίτερο όνομα (π.χ. name) με την ταυτότητα του αντικειμένου και την αναφορά (reference) η οποία “συνδέεται” με το αντικείμενο.
- Δημιουργία στιγμιότυπων – Μέθοδος κατασκευαστή και λειτουργία του.

- Σύγχυση με τη χρήση πολλαπλών κατασκευαστών (multiple constructors).
- Αρχικοποίηση των ιδιοχαρακτηριστικών των στιγμιοτύπων.
- Κατανόηση των συντεθειμένων κλάσεων (composed class).
- Κατανόηση της δημιουργίας στιγμιοτύπων συντεθειμένων κλάσεων.
- Κατανόηση της συνολικής ροής – εκτέλεσης προγραμμάτων (program flow).
- Δημιουργία προγραμμάτων με παραπάνω από μία κλάσεις.
- Επέκταση της γνώσης πέραν της ορθής περιοχής εφαρμογής της και ως εκ τούτου “οικοδόμηση” λανθασμένων κανόνων.
- Αντίληψη και εφαρμογή στην πράξη της αρχής της κελυφοποίησης (encapsulation).
- Δημιουργία νοητικών μοντέλων (mental models) για τα ζητήματα που σχετίζονται με το τι συμβαίνει στη μνήμη (memory-related) καθώς εκτελούνται τα προγράμματα.

Αξιοσημείωτο είναι να αναφερθεί, ότι δυσκολίες και παρανοήσεις που διαπιστώνονται σε κάποια/ες από τις παραπάνω έρευνες μπορεί να μην παρατηρούνται σε άλλες. Για παράδειγμα στην 3^η κατά σειρά αναφοράς έρευνα (πρωτοετών προπτυχιακών σπουδαστών στο Πανεπιστήμιο της κομητείας Κέντ στην πόλη Καντέρμπουρι της Μ. Βρετανίας (*University of Kent at Canterbury*), δεν υπήρξε πρόβλημα στη διάκριση των εννοιών κλάση και αντικείμενο, κάτι που παρατηρήθηκε στην αμέσως επόμενη έρευνα. Αυτό μπορεί να αποδοθεί στην ιδιαιτερότητα της οργάνωσης του κάθε μαθήματος, του μαθησιακού υλικού του, του τρόπου διεξαγωγής του, των διδασκόντων του κ.ά., δυστυχώς στοιχεία τα οποία δεν είναι διαθέσιμα για να σχολιασθούν.

2.3. Μακροχρόνιες ερευνητικές μελέτες για τη διδασκαλία του Αντικειμενοστρεφούς Προγραμματισμού

Κάποιες από τις έρευνες που αφορούν στον ΑΠ και αναφέρθηκαν προηγουμένως (ενότητα 2.2.), σχετίζονται με τα προβλήματα κατά τη διδασκαλία του που αναδείχθηκαν μέσω παρατήρησης ή συνεντεύξεων και με συστάσεις που έχουν διατυπωθεί για τη βελτίωση της διδασκαλίας του (Teif and Hazzan, 2004; Holland et al., 1997; Carter and Fowler, 1998; Fleury, 2000; Fleury, 2001). Άλλες, αφορούν στη διεξαγωγή ερευνών μέσω κάποιου ερωτηματολογίου για τη διερεύνηση

συγκεκριμένων θεμάτων (Milne and Rowe, 2002; Lahtinen et al., 2005) και άλλες στη μελέτη της αντίληψης προγραμμάτων (Wiedenbeck and Ramalingam, 1999; Wiedenbeck et al., 1999).

Στην ενότητα αυτή αναφέρονται έρευνες μακράς διάρκειας για τη διδασκαλία του ΑΠ. Η βιβλιογραφική μελέτη που πραγματοποιήθηκε οδήγησε στον εντοπισμό δύο διδακτορικών διατριβών που αφορούν στη διδασκαλία του γνωστικού αντικειμένου του ΑΠ, μιας διδακτορικής διατριβής που αφορά στη δημιουργία του προγραμματιστικού μικρόκοσμου objectKarel και στο σημαντικό μακροχρόνιο ερευνητικό έργο “Comprehensive Object Oriented Learning” (COOL project) που πραγματοποιήθηκε από τον Οκτώβριο του 2002 έως το 2005 στο Πανεπιστήμιο του Όσλο στη Νορβηγία και τα αποτελέσματά του δημοσιεύθηκαν στο Fjuk et al. (2006). Στη συνέχεια αυτής της ενότητας, αναφερόμαστε στις παραπάνω μακροχρόνιες ερευνητικές προσπάθειες.

➤ Στο τμήμα Διδασκαλίας των Επιστημών (*Department of Science Teaching*) στο Επιστημονικό Ινστιτούτο Βάιζμαν (*Weizmann Institute of Science*) της πόλης Ρεχόβοτ (*Rehovot*) του Ισραήλ, πραγματοποιήθηκε στο πλαίσιο διδακτορικής διατριβής (Ragonis, 2004), έρευνα για τη διδασκαλία του ΑΠ με τη χρήση της Java σε αρχάριους μαθητές της δευτεροβάθμιας εκπαίδευσης (High school students). Η επιδίωξη ήταν να ακολουθηθεί η “objects first” (OF) προσέγγιση στους αρχάριους (βλ. υποενότητα 2.4.1.3.) και να διερευνηθεί σε βάθος η μάθησή τους. Στην προσέγγιση αυτή η έμφαση στη διδασκαλία δίνεται πρωτίστως στη βασική για την αντικειμενοστρεφή φιλοσοφία προγραμματισμού έννοια του αντικειμένου (object) και γενικά στις έννοιες που συνιστούν τη φιλοσοφία αυτού του παραδείγματος προγραμματισμού.

Οι κύριοι στόχοι της έρευνας ήταν να προσδιοριστούν οι βασικές δυσκολίες των μαθητών, να εξακριβωθούν οι αιτίες τους, να ερευνηθεί ποιες έννοιες/θέματα θα περιλαμβάνονται και ποιες όχι σε ένα εισαγωγικό μάθημα διδασκαλίας του ΑΠ σε αυτό το επίπεδο και η σειρά με την οποία θα διδάσκονται οι/τα έννοιες/θέματα. Τα ερωτήματα στα οποία στόχευε να απαντήσει η έρευνα είναι:

- Αποτελεί η OF βιώσιμη προσέγγιση για τη διδασκαλία σε αρχάριους μαθητές Λυκείου;

- Ποιες έννοιες/θέματα είναι σημαντικά και μπορούν να συμπεριληφθούν σε ένα εισαγωγικό μάθημα διδασκαλίας του ΑΠ σε αυτό το επίπεδο;
- Τι αντιλήψεις οικοδομούν οι μαθητές;
- Ποιες είναι οι δυσκολίες τους;
- Ποια είναι η ενδεδειγμένη σειρά διδασκαλίας των εννοιών/θεμάτων;
- Ποιες συστάσεις μπορούν να γίνουν όσον αφορά στον τρόπο διδασκαλίας του ΑΠ σε αρχάριους;

Σε αποτελέσματα αυτής της έρευνας έχει ήδη γίνει αναφορά, στην ενότητα στην οποία παρουσιάστηκαν οι δυσκολίες και παρανοήσεις που παρουσιάζουν οι εκπαιδευόμενοι κατά την εκμάθηση του ΑΠ (ενότητα 2.2.) (Ragonis and Ben-Ari, 2002; Ragonis and Ben-Ari, 2005a; Ragonis and Ben-Ari, 2005b; Ben-Ari et al., 2002). Η ίδια η διατριβή είναι γραμμένη στην Εβραϊκή γλώσσα και συνεπώς είναι αδύνατη η ολοκληρωμένη μελέτη της. Στο σημείο αυτό θα γίνει εκτενέστερη αναφορά στο περιεχόμενο της δημοσίευσης Ragonis and Ben-Ari (2005b) που είναι και η πλουσιότερη στην παρουσίαση των ευρημάτων της έρευνας.

Η *μεθοδολογία έρευνας* που ακολουθήθηκε ήταν μια εποικοδομιστική ποιοτική ερευνητική μεθοδολογία (constructivism qualitative research methodology) μέσω παρατήρησης (observation), σημειώσεων (field notes) που οι ερευνητές κρατούσαν κατά τη διεξαγωγή της έρευνας, καταγεγραμμένων συνομιλιών και βιντεοσκοπήσεων (audio and video recordings) κατά τις διαλέξεις και τις εργαστηριακές ώρες, καθώς και ανάλυσης των εργασιών που οι μαθητές εκπονούσαν και παρέδιδαν (homework assignments, lab exercises, tests και projects) και ερωτηματολογίων που συμπλήρωσαν. Τα ευρήματα που αφορούν σε δυσκολίες των μαθητών χωρίζονται σε τέσσερις κύριες κατηγορίες που είναι:

- η έννοια της κλάσης (class) σε αντιδιαστολή με την έννοια του αντικειμένου (object),
- δημιουργία στιγμιοτύπων (instantiation) και κατασκευαστές (constructors),
- απλές (simple) κλάσεις σε αντιδιαστολή με συντεθειμένες (composed) κλάσεις,
- ροή προγράμματος (program flow)

και συνολικά προσδιορίστηκαν 58 δυσκολίες και εσφαλμένες αντιλήψεις των μαθητών.

Αυτό το οποίο αποτέλεσε το θέμα γύρω από το οποίο διεξήχθη η όλη έρευνα, ήταν οι διεργασίες των μαθητών και οι εντυπώσεις-αντιλήψεις που τους δημιουργήθηκαν και όχι οι βαθμολογίες τους.

Χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java και το εκπαιδευτικό προγραμματιστικό περιβάλλον BlueJ (βλ. ενότητα 2.5.). Η κεντρική επιδίωξη της διδασκαλίας ήταν οι μαθητές να κατανοήσουν τις έννοιες τμηματοποίηση (modularity), κελυφοποίηση (encapsulation) και απόκρυψη πληροφορίας (information hiding) και η σειρά που ακολουθήθηκε κατά τη διδασκαλία ήταν:

- Παρουσίαση των εννοιών κλάση και αντικείμενο με τη χρήση διαγραμμάτων.
- Διαδραστική κλήση μεθόδων στο περιβάλλον BlueJ.
- Διδασκαλία δημιουργίας κλάσεων Java, με ιδιότητες, κατασκευαστές και μεθόδους που περιελάμβαναν accessors, mutators και εντολές με απλές εκφράσεις και αναθέσεις τιμών.
- Η πολυπλοκότητα στις δραστηριότητες που οι μαθητές πραγματοποιούσαν αυξανόταν σταδιακά (από τη χρήση έτοιμων κλάσεων έως τη δημιουργία κλάσεων από την αρχή).
- Δημιουργία συντεθειμένων κλάσεων.

Στη συνέχεια η διδακτέα ύλη συνεχίστηκε με τα συνήθη θέματα όπως: δομές ελέγχου, πίνακες κ.ο.κ.

Η έρευνα διεξήχθη επί δύο ακαδημαϊκά έτη (2000-01 και 2001-02) σε μαθητές ηλικίας 15-16 ετών (18 μαθητές την πρώτη χρονιά και 29 τη δεύτερη) χωρίς προηγούμενη γνώση υπολογιστών, με δίωρη διδασκαλία ανά εβδομάδα (περίπου 50 ώρες διδασκαλίας συνολικά). Η μια ώρα ήταν διάλεξη και συζήτηση στην τάξη και η δεύτερη παρουσιάσεις και ασκήσεις στο εργαστήριο.

Συγκεντρώθηκε μεγάλος όγκος δεδομένων, τα οποία αναλύθηκαν για να προκύψουν τα *σημαντικά εκείνα γεγονότα (episodes)* –που μπορεί να ήταν μια συζήτηση στην τάξη, μια απάντηση σε ερωτηματολόγιο, μια λύση σε μια άσκηση κ.ο.κ– σε σχέση με τους στόχους της έρευνας. Στη συνέχεια ακολούθησε ανάλυση των συμβάντων αυτών, χρησιμοποιώντας τις σημειώσεις που οι ερευνητές είχαν κρατήσει, για να καθοριστεί το πλαίσιο στο οποίο συνέβησαν και να γίνουν οι απαραίτητες αποσαφηνίσεις για την κατανόησή τους. Τα γεγονότα που προέκυψαν από την ανάλυση, αφορούσαν σε ενδείξεις δυσκολιών και λαθών αλλά και σε μαρτυρίες κατανόησης εννοιών και συνδέθηκαν με μια η περισσότερες κατηγορίες εννοιών ΑΠ, οι οποίες αναλύθηκαν σε 18 υποκατηγορίες. Ήδη αναφέρθηκαν οι τέσσερις κύριες κατηγορίες δυσκολιών ή εσφαλμένων αντιλήψεων που επελέγησαν, οι οποίες θεωρήθηκαν σημαντικές και ικανές να δώσουν απαντήσεις στα ερευνητικά ερωτήματα που είχαν τεθεί.

Έγινε επιλογή κατηγοριών: (i) σχετικών με τις κύριες έννοιες ΑΠ, (ii) που επηρέασαν την κατανόηση και αντίληψη του αντικειμένου και (iii) που περιείχαν πολλά σημαντικά ευρήματα. Απορρίφθηκαν κατηγορίες που αφορούσαν στη γλώσσα προγραμματισμού αυτή καθ' εαυτή.

Δημιουργήθηκε έτσι ένα δένδρο κατηγοριών και υποκατηγοριών, όπως φαίνεται παρακάτω:

- Κατηγορία 1 – Αντικείμενο σε αντιδιαστολή με κλάση.
 - ◆ Υποκατηγορία 1.1: Η φύση της κλάσης ως πρότυπο (template).
 - ◆ Υποκατηγορία 1.2: Σχέσεις μεταξύ αντικειμένων και κλάσεων.
 - ◆ Υποκατηγορία 1.3: Δημιουργία αντικειμένου.
 - ◆ Υποκατηγορία 1.4: Ταυτότητα αντικειμένων.
- Κατηγορία 2 – Δημιουργία στιγμιotypών (υποστασιοποίηση) και κατασκευαστές
 - ◆ Υποκατηγορία 2.1: Γενική αντίληψη της υποστασιοποίησης.
 - ◆ Υποκατηγορία 2.2: Αντίληψη της υποστασιοποίησης σε συνθετιμένη κλάση (composed class).
 - ◆ Υποκατηγορία 2.3: Επηρεασμός της αντίληψης της υποστασιοποίησης από την υλοποίηση της μέσω της γλώσσας προγραμματισμού.
- Κατηγορία 3 – Απλές σε αντιδιαστολή με συνθετιμένες κλάσεις
 - ◆ Υποκατηγορία 3.1: Αντίληψη της κελυφοποίησης.
 - ◆ Υποκατηγορία 3.2: Αντίληψη της τμηματοποίησης (modularity).
 - ◆ Υποκατηγορία 3.3: Η κλάση ως συλλογή αντικειμένων.
 - ◆ Υποκατηγορία 3.4: Αντίληψη της απόκρυψη πληροφορίας (information hiding).
 - ◆ Υποκατηγορία 3.5: Παρανόηση για αυτόματη δημιουργία αντιγράφων των ιδιοχαρακτηριστικών μιας απλής κλάσης σε μια συνθετιμένη.
 - ◆ Υποκατηγορία 3.6: Αντίληψη της αποστολής μηνύματος (invocation) σε αντικείμενο.
- Κατηγορία 4 – Ροή προγράμματος
 - ◆ Υποκατηγορία 4.1: Αντίληψη της εκτέλεσης των μεθόδων.
 - ◆ Υποκατηγορία 4.2: Αντίληψη της ροής των δεδομένων.
 - ◆ Υποκατηγορία 4.3: Συμβάντα που λαμβάνουν χώρα χωρίς αίτιο.
 - ◆ Υποκατηγορία 4.4: “Πώς γνωρίζει ο Υπολογιστής;”
 - ◆ Υποκατηγορία 4.5: Δυσκολίες στην κατανόηση της συνολικής ροής της εκτέλεσης: Τι συμβαίνει και πότε;

Κάθε υποκατηγορία συνοδεύεται από το σύνολο των δυσκολιών και των παρανοήσεων που τη συνιστούν. Κάθε δυσκολία ή παρανόηση από αυτές παρουσιάζεται με έναν ορισμό και μια σύντομη περιγραφή του προβλήματος, ακολουθούμενα από έναν πίνακα συμβάντων. Κάθε συμβάν περιγράφεται με (i) την περίπτωση στην οποία συνέβη, (ii) μια περιγραφή του τι συνέβη, (iii) πότε συνέβη και (iv) έναν κωδικό για το όνομα του μαθητή στον οποίο αναφέρεται το συγκεκριμένο συμβάν.

Η διδακτορική διατριβή παρείχε συστάσεις για τη διδασκαλία και ένα προτεινόμενο πρόγραμμα διδασκαλίας του ΑΠ σε αρχάριους. Μερικές από τις συστάσεις αφορούσαν στους κατασκευαστές, στις συντεθειμένες κλάσεις, στη συνολική ροή του προγράμματος, στη διδασκαλία πλήρων προγραμμάτων (με main μέθοδο).

Συμπερασματικά η έρευνα αυτή συμβάλλει ουσιαστικά στη διαπίστωση αντιλήψεων και δυσκολιών των μαθητών και έχει πρωτότυπο τρόπο στην παρουσίασή τους. Αυτά αποτελούν πολύτιμη πηγή δεδομένων για τους διδάσκοντες και για τους σχεδιαστές μαθησιακού υλικού, προκειμένου να αναπτυχθεί ένα πρόγραμμα σπουδών με την ενδεδειγμένη σειρά παρουσίασης των εννοιών/θεμάτων και με την κατάλληλη διδακτική προσέγγιση για την αποφυγή τους. Επίσης απέδειξε, ότι η OF προσέγγιση είναι εφικτό να ακολουθηθεί στη δευτεροβάθμια εκπαίδευση με θετικά αποτελέσματα όσον αφορά στη μαθησιακή αποτελεσματικότητά της.

➤ Στο τμήμα *Τεχνολογίας της Πληροφορίας (Department of Information Technology)* στο *Πανεπιστήμιο της Ουψάλα (Uppsala University)* της Σουηδίας, πραγματοποιήθηκε στο πλαίσιο διδακτορικής διατριβής (Eckerdal, 2006), ποιοτική έρευνα των βιοματικών εμπειριών αρχάριων σπουδαστών κατά την εκμάθηση του ΑΠ.

Επελέγησαν 14 σπουδαστές από 22 που αρχικά συμπλήρωσαν ένα ερωτηματολόγιο που τους πληροφορούσε για την έρευνα. Οι σπουδαστές είχαν μόλις τελειώσει το πρώτο μάθημα προγραμματισμού σε Java, παρακολουθούσαν το πρόγραμμα “Aquatic and Environmental Engineering” και ένας από αυτούς το πρόγραμμα “Chemical Engineering” (Στη Σουηδία, μαθήματα Προγραμματισμού είναι υποχρεωτικά στα περισσότερα Προγράμματα Σπουδών Τμημάτων Τεχνικών και Φυσικών επιστημών και όχι μόνο σε Τμήματα της Επιστήμης των Υπολογιστών). Τα ερωτήματα στα οποία στόχευε να απαντήσει η έρευνα είναι:

- *Τι μαθαίνουν* οι αρχάριοι σπουδαστές που διδάσκονται ΑΠ;

- ◆ Τι σημαίνει μαθαίνω να προγραμματίζω από την πλευρά των σπουδαστών;
- ◆ Πώς αντιλαμβάνονται οι σπουδαστές αφηρημένες έννοιες στον ΑΠ (κλάση, αντικείμενο);
- *Πώς μαθαίνουν;*
 - ◆ Πώς χρησιμοποιούν τους διάφορους πόρους καθώς μαθαίνουν Προγραμματισμό υπολογιστών και πώς βιώνουν την υποστήριξη που τους παρέχουν;
 - ◆ Ποια είναι από την πλευρά των σπουδαστών τα θετικά κίνητρα –δηλαδή αυτά που οδηγούν σε ουσιαστική και ενεργή μάθηση– για να μάθουν Προγραμματισμό υπολογιστών;

Τα παραπάνω διερευνώνται σε σχέση με την επίπτωσή τους στη διδασκαλία. Τα συμπεράσματα της έρευνας όσον αφορά στις επιπτώσεις των ευρημάτων της στη διδασκαλία, συμπίπτουν σημαντικά με άλλων ερευνών και παρέχουν μια θεωρητική βάση για την εξήγηση και τη γενίκευσή τους.

Τα δεδομένα συγκεντρώθηκαν με ημιδομημένες συνεντεύξεις, οι οποίες αναλύθηκαν κυρίως με τη χρήση της φαινομενογραφικής (phenomenographic) ερευνητικής προσέγγισης που αποσκοπεί στον εντοπισμό ποιοτικά διαφορετικών αντιλήψεων σε μια ομάδα ανθρώπων. Αυτό που ερευνάται είναι οι αντιλήψεις των σπουδαστών, όχι οι παρανοήσεις τους (“conceptions rather than misconceptions”, Eckerdal and Thune’, 2005, p. 89). Η έρευνα ήταν ποιοτική και δε στόχευε στην εξαγωγή στατιστικώς σημαντικών αποτελεσμάτων.

- Η έρευνα πρεσβεύει τη σημασία που έχει στην εκμάθηση του ΑΠ, η αντίληψη πρωταρχικής σημασίας εννοιών οι οποίες συνιστούν ουσιώδη γνώση του αντικείμενου, όπως *κλάση* και *αντικείμενο*. Το σημαντικό που εισηγείται, είναι η σημασία που έχει να σχεδιασθεί έτσι η διδασκαλία ώστε οι σπουδαστές να αντιληφθούν τις *ζωτικές (σημαντικές, καίριες) πλευρές* των εννοιών *αντικείμενο* και *κλάση* και ανιχνεύει τους διαφορετικούς τρόπους με τους οποίους οι σπουδαστές αντιλαμβάνονται αυτές τις έννοιες. Ουσιαστικά οι διαφορετικές αυτές αντιλήψεις, αποτελούν διαφορετικές πλευρές της ορθής αντίληψης των εννοιών. Προσεκτικά σχεδιασμένα δείγματα προγραμμάτων για χρήση τους στη διδασκαλία, σημαίνει παραλλαγή (variation) των διαφορετικών διαστάσεων (dimensions) που αντιστοιχούν σε αυτές τις σημαντικές πλευρές. Και αυτό είναι εφαρμόσιμο σε όλες τις πλευρές-όψεις του μαθησιακού περιβάλλοντος και της διδασκαλίας. Ο όρος διδασκαλία έχει μια ευρεία έννοια και περιλαμβάνει

οτιδήποτε παρέχει πόρους για μάθηση, όπως προγραμματιστικές εργασίες, εργαλεία λογισμικού, διαλέξεις, Διαδίκτυο (Internet), συμφοιτητές, οτιδήποτε οι σπουδαστές επιλέγουν να χρησιμοποιήσουν κατά τη μαθησιακή διαδικασία, δηλαδή την όλη οργάνωση του μαθησιακού περιβάλλοντος. Για να καταστεί δυνατό οι σπουδαστές να αντιληφθούν όλες τις διαφορετικές πλευρές των φαινομένων, θα πρέπει να τους δίνεται η ευκαιρία να ακολουθήσουν ένα πλήρες προγραμματιστικό έργο, που περιλαμβάνει την ανάλυση, το σχεδιασμό, την υλοποίηση και τον έλεγχο του προγράμματος.

- Από τα αποτελέσματα τα σχετικά με τις αντιλήψεις για τις *βασικές έννοιες κλάση και αντικείμενο* και περί του *τι σημαίνει μαθαίνω να προγραμματίζω από την πλευρά των σπουδαστών*, αναδείχθηκαν δύο σαφείς κατηγορίες.
 - ◆ Στην πρώτη ανήκουν οι αντιλήψεις που περιορίζονται στη γλώσσα προγραμματισμού αυτή καθαυτή και σε μια επιφανειακή περιγραφή του τι σημαίνει μαθαίνω να προγραμματίζω. Σπουδαστές που εκφράζουν αντιλήψεις μόνον αυτής της κατηγορίας, αντιμετωπίζουν προβλήματα και παρουσιάζουν συγχύσεις.
 - ◆ Στη δεύτερη ανήκουν αντιλήψεις που εκτείνονται πέραν της γλώσσας προγραμματισμού και του μαθήματος του ίδιου.
- Η έρευνα κατέδειξε σχέση μεταξύ των αντιλήψεων για τις έννοιες και των αντιλήψεων περί του τι σημαίνει μαθαίνω να προγραμματίζω.

Με τον όρο *πόρο (resource)*, αυτή η έρευνα αναφέρεται σε οτιδήποτε αναφέρθηκε από τους σπουδαστές ότι χρησιμοποίησαν και βίωσαν ότι τους υποστήριξε κατά τη διαδικασία της μάθησης, όπως ο ίδιος ο διδάσκων, οι διαλέξεις, οι σημειώσεις που κρατούσαν κατά τη διάρκεια των διαλέξεων, τα παραδείγματα που τους δόθηκαν από το διδάσκοντα, το βιβλίο του μαθήματος, οι υποχρεωτικές εργασίες, ο υπολογιστής, οι φίλοι τους, οι συμφοιτητές τους, ακόμα και ο όρος “υπομονή”. Ο μόνος τεχνολογικός πόρος που αναφέρεται από τους σπουδαστές που συμμετείχαν στην έρευνα, είναι ο μεταγλωττιστής (compiler). Πολύ λίγοι αναφέρουν το Διαδίκτυο (Internet) ως πηγή αναζήτησης πληροφοριών και ένας/μία μόνον ότι θα προτιμούσε ένα περισσότερο εξελιγμένο περιβάλλον ανάπτυξης λογισμικού για το μάθημα. Τα παραπάνω αποδίδονται από τη συγγραφέα της διατριβής στο ότι πολλοί από τους σπουδαστές δεν είχαν προηγούμενη σχέση με τον Προγραμματισμό και δεν ήταν ενήμεροι περί άλλων πόρων και ορολογίας πέραν αυτών που τους προσφέρθηκαν από το/τη διδάσκοντα/ουσα (Eckerdal, 2006 p. 5). Αναζητήθηκαν κατά την ανάλυση των

δεδομένων ποιοτικές διαφορές στα βιώματα των σπουδαστών.

- Πόροι που χρησιμοποιήθηκαν ουσιαστικά κατά τη μαθησιακή διαδικασία αναδείχθηκαν, ο ίδιος ο διδάσκων, ο υπολογιστής, οι υποχρεωτικές εργασίες, οι φίλοι, οι συμφοιτητές.
- Πόροι που χρησιμοποιήθηκαν με επιφανειακό-επιδερμικό τρόπο κατά τη μαθησιακή διαδικασία αναδείχθηκαν, το βιβλίο του μαθήματος, οι σημειώσεις που κρατούσαν κατά τη διάρκεια των διαλέξεων, τα παραδείγματα που τους δόθηκαν από το διδάσκοντα.
- Αναδείχθηκε μια ξεκάθαρη σύνδεση μεταξύ των πόρων που χρησιμοποιήθηκαν με ουσιαστικό τρόπο και του πώς οι σπουδαστές αντιλήφθηκαν την ουσιαστική υποστήριξη κατά την εκμάθηση του ΑΠ. Για παράδειγμα, σπουδαστές περιγράφουν το διδάσκοντα και τη βοήθεια που τους προσέφερε ως σημαντικό παράγοντα ώστε να σκεφθούν μόνοι τους, ή την ενασχόλησή τους με την εκπόνηση των υποχρεωτικών ασκήσεων ως βασικό παράγοντα μάθησης.
- Επίσης οι πόροι που χρησιμοποιήθηκαν ουσιαστικά κατά τη μαθησιακή διαδικασία είναι στενά συνδεδεμένοι μεταξύ τους και με το γνωστικό αντικείμενο. Ο υπολογιστής χρησιμοποιείται για την εκπόνηση των εργασιών. Οι σπουδαστές συνεργάζονται με τους συμφοιτητές τους για τις περισσότερες εργασίες. Οι διδάσκοντες και οι συμφοιτητές αποτελούν πολύ σημαντικούς πόρους για την κατανόηση εννοιών και τη συζήτηση και επίλυση προβλημάτων καθώς ασχολούνται με τις εργασίες.
- Τα διαφορετικά μεταξύ τους κίνητρα από την πλευρά των σπουδαστών για την εκμάθηση του Προγραμματισμού που ανέδειξε η έρευνα (κοινωνικής συναναστροφής κατά την παρακολούθηση των διαλέξεων, ενδιαφέρον για την απόκτηση γενικότερης ικανότητας επίλυσης προβλημάτων, οφέλη που θα αποκομισθούν από τη γνώση του Προγραμματισμού σε επόμενα μαθήματα αργότερα και γενικότερα στην καριέρα τους) *συνεπάγονται για τους διδάσκοντες τη μεγάλη αξία τού να προσπαθούν για μια διδασκαλία και ένα μαθησιακό περιβάλλον τα οποία σαφώς εστιάζουν στην problem solving πλευρά του ζητήματος, ενθαρρύνουν την κοινωνική αλληλεπίδραση και συναναστροφή και αναδεικνύουν τα οφέλη και τη χρησιμότητα που οι σπουδαστές μπορούν να αποκομίσουν από τη γνώση του Προγραμματισμού.* Έτσι αναδεικνύεται ο πολύ σημαντικός ρόλος του διδάσκοντα στο να βοηθήσει τους σπουδαστές με αρχικά χαμηλό ενδιαφέρον να το αυξήσουν.

- Αναδείχθηκαν ενδιαφέρουσες συνδέσεις μεταξύ των κινήτρων και των αντιλήψεων περί του τι σημαίνει μαθαίνω να προγραμματίζω.
 - Επίσης από τα αποτελέσματα προκύπτει ότι *θα πρέπει να παρέχεται μια ποικιλία πόρων (διαλέξεις, ομαδικές δραστηριότητες, προγραμματιστικά προβλήματα άμεσα συνδεδεμένα με την πραγματικότητα), ώστε να καλύψουν τα διαφορετικά κίνητρα που κινητοποιούν τους σπουδαστές και ανιχνεύθηκαν από την έρευνα.*
 - *Συνολικά διαπιστώθηκε η αλληλεξάρτηση των τεσσάρων ερευνητικών ερωτήσεων της έρευνας και η σημαντικότητά τους στη μαθησιακή διαδικασία.*
- Η διδακτορική διατριβή που αφορά στην ανάπτυξη του προγραμματιστικού μικρόκοσμου objectKarel για την εισαγωγή στον ΑΠ (Ξυνόγαλος, 2002), περιλαμβάνει και την πιλοτική εφαρμογή και αξιολόγηση του περιβάλλοντος στη διδακτική πράξη. Συγκεκριμένα, αυτή έγινε σε 19 φοιτητές του Τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας, με πέντε δίωρα μαθήματα και ένα έκτο μάθημα για την αξιολόγηση της γνώσης τους και την καταγραφή της γνώμης τους για το περιβάλλον και τα μαθήματα.

Στην εφαρμογή αυτή, εξετάσθηκε κατά πόσο συγκεκριμένα χαρακτηριστικά του περιβάλλοντος που αναπτύχθηκε, βοήθησαν τους σπουδαστές να αντιμετωπίσουν δυσκολίες και παρανοήσεις τους που έχουν καταγραφεί στη διεθνή βιβλιογραφία.

Επίσης, με τη χρήση ερωτήσεων ανοικτού τύπου διερευνήθηκε ο βαθμός κατανόησης βασικών εννοιών του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού (κλάση, αντικείμενο, μήνυμα, κληρονομικότητα, πολυμορφισμός, υπερίσχυση) και των βασικών δομών ελέγχου και αν δημιουργήθηκαν κάποιες παρανοήσεις που έχουν αναφερθεί στη διεθνή βιβλιογραφία.

Τέλος, οι φοιτητές απάντησαν ατομικά σε ερωτηματολόγιο που περιελάμβανε ερωτήσεις τόσο ανοικτού όσο και κλειστού τύπου, οι οποίες αφορούσαν στην αποδοχή του προγραμματιστικού περιβάλλοντος και των μαθημάτων που πραγματοποιήθηκαν, από την πλευρά τους.

Η διεξαγωγή της έρευνας κατά τη διάρκεια της πιλοτικής εφαρμογής του περιβάλλοντος, αφορά συνεπώς αποκλειστικά στην αξιολόγησή του και δεν αποφέρει γενικού ενδιαφέροντος ευρήματα ως προς τη διδασκαλία του γνωστικού αντικειμένου του ΑΠ.

- Στο Fjuk et al. (2006) παρουσιάζονται με τη μορφή άρθρων, αναφορές από την

έρευνα που διεξήχθη στο πλαίσιο του έργου “Comprehensive Object Oriented Learning (COOL)”. Το έργο διεξήχθη με τη συνεργασία τεσσάρων ερευνητικών ιδρυμάτων:

- Του διεπιστημονικού κέντρου Intermedia του Πανεπιστημίου του Όσλο, που αντικείμενό του αποτελούν οι κοινωνικές, πολιτιστικές, παιδαγωγικές και τεχνολογικές πλευρές των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας.
- Του ερευνητικού εργαστηρίου Simula (Simula Research Laboratory) που ιδρύθηκε από τα Τμήματα Πληροφορικής των Πανεπιστημίων του Oslo, Bergen, Trondheim και Tromsø, το Νορβηγικό Κέντρο Υπολογιστών και το SINTEF (The Foundation for Scientific and Industrial Research at the Norwegian Institute of Technology).
- Του Νορβηγικού Κέντρου Υπολογιστών (Norwegian Computing Center, NCC), που αποτελεί ένα ιδιωτικό, ανεξάρτητο μη κερδοσκοπικού χαρακτήρα ίδρυμα εφαρμοσμένης έρευνας.
- Του Τμήματος Πληροφορικής του Πανεπιστημίου του Όσλο.

Συγκεκριμένα οκτώ άρθρα, παρουσιάζουν ευρήματα από τέσσερις μελέτες περίπτωσης (case studies), δύο σχεδιασμένα πειράματα (design experiments) και τρία πειράματα με ομάδα ελέγχου (controlled experiments) σε συνολικά 187 εκπαιδευόμενους με διαφορετικά κίνητρα για την εκμάθηση του ΑΠ. Ο μέγιστος αριθμός συμμετεχόντων σε πείραμα ήταν 42 άτομα και ο ελάχιστος 5 άτομα. Τα πειράματα είχαν διάρκεια από έξι ώρες μέχρι ένα εξάμηνο. Οι εκπαιδευόμενοι ήταν ηλικίας από 11 ετών έως επαγγελματίες προγραμματιστές.

Τα θέματα που αφορούν στη διδασκαλία και εκμάθηση του ΑΠ άπτονται ζητημάτων όπως, τα προγραμματιστικά περιβάλλοντα και εργαλεία που χρησιμοποιούνται, τα ιδιαίτερα χαρακτηριστικά των εκπαιδευομένων, οι παιδαγωγικές προσεγγίσεις και οι διδακτικές στρατηγικές που ακολουθούνται, οι μαθησιακοί πόροι που διατίθενται, η χρήση των τεχνολογιών της Πληροφορίας και της Επικοινωνίας (Information and Communication Technologies, ICTs), καθώς και των σχέσεων μεταξύ των παραπάνω. Κάτω από το πρίσμα αυτών των πλευρών μελετήθηκε στο έργο COOL η διδασκαλία του ΑΠ και από την πλευρά του εκπαιδευόμενου (the learner’s perspective), δηλαδή των ενεργειών, των στρατηγικών και των σκέψεών του καθώς επιλύει Αντικειμενοστρεφή προβλήματα.

Τα ευρήματα του έργου αφορούν: (i) στις προκλήσεις που οι εκπαιδευόμενοι αντιμετωπίζουν κατά την εκμάθηση αντικειμενοστρεφών εννοιών, προγραμματισμού και μοντελοποίησης και τις συνέπειες αυτών στη διδασκαλία, (ii) σε ζητήματα σχετικά με προγραμματιστικά εργαλεία και περιβάλλοντα και πώς τα σχετικά με αυτά ευρήματα επηρεάζουν την ανάπτυξη εργαλείων, (iii) στην αποκτηθείσα εμπειρία από την εφαρμογή διαφορετικών ΟΦ στρατηγικών και των επιπτώσεών τους στον παιδαγωγικό σχεδιασμό μαθημάτων και (iv) στις συνέπειες της συνεργασίας (collaboration) στην εκπόνηση έργων από την οπτική των εκπαιδευομένων και των συνεπειών που η συνεργατική μάθηση (collaborative learning) έχει στο σχεδιασμό μαθημάτων (Fjuk et al., 2006, Chapter 11).

Στη συνέχεια παρουσιάζονται συνοπτικά τα συμπεράσματα του έργου για τη διδασκαλία του ΑΠ, την ανάπτυξη εργαλείων που την υποστηρίζουν και το σχεδιασμό μαθημάτων ΑΠ.

- Στα μαθήματα θα πρέπει να τονίζονται η διάκριση μεταξύ εννοιών. Θα πρέπει να επινοούνται παραδείγματα και ασκήσεις τα οποία θα στοχεύουν ειδικά στις συγχύσεις των εκπαιδευομένων. Αυτή η σύσταση επαληθεύει παρόμοια που έχει διατυπωθεί από τους Holland et al. (1997).
- Για να καταστούν ικανοί οι εκπαιδευόμενοι να δημιουργήσουν τις δικές τους κλάσεις, θα πρέπει αρχικά να εξοικειωθούν με αντικειμενοστρεφείς έννοιες –όπως κλάσεις, υποκλάσεις, αντικείμενα και αναφορές– μέσω έτοιμων κλάσεων.
- Είναι χρήσιμο, στις ασκήσεις να υπάρχει συνδυασμός ανάγνωσης και δημιουργίας κώδικα για να αναπτύσσονται δεξιότητες δημιουργίας προγραμμάτων αλλά και κατανόησης έτοιμων προγραμμάτων και οι δύο απαιτούμενες στην επαγγελματική πρακτική.
- Τα προγραμματιστικά εργαλεία τα σχεδιασμένα για διδασκαλία, αποκρύπτουν λεπτομέρειες της προγραμματιστικής διαδικασίας και μειώνουν την πολυπλοκότητα του περιβάλλοντος. Όμως κάποια στιγμή, οι εκπαιδευόμενοι θα πρέπει να έρχονται σε επαφή με έναν συντάκτη κώδικα (text editor) και τη γραμμή εντολών του λειτουργικού συστήματος (operating system command level) και να εξοικειώνονται πλήρως με τη διαδικασία και με ένα επαγγελματικό περιβάλλον, ιδίως όταν πρόκειται να ασχοληθούν επαγγελματικά με τον Προγραμματισμό.
- Για να αποφευχθεί η επίπτωση που ενδεχομένως να έχει η προηγούμενη γνώση διαδικασιακού προγραμματισμού, θα πρέπει να σχεδιάζονται ειδικές

δραστηριότητες γι' αυτό το σκοπό. Για παράδειγμα, να δίνεται στους εκπαιδευόμενους μια μέθοδος main έκτασης μιας σελίδας και να τους ζητείται να δομήσουν τις εντολές της σε μεθόδους και κλάσεις. Αλλά επιπροσθέτως αυτού, θα πρέπει να επιδεικνύονται στους εκπαιδευόμενους τα οφέλη της Αντικειμενοστρεφούς δόμησης για να αντιληφθούν επιπλέον του πώς και το γιατί πρέπει να δομήσουν το πρόγραμμα τους με αντικειμενοστρεφή λογική.

- Είναι σημαντικό να αντιληφθούν οι εκπαιδευόμενοι την εκτέλεση του κώδικά τους, δηλαδή τη δημιουργία αντικειμένων, τη σύνδεση αναφορών σε αυτά και την κλήση μεθόδων καθώς το πρόγραμμα εκτελείται. Αυτό αποκρύπτεται στο εσωτερικό του υπολογιστή και το μόνο που γίνεται φανερό είναι η έξοδος που παράγεται από συγκεκριμένες εντολές (π.χ. `System.out.println(...)` στην Java). Αυτού του είδους η οπτικοποίηση, θα πρέπει να αποτελέσει στόχο ενός εργαλείου σχεδιασμένου για τη διδασκαλία του ΑΠ.
- Μια προτεινόμενη ακολουθία στη διδασκαλία εισαγωγικών μαθημάτων ΑΠ είναι:
 - ◆ Διδασκαλία αντικειμενοστρεφών εννοιών με το μικρόκοσμο Karel J (βλ. υποενότητα 2.5.1.) ή ένα παρόμοιο περιβάλλον που παρέχει οπτική και άμεση ανατροφοδότηση της εκτέλεσης του προγράμματος, ενισχύει τον ορισμό υποκλάσεων και μεθόδων χωρίς να ευνοεί την ανάπτυξη user-defined κλάσεων.
 - ◆ Διδασκαλία χρήσης έτοιμων κλάσεων στο BlueJ, ή ένα παρόμοιο περιβάλλον που υποστηρίζει την οπτικοποίηση.
 - ◆ Διδασκαλία δημιουργίας κλάσεων από την αρχή στο BlueJ, ή ένα παρόμοιο περιβάλλον το οποίο ενθαρρύνει την ανάπτυξη μεθόδων στα αντικείμενα.
 - ◆ Μετά τα παραπάνω οι εκπαιδευόμενοι μπορούν να περάσουν στη γραμμική εντολών (command line mode) και επίσης σε πιο επαγγελματικά περιβάλλοντα (όπως το JBuilder).

Δε συνιστάται η χρήση περισσότερων από δύο περιβαλλόντων στο ίδιο εξάμηνο, αφού η εκμάθηση χρήσης τους απαιτεί χρόνο που αποβαίνει εις βάρος του ίδιου του διδασκόμενου αντικειμένου.

- Η κοινωνική αλληλεπίδραση (social interaction) και η συνεργασία (collaboration) είναι ζωτικής σημασίας για την ανάπτυξη των δεξιοτήτων των εκπαιδευομένων στην Αντικειμενοστρέφεια και για τη δημιουργία μιας παραγωγικής μαθησιακής κατάστασης. Τα ευρήματα αυτά έχουν σημαντικές επιπτώσεις στο σχεδιασμό μαθημάτων, κατά τον οποίο πρέπει να λαμβάνονται υπόψη ένα σύνολο από

παράγοντες, όπως ποίοι είναι οι εκπαιδευόμενοι και οι εκπαιδευτές, ο διαθέσιμος χρόνος, ποίοι είναι οι μαθησιακοί στόχοι, το περιεχόμενο του μαθήματος, τα εργαλεία που χρησιμοποιούνται, η μέθοδος αξιολόγησης.

2.4. Διδακτικές προσεγγίσεις

Στο διάστημα που ο ΑΠ έχει εμφανισθεί ως ιδιαίτερο παράδειγμα προγραμματισμού και περιλαμβάνεται στα Προγράμματα Σπουδών των Πανεπιστημιακών κυρίως τμημάτων, έχουν αναπτυχθεί, προταθεί και ακολουθηθεί τρόποι για να διδαχθεί, με διαφορές στις αρχές και τα χαρακτηριστικά τους. Στη συνέχεια παρατίθενται οι κύριες διδακτικές προσεγγίσεις που αναφέρονται στη βιβλιογραφία για τη διδασκαλία του ΑΠ.

2.4.1. Κύριες διδακτικές προσεγγίσεις

2.4.1.1. Διδασκαλία ΑΠ με τη χρήση της γλώσσας προγραμματισμού Java και ενός επαγγελματικού ολοκληρωμένου περιβάλλοντος ανάπτυξης (IDE) – Κλασική διδακτική προσέγγιση

Η αναφορά αυτής της προσέγγισης –που αποδίδεται με τον όρο κλασική διδακτική προσέγγιση (classic approach)– περιορίζεται στη γλώσσα Java, αφού είναι η επικρατούσα γλώσσα προγραμματισμού για το αντικειμενοστρεφές παράδειγμα (Ragonis and Ben-Ari, 2005b; Benander et al., 2004; Kölling and Rosenberg, 2001b). Βασίζεται στη χρήση της γλώσσας προγραμματισμού και ενός επαγγελματικού IDE. Η έμφαση δίνεται στις συντακτικές και σημασιολογικές λεπτομέρειες της γλώσσας προγραμματισμού και την υλοποίηση προγραμμάτων στο επαγγελματικό περιβάλλον που χρησιμοποιείται και όχι στην ανάπτυξη ικανοτήτων σχεδίασης αλγορίθμων για την επίλυση προβλημάτων. Το περιβάλλον ανάπτυξης, δε διαθέτει εκπαιδευτικά χαρακτηριστικά ώστε να ενισχυθούν οι εκπαιδευόμενοι στη δημιουργία και την εκτέλεση των προγραμμάτων τους. Η προσέγγιση αυτή έχει διερευνηθεί αρκετά και για το διαδικασιακό παράδειγμα προγραμματισμού και η σχετική βιβλιογραφία (Brusilovsky et al., 1997) την έχει αναδείξει ως έναν πολύ σημαντικό παράγοντα δυσκολίας στη διδασκαλία. Τα προβλήματα αυτής της προσέγγισης περιγράφονται στο Ξυνόγαλος (2002), σελ. 62-64.

Στο πλαίσιο μιας τέτοιας διδακτικής προσέγγισης, τα συγγράμματα που συνήθως χρησιμοποιούνται αποτελούν έναν οδηγό της γλώσσας Java και όχι ένα βιβλίο για τον ΑΠ. Κάποια από αυτά (Lemay and Cadenhead, 1999), ακολουθούν την προσέγγιση να παρουσιάζουν τις δομές της γλώσσας που υλοποιούν έννοιες, χωρίς όμως να παρέχουν μια παρουσίαση των αντικειμενοστρεφών εννοιών αυτών καθ' εαυτών σε ένα πλαίσιο προσέγγισης της αντικειμενοστρεφούς φιλοσοφίας. Σε άλλα πάλι (Linden, 1996), παρατίθενται οι έννοιες θεωρητικά στην αρχή και χωριστά από ένα πραγματικό πλαίσιο εφαρμογής τους και στη συνέχεια παρουσιάζονται οι δομές της γλώσσας (Becker, 2000). Αντιπροσωπευτικά παραδείγματα επαγγελματικών ολοκληρωμένων προγραμματιστικών περιβαλλόντων ανάπτυξης εφαρμογών σε Java (IDEs), είναι τα SUN One Studio, Eclipse, JBuilder, VisualCafe, Visual J++, NetBeans.

2.4.1.2. Η “imperatives first” ή “procedurals first” προσέγγιση

Η προσέγγιση αυτή δε λαμβάνει υπόψη τη διαφορετικότητα της φιλοσοφίας του Αντικειμενοστρεφούς Προγραμματισμού σε σχέση με αυτήν του διαδικασιακού (Bergin, 2000a) και δεν εστιάζει στις θεμελιώδεις εκείνες έννοιες και αρχές (κλάση, αντικείμενο, κελυφοποίηση κ.ά.) που αποτελούν τη βάση αυτού του παραδείγματος προγραμματισμού. Οι έννοιες και αρχές αυτές παίζουν σημαντικό ρόλο στη μέθοδο διδασκαλίας που πρέπει να ακολουθηθεί προκειμένου να αφομοιωθούν και αυτές και η ιδιαιτερότητα της Αντικειμενοστρέφειας. Η προσέγγιση αυτή δεν εστιάζει στις αρχές του ΑΠ από νωρίς. Αντίθετα, διδάσκεται ο ΑΠ χωρίς να δίνεται έμφαση στις ιδιαίτερες αρχές που τον διέπουν, αλλά αρχίζοντας με την εισαγωγή των παραδοσιακών στοιχείων της γλώσσας προγραμματισμού, όπως είναι οι μεταβλητές, οι εντολές ανάθεσης, οι δομές επιλογής και επανάληψης. Οι αντικειμενοστρεφείς έννοιες που έχουν βασική σημασία, μένουν κρυμμένες στο παρασκήνιο και είτε καθυστερείται η αναφορά τους, είτε/και παρουσιάζονται στα σημεία που είναι αναπόφευκτο, ως “απαραίτητο” στοιχείο για την κατασκευή προγραμμάτων, αλλά μετατίθεται η κατανόησή τους στο μέλλον.

Με τη μέθοδο αυτή οι εκπαιδευόμενοι δεν “μυούνται” στη φιλοσοφία του ΑΠ και δεν αφομοιώνουν το ρόλο και τη σημασία των θεμελιωδών αντικειμενοστρεφών εννοιών και αρχών στην επίλυση προβλημάτων με προγραμματιστικό τρόπο (Westfall, 2001; Kölling, 2003). Παρότι διδάσκονται –με κάποιο τρόπο– αντικειμενοστρεφείς έννοιες και αρχές και μπορούν από συντακτική άποψη να τις χρησιμοποιήσουν, δεν

αφομοιώνουν την αντικειμενοστρεφή φιλοσοφία επίλυσης προβλημάτων και τη διαφορετικότητα της από τη διαδικασιακή, δηλαδή δεν αποκτούν τη δεξιότητα να χρησιμοποιούν αποδοτικά τις έννοιες και αρχές που διδάχθηκαν.

2.4.1.3. Η “objects first” προσέγγιση

Τα τελευταία χρόνια έχει πραγματοποιηθεί μια μετακίνηση του τρόπου διδασκαλίας του ΑΠ, υιοθετώντας αυτή την προσέγγιση (Kölling, 1999a and b; CC2001, 2001; Proulx et al., 2002; Cooper et al., 2003; Bruce et al., 2001). Αυτό που τη χαρακτηρίζει είναι η έμφαση στις αντικειμενοστρεφείς έννοιες της κλάσης (class) και του αντικειμένου (object) και η από νωρίς παρουσίασή τους. Εδώ, η σημασία της βασικής για την αντικειμενοστρεφή φιλοσοφία προγραμματισμού έννοιας του αντικειμένου (object), αποτελεί το βασικό στοιχείο διδασκαλίας από την αρχή. Όπως αναφέρεται στο IEEE/ACM Computing Curricula (CC2001, 2001) “What distinguishes the IF (Imperatives First) approach from the objects-first model is the emphasis and ordering of the early topics.”.

Η έμφαση δίνεται στις έννοιες που συνιστούν τη φιλοσοφία αυτού του παραδείγματος προγραμματισμού και αναδεικνύει τη σημασία τους ως δομικά συστατικά των προγραμμάτων. Οι δομές της γλώσσας προγραμματισμού που χρησιμοποιείται μαζί με τις λεπτομέρειες των εντολών και της σύνταξης τους, παρουσιάζονται όταν απαιτούνται για την εφαρμογή των εννοιών και την ανάδειξη του ρόλου τους στην επίλυση προβλημάτων. Ο τρόπος αυτός βοηθά στην εμπέδωση της Αντικειμενοστρέφειας ως ιδιαίτερου παραδείγματος Προγραμματισμού σε πρόωρη φάση κατά τη διάρκεια της διδασκαλίας και συνεπώς της φιλοσοφίας της.

Ένα πλήθος εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων έχει σχεδιασθεί για την υποστήριξη αυτής της διδακτικής προσέγγισης στη διδασκαλία του ΑΠ. Αυτά, αποτελούν προγραμματιστικούς μικρόκοσμους –όπως οι KarelJ.Robot, Alice, objectKarel, Jeroo–, ολοκληρωμένα προγραμματιστικά περιβάλλοντα με εκπαιδευτικά χαρακτηριστικά όπως το BlueJ και το greenfoot (για μαθητές σχολικής ηλικίας), εργαλεία που υποστηρίζουν την προσέγγιση μέσω γραφικού περιβάλλοντος διασύνδεσης χρήστη (Graphics User Interaction, GUI) και εργαλεία που συνίστανται σε γραφικές βιβλιοθήκες (Graphics Libraries). Αναλυτική παρουσίαση όλων αυτών των εργαλείων και περιβαλλόντων γίνεται στην επόμενη ενότητα (ενότητα 2.5.).

Η προσέγγιση αυτή έχει επικρατήσει, γιατί επιδιώκει την ουσιαστική μετάδοση της Αντικειμενοστρεφούς σκέψης και φιλοσοφίας στην επίλυση προβλημάτων με προγραμματιστικό τρόπο. Όπως αναφέρεται στο CC2001 (2001), p. 30, “the principal advantage in the objects-first strategy is the early exposure to object-oriented programming, which has become increasingly important in both academia and industry. The December 2000 announcement by the College Board that they plan to introduce a more object-oriented approach in the Advanced Placement curriculum underscores the importance of this approach”.

Στο Lewis (2000), παρουσιάζεται η άποψη πως αποτελεί μύθο ότι “η φράση “objects first” είναι καλά ορισμένη”. Η συγκεκριμένη δημοσίευση αναφέρει και συζητά ιδέες και θέματα που κυκλοφορούν και άπτονται του ΑΠ ως προς την ορθότητά τους, από την άποψη πάντα του συγγραφέα. Η άποψη του για το συγκεκριμένο θέμα υποδηλώνει τη διαφορετικότητα με την οποία μπορεί να υλοποιηθεί η OF προσέγγιση και ενισχύει την άποψη της ανάπτυξης διαφορετικών μεθόδων.

Οι στρατηγικές που θα αναφερθούν στη συνέχεια, αποτελούν ουσιαστικά υλοποιήσεις της OF προσέγγισης στην πράξη.

2.4.2. Στρατηγικές της “objects first” προσέγγισης

2.4.2.1. Η “concepts first” στρατηγική

Με τον όρο αυτό αποδίδεται (Goldman, 2004), η στρατηγική για την υλοποίηση της οποίας χρησιμοποιείται το ολοκληρωμένο προγραμματιστικό περιβάλλον JPie (Goldman, 2003), που έχει αναπτυχθεί στο Τμήμα Computer Science and Engineering του Washington University στο St. Louis του Missouri. Τα χαρακτηριστικά αυτού του περιβάλλοντος ενισχύουν σε μεγάλο βαθμό τη συγκέντρωση της προσοχής των εκπαιδευομένων στις σημαντικές έννοιες.

Το διδακτικό περιεχόμενο δομείται με ένα πρόγραμμα μαθημάτων οργανωμένο με βάση τις θεμελιώδεις προγραμματιστικές έννοιες (“big ideas”) (Fjuk et al., 2006, Chapter 6, p. 81) (top down παρουσίαση) και οι εκπαιδευόμενοι χωρίς προηγούμενη προγραμματιστική γνώση, εισάγονται στην Επιστήμη των Υπολογιστών μέσω πρακτικής εξάσκησης στην ανάπτυξη λογισμικού.

Τα καίρια θέματα της διδασκαλίας ενισχύονται μέσω υλοποίησης έργων, με χειρισμό γραφικών αναπαραστάσεων προγραμματιστικών εννοιών και δομών και όχι κωδικοποίησής τους με κείμενο. Έτσι η κατασκευή λογισμικού πραγματοποιείται αμεσότερα και αποδοτικότερα και αυτό επιτρέπει στους εκπαιδευόμενους να συγκεντρώνονται σε υψηλότερου επιπέδου θέματα και στις σημαντικές ιδέες του Προγραμματισμού. Η ύλη που αφορά στις έννοιες καλύπτεται αποτελεσματικότερα, χωρίς ενασχόληση με σύνταξη γλώσσας και αρκετά πιο γρήγορα σε σύγκριση με προσεγγίσεις που χρησιμοποιούν σύνταξη γλωσσών προγραμματισμού.

Η δυνατότητα δημιουργίας κώδικα Java που αντιστοιχεί στις γραφικές αναπαραστάσεις, την οποία παρέχει το περιβάλλον JPie, αποτελεί για τους εκπαιδευόμενους όταν συμβεί προς το τέλος των μαθημάτων, μια ομαλή μετάβαση σε κείμενο προγράμματος χωρίς να συνιστά εκφοβιστικό παράγοντα.

Η στρατηγική αποτελεί υλοποίηση της OF προσέγγισης, υποστηριζόμενη από το συγκεκριμένο περιβάλλον με τα ιδιαίτερα χαρακτηριστικά του.

Στην επόμενη ενότητα (ενότητα 2.5.) των εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων, παρουσιάζεται το περιβάλλον JPie.

2.4.2.2. Η “design driven” στρατηγική

Αυτή εστιάζει στην απόκτηση κυρίως σχεδιαστικών δεξιοτήτων (design skills) από την πλευρά των εκπαιδευομένων. Η έμφαση δίνεται στη σχεδίαση (design) παρά στη σύνταξη (Fjuk et al., 2006, Chapter 6, p. 81) και η στρατηγική αναπτύσσεται σε δύο σειρές μαθημάτων (CS1) και CS2). Ουσιαστικά αποτελεί υλοποίηση της OF προσέγγισης σε συνδυασμό με ανάλυση και σχεδιασμό από τα αρχικά στάδια εκπαίδευσης.

Στη συνέχεια αναφέρονται πιο αναλυτικά, λεπτομέρειες της στρατηγικής αυτής από τη μελέτη που την παρουσίασε.

Στο Alphonse and Ventura (2002), παρουσιάζεται μια “design driven” –όπως αναφέρεται– στρατηγική που εφαρμόζεται σε σειρά μαθημάτων CS1 και CS2 στην Αντικειμενοστρέφεια. Αυτή εστιάζει στην *ανάπτυξη σχεδιαστικών δεξιοτήτων* από τη μεριά των εκπαιδευομένων –οι οποίες αναμφισβήτητα είναι ζωτικής σημασίας για την ακαδημαϊκή και επαγγελματική τους πορεία–, *δίνοντας έμφαση στη σχεδίαση παρά στη σύνταξη* και περιλαμβάνοντας τη *χρήση ενός υποσυνόλου της UML (Unified Modeling Language), σχεδιαστικών προτύπων (design patterns) και μη τετριμμένων προβλημάτων για την παρακίνηση των εκπαιδευομένων*. Στο άρθρο αυτό

υποστηρίζεται, ότι με τη στρατηγική αυτή αντιμετωπίζονται αρκετά μειονεκτήματα του programming-first (PF) μοντέλου –όπως αυτό εννοείται στο CC2001 (2001) (υιοθέτηση της πρωταρχικής σημασίας ανάπτυξης προγραμματιστικών δεξιοτήτων από οποιονδήποτε σπουδάζει CS)– στο εισαγωγικό πρόγραμμα σπουδών (introductory curriculum) του αντικειμένου CS. Τα μειονεκτήματα αυτά αναφέρονται στο CC2001 (2001) p. 23 (περιορισμένη αντίληψη του επιστημονικού αντικειμένου, καθυστέρηση της παρουσίασης στη διδακτέα ύλη θεωρητικών θεμάτων του αντικειμένου που αργότερα φαίνονται να μην έχουν την ίδια άμεση σχέση, εστίαση στη σύνταξη, μικρό βάρος στη σχεδίαση, ανάλυση και έλεγχο, υστέρηση των εκπαιδευομένων που δεν έχουν προηγούμενη εμπειρία με υπολογιστές και δημιουργία ψευδαίσθησης στους άλλους ότι ξέρουν περισσότερα από ότι πραγματικά ξέρουν, δημιουργία εσφαλμένης εντύπωσης ότι ο μόνος εφικτός τρόπος επίλυσης ενός προβλήματος με τη χρήση H/Y είναι η συγγραφή προγράμματος), τονίζοντας ταυτόχρονα την εξαιρετική διαχρονικότητα του μοντέλου αυτού. Η προσέγγιση τονίζει τη σημασία της περιγραφής συστημάτων από τους εκπαιδευόμενους σχεδιάζοντας μοντέλα τους, της αντίληψης των προβλημάτων με έναν αντικειμενοστρεφή τρόπο σκέψης και της εστίασης στις αντικειμενοστρεφείς έννοιες και όχι στην αντικειμενοστρεφή γλώσσα προγραμματισμού. Το τελευταίο επιτυγχάνεται με έμφαση αρχικά σε θέματα σχεδίασης και καθυστέρηση της παρουσίασης διαδικασιακών χαρακτηριστικών. Την έκθεση δηλαδή των εκπαιδευομένων στην επίλυση προβλημάτων και στις αντικειμενοστρεφείς έννοιες ώστε να μάθουν να σκέφτονται “αντικειμενοστρεφώς”. Τα παραπάνω επιτυγχάνονται μέσω μιας έντονα σχεδιαστικής και θεματικής αντικειμενοστρεφούς φιλοσοφίας στα CS1 και CS2 και συγκεκριμένα:

- Στο CS1 συνδυάζονται OF προσέγγιση με ανάλυση και σχεδιασμό από τα αρχικά στάδια εκπαίδευσης. Για τη σχεδίαση χρησιμοποιείται ένα υποσύνολο των διαγραμμάτων της UML μαζί με την ιδιαίτερη σημασιολογία των σχέσεων μεταξύ των κλάσεων χωρίς την αναφορά σε γλώσσα προγραμματισμού. Θέματα όπως αντικείμενα, κλάσεις, UML, κληρονομικότητα, πολυμορφισμός, διαπροσωπείες (interfaces) και απλά σχεδιαστικά πρότυπα, παρουσιάζονται πριν από τις διαδικασιακές δομές όπως δομές επιλογής, επανάληψης και πίνακες. Συγκεκριμένα στους εκπαιδευόμενους ως πρώτες ασκήσεις τους δίνεται η σχεδίαση σε UML μαζί με μια “σκελετική” λύση και τους ζητείται να την ολοκληρώσουν με κώδικα. Αργότερα τους δίνονται ασκήσεις που περιέχουν τις περιγραφές των απαιτήσεων του προβλήματος και τους ζητείται να

σχεδιάσουν τη λύση σε UML πριν αρχίσουν να γράφουν κώδικα. Με αυτό τον τρόπο μαθαίνουν να σκιαγραφούν ένα προσχέδιο για τη λύση τους πριν αρχίσουν να κωδικοποιούν. Οι συντακτικές δομές δεν είναι στο επίκεντρο της προσοχής τους και εισάγονται μόνον όταν χρειαστούν. Οι πίνακες (arrays) που παρουσιάζονται στο τέλος του course αποτελούν μια καλή εισαγωγή στο CS2 course που ξεκινά με τις συλλογές αντικειμένων (collections of objects).

- Επιπλέον για την ανάπτυξη καλών σχεδιαστικών δεξιοτήτων από τους εκπαιδευόμενους, εισάγονται αρκετά σχεδιαστικά πρότυπα (design patterns) τα οποία παρουσιάζονται ως λύσεις σε συγκεκριμένα προβλήματα. Στο CS1 course παρουσιάζονται τα Holder, Proxy, State και μερικές φορές μερικά άλλα όπως τα Singleton και Factory (Gamma et al., 1995). Με τη χρήση τους, γίνεται δυνατή η διαχείριση της πολυπλοκότητας μη τετριμμένων προβλημάτων που αναδεικνύουν τα οφέλη της αντικειμενοστρέφειας. Επίσης οι εκπαιδευόμενοι απελευθερώνονται από το να σκέφτονται για τη σχεδίαση προγραμμάτων με όρους της γλώσσας υλοποίησης.
- Στο CS2 οι εκπαιδευόμενοι δουλεύουν κατά μικρές ομάδες, σε ένα project διάρκειας εξαμήνου, που τους δίνει την ευκαιρία να βιώσουν τη σημασία της σχεδίασης και της τεκμηρίωσης και επίσης τους παρουσιάζονται κάποια πολυπλοκότερα σχεδιαστικά πρότυπα. Το project επιλέγεται ώστε να κεντρίζει το ενδιαφέρον των εκπαιδευομένων, να είναι σημαντικό ως προς τη συνολική προσπάθεια που απαιτεί για τη λύση του και τη γνώση που θα αποκομίσουν ολοκληρώνοντας το.

Τέλος η έρευνα αυτή αναφέρει τη *σπουδαιότητα χρήσης ενός UML εργαλείου για σχεδίαση και παραγωγή κώδικα* και τη βελτίωση των σχεδιαστικών δεξιοτήτων που θα επιφέρει, εστιάζοντας την προσοχή των εκπαιδευομένων στη σχεδίαση του προβλήματος παρά στους μηχανισμούς μετατροπής της αφηρημένης σχεδίασης σε υλοποιημένο κώδικα. Συνολικά η παραπάνω μελέτη εστιάζει στη σημασία της σχεδίασης (design) για τη διδασκαλία του ΑΠ

2.4.2.3. Η “model first” στρατηγική

Πρόκειται για μια στρατηγική που ακολουθεί τη Σκανδιναβική παράδοση (“Scandinavian tradition”) τη βασισμένη στην “κληρονομιά” του Kristen Nygaard (Fjuk et al., 2006, Chapter 6, p. 81) ο οποίος είναι και ο θεμελιωτής της αντικειμενοστρέφειας από τη δεκαετία του 60 που αναπτύχθηκαν στη Νορβηγία οι εκδόσεις της γλώσσας Simula, της πρώτης αντικειμενοστρεφούς γλώσσας

προγραμματισμού. Έχει χαρακτηριστεί από τον Kristen Nygaard ως Σκανδιναβική προσέγγιση στην αντικειμενοστρέφεια (Scandinavian approach to object-orientation) (Madsen, 1995; Smørdal, 1999).

Αποτελεί μια υλοποίηση της OF προσέγγισης με έμφαση στη μοντελοποίηση. Εστιάζει στην *εννοιολογική μοντελοποίηση (conceptual modeling)* ως καθοριστικό χαρακτηριστικό της αντικειμενοστρέφειας και παρέχει μια ενοποιημένη όψη και μια παιδαγωγική προσέγγιση που επικεντρώνεται στις *modeling* “πλευρές” της αντικειμενοστρέφειας (Bennedsen and Caspersen, 2004a). Στοχεύει στην απόκτηση δεξιοτήτων για την επίλυση προβλημάτων εφαρμόζοντας την αντικειμενοστρεφή φιλοσοφία μέσω σχεδίασης μοντέλων και επίσης δεξιοτήτων ανάπτυξης κώδικα που βασίζεται σε αυτά τα μοντέλα. Περιλαμβάνει τον πλήρη κύκλο από τη φάση της ανάλυσης έως την υλοποίηση με κώδικα.

Η θεμελιώδης ιδέα είναι ότι ένα αντικειμενοστρεφές πρόγραμμα αποτελεί ένα *μοντέλο (model)*, το οποίο προσομοιώνει τη συμπεριφορά ενός πραγματικού ή ενός φανταστικού τμήματος του κόσμου. Αυτό το μοντέλο μπορεί να εξετασθεί σε διαφορετικά επίπεδα: του εννοιολογικού μοντέλου που περιγράφει τα βασικά στοιχεία ενός προβλήματος και τις μεταξύ τους σχέσεις, ενός πιο λεπτομερούς μοντέλου κλάσεων (*class model*) που παρέχει μια σφαιρική θεώρηση της λύσης και της πραγματικής υλοποίησης της λύσης σε μια αντικειμενοστρεφή γλώσσα προγραμματισμού. Τα παραπάνω επίπεδα συνιστούν τις τρεις φάσεις της ανάλυσης (*analysis*), σχεδίασης (*design*) και υλοποίησης (*implementation*).

Βασική μελέτη παρουσίασης της “*model first*” (ή *model-driven*) προσέγγισης διδασκαλίας για ένα CS1 μάθημα αποτελεί το Bennedsen and Caspersen (2004a). Σε αυτήν αναφέρεται η *εννοιολογική μοντελοποίηση (conceptual modelling)* ως ένα στοιχείο που λείπει από τις συστάσεις για ένα CS1 μάθημα που το CC2001 (2001) περιλαμβάνει και επίσης ότι στο CC2001 (2001) θίγονται σύντομα μόνον κάποιες “πλευρές” της και ότι ο χρόνος που προτείνεται να αφιερωθεί σε αυτήν είναι μόνον τέσσερις ώρες. Υποστηρίζει την εξισορρόπηση στο σχεδιασμό ενός μαθήματος προγραμματισμού, ανάμεσα στη μοντελοποίηση με τη χρήση, για παράδειγμα, ενός CASE εργαλείου που παράγει πλήρως τον κώδικα –και που έχει ως αποτέλεσμα οι εκπαιδευόμενοι να μην μπορούν να συντάξουν κώδικα μόνοι τους– και στην εστίαση στις λεπτομέρειες της γλώσσας προγραμματισμού που συνεπάγεται τη μη εκμάθηση του προγραμματιστικού παραδείγματος. Δηλαδή, *πραγματεύεται την ενοποίηση της*

εννοιολογικής μοντελοποίησης και της κωδικοποίησης προς μια συστηματική προσέγγιση της ανάπτυξης προγράμματος.

Προτείνει μια γενική δομή ενός CS1 μαθήματος με την εννοιολογική μοντελοποίηση ως “κινητήρια δύναμη” και συζητά απόψεις αυτής της δομής προς μια κατεύθυνση συστηματικής προσέγγισης του προγραμματισμού. Σε γενικές γραμμές, στο πρώτο μισό του μαθήματος προτείνει την έμφαση συγχρόνως στην εννοιολογική μοντελοποίηση και την κωδικοποίηση (η κωδικοποίηση και η εννοιολογική μοντελοποίηση προχωράνε μαζί, με τη δεύτερη να “δείχνει” το δρόμο) και στο δεύτερο μισό την έμφαση στην ποιότητα του κώδικα που επιτυγχάνεται με συμπληρωματικές τεχνικές σε αυτές του πρώτου μισού του μαθήματος. Συγκεκριμένα τα επίπεδα των τεχνικών για τη συστηματική δημιουργία αντικειμενοστρεφών προγραμμάτων καλής ποιότητας είναι:

- Χώρος προβλήματος → Μοντέλο (με UML)
- Μοντέλο → Σκελετός κώδικα Java (με τη χρήση του coding pattern)
- Λειτουργικότητα → Κώδικας Java (καθορισμός attributes και μεθόδων των κλάσεων)
- Υλοποίηση των κλάσεων (εσωτερικοί περιορισμοί των κλάσεων που πρέπει να πληρούνται πριν και μετά την κλήση κάθε μεθόδου)
- Υλοποίηση των μεθόδων (με τη χρήση algorithmic patterns)

Στην υλοποίηση της “model first” στρατηγικής στην παραπάνω μελέτη, γίνεται χρήση της UML και ενός γραφικού πακέτου για την εισαγωγή στη βασική κωδικοποίηση. Μετά την εισαγωγή μιας έννοιας από το εννοιολογικό πλαίσιο, εισάγεται ένα αντίστοιχο πρότυπο κωδικοποίησης (coding pattern) για τη μετάφραση από UML σε κώδικα. Η μελέτη αναφέρει ότι κίνητρο για το COOL project (ενότητα 2.3.), αποτέλεσε το γεγονός της έλλειψης, από τους παραδοσιακούς τρόπους διδασκαλίας της αντικειμενοστρέφειας, μιας ενιαίας “προοπτικής” (αντίληψης) και μιας παιδαγωγικής προσέγγισης που να εστιάζει στις modeling “πλευρές” της αντικειμενοστρέφειας και ότι επίσης το COOL project, σκόπευε να διερευνήσει τη σχέση μεταξύ παιδαγωγικών προσεγγίσεων, διδακτικών τεχνικών, κατάλληλων παραδειγμάτων, εργαλείων κ.λ.π.

Η “model first” στρατηγική υλοποίησης, θεωρεί ζωτικής σημασίας την *ενσωμάτωση της εκμάθησης του προγραμματισμού σε ένα πλαίσιο, το οποίο πρωταρχικά θα εστιάζει στην εκμάθηση συστηματικών τεχνικών για την ανάπτυξη προγραμμάτων ξεκινώντας από ένα εννοιολογικό μοντέλο του χώρου του προβλήματος και επίσης στην άσκηση*

των εκπαιδευομένων στη διαδικασία εφαρμογής αυτών των τεχνικών (Bennedsen and Caspersen, 2004b).

Στο COOL project –στο πλαίσιο του οποίου κυρίως διατυπώθηκε η “model first” στρατηγική–, ακολουθήθηκε η θεώρηση ότι ο ΑΠ θα αντιμετωπίζεται ως μοντελοποίηση (modeling) και ότι οι βασικές έννοιες του αντικειμενοστρεφούς παραδείγματος προγραμματισμού θα εμπεδώνονται από τα αρχικά στάδια εκπαίδευσης (Groven et al., 2003; Berge et al., 2003b).

Για την υλοποίηση μιας τέτοιας στρατηγικής απαιτείται η υποστήριξη από κατάλληλα εργαλεία που θα παρουσιάζουν τις σημαντικές όψεις της αντικειμενοστρέφειας και θα μειώνουν την ενασχόληση των εκπαιδευομένων με μη ουσιώδεις λεπτομέρειες, σε συνδυασμό με τη χρήση κατάλληλα επιλεγμένων παραδειγμάτων (Groven et al., 2003).

2.5. Εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα για τον Αντικειμενοστρεφή Προγραμματισμό

Τα εργαλεία λογισμικού που έχουν σχεδιασθεί με στόχο τη διδασκαλία του ΑΠ και γενικά το υποστηρικτικό μαθησιακό υλικό γι’ αυτό το σκοπό, είναι αναμφίβολα λιγότερα και λιγότερο ώριμα από τα αντίστοιχα για το διαδικασιακό παράδειγμα προγραμματισμού (procedural programming paradigm) (Kölling and Rosenberg, 2001b), το οποίο έχει διδαχθεί και ερευνηθεί η διδασκαλία και η εκμάθησή του για πολύ περισσότερο χρόνο.

Στην ενότητα αυτή θα αναφερθούν οι κύριες κατηγορίες εργαλείων λογισμικού και υποστηρικτικού υλικού που έχουν αναπτυχθεί για τη διδασκαλία του ΑΠ και εντοπίστηκαν στη βιβλιογραφία. Αυτές είναι:

- οι προγραμματιστικοί μικρόκοσμοι (programming microworlds),
- εκπαιδευτικά προγραμματιστικά περιβάλλοντα (educational programming environments) για τη διδασκαλία του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού που χρησιμοποιούν τη γλώσσα Java ή υποστηρίζουν ένα τμήμα της γλώσσας,
- εργαλεία για την αναβάθμιση των δυνατοτήτων των προγραμματιστικών περιβαλλόντων,
- εργαλεία λογισμικού για την υποστήριξη της OF προσέγγισης,

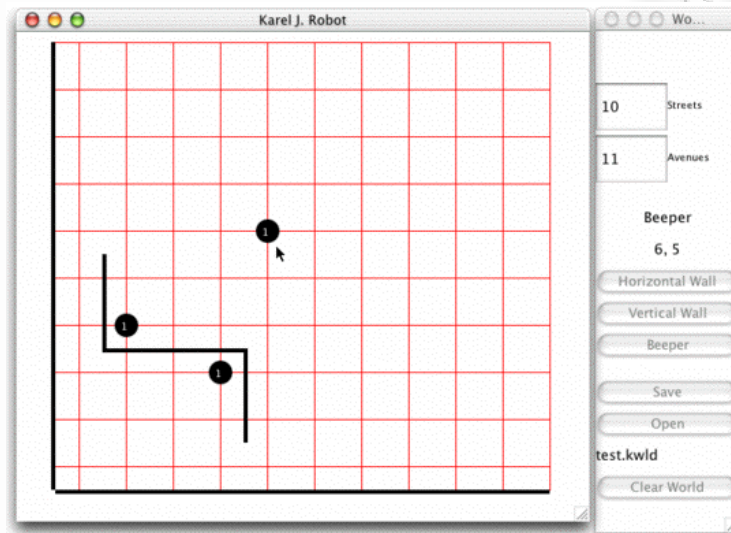
- περιβάλλοντα παιχνιδιών,
- εκπαιδευτικές γλώσσες προγραμματισμού για την εκμάθηση του ΑΠ
- και διδακτικοί πόροι ειδικά αναπτυγμένοι για την υποστήριξη της χρήσης της γλώσσας Java σε εισαγωγικά μαθήματα.

2.5.1. Προγραμματιστικοί μικρόκοσμοι

Οι προγραμματιστικοί μικρόκοσμοι (*programming microworlds*) αναπτύσσονται για καθαρά εκπαιδευτικούς σκοπούς, βασίζονται σε μια φυσική μεταφορά (*metaphor*) (χελώνα, ρομπότ, καγκουρό, τρισδιάστατα αντικείμενα σε εικονικούς κόσμους) – δηλαδή πρωταγωνιστές που “ζουν” μέσα στο περιβάλλον του μικρόκοσμου–, είναι εύχρηστοι, ενσωματώνουν δυνατότητες οπτικοποίησης και κίνησης (*animation*), συνήθως χρησιμοποιούν μια εκπαιδευτική γλώσσα προγραμματισμού και παρέχουν τη δυνατότητα της δυναμικής προσομοίωσης της εκτέλεσης των προγραμμάτων, δηλαδή της βήμα προς βήμα εκτέλεσης τους και συγχρόνως απεικόνιση του αποτελέσματος της εκτέλεσης στην κατάσταση του μικρόκοσμου. Στην κατηγορία αυτή, οι μικρόκοσμοι που έχουν αναπτυχθεί για την εισαγωγή στον ΑΠ είναι οι *Karel the Robot in Java*, *KarelJ.Robot*, *JKarelRobot*, *Jeroo*, *Alice* και *objectKarel*. Σε μεγάλο βαθμό –όπως και από τα ονόματά τους δηλώνεται– οι περισσότεροι βασίζονται στο περιβάλλον του μικρόκοσμου *Karel the Robot* (Pattis, 1981) που αναπτύχθηκε για την εισαγωγή στο διαδικασιακό προγραμματισμό με μια γλώσσα τύπου Pascal (*Pascal-like*) και της μεταγενέστερης έκδοσης του *Karel++* που ο Bergin έγραψε με τη βοήθεια των Stehlik, Roberts και Pattis για την εισαγωγή στον ΑΠ. Η δεύτερη αυτή έκδοση, ουσιαστικά εισήγαγε το μικρόκοσμο *Karel* στην αντικειμενοστρεφή εποχή με μια γλώσσα τύπου C++ (Bergin et al., 1997).

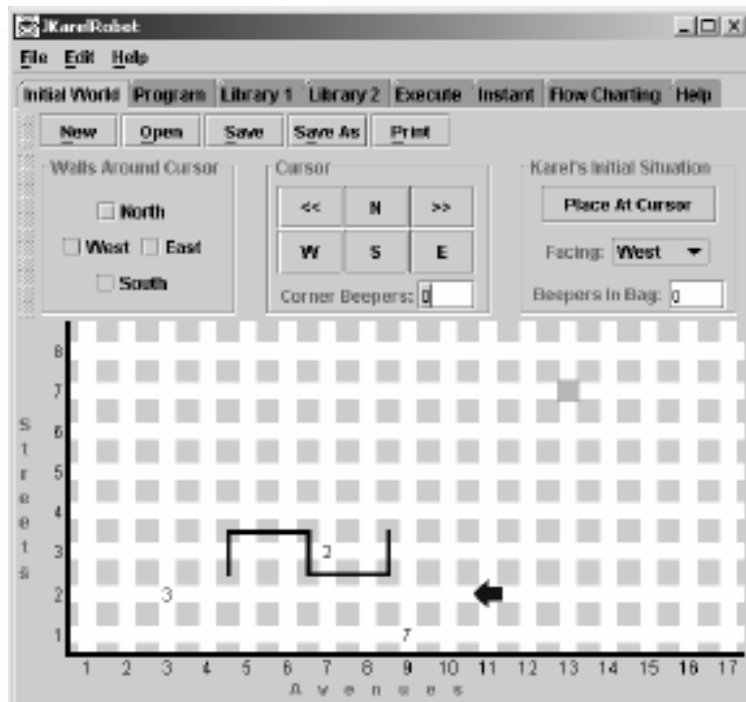
- Ο ***Karel the Robot in Java*** αποτελεί τη μετάφραση του *Karel++* σε γλώσσα *Java*, για την πλήρη υποστήριξη της γλώσσας αυτής χρησιμοποιώντας έτοιμες κλάσεις (Becker, 2001). Οι σπουδαστές σε αυτό το μικρόκοσμο δουλεύουν καθαρά σε περιβάλλον ανάπτυξης *Java* δηλαδή με κώδικα *Java*. Αναπτύχθηκε στο Πανεπιστήμιο του Waterloo στο Ontario του Καναδά, στο τμήμα της Επιστήμης των Υπολογιστών.
- Ο ***KarelJ.Robot*** (<http://csis.pace.edu/~bergin/KarelJava2ed/karelexperimental.html>) αποτελεί τη

μετάφραση που ο Bergin έκανε στο μικρόκοσμο Karel++ σε αμιγώς γλώσσα Java δηλαδή ένα αντικειμενοστρεφές Java περιβάλλον (Σχήμα 2.1.) (Bergin, 2000b; Bergin et al., 2003). Πρόσφατα εκδόθηκε βιβλίο με παραδείγματα και ασκήσεις γι' αυτόν το μικρόκοσμο (Bergin et al., 2005).

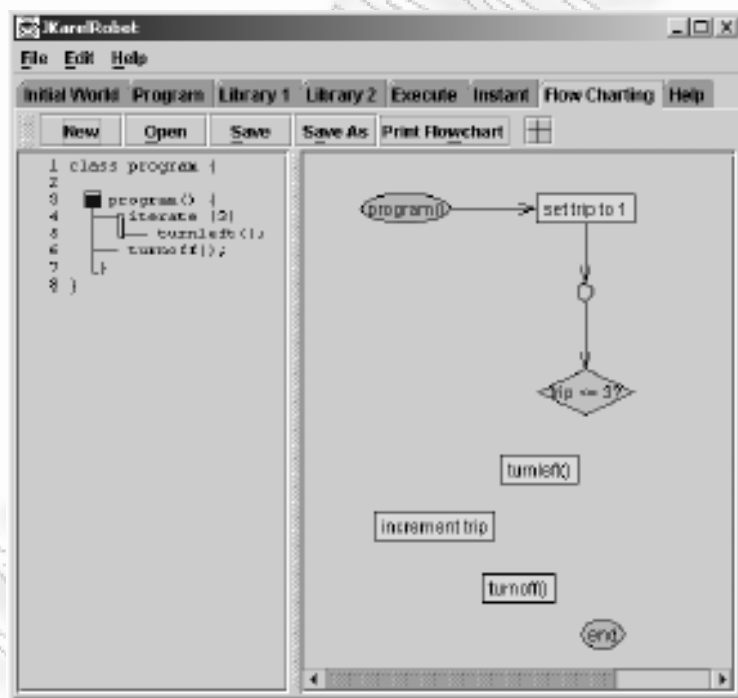


Σχήμα 2.1. Περιβάλλον KarelJ.Robot

- Ο *JKarelRobot* (<http://math.otterbein.edu/JKarelRobot/>) (Σχήμα 2.2.) είναι ένας μικρόκοσμος ανεξάρτητος πλατφόρμας (γραμμένος σε Java) και παραδείγματος προγραμματισμού, ο οποίος υποστηρίζει τύπου *Pascal*, *Java* και *Lisp* περιβάλλοντα και επίσης διαγράμματα ροής (*flow charts*) (με αντίστροφη κατεύθυνση από την παραδοσιακή, δηλαδή για τη μετατροπή έτοιμων προγραμμάτων σε διάγραμμα ροής) (Σχήμα 2.3.) και τη δυνατότητα διαγραμματικής παρουσίασης του προγράμματος με τη χρήση διαγραμμάτων δομής (*control structure diagrams, CSD*). Τα τελευταία έχουν στόχο τη βελτίωση της καταληπτότητας του πηγαίου κώδικα και είναι διαγράμματα που απεικονίζουν γενικά τη συνολική δομή κάθε τμήματος του προγράμματος. Αναπτύχθηκε από τους Buck και Stucki και χρησιμοποιήθηκε στο Otterbein College (Buck and Stucki, 2001).



Σχήμα 2.2. JKarelRobot World



Σχήμα 2.3. JKarelRobot Flow Chart

- Ο προγραμματιστικός μικρόκοσμος *Jeroo* (<http://www.jeroo.org/>) (Sanders and Dorn, 2003a; b) χρησιμοποιεί ως μεταφορά ένα σπάνιο είδος θηλαστικού παρόμοιο με τα μικρόσωμα καγκουρό της Αυστραλίας. Από κατασκευής το περιβάλλον εστιάζει στις έννοιες “αντικείμενο” και “μέθοδος” και στις δομές ελέγχου (*control structures*). Υποστηρίζει δύο στυλ γλωσσών προγραμματισμού,

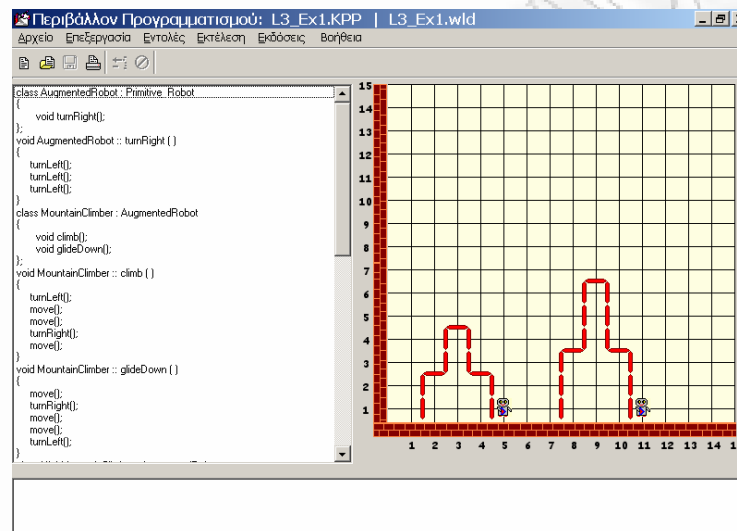
- Το εργαλείο λογισμικού *Alice* (<http://www.alice.org>) είναι ένα τρισδιάστατο διαδραστικό περιβάλλον απεικόνισης (*animation*) στο οποίο οπτικοποιούνται τα αντικείμενα και οι συμπεριφορές τους και το οποίο επιχειρεί να υλοποιήσει την OF προσέγγιση στη διδασκαλία εισαγωγικών μαθημάτων προγραμματισμού (Cooper et al., 2000; Cooper et al., 2003; Moskal et al., 2004) (Σχήμα 2.6). Η τριών διαστάσεων απεικόνιση “ενισχύει” την οπτικοποίηση των αντικειμένων παρέχοντας μια αίσθηση πραγματικότητας γι’ αυτά και παρέχει ένα ευέλικτο και εποικοδομητικό πλαίσιο για την κατανόηση των αντικειμενοστρεφών εννοιών. Ουσιαστικά ακολουθεί την παράδοση των μικρόκοσμων Karel παρέχοντας μια επιπλέον διάσταση. Οι τριών διαστάσεων κόσμοι είναι περισσότερο ρεαλιστικοί από τους αντίστοιχους δισδιάστατους. Στο περιβάλλον αυτό οι σπουδαστές μπορούν να δημιουργούν τους δικούς τους εικονικούς κόσμους. Το εργαλείο αναπτύχθηκε από τον Randy Pausch και την ερευνητική ομάδα Stage 3 research στο πανεπιστήμιο Carnegie Mellon (*Carnegie Mellon University, CMU*).



Σχήμα 2.6. Η διαπροσωπεία χρήση του περιβάλλοντος Alice

- Ο προγραμματιστικός μικρόκοσμος *objectKarel*, αποτελεί ένα αξιόλογο εκπαιδευτικό περιβάλλον το οποίο βασίζεται στον Karel++ (Ξυνόγαλος, 2002; Xinogalos et al., 2006, Xinogalos and Satratzemi, 2002) (<http://csis.pace.edu/~bergin/temp/findkarel.html>) και αναπτύχθηκε στο τμήμα Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας (Σχήμα 2.7.). Σημαντικές του πρωτοτυπίες αποτελούν, η ύπαρξη του εκδότη δομής (*structure editor*) για τη δημιουργία των προγραμμάτων, η δυνατότητα της επεξηγηματικής

οπτικοποίησης (δηλαδή της εμφάνισης επεξηγήσεων για την τρέχουσα εντολή), η σειρά μαθημάτων (*e-lessons*) που διαθέτει στα ελληνικά και στα αγγλικά και των δραστηριοτήτων για την εξοικείωση των σπουδαστών με τις βασικές αντικειμενοστρεφείς έννοιες και τις δομές ελέγχου και η δυνατότητα καταγραφής των ενεργειών των σπουδαστών καθώς δημιουργούν προγράμματα (καταγραφή των προγραμμάτων και όλων των μεταγλωττίσεών τους) για τον προσδιορισμό δυσκολιών και παρανοήσεών τους.



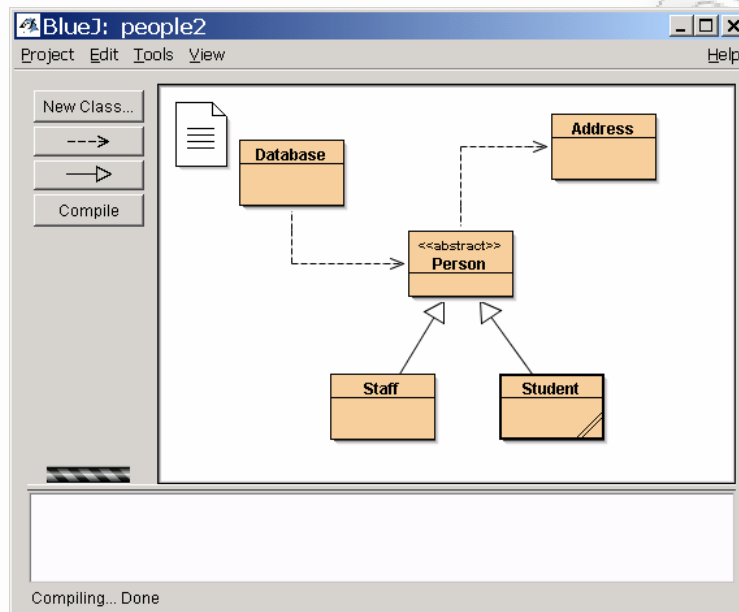
Σχήμα 2.7. Περιβάλλον προγραμματισμού objectKarel

2.5.2. Εκπαιδευτικά προγραμματιστικά περιβάλλοντα

Στη συνέχεια αναφέρονται εκπαιδευτικά προγραμματιστικά περιβάλλοντα για τη διδασκαλία του Αντικειμενοστρεφούς παραδείγματος Προγραμματισμού που χρησιμοποιούν τη γλώσσα *Java* (δηλαδή για την ανάπτυξη προγραμμάτων σε *Java*).

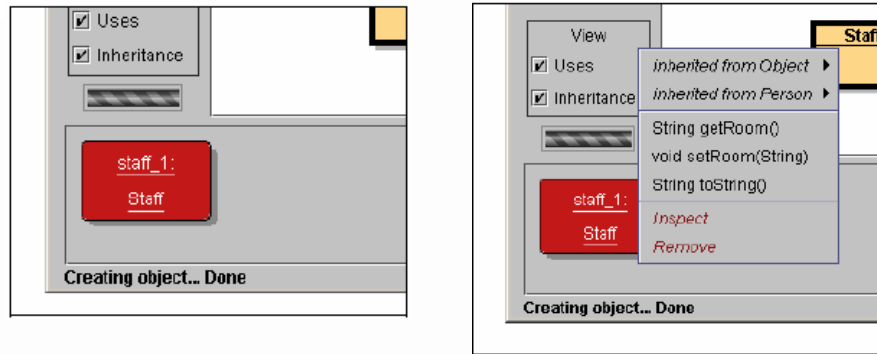
- Το **BlueJ** (<http://www.bluej.org/>) (Kölling et al., 2003) αποτελεί ένα πολύ αξιόλογο εκπαιδευτικό προγραμματιστικό περιβάλλον για την υιοθέτηση της “object first” προσέγγισης στη διδασκαλία. Στη βιβλιογραφία η χρήση του στη διδασκαλία αναφέρεται και ως “BlueJ approach” (Fjuk et al., 2006, Chapter 6, p. 80). Εστιάζει στην αντικειμενοστρέφεια και αποτελεί ένα πλήρες περιβάλλον ανάπτυξης εφαρμογών σε *Java*, υλοποιημένο καθ’ ολοκληρίαν σε *Java*. Σημαντικά χαρακτηριστικά του:
 - ♦ Είναι εύκολο στη χρήση του σε σχέση με άλλα ολοκληρωμένα προγραμματιστικά περιβάλλοντα (IDEs).

- ◆ Παρέχει τη δυνατότητα να αντιληφθείς ότι μια εφαρμογή είναι ένα σύνολο από συνεργαζόμενες κλάσεις (Σχήμα 2.8).
- ◆ Αποκτούν σε αυτό οπτική υπόσταση οι κλάσεις και τα αντικείμενα και δεν αποτελούν απλά γραμμές κώδικα. Όταν ανοίγεται ένα project παρουσιάζεται στο βασικό παράθυρο του περιβάλλοντος ένα διάγραμμα UML το οποίο οπτικοποιεί τη δομή της εφαρμογής (Σχήμα 2.8).

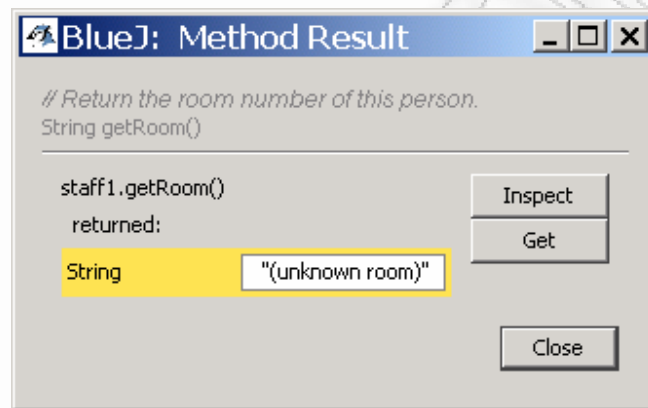


Σχήμα 2.8. Οπτικοποίηση εφαρμογής στο περιβάλλον BlueJ

- ◆ Διαθέτοντας μόνο τον κώδικα μιας κλάσης, μπορείς να δημιουργήσεις αντικείμενα και να εκτελέσεις μεθόδους χωρίς τη συγγραφή ούτε μιας γραμμής κώδικα γι' αυτό το σκοπό.
- ◆ Επιτρέπεται να αλληλεπιδράς με τις κλάσεις δημιουργώντας αντικείμενα. Μπορεί κανείς να “δει” τα αντικείμενα, να ελέγξει (inspect) τις τιμές των ιδιοχαρακτηριστικών τους (attributes), να αλληλεπιδράσει με αυτά καλώντας και εκτελώντας μεθόδους και να παρατηρήσει και ελέγξει τα αποτελέσματα της εκτέλεσής τους (Σχήματα 2.9, 2.10.).

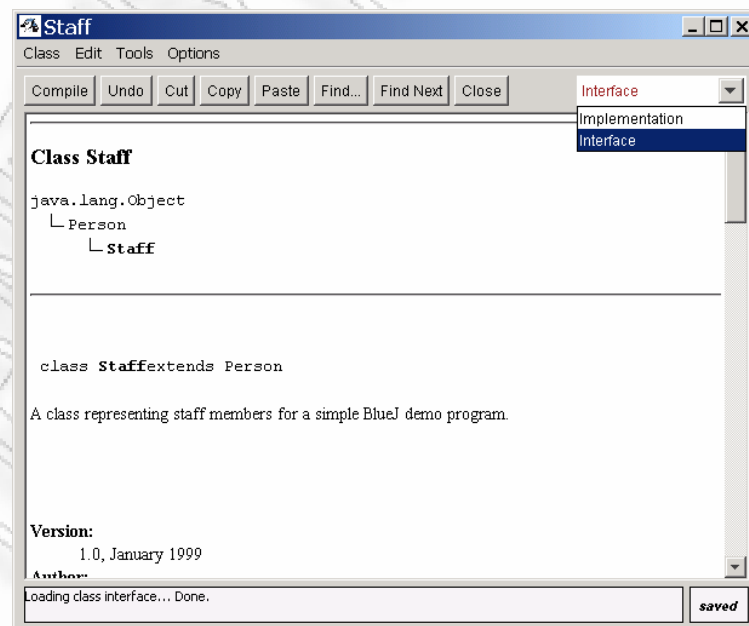


Σχήμα 2.9. Object Workbench και μέθοδοι αντικειμένου (BlueJ)



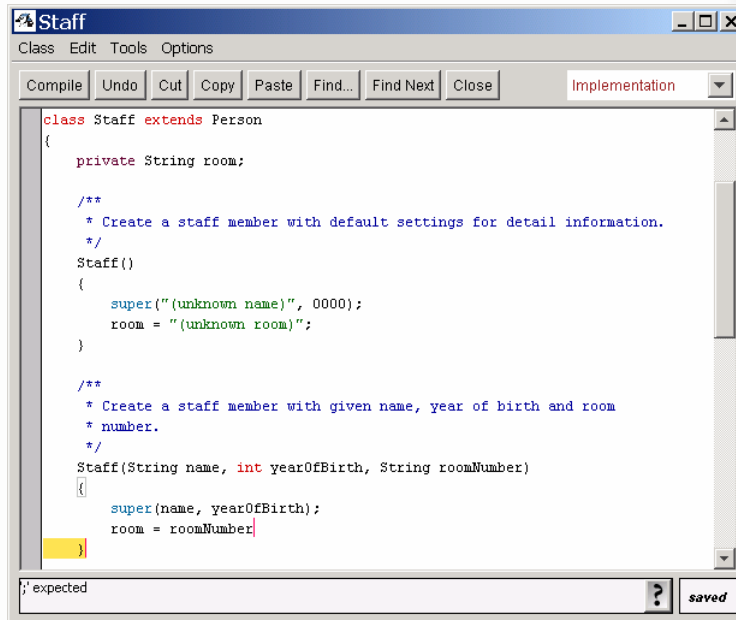
Σχήμα 2.10. Αποτέλεσμα κλήσης μεθόδου (BlueJ)

- ◆ Διαθέτει όψη υλοποίησης (implementation) και όψη διαπροσωπείας (interface) των κλάσεων (Σχήμα 2.11.).



Σχήμα 2.11. Παρουσίαση interface κλάσης (BlueJ)

- ◆ Παρέχει εύκολο πέρασμα από το οπτικό περιβάλλον στο περιβάλλον κώδικα και εύκολη μεταγλώττιση με επισήμανση των λαθών κατευθείαν στο συντάκτη του κώδικα, με τονισμό της γραμμής του λάθους και επίδειξη του κειμένου του μηνύματος λάθους (Σχήμα 2.12.).



```
class Staff extends Person
{
    private String room;

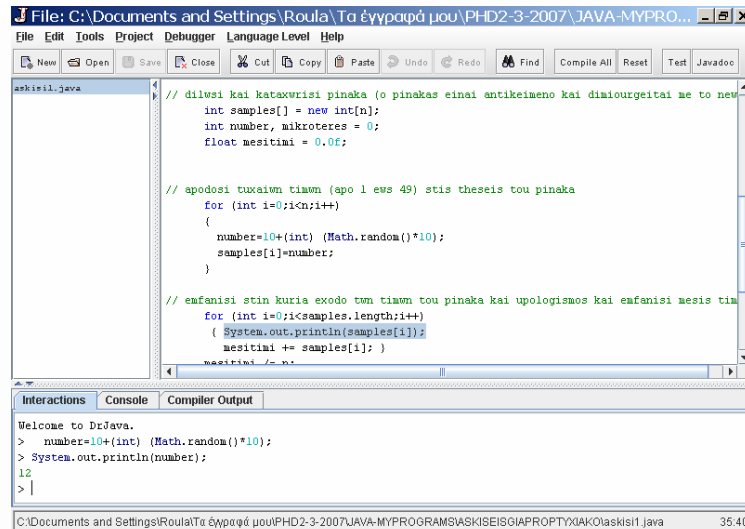
    /**
     * Create a staff member with default settings for detail information.
     */
    Staff()
    {
        super("(unknown name)", 0000);
        room = "(unknown room)";
    }

    /**
     * Create a staff member with given name, year of birth and room
     * number.
     */
    Staff(String name, int yearOfBirth, String roomNumber)
    {
        super(name, yearOfBirth);
        room = roomNumber;
    }
}
```

Σχήμα 2.12. Implementation κλάσης - Επισήμανση λάθους κώδικα (BlueJ)

Το περιβάλλον BlueJ συνοδεύεται και από αντίστοιχο βιβλίο για την πλήρη υποστήριξη μέσω αυτού της ΟΕ προσέγγισης (Barnes and Kölling, 2005). Στο βιβλίο ακολουθείται μια “project driven” ακολουθία για την παρουσίαση του υλικού του.

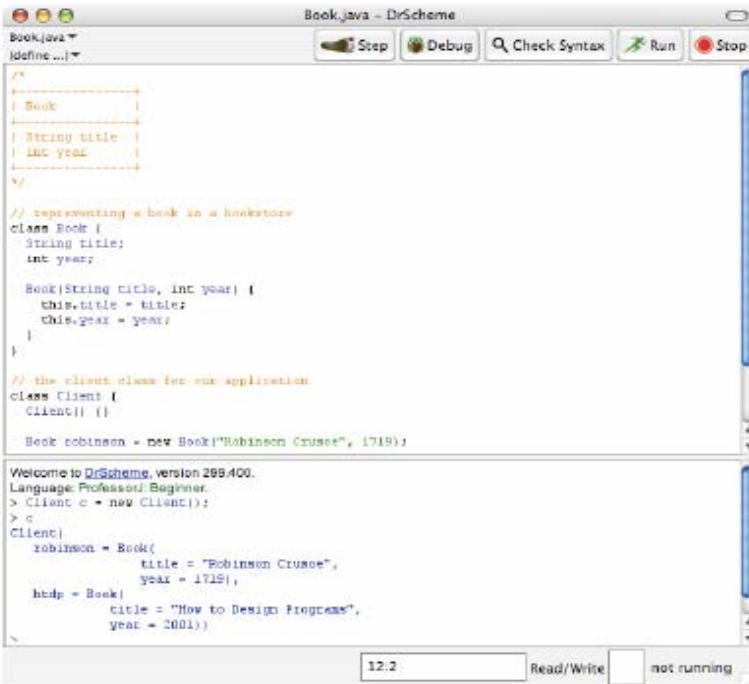
- Το *DrJava* (<http://drjava.sourceforge.net/>, προσπέλαση 2/1/2007) αποτελεί ένα παιδαγωγικό προγραμματιστικό περιβάλλον Java το οποίο εισάγει ομαλά τους σπουδαστές στη συγγραφή προγραμμάτων (Allen et al., 2002). Επιτρέπει την ανάπτυξη, τον έλεγχο και την αποσφαλμάτωση προγραμμάτων Java με διαδραστικό και επαυξητικό τρόπο. Υποστηρίζει τον κύκλο “read-eval-print loop” (read, evaluate, print, REPL) ο οποίος και διευκολύνει την επαυξητική ανάπτυξη των προγραμμάτων. Ουσιαστικά αποτελεί ένα διερμηνευτή Java (Java interpreter) που επιτρέπει διαδραστικό έλεγχο της εκτέλεσης εντολών και αλληλεπίδραση σε επίπεδο αντικειμένου, μέσω ενός διαδραστικού παραθύρου (Kölling et al., 2003) στο οποίο οι σπουδαστές μπορούν να εισάγουν εκφράσεις και εντολές Java και να δουν άμεσα τα αποτελέσματά τους χωρίς την ανάγκη μεθόδου main() και της χρήσης ενός εξωτερικού μεταγλωττιστή (Σχήμα 2.13.). Δεν παρέχει γραφική αναπαράσταση των κλάσεων και των αντικειμένων όπως το BlueJ.



Σχήμα 2.13. Περιβάλλον DrJava

- Παρόμοιο περιβάλλον, είναι το **BeanShell** (<http://www.beanshell.org/>, προσπέλαση 2/1/2007). Και αυτό αποτελεί ένα διερμηνευτή Java (Java interpreter) που επιτρέπει διαδραστικό έλεγχο της εκτέλεσης εντολών και αλληλεπίδραση σε επίπεδο αντικειμένου (Kölling et al., 2003). Δεν παρέχει γραφική αναπαράσταση των κλάσεων και των αντικειμένων όπως το BlueJ.
- Το **ProfessorJ** (Gray and Flatt, 2003; Proulx and Gray, 2006) (<http://www.professorj.org/>, προσπέλαση 15/1/2007) (Σχήμα 2.14.), αποτελεί ένα προγραμματιστικό περιβάλλον για την Java, που διαθέτει επίπεδα παρουσίασης των δομών της γλώσσας (Beginner, Intermediate, Advanced), παρέχοντας ένα απλοποιημένο ενδιαμέσο χρήστη στο μεταφραστή Java και το σύστημα εκτέλεσης Java (Java Virtual Machine, JVM). Είναι επέκταση του προγραμματιστικού περιβάλλοντος DrScheme της γλώσσας Scheme, αλλά αποτελεί περιβάλλον Java. Κάθε επίπεδο υποστηρίζει ένα υποσύνολο της γλώσσας. Με αυτό τον τρόπο οι σπουδαστές εισάγονται σταδιακά στις συντακτικές και σημασιολογικές λεπτομέρειες της γλώσσας, έχοντας στη διάθεσή τους ένα περιβάλλον το οποίο ακολουθεί την πρόοδο του μαθήματος που παρακολουθούν, που είναι μια βαθμιαία έκθεση σε αυξανόμενης πολυπλοκότητας δομές. Επιπλέον τα μηνύματα λάθους που το περιβάλλον *ProfessorJ* εμφανίζει, βασίζονται στο επίπεδο γνώσεων που αναμένεται να διαθέτει ένας αρχάριος προγραμματιστής και προσαρμόζονται κατάλληλα σε κάθε υποσύνολο του περιβάλλοντος με το τρέχον επίπεδο κατανόησης της γλώσσας που οι σπουδαστές διαθέτουν. Όλα τα τμήματα του περιβάλλοντος –η γλώσσα, τα μηνύματα λάθους, το περιβάλλον ελέγχου, οι

βιβλιοθήκες υποστήριξης— είναι σε συμφωνία με το επίπεδο του κάθε υποσυνόλου (Powers et al., 2006). Είναι πλήρως ανεπτυγμένα τα Beginner και Intermediate επίπεδα και διαθέσιμα στη διεύθυνση download.plt-scheme.org/drscheme. Το επίπεδο Advanced έχει αναπτυχθεί και είναι στη φάση ελέγχου (Proulx and Gray, 2006).



```
Book.java - DrScheme
[define ...]
[Stop] [Debug] [Check Syntax] [Run] [Stop]

//
//
//
//
// representing a book in a bookstore
class Book {
  String title;
  int year;

  Book(String title, int year) {
    this.title = title;
    this.year = year;
  }
}

// the client class for our application
class Client {
  Client() {}

  Book robinson = new Book("Robinson Crusoe", 1719);
}

Welcome to DrScheme, version 299.409.
Language: Professor/Beginner.
> Client c = new Client();
> c
Client()
  robinson = Book(
    title = "Robinson Crusoe",
    year = 1719);
  hudp = Book(
    title = "How to Design Programs",
    year = 2001);
```

Σχήμα 2.14. Περιβάλλον DrScheme – ProfessorJ

- Το περιβάλλον **JPie** (*Java Programmer's Interactive Environment*) (<http://jpie.cse.wustl.edu/>, προσπέλαση 15/1/2007) είναι ένα εκπαιδευτικό ολοκληρωμένο προγραμματιστικό περιβάλλον σχεδιασμένο να κάνει προσιτή σε ευρύτερο “κοινό” τη δύναμη της αντικειμενοστρεφούς ανάπτυξης λογισμικού (Goldman, 2003). Είναι υλοποιημένο σε Java και είναι συμβατό με μια ποικιλία από πλατφόρμες. Αναπαριστά τα προγράμματα γραφικά και όχι με κείμενο. Υποστηρίζει τη “ζωντανή” κατασκευή κώδικα σε Java με οπτικό τρόπο, μέσω άμεσου χειρισμού γραφικών αναπαραστάσεων προγραμματιστικών εννοιών (αφαιρέσεων) και δομών (όπως κλάσεις, υποκλάσεις, μεταβλητές στιγμιότυπου (type και scope), μέθοδοι, event handler μέθοδοι κ.ά.). Η υψηλού επιπέδου διαδραστικότητα που υποστηρίζει, στηρίζεται στην ιδέα της *δυναμικής κλάσης* (*dynamic class*) που η “υπογραφή” της (signature) και η υλοποίησή της μπορούν να μεταβάλλονται “ζωντανά” ακόμα και όταν έχουν δημιουργηθεί αντικείμενά

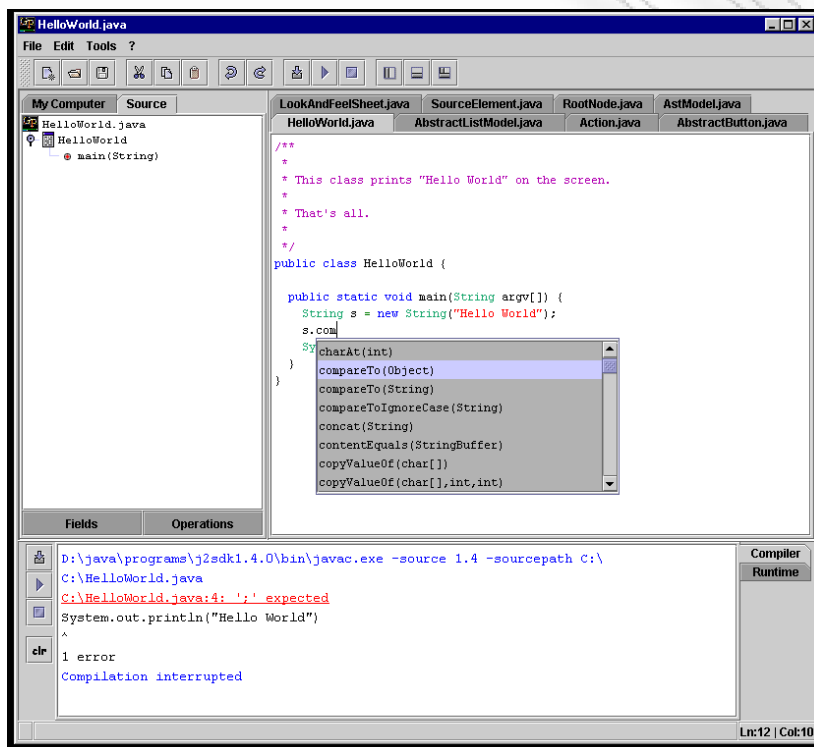
της κατά την εκτέλεση του προγράμματος, εξαλείφοντας έτσι τον κύκλο συγγραφή-μετάφραση-εκτέλεση (edit-compile-run). Οι χρήστες χειρίζονται τις γραφικές αναπαραστάσεις για να επιφέρουν αλλαγές στο εκτελούμενο πρόγραμμα. Πολλές λειτουργίες εκτελούνται με τη μέθοδο “σύρε και άφησε” (“drag and drop”). Η δυνατότητα χειρισμού εννοιών γραφικά και όχι ως μια γραμμική σειρά χαρακτήρων, σε συνδυασμό με την υποστήριξη “ζωντανής” μεταβολής του λογισμικού παρέχει έναν φυσικότερο τρόπο ανάπτυξης λογισμικού και αυξάνει το επίπεδο της αφαίρεσης. Έχει αναπτυχθεί στο Τμήμα Computer Science and Engineering του Washington University στο St. Louis του Missouri. Υποστηρίζει τη διδασκαλία ενός “concepts first” προγράμματος μαθημάτων οργανωμένου με γνώμονα τις θεμελιώδεις προγραμματιστικές έννοιες (“big ideas”) (Goldman, 2004) που παρουσιάστηκε στην προηγούμενη ενότητα αυτού του κεφαλαίου (“concepts first” στρατηγική, υποενότητα 2.4.2.1.).

- Το **Javina** αποτελεί ένα μεταγλωττιστή (compiler) για προγράμματα Java γραμμένα από αρχάριους σπουδαστές. Παρέχει τη δυνατότητα να συμπεριλαμβάνεις στον πηγαίο κώδικα Java, προϋποθέσεις (pre-conditions) και μεταπροϋποθέσεις (post-conditions) που πρέπει να ισχύουν σε μεθόδους κλάσεων, με τη μορφή συγκεκριμένου στυλ σχολίων. Το Javina, επεξεργάζεται αυτά τα σχόλια και ενσωματώνει τις προϋποθέσεις και μεταπροϋποθέσεις στις κλάσεις που παράγει. Όταν οι κλάσεις αυτές χρησιμοποιούνται στο χρόνο εκτέλεσης, ανιχνεύεται και αναφέρεται κάθε παραβίαση αυτών των προϋποθέσεων και επιπλέον παράγονται αφηρημένες οπτικοποιήσεις των αντικειμένων που δημιουργούνται από τις κλάσεις (Turner and Zachary, 2001). Οι δυνατότητες αυτές, ενισχύουν την ικανότητα των σπουδαστών να σκέφτονται αφηρημένα για τα συστατικά μέρη ενός προγράμματος (π.χ. τι κάνει μια μέθοδος, τι αναπαριστά μια κλάση) και να αποφεύγεται η τάση να αντιλαμβάνονται ένα πρόγραμμα ως μια αδόμητη συλλογή εντολών και εκφράσεων.
- Παρόμοια συστήματα αποτελούν τα **iContract** και **JML** (Turner and Zachary, 2001). Το πρώτο επιτρέπει την προσθήκη ελέγξιμων ισχυρισμών (assertions) σε προγράμματα Java με τη χρήση αναγνωριστικών σχολίων. Αυτά τα σχόλια ενσωματώνονται στον παραγόμενο κώδικα byte (byte code) και ελέγχονται κατά τη διάρκεια της εκτέλεσης. Το δεύτερο είναι αρκετά όμοιο με το Javina.

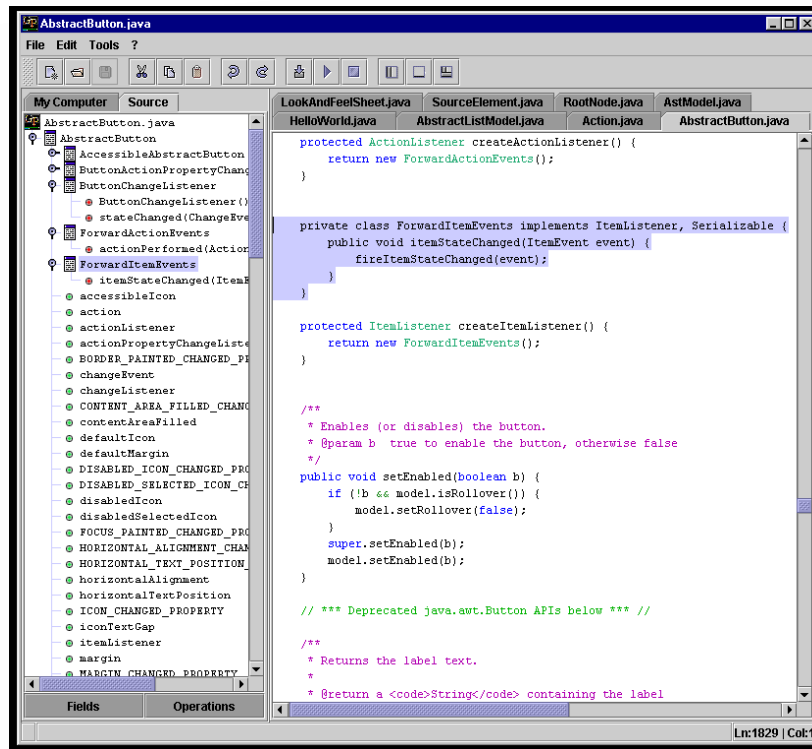
Σε πρόσφατη ανακοίνωση που έγινε στο 5^ο Πανελλήνιο Συνέδριο “Οι ΤΠΕ στην Εκπαίδευση” (Σατρατζέμη κ.ά., 2006), παρουσιάστηκε μια επισκόπηση μερικών

εκπαιδευτικών προγραμματιστικών περιβαλλόντων. Η ανακοίνωση αυτή περιελάμβανε τη σύντομη παρουσίαση και συγκριτική ανάλυση των περιβαλλόντων *BlueJ*, *DrJava*, *Ginipad*, *JGrasp* και *jCreator LE*.

- Το **GiniPad** (<http://www.mokabyte.it/ginipad/english.htm>, προσπέλαση 3/1/2007) διαθέτει παράθυρο στο οποίο παρουσιάζει τις κλάσεις, τις μεθόδους, τις ιδιότητες και τις διασυνδέσεις του πηγαίου κώδικα με δενδρική μορφή. Επισημαίνει στοιχεία του κώδικα με χρώματα και επιτρέπει τη μετάβαση από ένα λάθος μεταγλώττισης στο σημείο του κώδικα που βρίσκεται το λάθος. Δε διαθέτει αποσφαλματωτή και δυνατότητα βηματικής εκτέλεσης (Σχήματα 2.15., 2.16.).

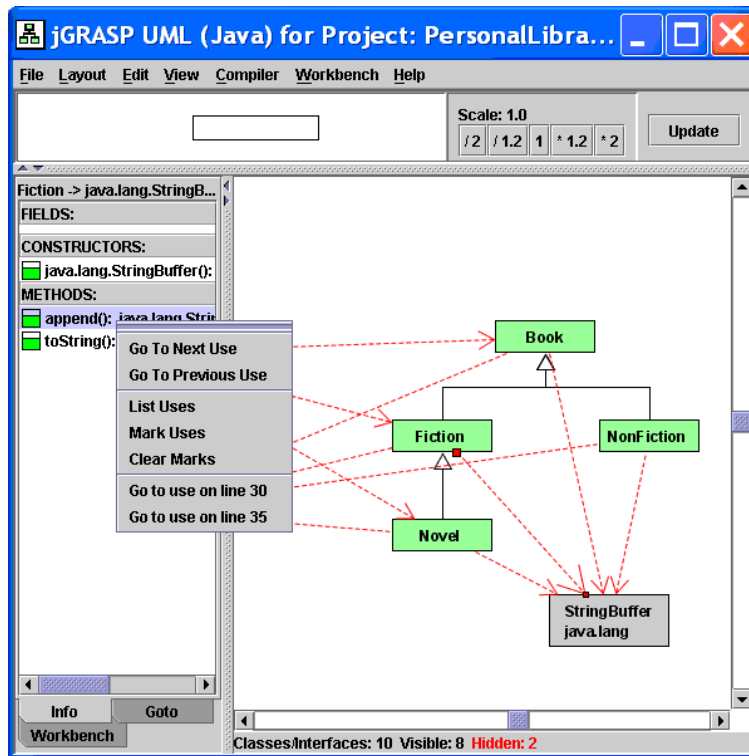


Σχήμα 2.15. Περιβάλλον GiniPad

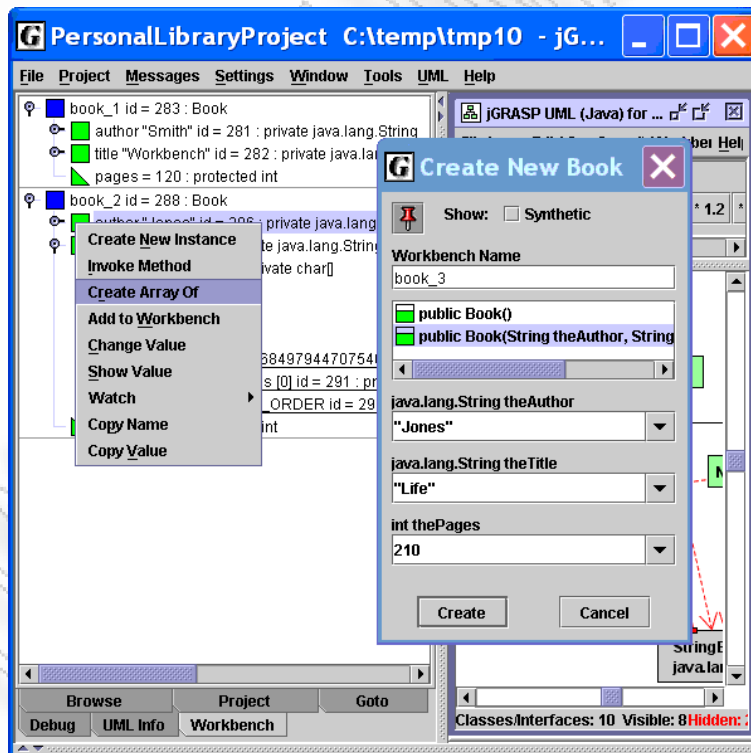


Σχήμα 2.16. Ginipad - Source browser

- Το *JGrasp* (Hendrix et al., 2004) (<http://www.eng.auburn.edu/grasp/>, προσπέλαση 3/1/2007) υποστηρίζει την επισήμανση στοιχείων του κώδικα με χρώματα και επίσης τη διαγραμματική παρουσίαση του προγράμματος με τη χρήση διαγραμμάτων δομής (control structure diagrams (CSD)). Τα τελευταία έχουν στόχο τη βελτίωση της καταληπτότητας του πηγαιού κώδικα και είναι διαγράμματα που απεικονίζουν τις δομές ελέγχου, τη ροή των εντολών και γενικά τη συνολική δομή κάθε τμήματος του προγράμματος και όλο αυτό μέσα στο χώρο που προκύπτει από το indentation του πηγαιού κώδικα. Διαθέτει πρότυπα κώδικα (templates) για τη δημιουργία κλάσεων, μεθόδων και δομών ελέγχου και επανάληψης, επισημαίνει το πρώτο λάθος μεταγλώττισης με αλλαγή χρώματος καθώς και τη γραμμή που πιθανά βρίσκεται το λάθος και παρέχει έναν οπτικό αποσφραματωτή που επιτρέπει την εισαγωγή σημείων διακοπής και τη βηματική εκτέλεση του προγράμματος. Δημιουργεί διαγράμματα UML και οι σπουδαστές έχουν τη δυνατότητα να δημιουργήσουν αντικείμενα των κλάσεων του διαγράμματος UML ή γενικά οποιασδήποτε κλάσης Java και να αλληλεπιδράσουν με αυτά καλώντας και εκτελώντας μεθόδους στην περιοχή workbench του περιβάλλοντος (Σχήματα 2.17, 2.18.).



Σχήμα 2.17. jGrasp – UML

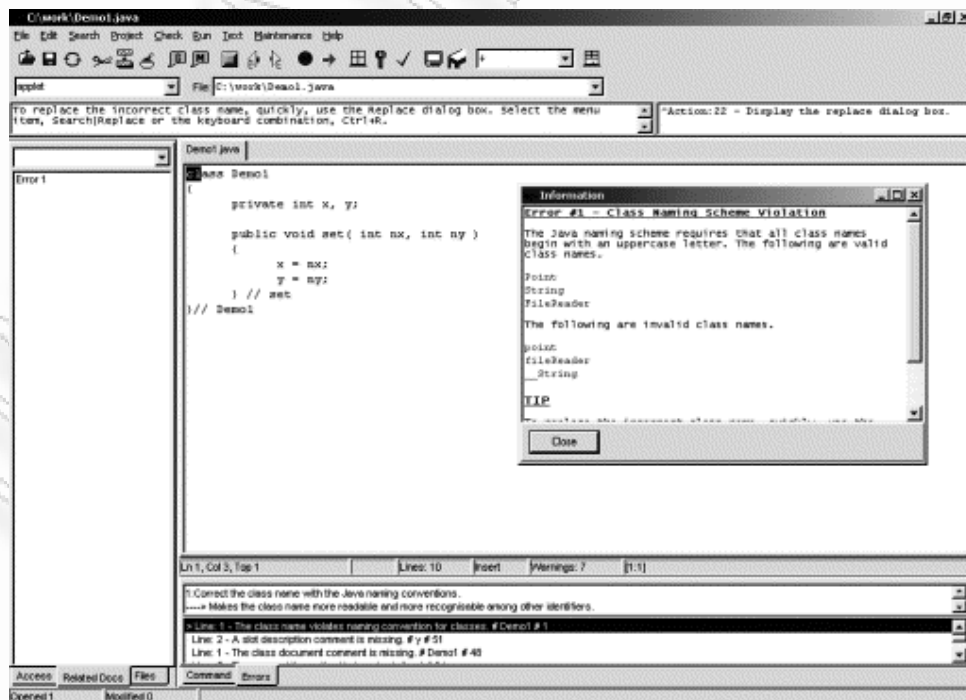


Σχήμα 2.18. jGrasp - Workbench

- Το *jCreator LE* (<http://www.jcreator.com/index.htm>, προσπέλαση 3/1/2007) παριστάνει σε δενδρική μορφή τα αρχεία ενός project, επιτρέπει την προσθήκη κλάσεων μέσω προτύπων, παρέχει μια σύνοψη της δομής κάθε κλάσης και

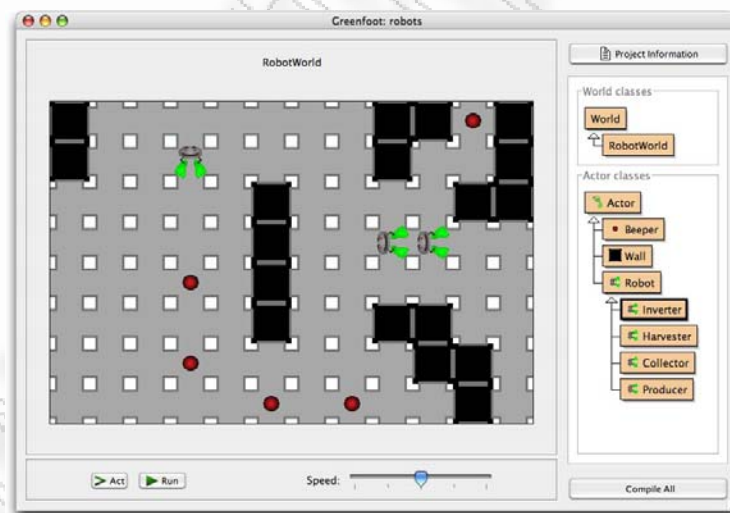
δυνατότητα προσθήκης στοιχείων στην κλάση μέσω προτύπων. Δε διαθέτει αποσφαλματωτή και δυνατότητα βηματικής εκτέλεσης.

- Το σύστημα **D-ChK** (Σχήμα 2.19.) αποτελεί ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για τη γλώσσα Java που επιτρέπει ισχυρότερη σύνδεση μεταξύ θεωρίας και πράξης, ενσωματώνοντας αρκετές βελτιώσεις στη λειτουργία και τη διεπιφάνεια χρήσης (Depradine and Gay, 2004). Το περιβάλλον λειτουργεί ως εμπρόσθιο άκρο (front end) για το Java Software Development Kit (SDK). Μερικά από τα σημαντικά χαρακτηριστικά του είναι: (i) ελέγχει τον κώδικα καθώς πληκτρολογείται και εμφανίζει τα συντακτικά λάθη σε πραγματικό χρόνο, δηλαδή κατά το χρόνο συγγραφής του προγράμματος, (ii) παρέχει πρότυπα (templates) τμημάτων κώδικα, εξοικονομώντας έτσι πολύτιμο χρόνο και προσπάθεια των σπουδαστών ελαττώνοντας λάθη που οφείλονται σε εσφαλμένη δομή του προγράμματος, (iii) παρέχει απευθείας πρόσβαση στα Java SDK εργαλεία, μειώνοντας έτσι την πολυπλοκότητα της διεπιφάνειάς τους και αποτρέποντας τον κίνδυνο να αποσπαστεί η προσοχή των σπουδαστών από τη μαθησιακή διαδικασία. Αποτελεί συνολικά ένα ολοκληρωμένο προγραμματιστικό περιβάλλον που μπορεί να έχει έναν ενεργό ρόλο στη μαθησιακή διαδικασία παρέχοντας τις απαραίτητες βελτιώσεις στη διεπιφάνεια χρήσης.

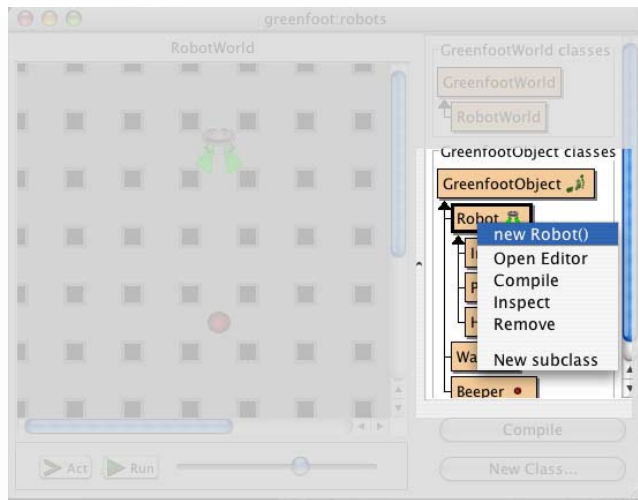


Σχήμα 2.19. Version 3 του D-ChK εργαλείου

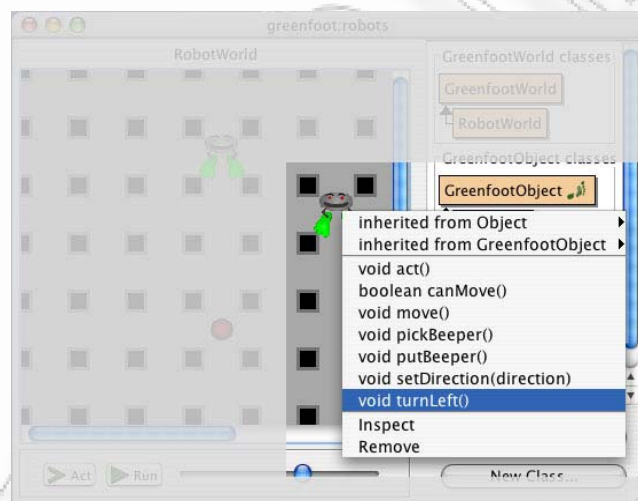
- Το **greenfoot** (<http://www.greenfoot.org>, προσπέλαση 23/1/2007) αποτελεί ένα εργαλείο που στοχεύει στην υποστήριξη της διδασκαλίας του ΑΠ σε εκπαιδευόμενους σχολικής ηλικίας (Λυκείου ή χαμηλότερης βαθμίδας). Σχεδιάστηκε συνδυάζοντας τα πλέον χρήσιμα χαρακτηριστικά μερικών ήδη υπαρχόντων εργαλείων. Κυρίως ο σχεδιασμός του εμπνεύστηκε από την απλότητα και την πολύ καλή οπτικοποίηση των αντικειμένων, της κατάστασής τους και της συμπεριφοράς τους (object visualization), που διαθέτει το περιβάλλον του Karel the Robot (microworld) και από την ευελιξία και τη δυνατότητα άμεσης αλληλεπίδρασης (direct interaction) με τα αντικείμενα και τις κλάσεις που διαθέτει το περιβάλλον BlueJ (Henriksen and Kölling, 2004). Αποτελεί συνδυασμό ενός πλαισίου για τη δημιουργία προγραμμάτων σε Java που μπορούν να οπτικοποιηθούν σε δισδιάστατο πλέγμα (two-dimensional grid) και ενός ολοκληρωμένου περιβάλλοντος ανάπτυξης (με class browser, editor, compiler, execution control, debugger κ.λ.π.) κατάλληλο για αρχάριους προγραμματιστές (Σχήματα 2.20., 2.21., 2.22.).



Σχήμα 2.20. Class structure και οπτικοποίηση κατάστασης-συμπεριφοράς αντικειμένων (greenfoot)



Σχήμα 2.21. Δημιουργία αντικειμένου (greenfoot)



Σχήμα 2.22. Μέθοδοι αντικειμένου (greenfoot)

- Το εργαλείο *P-coder* (Roy and Armarego, 2004) παρέχει ένα πλήρες περιβάλλον για την επίδειξη και υλοποίηση πολλών αντικειμενοστρεφών εννοιών για τους αρχάριους χρήστες. Εστιάζει στην ανάπτυξη και κατανόηση των βασικών αρχών μέσω γραφικής και κειμενικής παράστασης ψευδοκώδικα, αναπαριστώντας τα σημαντικά προγραμματιστικά στοιχεία σε διάγραμμα δενδρικής δομής. Παρέχει πολλαπλές όψεις όπως Design View, Class View, Code View και Object View. Οι χρήστες του συστήματος αυτού έχουν τη δυνατότητα σχεδίασης, υλοποίησης, μετάφρασης και εκτέλεσης πλήρων προγραμμάτων με το συνήθη τρόπο και υποστηρίζει τη διαδραστική δημιουργία αντικειμένων. Παράγει κώδικα Java χωρίς να την υποστηρίζει πλήρως, αλλά αυτό δεν επηρεάζει τους αρχάριους χρήστες.

2.5.3. Εργαλεία αναβάθμισης των δυνατοτήτων προγραμματιστικών περιβαλλόντων

- Το *Java Tool set* είναι ένα σύνολο από τρεις σχετικές μεταξύ τους εφαρμογές γραμμένες σε Java για την ενίσχυση των αρχάριων στη μεταγλώττιση προγραμμάτων Java και κατ' επιλογή τον έλεγχο των προγραμμάτων για το στυλ του κώδικα, τη μορφή του, το σχολιασμό του, την αναμόρφωση και τη δημιουργία εσοχών σε αυτόν (indentation) (τη στοίχισή του σύμφωνα με τη δομή του) (Lang, 2002). Οι εφαρμογές είναι:
 - ◆ Ο *Java Pre-Scanner* για την παραγωγή κατά τη μεταγλώττιση μηνυμάτων λάθους που είναι καταλληλότερα για τους αρχάριους προγραμματιστές από αυτά που παράγει το Software Development Kit της Java (SDK). Όταν μεταγλωττίζεται ένα πρόγραμμα ο Java pre-scanner εκτελείται πριν τον SDK javac μεταφραστή και αν αναφέρει λάθη, η μεταγλώττιση από τον SDK μεταφραστή μπορεί να παραλειφθεί.
 - ◆ Ο *Instant Feedback* εκτελεί ειδικούς ελέγχους και εκτελείται κατ' απαίτηση του σπουδαστή. Οι έλεγχοι αφορούν σε προβλήματα σχετικά με το στυλ του κώδικα, τη μορφή του και το κατά πόσο είναι επαρκώς σχολιασμένος. Οι κύριες λειτουργίες του είναι η επιβεβαίωση ότι ο κώδικας είναι γραμμένος με εσοχές (στοίχιση σύμφωνα με τη δομή του, indentation) και ότι τα σχόλια έχουν τοποθετηθεί στα σημεία που πρέπει.
 - ◆ Ο *Pretty Printer* εκτελείται επίσης κατ' απαίτηση του σπουδαστή, αναμορφώνει και δημιουργεί εσοχές (indentation) στον κώδικα.
- Το *Espresso* είναι ένα εκπαιδευτικό εργαλείο για προγραμματισμό με Java (Hristova et al., 2003). Αποτελεί έναν προ-μεταγλωττιστή (pre-compiler) δηλαδή ένα πρόγραμμα που εκτελείται πριν τη μετάφραση του προγράμματος για να επισημάνει με τρόπο πιο κατανοητό στους αρχάριους προγραμματιστές τα λάθη απ' ό,τι ο καθαυτό μεταγλωττιστής (compiler) και για να προτείνει τρόπους διόρθωσης των λαθών. Με αυτό τον τρόπο ενισχύει τη λειτουργία της μετάφρασης.

2.5.4. Εργαλεία λογισμικού για την υποστήριξη της “objects first” προσέγγισης

Στη συνέχεια αναφέρονται εργαλεία λογισμικού για την υποστήριξη της OF προσέγγισης στη διδασκαλία εισαγωγικών μαθημάτων. Αυτά είναι:

- Το *Java power tools (JPT)* (http://www.ccs.neu.edu/jpt/jpt_2_3/index.htm, προσπέλαση 6/1/2007) (Rasala et al., 2001; Proulx et al., 2002; Cooper et al., 2003) παρέχει ένα περιεκτικό, γραφικό περιβάλλον διασύνδεσης χρήστη (Graphics User Interaction, GUI) το οποίο αποτελείται από αρκετές κλάσεις. Οι σπουδαστές αλληλεπιδρούν με το GUI και μαθαίνουν τη συμπεριφορά των κλάσεων του μέσω αυτής της αλληλεπίδρασης. Είναι εργαλείο που βασίζεται στον οδηγούμενο από γεγονότα προγραμματισμό (event-driven programming), ο οποίος απαιτεί το ξεκίνημα με υπάρχοντα κώδικα τον οποίο οι σπουδαστές μετατρέπουν, αλλά χρειάζεται και να γράψουν οι ίδιοι κώδικα. Επιτρέπει τη γρήγορη δημιουργία γραφικών περιβαλλόντων διασύνδεσης χρήστη (GUI) σε Java με αυτόματο έλεγχο λαθών για ό,τι εισάγει ο χρήστης. Το Java Power Tools περιλαμβάνει το *Java Power Framework (JPF)* το οποίο επιτρέπει τον άμεσο πειραματισμό (instant experimentation) και το συστηματικό έλεγχο (systematic testing). Ο χρήστης μπορεί να εισάγει κάποιες μεθόδους σε μια κλάση η οποία επεκτείνει (extends) το JPF και αυτομάτως αυτές δημιουργούν “κουμπιά” (buttons) στο JPF Button Panel που καλούν τις αντίστοιχες μεθόδους όταν πιέζονται. Με αυτό τον τρόπο γίνονται έλεγχοι και εκτελούνται οι μέθοδοι σχεδόν άμεσα.
- Διάφορες *γραφικές βιβλιοθήκες (Graphics Libraries)* (Bruce et al., 2001; Roberts and Picard, 1998) στις οποίες υπάρχει ένα είδος καμβά (canvas) επί του οποίου σχεδιάζονται αντικείμενα. Παρά το ότι παρέχονται βιβλιοθήκες, οι σπουδαστές χρειάζεται να γράψουν προγράμματα Java/C++ από κενή οθόνη (from scratch) δηλαδή να κατέχουν αρκετά τη σύνταξη κώδικα (Cooper et al., 2003).
 - ◆ Η βιβλιοθήκη *objectdraw* (<http://cortland.cs.williams.edu/~cs134/eof/library.html>, προσπέλαση 6/1/2007) αναπτύχθηκε από τους Bruce, Danyluk και Murtagh (Bruce et al., 2001) για την υποστήριξη της διδασκαλίας της Java σε αρχαίους με τη χρήση της “objects from the beginning” προσέγγισης. Υποστηρίζει

αντικειμενοστρεφή γραφικά και καθιστά δυνατή την ενσωμάτωση τεχνικών event-driven προγραμματισμού. Τα γραφικά αντικείμενα χρησιμεύουν ως πολύ καλά παραδείγματα αντικειμένων αλλά και παρέχουν οπτικό αποτέλεσμα, γεγονός που διευκολύνει τους σπουδαστές να διαπιστώσουν το αποτέλεσμα του κώδικά τους και να ανιχνεύσουν λάθη στα προγράμματά τους.

- ♦ Ένα μικρό και καλαίσθητο πακέτο γραφικών είναι και το *Nice Graphics Package (NGP)* (Alphonse and Ventura, 2003) το οποίο αναπτύχθηκε στο Πανεπιστήμιο Brown από τον Andries Van Dam, με κύριο στόχο την προσέλκυση του ενδιαφέροντος και της προσοχής των σπουδαστών και συγχρόνως την υποστήριξη της διδασκαλίας θεμελιωδών αντικειμενοστρεφών εννοιών. Επιτρέπει τη δημιουργία event-driven γραφικών προγραμμάτων με τη χρήση μόνον της κληρονομικότητας και της υπερίσχυσης μεθόδων (methods overriding). Παρέχει κλάσεις οι οποίες αμβλύνουν την πολυπλοκότητα των πακέτων AWT/Swing του Java SDK και έτσι αποτελεί έναν πολύ καλό τρόπο να χρησιμοποιηθούν event-driven γραφικά προγράμματα για τη σε βάθος κατανόηση αντικειμενοστρεφών εννοιών από τους σπουδαστές, απαλλάσσοντας τους από τη δυσκολία που παρουσιάζουν οι γραφικές βιβλιοθήκες και οι event-driven προγραμματιστικοί μηχανισμοί.

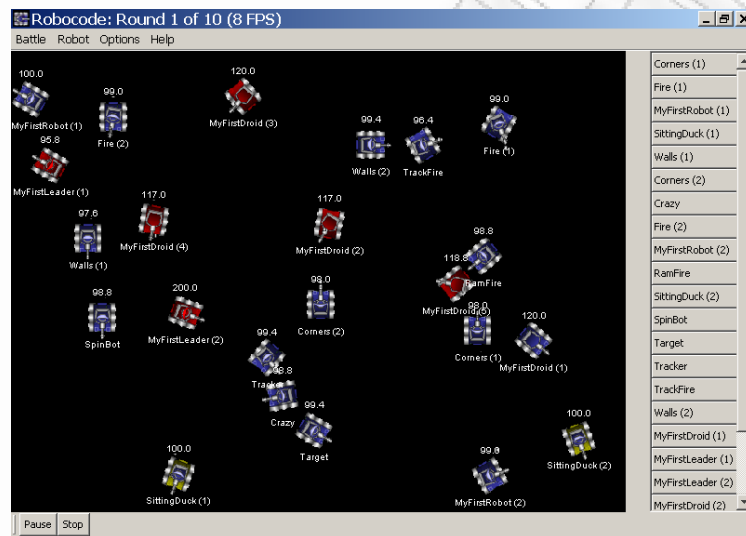
Φυσικά στην κατηγορία αυτή ανήκουν και προγραμματιστικά περιβάλλοντα που αναφέρθηκαν προηγουμένως όπως το BlueJ, το greenfoot, το JPie, το JGrasp, οι προγραμματιστικοί μικρόκοσμοι KarelJ.Robot, Jeroo, Alice και objectKarel.

2.5.5. Περιβάλλοντα παιχνιδιών

Ένα περιβάλλον παιχνιδιού για την εισαγωγή στον ΑΠ είναι το *Robocode* (<http://robocode.sourceforge.net/>). Πρόκειται για έναν προγραμματιστικό χώρο που συμβαίνουν εκσφενδονισμοί (shooting game) μεταξύ αρμάτων μάχης (tanks) (μικρά αυτόματα robots με έξι τροχούς), στον οποίο μπορεί να υλοποιηθεί λίγος κώδικας Java για να προγραμματιστεί ένα περίπλοκο παιχνίδι (Borge and Kaasbøll, 2003). Βασική έννοια αποτελούν τα ενεργά αντικείμενα (active objects), μοναδικές οντότητες με τη συμπεριφορά τους, τα οποία μπορούν να αλληλεπιδρούν με άλλα αντικείμενα με διαφορετικούς τρόπους ανάλογα με την κατάσταση. Το περιβάλλον

έχει σχεδιασθεί με τον εξής τρόπο: όταν ένα άρμα δει κάποιο άλλο ή δεχθεί πλήγμα από κάποιο άλλο κ.λ.π. δημιουργείται ένα συμβάν (event) και οι σπουδαστές έχουν τη δυνατότητα να προγραμματίσουν την αντίδραση του άρματος με συγκεκριμένο τρόπο. Όσο καλύτερα το άρμα χειρίζεται αυτά τα συμβάντα, τόσο καλύτερα αντεπεξέρχεται στη μάχη. Τα άρματα μάχονται μέχρι να απομείνει το τελευταίο από αυτά. (Σχήμα 2.23.).

Με το Robocode οι σπουδαστές μπορούν να εισαχθούν σε πολλές αντικειμενοστρεφείς έννοιες όπως υποκλάσεις, κλήση μεθόδων, ορισμός μεθόδων, χειρισμός συμβάντων, κληρονομικότητα, κελυφοποίηση, πολυμορφισμός.



Σχήμα 2.23. Περιβάλλον Robocode

2.5.6. Εκπαιδευτικές γλώσσες προγραμματισμού

- Η **PIPOO** είναι μια προσαρμοστική παιδαγωγική γλώσσα προγραμματισμού, για το πέρασμα από το διαδικασιακό στον Αντικειμενοστρεφή Προγραμματισμό (Fernández Muñoz and Peña, 2002). Είναι απόγονος της Pascal και κληρονομεί τα περισσότερα από τα χαρακτηριστικά της. Σχεδιάστηκε κατά το δυνατόν παρόμοια της και περιλαμβάνει τις ελάχιστες εκείνες αλλαγές που απαιτούνται για να διευκρινίσουν τη διαφορά μεταξύ των δύο παραδειγμάτων προγραμματισμού. Προσφέρεται για τη διδασκαλία αντικειμενοστρεφών εννοιών όταν προηγουμένως έχει προηγηθεί η διδασκαλία του διαδικασιακού παραδείγματος με τη χρήση της Pascal.

- Η **IOPL (Initial Object-Oriented Programming Language)** (Harrison et al., 2005) και η υλοποίησή της αποτελούν ένα ειδικής κατασκευής σύστημα-γλώσσα σχεδιασμένο για εισαγωγικά μαθήματα, το οποίο επιπλέον αποκλείει όλα τα συντακτικά λάθη και με έλεγχο επαυξητικού τύπου όλα τα λάθη πληκτρολόγησης στη φάση της συγγραφής του προγράμματος και όχι μετά από αυτή όπως συμβαίνει στα τυπικά περιβάλλοντα. Αυτό επιτυγχάνεται με τη χρήση ενός διαδραστικού εκδότη δομής (structure editor) ο οποίος επιτρέπει τη δημιουργία κλάσεων, κάνει ελέγχους λαθών και ο οποίος διαθέτει πρότυπα (templates) για τη συμπλήρωση (fill in) των τμημάτων που λείπουν κατά την ανάπτυξη του κώδικα. Η γλώσσα είναι ένας συνδυασμός Pascal-like και Smalltalk-like σύνταξης. Παρέχει τη δυνατότητα δημιουργίας αντικειμένων με διαδραστικό τρόπο, την κλήση των μεθόδων τους και τον έλεγχο της κατάστασής τους.

2.5.7. Διδακτικοί πόροι για την υποστήριξη της χρήσης της γλώσσας Java σε εισαγωγικά μαθήματα

Πρόσφατα έχουν αναπτυχθεί διδακτικοί πόροι για την υποστήριξη της χρήσης της γλώσσας Java σε εισαγωγικά μαθήματα. Συγκεκριμένα, το 2004 δημιουργήθηκε η ομάδα *ACM Java Task Force* με σκοπό την ανάπτυξη παιδαγωγικών πόρων για τη διδασκαλία της Java σε πρωτοετείς σπουδαστές. Τα αρχικά παραδοτέα του έργου είναι ένα σύνολο **πακέτων βιβλιοθηκών (library packages)** (σύνολα κλάσεων σχεδιασμένα για ειδικούς σκοπούς και έτοιμα για χρήση) για την αντιμετώπιση των προβλημάτων διδασκαλίας που έχουν αναφερθεί από τη διδακτική κοινότητα ως πιο σημαντικά. Επιπλέον έχει αναπτυχθεί ένα σύνολο εργαλείων λογισμικού για την υποστήριξη της διδασκαλίας της Java με άλλους τρόπους. Τέτοιο είναι ένα **on-line tutorial σύστημα (Java Task Force tutorial system)** που αναπτύχθηκε αρχικά για την τεκμηρίωση των πακέτων βιβλιοθηκών και το οποίο παρέχει τη δυνατότητα της εξερεύνησης τους από διδάσκοντες και σπουδαστές με έντονα διαδραστικό τρόπο. Παρέχει όμως επιπλέον και τη δυνατότητα στους διδάσκοντες να δημιουργήσουν τα δικά τους διαδραστικά συστήματα εκπαίδευσης για τους σπουδαστές και επίσης επεκτείνεται για να υποστηρίζει τη δημιουργία applets διαλέξεων από απλές παρουσιάσεις PowerPoint διαφανειών (Roberts, 2006).

Με την παραπάνω παρουσίαση σε αυτή την ενότητα, έγινε προσπάθεια να καταγραφούν εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα που άπτονται της διδασκαλίας του ΑΠ και προέκυψαν από τη βιβλιογραφική μελέτη κατά τη διάρκεια εκπόνησης της διατριβής. Η χρήση εργαλείων που ανήκουν σε κάποιες από τις αναφερθείσες κατηγορίες, συνεπάγονται κατά κανόνα και την υιοθέτηση μιας συγκεκριμένης μεθοδολογίας για τη διδασκαλία του γνωστικού αυτού αντικειμένου. Για παράδειγμα, μια τέτοια μεθοδολογία που έχει προταθεί γενικά για τη διδασκαλία του Προγραμματισμού, είναι η χρήση ενός προγραμματιστικού μικρόκοσμου με τα ιδιαίτερα χαρακτηριστικά που ένα τέτοιο περιβάλλον διαθέτει και τα οποία αναφέρθηκαν. Προσεγγίσεις διδακτικές που επίσης αναφέρθηκαν στην αντίστοιχη ενότητα (ενότητα 2.4.), είναι η κλασική που κάνει χρήση μιας γλώσσας προγραμματισμού και ενός τυπικού επαγγελματικού προγραμματιστικού περιβάλλοντος καθώς επίσης και η OF.

Η επιλογή των εργαλείων που κρίνονται κατάλληλα να ενσωματωθούν στη διδασκαλία, είναι άμεσα εξαρτώμενη από την προσέγγιση που θα ακολουθηθεί, τους στόχους που επιδιώκει να επιτύχει, τη στρατηγική υλοποίησής της, τις επιλογές ως προς τη γλώσσα προγραμματισμού και το ρόλο της στην όλη φιλοσοφία της προσέγγισης.

Στους Πίνακες 2.1. και 2.2. δίνεται μια συνολική παρουσίαση των κύριων εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων που παρουσιάστηκαν, με συγκριτική εμφάνιση των διδακτικών προσεγγίσεων που υποστηρίζουν, των σημαντικών χαρακτηριστικών που διαθέτουν ή όχι και της ωριμότητας χρήσης τους (Georgantaki and Retalis, 2007b). Ως βασικά χαρακτηριστικά που θα πρέπει να ληφθούν υπόψη στη συγκριτική παρουσίαση των εργαλείων-περιβαλλόντων θεωρήθηκαν τα: απλότητα (simplicity) του περιβάλλοντος, δυνατότητα αλληλεπίδρασης (interaction) (δημιουργία αντικειμένων και κλήση μεθόδων επί αυτών), οπτικοποίηση (visualization) των κλάσεων και των αντικειμένων, υποστήριξη της μοντελοποίησης (modeling), οπτικοποίηση της δομής του προγράμματος με UML ή UML-like διαγράμματα, οπτικοποίηση της δυναμικής εκτέλεσης του προγράμματος, δυνατότητα δημιουργίας skeletons κώδικα από τις γραφικές αναπαραστάσεις των κλάσεων, ύπαρξη γραφικού ενδιαμέσου, πλήρης υποστήριξη της γλώσσας Προγραμματισμού Java, εύκολη μεταγλώττιση (compilation), επισήμανση της γραμμής κώδικα με συντακτικό λάθος, δυνατότητα

αποσφαλμάτωσης (debugging). Τα περισσότερα από τα παραπάνω αναφέρονται ως βασικές κατηγορίες χαρακτηριστικών οι οποίες θα πρέπει να λαμβάνονται υπόψη κατά τη συγκριτική ανάλυση εκπαιδευτικών προγραμματιστικών περιβαλλόντων στο McIver (2002).

Ο πρώτος πίνακας περιέχει τις δύο πρώτες κατηγορίες εργαλείων λογισμικού, τους προγραμματιστικούς μικρόκοσμους και τα εκπαιδευτικά προγραμματιστικά περιβάλλοντα. Από τα περιβάλλοντα-εκπροσώπους του προγραμματιστικού μικρόκοσμου Karel the Robot, στον πίνακα έχει περιληφθεί η υλοποίηση του JKarelRobot.

Καταρχάς, η πλειοψηφία των περιβαλλόντων υποστηρίζει την OF προσέγγιση, αναδεικνύοντας και την επικράτηση αυτής της προσέγγισης στην τεχνολογία λογισμικού για την κατασκευή εργαλείων για τη διδασκαλία του ΑΠ. Από τη συγκριτική παρουσίαση, προκύπτει η υπεροχή του περιβάλλοντος BlueJ, σε χαρακτηριστικά και ωριμότητα χρήσης (Hagan and Markham 2000; Van Haaster and Hagan, 2004). Αναδεικνύεται η έλλειψη του χαρακτηριστικού της δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων από τα εκπαιδευτικά προγραμματιστικά περιβάλλοντα. Το περιβάλλον greenfoot που ενσωματώνει επιπλέον αυτό το χαρακτηριστικό, όπως ήδη αναφέρθηκε αποτελεί ένα περιβάλλον σχεδιασμένο για εκπαιδευόμενους σχολικής ηλικίας, που σε άλλη βαθμίδα εκπαίδευσης η χρήση του ίσως χαρακτηριζόταν “παιδική” και επίσης δεν υπάρχουν γι’ αυτό, αποτελέσματα χρήσης στη διδακτική πράξη.

Του περιβάλλοντος BlueJ ακολουθεί το jGrasp, αλλά δεν έχει δοκιμαστεί και δεν υποστηρίζει πλήρως την οπτικοποίηση ούτε τη σχεδίαση μοντέλων.

Από τους προγραμματιστικούς μικρόκοσμους ο Jeroo διαθέτει αρκετά σημαντικά χαρακτηριστικά, αλλά δεν προσφέρει τη δυνατότητα της πλήρους παρουσίασης της φιλοσοφίας του ΑΠ, αφού από κατασκευής δεν υποστηρίζει κληρονομικότητα και οι Jeroo γλώσσες υποστηρίζουν μόνο μια κλάση και τη δημιουργία μέχρι τεσσάρων αντικειμένων από αυτήν. Σημαντικό χαρακτηριστικό αποτελεί η συνάφεια της γλώσσας τύπου JAVA/C++/C# του περιβάλλοντος, με τη Java.

Οι μικρόκοσμοι Alice και objectKarel ακολουθούν σε χαρακτηριστικά. Σχετικά με τον objectKarel, αποτελεί σαφώς ένα πληρέστατο τέτοιου είδους περιβάλλον (προγραμματιστικός μικρόκοσμος) για την παρουσίαση της Αντικειμενοστρέφειας και των βασικών εννοιών-αρχών της, με τα πολύ σημαντικά χαρακτηριστικά του εκδότη δομής (structure editor) τον οποίο διαθέτει (και ο οποίος ελαχιστοποιεί την

πιθανότητα συντακτικών λαθών) και των μαθημάτων που ενσωματώνει κατά ενότητες (θεωρία και δραστηριότητες για την εξοικείωση του χρήστη με τις έννοιες της κάθε ενότητας).

Ο δεύτερος πίνακας περιέχει τις υπόλοιπες κατηγορίες εργαλείων λογισμικού, που η βιβλιογραφική αναζήτηση ανέδειξε. Από τη συγκριτική παρουσίαση, το Expresso προκύπτει να αποτελεί ένα καλό εργαλείο για την ενίσχυση της λειτουργίας της μετάφρασης στους αρχάριους προγραμματιστές, σε προγράμματα γραμμένα σε γλώσσα Java, χωρίς να είναι αρκετά δοκιμασμένο. Επίσης η χρήση γραφικών βιβλιοθηκών συγκεντρώνει καλά χαρακτηριστικά εργαλείου που υποστηρίζει την OF προσέγγιση, με μικρή ωριμότητα χρήσης και αυτό. Συνολικά για τα εργαλεία του δεύτερου πίνακα δεν υπάρχουν αποτελέσματα χρήσης στη διδακτική πράξη.

Όλα τα παραπάνω συμπεράσματα, αλλά και παράγοντες όπως η στρατηγική υλοποίησης της “object first” προσέγγισης και συγκεκριμένες αποφάσεις ως προς τη γλώσσα προγραμματισμού και το ρόλο της στη διδακτική προσέγγιση, έχουν οδηγήσει σε συγκεκριμένες επιλογές από τα εκπαιδευτικά εργαλεία-προγραμματιστικά περιβάλλοντα που αναφέρθηκαν, στη διδακτική προσέγγιση που σχεδιάστηκε. Οι επιλογές που έχουν γίνει και οι λόγοι που οδήγησαν σε αυτές θα εξηγηθούν στο επόμενο κεφάλαιο, στο οποίο γίνεται η πλήρης και αναλυτική παρουσίαση της προσέγγισης που σχεδιάστηκε.

Πίνακας 2.1. Συγκριτική παρουσίαση Προγραμματιστικών Μικρόκοσμων-Εκπαιδευτικών Προγραμματιστικών περιβαλλόντων για τη διδασκαλία του ΑΠ

		Προγραμματιστικοί Μικρόκοσμοι				Εκπαιδευτικά Προγραμματιστικά Περιβάλλοντα											
		JKarelRobot	Jeroo	Alice	objectKarel	BlueJ	DrJava	ProfessorJ	JPie	Javiva	GiniPad	jGrasp	jCreator	D-Chk	greenfoot	P-Coder	
Διδακτική προσέγγιση που υποστηρίζει	Classical																
	Imperative first																
	Objects first		✓	✓	✓	✓			✓	✓	✓	✓	✓		✓	✓	
Σημαντικά χαρακτηριστικά	Απλότητα	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓		
	Δυνατότητα αλληλεπίδρασης					✓	✓		✓		✓				✓	✓	
	Οπτικοποίηση		μα ¹	μα ¹	μα ¹	✓			γαε ²	ασα ³	δδκμ ⁴	αμμκ ⁵	δδκμ ⁴		✓		
	Μοντελοποίηση					✓									✓		
	Οπτικοποίηση δομής προγράμματος					✓						✓			✓		
	Οπτικοποίηση δυναμικής εκτέλεσης προγράμματος	✓	✓	✓	✓											✓	

¹ μα = μόνο οπτικοποίηση αντικειμένων
² γαε = γραφικές αναπαραστάσεις εννοιών
³ ασα = αφηρημένες οπτικοποιήσεις αντικειμένων
⁴ δδκμ = δενδρική δομή κλάσεων - ιδιοτήτων - μεθόδων
⁵ αμμκ = αντικείμενα με μορφή κειμένου

		Προγραμματιστικοί Μικρόκοσμοι				Εκπαιδευτικά Προγραμματιστικά Περιβάλλοντα										
		JKarelRobot	Jeroo	Alice	objectKarel	BlueJ	Dr.Java	ProfessorJ	JPie	Javiva	GiniPad	jGrasp	jCreator	D-Chk	greenfoot	P-Coder
	Δημιουργία skeletons κώδικα					✓									✓	
Σημαντικά χαρακτηριστικά	Γραφικό ενδιάμεσο	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	✓	
	Πλήρης υποστήριξη Java					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Εύκολο compilation		✓		✓	✓	✓	✓			✓	✓	✓	✓	✓	
	Επισήμανση γραμμής με συντακτικό λάθος		✓		✓	✓	✓				✓	✓	✓	✓	✓	
	αποσφαλματοτής (debugger)	✓				✓	✓					✓			✓	
Ωριμότητα χρήσης	Μικρή		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Αρκετή - Μεγάλη	✓				✓										

Πίνακας 2.2. Συγκριτική παρουσίαση άλλων εργαλείων και γλωσσών Προγραμματισμού για τη διδασκαλία του ΑΠ

		Αναβάθμισης πρ/κών περιβαλλόντων		Για την υποστήριξη της “objects first” προσέγγισης		games	Εκπαιδευτικές γλώσσες πρ/σμού		Διδακτικοί πόροι για την υποστήριξη της Java
		Java Tool Set	Espresso	Java Power Tools	Graphics Libraries	Robocode	PIIPOO	IOPL	Java Task Force library packages
Διδακτική προσέγγιση που υποστηρίζει	Classical								
	Imperative first								
	Objects first			✓	✓	✓			
Σημαντικά χαρακτηριστι κά	Απλότητα								
	Δυνατότητα αλληλεπίδρασης							✓	
	Οπτικοποίηση				γα ⁶	μα ⁷			
	Μοντελοποίηση								
	Οπτικοποίηση δομής πρ/τος								
	Οπτικοποίηση δυναμικής εκτέλεσης προγράμματος						✓		

⁶ γα = γραφικά αντικείμενα

⁷ μα = μόνο οπτικοποίηση αντικειμένων

		Αναβάθμισης πρ/κών περιβαλλόντων		Για την υποστήριξη της “objects first” προσέγγισης		games	Εκπαιδευτικές γλώσσες προγραμματισμού		Διδακτικοί πόροι για την υποστήριξη της Java
		Java Tool Set	Espresso	Java Power Tools	Graphics Libraries	Robocode	ΠΙΠΟΟ	ΙΟΠΛ	Java Task Force library packages
Σημαντικά χαρακτηριστικά	Δημιουργία skeletons κώδικα								
	γραφικό ενδιάμεσο								
	Πλήρης υποστήριξη Java	✓	✓		✓				✓
	Εύκολο compilation	✓	✓	✓					
	Επισήμανση γραμμής με συντακτικό λάθος		✓					✓	
	αποσφαλματοτής (debugger)								
Ωριμότητα χρήσης	Μικρή	✓	✓	✓	✓	✓	✓	✓	✓
	Αρκετή - Μεγάλη								

2.6. Συστάσεις διδασκαλίας

Σε ένα σύνολο ερευνών που αφορούν στον εντοπισμό δυσκολιών-παρανοήσεων των εκπαιδευομένων στον ΑΠ και σε βιβλιογραφικές μελέτες σχετικές με δυσκολίες στη διδασκαλία και την εκμάθηση του Προγραμματισμού γενικά και στην ανάπτυξη εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων, περιέχονται οδηγίες και συστάσεις για τη μαθησιακά αποτελεσματικότερη διδασκαλία του ΑΠ.

➤ Σημαντικές *συστάσεις διδασκαλίας* διατυπώνονται στην έρευνα, που αφορά σε αρχάριους σπουδαστές οι οποίοι παρακολουθούσαν εισαγωγικά μαθήματα ΑΠ με τη χρήση της γλώσσας Java (Fleury, 2000), όπως:

- Να δίνεται έμφαση σε δραστηριότητες ανάγνωσης κώδικα και αποσφαλμάτωσης (debugging) καθώς και σε προγραμματιστικές δραστηριότητες (programming activities).
- Είναι πολύ χρήσιμο να παρέχεται στους σπουδαστές ένα μοντέλο της μηχανής, όταν εξηγούνται μερικές ενοχλητικές κρυμμένες παρενέργειες των εντολών του προγράμματος.
- Με τον προσεκτικό και με φροντίδα σχεδιασμό προγραμμάτων, είναι δυνατό να αποφευχθούν κοινές παρανοήσεις.

Μερικοί σπουδαστές στις απαντήσεις τους σε αυτή την έρευνα, έκαναν συγκεκριμένες, αυθόρμητες συστάσεις διδασκαλίας, όπως:

- Να τους παρέχεται ένα βιβλίο ασκήσεων (workbook) με πολύ απλά προβλήματα και τις λύσεις τους, πριν αντιμετωπίσουν πιο δύσκολα δείγματα προγραμμάτων και τους ανατεθούν προγραμματιστικές ασκήσεις.
- Να χρησιμοποιούνται διαφάνειες και κείμενο που θα έχουν χρωματιστά κωδικοποιημένα τα προγράμματα (color-coded the program listings) για να διαφοροποιούνται τα συστατικά των εντολών της Java.
- Να έχουν χρόνο να ασχοληθούν με τα προγράμματα.
- Θεωρούν μεγάλο πλεονέκτημα να έχουν εξατομικευμένη βοήθεια.

Όλες οι συστάσεις που οι σπουδαστές διατυπώνουν είναι συνεπείς με τη θεωρία του εποικοδομητισμού (constructivism learning theory):

- Το βιβλίο ασκήσεων (workbook) και τα χρωματιστά κωδικοποιημένα προγράμματα (color-coded program listings) μέσα σε ένα κείμενο,

προορίζονται να ενισχύσουν την προηγούμενη γνώση πριν αποπειραθούν να καταλάβουν περίπλοκα προγράμματα.

- Το να ασχοληθούν με τα προγράμματα, τους επιτρέπει να αναρωτηθούν και να απαντήσουν ερωτήσεις σχετικές με το τρέχον επίπεδο κατανόησης που διαθέτουν.
- Η έναν προς έναν συνδρομή παρέχει βοήθεια συντονισμένη με το τρέχον επίπεδο κατανόησης.

Πολλές συστάσεις προέκυπταν έμμεσα από τις απαντήσεις των φοιτητών:

- Επειδή έδιναν μεγαλύτερη σημασία στο αν απαντούσαν σωστά ή λάθος, παρά στο αν οι εξηγήσεις που επίσης τους ζητήθηκαν να δίνουν στις απαντήσεις τους ήταν σωστές, φαίνεται ότι δεν ενδιαφέρονταν για την πραγματική κατανόηση αλλά για την απόκρυψη πιθανόν της σύγχυσής τους. Αυτό αναδεικνύει ότι δίνοντας περισσότερες ευκαιρίες στους σπουδαστές να εξηγούν τις σκέψεις τους –σε συζητήσεις, quizzes, εξετάσεις– θα τους ενθαρρύνει στο να στοχεύουν στην πραγματική κατανόηση και όχι στην απόκρυψη της σύγχυσης με επιτυχία.
- Ο εποικοδομητισμός (constructivism) προτείνει ως πιο αποδοτικό το να σχεδιάσεις κατάλληλες εργαστηριακές εμπειρίες, από το να μιλάς απλώς στους σπουδαστές.

Επίσης από την έρευνα, συμπεραίνεται ότι:

- Είναι πολύ σημαντικό να ακούς τις σκέψεις των σπουδαστών σε μη απειλητικές/δυσοίωνες γι' αυτούς συνθήκες και να μεταφράζεις τις σκέψεις και τις ενέργειες τους από τη δική τους πλευρά.
- Στη δεύτερη έρευνα που αφορά στους ίδιους σπουδαστές (Fleury, 2001) επίσης διατυπώνονται σημαντικές συστάσεις διδασκαλίας. Συγκεκριμένα:
- Η κατανόηση των καλών προγραμματιστικών τεχνικών που διδάσκουμε στους σπουδαστές έρχεται μόνο με την εμπειρία, δηλαδή πρέπει να σχεδιάζουμε κατάλληλες εργαστηριακές ασκήσεις.
 - Να δείχνουμε στους σπουδαστές την καλή αντικειμενοστρεφή σχεδίαση (OO design) από τα αρχικά στάδια της εκπαιδευτικής διαδικασίας.
 - Όταν διδάσκουμε τους σπουδαστές να διαβάζουν κώδικα, να δίνεται έμφαση στην κατανόηση του όλου προγράμματος και όχι στην κατανόηση του κώδικα γραμμή προς γραμμή.

- Να χρησιμοποιούνται στη διδασκαλία μελέτες περίπτωσης (case studies).
 - Να δίνονται ασκήσεις αποσφαλμάτωσης (debugging) και συντήρησης (maintenance) κώδικα ο οποίος δεν είναι σύμφωνος με την αρχή της κελυφοποίησης (poorly encapsulated) και στον οποίο επαναλαμβάνονται ίδια τμήματα (duplicate code).
 - Να χρησιμοποιούνται τεχνικές που θα ενθαρρύνουν τους σπουδαστές να στοχάζονται πάνω στο τι μαθαίνουν.
- Όπως ήδη έχει αναφερθεί στην ενότητα 2.2., στην έρευνα που έγινε μέσω παρατήρησης των σπουδαστών στο πλαίσιο του εξ' αποστάσεως μεταπτυχιακού μαθήματος "Object-oriented Software Technology (M868)" και κατά τη διάρκεια του επίσης εξ' αποστάσεως εισαγωγικού προπτυχιακού μαθήματος "Computing: An Object-oriented approach (M206)" και τα δύο με τη χρήση της γλώσσας Smalltalk ως πρώτη γλώσσα προγραμματισμού, οι Holland et al. (1997) έχουν καταγράψει ένα πλήθος δυσκολιών που οι σπουδαστές αντιμετωπίζουν. Στην ίδια έρευνα αναφέρονται συγκεκριμένα μέτρα που μπορούν να ληφθούν ώστε να αποφεύγονται οι δυσκολίες και να μη δημιουργηθούν παρανοήσεις, γεγονός που είναι δύσκολο να αντιμετωπιστεί όταν ήδη έχει συμβεί. Τα μέτρα αυτά αφορούν στον προσεκτικό σχεδιασμό των παραδειγμάτων που παρουσιάζονται και των ασκήσεων που ανατίθενται στους σπουδαστές, αφού ληφθούν σοβαρά υπόψη τα προβλήματα που έχουν διαπιστωθεί. Π.χ.
- Στα παραδείγματα των κλάσεων θα πρέπει:
 - ◆ για την αποφυγή της ταύτισης αντικειμένου/μεταβλητής, να υπάρχουν τουλάχιστον δύο μεταβλητές στιγμιοτύπου,
 - ◆ για την αποφυγή της παρανόησης ότι όλες οι μεταβλητές στιγμιοτύπου μιας κλάσης είναι του ίδιου τύπου, να υπάρχουν τουλάχιστον δύο μεταβλητές στιγμιοτύπου που θα αποτελούν αντικείμενα διαφορετικών κλάσεων,
 - ◆ για την αποφυγή της ταύτισης κλάσης/αντικειμένου θα πρέπει να δημιουργούνται αρκετά αντικείμενα από κάθε κλάση
 - ◆ κ.λ.π. (βλ. ενότητα 2.2.).
 - Οι σπουδαστές θα πρέπει να πειραματίζονται με ένα σύνολο παραδειγμάτων όπως:
 - ◆ Αναθέσεις πολλών μεταβλητών αναφοράς στο ίδιο αντικείμενο.
 - ◆ Ανάθεση άλλου αντικειμένου από το αρχικό σε μεταβλητές αναφοράς.

- ◆ Αντιμετάθεση των μεταβλητών αναφοράς που “συνδέονται” σε δύο αντικείμενα με τη χρήση ενδιάμεσης μεταβλητής.
 - ◆ κ.λ.π. (βλ. ενότητα 2.2.).
- Στη βιβλιογραφική μελέτη “Problems in Learning and Teaching Programming” αναφέρονται σημαντικές συστάσεις, για την ανάπτυξη αποτελεσματικού δυναμικού οπτικού μαθησιακού υλικού (δηλαδή με τη χρήση δυναμικής οπτικοποίησης) και εκπαιδευτικών στρατηγικών για τη χρήση του στο πλαίσιο του έργου “Codewitz-Minerva” (Codewitz-Minerva project), με στόχο την ενίσχυση της εκμάθησης των βασικών εννοιών και δομών του Προγραμματισμού γενικά (Ala-Mutka). Γενικά η οπτικοποίηση έχει χρησιμοποιηθεί αρκετά στο χώρο της εκπαίδευσης στην Επιστήμη των Υπολογιστών (CS Education), αφού έχει θεωρηθεί ωφέλιμη και αποτελεσματική στην κατανόηση και εκμάθηση αφηρημένων και περίπλοκων εννοιών και θεμάτων. Οι συστάσεις αυτές δεν περιορίζονται αποκλειστικά στη δημιουργία οπτικοποιημένου μαθησιακού υλικού, αλλά εμπίπτουν και στην περίπτωση γενικών συστάσεων διδασκαλίας, όπως η πρώτη από τις παρακάτω αναφερόμενες:
- Τα μαθήματα Προγραμματισμού θα πρέπει να περιέχουν εκτός από τη συγγραφή προγραμμάτων και ανάγνωση τους, γιατί συχνά παρατηρείται μικρή σχέση μεταξύ της ικανότητας να γράφουν οι σπουδαστές ένα πρόγραμμα και αυτής του να διαβάζουν και να κατανοούν ένα πρόγραμμα. Επιπλέον, θα πρέπει να διδάσκονται βασικές στρατηγικές ελέγχου (testing) και αποσφαλμάτωσης (debugging) προγραμμάτων.
 - Η οπτικοποίηση της εκτέλεσης μικρών αντικειμενοστρεφών προγραμμάτων και η επίδειξη του πλήρους κύκλου ζωής των αντικειμένων, πολύ πιθανά θα βοηθούσε τους σπουδαστές να κατανοήσουν την κατάσταση του προγράμματος. Η οπτικοποίηση θα μπορούσε να αποβεί ιδιαίτερα ωφέλιμη για την επίδειξη της ροής του προγράμματος με τη χρήση “δεικτών” (pointers) και των περιεχομένων της μνήμης, καθώς και για περίπλοκα χαρακτηριστικά της κληρονομικότητας (inheritance).

Ειδικότερες συστάσεις που διατυπώνονται στην παραπάνω μελέτη για τη δημιουργία οπτικοποιημένου μαθησιακού υλικού, δεν εμπίπτουν στην περιοχή ενδιαφέροντος της διατριβής και γι’ αυτό το λόγο δεν αναφέρονται.

➤ Μια σειρά συστάσεων διδασκαλίας έχουν διατυπωθεί με τη μορφή οδηγιών (guidelines) (Kölling and Rosenberg, 2001b) από τους ερευνητές που ασχολήθηκαν με την ανάπτυξη του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ, το οποίο περιγράφηκε στην προηγούμενη ενότητα 2.5. Αυτές είναι σε κάποιο βαθμό εξαρτημένες από τη χρήση του και στοχεύουν στη δημιουργία μιας σειράς προγραμματιστικών δραστηριοτήτων ειδικά σχεδιασμένων για την αξιοποίηση των δυνατοτήτων του περιβάλλοντος αυτού και αντιστρόφως, δηλαδή κάποιες από αυτές τις συστάσεις είναι δύσκολο να υιοθετηθούν χωρίς τη χρήση του (Kölling and Rosenberg, 2001b). Οι οδηγίες αυτές παρατίθενται στη συνέχεια.

- *Οδηγία 1:* Υιοθέτησε την “objects first” (OF) προσέγγιση.

Να διδάσκονται δηλαδή τα αντικείμενα από τα αρχικά στάδια της εκπαίδευσης και όχι να ακολουθείται μια περιορισμένης έκτασης δομημένη (structured) προγραμματιστική προσέγγιση αρχικά και αργότερα να προστίθεται η έννοια των αντικειμένων. Η προσέγγιση περιγράφηκε στην υποενότητα 2.4.1.3.

- *Οδηγία 2:* Μην ξεκινάς με “κενή οθόνη” (Don't start with a blank screen).

Η συγγραφή κώδικα από το μηδέν, αποτελεί δύσκολο έργο για τους εκπαιδευόμενους. Αυτό προϋποθέτει σχεδιασμό (design), δηλαδή να αποφασίσεις ποιες κλάσεις θα υπάρχουν και με ποιες μεθόδους. Ακόμα και αν αυτά δοθούν στους εκπαιδευόμενους, δεν είναι σε θέση να αντιληφθούν γιατί ελήφθησαν αυτές οι σχεδιαστικές αποφάσεις. Αντίθετα, μπορούν να ξεκινήσουν κάνοντας μικρές αλλαγές σε υπάρχοντα κώδικα και να γράψουν εντελώς νέο κώδικα ως τμήμα υπάρχουσας κλάσης, δηλαδή να επεκτείνουν υπάρχουσες κλάσεις.

- *Οδηγία 3:* Διάβασε κώδικα (Read code).

Αν οι εκπαιδευόμενοι μόνο γράφουν κώδικα χωρίς ποτέ να διαβάσουν, χάνουν μια σπουδαία ευκαιρία να επωφεληθούν και να μάθουν, διαβάζοντας “καλογραμμένα” προγράμματα και αντιγράφοντας τεχνικές από αυτά.

Η σύσταση αυτή περιέχεται και στη βιβλιογραφική μελέτη “Problems in Learning and Teaching Programming” (Ala-Mutka) που αναφέρθηκε προηγουμένως.

- *Οδηγία 4:* Κάνε χρήση “μεγάλων” projects (Use "large" projects).

Για να αναδειχθούν τα οφέλη του ΑΠ, απαιτούνται “μεγάλα” (για τα δεδομένα των εκπαιδευομένων) προγράμματα, τα οποία αποτελούνται από ένα σύνολο συνεργαζόμενων κλάσεων με ένα λογικό αριθμό μεθόδων. Γι’ αυτό συνιστάται να βλέπουν τέτοια παραδείγματα.

- *Οδηγία 5:* Μην ξεκινάς με τη μέθοδο “main” (Don't start with "main").

Η ύπαρξη αυτής της μεθόδου έχει να κάνει με τη σύνδεση της εφαρμογής με το λειτουργικό σύστημα και δεν έχει σχέση με την Αντικειμενοστρέφεια. Έτσι δεν αποτελεί κατάλληλο παράδειγμα για την αρχή. Απαιτείται βέβαια να συμπεριληφθεί στη διδασκαλία, για να είναι σε θέση οι εκπαιδευόμενοι να γράφουν προγράμματα σε οποιοδήποτε προγραμματιστικό περιβάλλον.

- *Οδηγία 6:* Μη χρησιμοποιείς το "Hello World" παράδειγμα (Don't use "Hello World").

Η οδηγία αυτή αποτελεί απόρροια των προηγούμενων. Σε αυτό το τόσο κοινό πρόγραμμα (ή κάποιο αντίστοιχό του), ο εκπαιδευόμενος δε γράφει αντικειμενοστρεφή κώδικα. Γράφει μια κλάση από την οποία κανένα αντικείμενο δε δημιουργείται ποτέ και μια μέθοδο που δεν υλοποιεί λειτουργία αντικειμένου.

- *Οδηγία 7:* Δείξε με οπτικούς τρόπους τη δομή του προγράμματος (Show program structure).

Οι κλάσεις μιας εφαρμογής και οι μεταξύ τους σχέσεις είναι πολύ καίρια θέματα και είναι αρκετά δύσκολο να συζητηθούν αφηρημένα. Η οπτική αναπαράσταση τους είναι ζωτικής σημασίας για την κατανόηση σημαντικών εννοιών.

- *Οδηγία 8:* Δώσε προσοχή στη διεπαφή χρήστη δηλαδή στο χειρισμό εισόδου-εξόδου (Be careful with the user interface).

Από την αρχή της εκπαίδευσης (δηλαδή για τον κώδικα που οι εκπαιδευόμενοι δημιουργούν στα πρώιμα στάδια), χρειάζεται να δοθεί σημασία σε θέματα όπως πώς θα γίνονται ορατά τα αποτελέσματα του κώδικα των εκπαιδευομένων και πώς θα εισάγονται οι παράμετροι.

- Μια σειρά από παιδαγωγικά σχεδιαστικά πρότυπα (pedagogical patterns) έχουν διατυπωθεί στα Bergin (2006a) και Bergin (2006b) που αφορούν στη διδασκαλία γενικώς και βέβαια βρίσκουν εφαρμογή και στη διδασκαλία του ΑΠ. Επίσης σχεδιαστικά πρότυπα για εκπαίδευση στην τάξη (Classroom Education)

διατυπώθηκαν στο Anthony (1996) καθώς και παιδαγωγικά σχεδιαστικά πρότυπα για την αποδοτική διδασκαλία σεμιναρίων (A Pedagogical Pattern Language About Teaching Seminars Effectively) στο Fricke and Voelter (2000).

2.7. Σύνοψη

Στο κεφάλαιο αυτό έγινε μια εκτενής βιβλιογραφική επισκόπηση σχετικά με τη διδασκαλία του γνωστικού αντικείμενου του ΑΠ, η οποία περιελάμβανε την παρουσίαση ερευνών μικρής διάρκειας αλλά και μακροχρόνιων που έχουν διεξαχθεί για τη διδασκαλία του, διδακτικές προσεγγίσεις που έχουν αναπτυχθεί για τη διδασκαλία του, την περιγραφή αρκετών προγραμματιστικών περιβαλλόντων και εργαλείων που αφορούν στη διδασκαλία του και τέλος συστάσεις διδασκαλίας που έχουν διατυπωθεί σε όλη την παραπάνω αναφερθείσα βιβλιογραφία.

Συμπερασματικά, από τη μελέτη που έγινε προκύπτει ότι οι έρευνες που έχουν διεξαχθεί τόσο οι μακροχρόνιες όσο και οι μικρής διάρκειας, έχουν συνεισφέρει σημαντικά στη διερεύνηση των αντιλήψεων και των δυσκολιών των εκπαιδευομένων σχετικά με το αντικείμενο του ΑΠ. Ουσιαστικά, αντικείμενό τους αποτέλεσε αυτή η διερεύνηση από διάφορες σκοπιές, η οποία απέφερε σημαντική γνώση για το πώς οι εκπαιδευόμενοι αντιλαμβάνονται το αντικείμενο του ΑΠ, πού δυσκολεύονται και επίσης απέφερε σημαντικές συστάσεις για τη διδασκαλία του.

Όσα παρουσιάστηκαν σε αυτό το κεφάλαιο –όσα είχαν δημοσιευθεί και μελετηθεί στην εκάστοτε φάση εκπόνησης της διδακτορικής διατριβής– αποτέλεσαν τα θεμέλια για τις αποφάσεις που ελήφθησαν και αφορούν στη σχεδίαση-συγκρότηση της διδακτικής προσέγγισης που παρουσιάζεται στην παρούσα διατριβή και τον τρόπο διεξαγωγής των πειραματικών μελετών που περιλαμβάνουν μελέτη αξιολόγησής της σε βάθος.

Η προτεινόμενη προσέγγιση οριστικοποιήθηκε κατά φάσεις μετά από επαναλαμβανόμενες εφαρμογές και αξιολογήσεις. Η τελική μορφή στην οποία κατέληξε, αποτελεί απόρροια διαφορετικών κύκλων αξιολόγησης και βελτιστοποίησης. Οι μελέτες αξιολόγησης για τις εφαρμογές της διδακτικής προσέγγισης που πραγματοποιήθηκαν, παρουσιάζονται στο τέταρτο κεφάλαιο της διατριβής.

Στο επόμενο κεφάλαιο παρουσιάζεται η διδακτική προσέγγιση που σχεδιάστηκε, με το διδακτικό της περιεχόμενο, την εκπαιδευτική της φιλοσοφία, το μαθησιακό υλικό που τη συνοδεύει, τα μαθησιακά εργαλεία που χρησιμοποιεί και το ρόλο κάθε εμπλεκόμενου στη μαθησιακή διαδικασία.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

РАНЕЕ НЕ ПЕРПА

ΚΕΦΑΛΑΙΟ 3

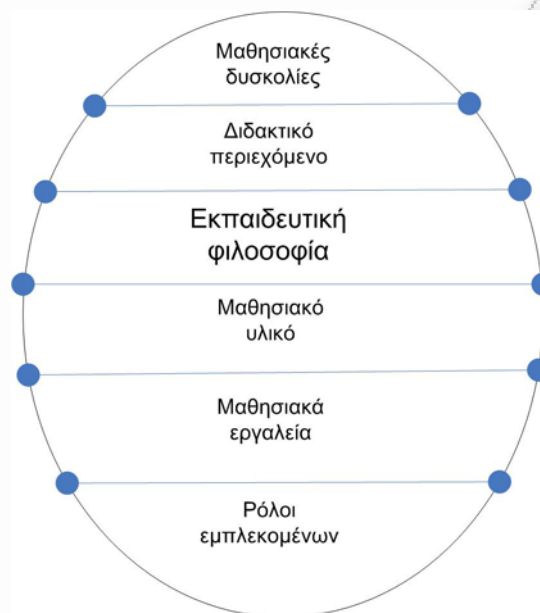
Η Προτεινόμενη Διδακτική Προσέγγιση για τον ΑΠ

3.1. Εισαγωγή

Η προσέγγιση που σχεδιάστηκε και παρουσιάζεται σε αυτό το κεφάλαιο, στοχεύει στην επίλυση προβλημάτων της διδασκαλίας για την εισαγωγή στον ΑΠ. Αντιμετωπίζονται σφαιρικά τα ζητήματα που αφορούν στη διδασκαλία του ΑΠ και επιχειρείται να εμπεδωθεί από τους εκπαιδευόμενους η φιλοσοφία και ο Αντικειμενοστρεφής τρόπος σκέψης στην επίλυση προβλημάτων με προγραμματιστικό τρόπο. Η προτεινόμενη προσέγγιση υιοθετεί τις αρχές της “objects first” φιλοσοφίας με “model first” στρατηγική υλοποίησης, χαρακτηριστικά της παιδαγωγικής στρατηγικής της γνωστικής μαθητείας (cognitive apprenticeship), προτείνει να πραγματοποιούνται μαθησιακές δραστηριότητες με μια συγκεκριμένη σειρά και αξιοποιεί διαδικτυακές τεχνολογίες και εργαλεία όπως το BlueJ, forum, το SUN One Studio κ.ά. Επίσης, για τις ανάγκες καλύτερης υποστήριξης των μαθησιακών δραστηριοτήτων σχεδιάστηκε και κατασκευάστηκε κατάλληλο μαθησιακό υλικό (σε μορφή ασκήσεων, σημειώσεων θεωρίας, κ.ά.) αφού ελήφθησαν υπόψη συστάσεις και οδηγίες με προσήλωση στο στόχο της “θεραπείας” μαθησιακών προβλημάτων.

Στην παρουσίαση της προτεινόμενης διδακτικής προσέγγισης περιλαμβάνονται: (i) *τα προβλήματα (μαθησιακές δυσκολίες) που έχουμε να επιλύσουμε* (ενότητα 3.2.), (ii) *το διδακτικό περιεχόμενο ή διδακτέα ύλη* (ενότητα 3.3.), (iii) *η εκπαιδευτική φιλοσοφία της με τις αρχές και τις στρατηγικές υλοποίησης της* (ενότητα 3.4.), βάσει των οποίων έχουν προταθεί συγκεκριμένες *δραστηριότητες που λαμβάνουν χώρα με συγκεκριμένη σειρά* (ενότητα 3.5.), (iv) *το μαθησιακό περιβάλλον* μέσα στο οποίο αυτή εντάσσεται (ενότητα 3.6.), (v) *το μαθησιακό υλικό που σχεδιάστηκε και συνοδεύει την προσέγγιση* (ενότητα 3.7.), (vi) *τα μαθησιακά εργαλεία που χρησιμοποιούνται* (ενότητα 3.8.) και (vii) *ο ρόλος όσων εμπλέκονται στη μαθησιακή διαδικασία* (ενότητα 3.9.).

Συνολικά με την παρουσίαση της προτεινόμενης διδακτικής προσέγγισης, θα έχουν καλυφθεί ερωτήματα όπως: ποιο πρόβλημα έχουμε να επιλύσουμε, τι θα διδάξουμε, πώς θα το διδάξουμε (με ποια φιλοσοφία, τι πόρους, εργαλεία, υποδομή, μέσα σε ποιο περιβάλλον, κάνοντας τι και πότε ο κάθε εμπλεκόμενος στη μαθησιακή διαδικασία). Αυτός ο τρόπος παρουσίασης αποτυπώνεται στο Σχήμα 3.1. (Goodyear, 1994).



Σχήμα 3.1. Παρουσίαση των συστατικών της προτεινόμενης διδακτικής προσέγγισης

Τα θεμέλια για τη σχεδίαση της προτεινόμενης προσέγγισης, αποτέλεσαν το διδακτικό περιεχόμενο (διδασκτέα ύλη), οι μαθησιακές δυσκολίες του γνωστικού αντικείμενου και η εκπαιδευτική φιλοσοφία που τη διέπει. Δομικές συνιστώσες της αποτελούν το μαθησιακό υλικό, η τεχνολογική της υποδομή και το ανθρώπινο δυναμικό που εμπλέκεται στην εφαρμογή της με τους ρόλους που ο καθένας επιτελεί.

3.2. Μαθησιακές δυσκολίες στον ΑΠ (ή προβλήματα προς λύση)

Στο προηγούμενο κεφάλαιο στις ενότητες 2.2. και 2.3. έγινε εκτενής αναφορά των μαθησιακών δυσκολιών που έχουν διαπιστωθεί κατά τη διδασκαλία του ΑΠ, τόσο από έρευνες μικρές και σύντομης διάρκειας όσο και από μακροχρόνιες και πιο εκτενείς. Μια κατηγοριοποίηση των δυσκολιών και παρανοήσεων έχει ήδη γίνει στην ενότητα 2.2.

Σημαντικά ζητήματα στη διδασκαλία του ΑΠ, προκύπτουν να είναι:

- Η ορθή αντίληψη των θεμελιωδών για το παράδειγμα αυτό Προγραμματισμού εννοιών, του αντικειμένου και της κλάσης καθώς και της διάκρισης μεταξύ τους.
- Η ορθή αντίληψη της έννοιας του *ιδιοχαρακτηριστικού των κλάσεων* και η *μη σύγχυση της με άσχετες έννοιες* όπως π.χ. οι ενέργειες που μπορεί να εκτελέσει (actions it performs) ένα αντικείμενο της κλάσης.
- Η *διάκριση σχέσεων* όπως όλου-μέρους, κλάσης-αντικειμένου και ιεραρχίας μεταξύ τους.
- Η αντιμετώπιση φαινομένων *ταύτισης* διαφορετικών αντικειμένων μεταξύ τους.
- Η κατανόηση της *αρχής της κελυφοποίησης – απόκρυψης πληροφορίας* (encapsulation-information hiding) και η εφαρμογή της στη σχεδίαση κλάσεων.
- Η αντίληψη της περίπλοκης και “κρυμμένης” διαδικασίας της *δημιουργίας αντικειμένων*, με όλα τα ζητήματα που σχετίζονται με αυτήν (κατασκευαστής, αρχικοποίηση, κατανομή μνήμης) αλλά και της καταστροφής τους με τα σχετικά με αυτή θέματα.
- Η αντίληψη των *τύπων αναφοράς*, της διαφοράς τους από τους βασικούς τύπους δεδομένων και της σημασίας τους στον προσδιορισμό της ταυτότητας του αντικειμένου.
- Η αντίληψη του τι αποτελεί την “ταυτότητα” του αντικειμένου.
- Η αντίληψη των *συντεθειμένων (composed) κλάσεων* και της δημιουργίας στιγμοτύπων τους.
- Η κατανόηση της *συνολικής ροής – εκτέλεσης προγραμμάτων (program flow)*.

Όλα τα παραπάνω, που αφορούν σε δυσκολίες κατανόησης μεμονωμένων ΟΟ εννοιών, σε διακρίσεις μεταξύ εννοιών, σε λανθασμένες συγχύσεις-ταυτίσεις, στην κατανόηση αρχών που διέπουν την Αντικειμενοστρεφή φιλοσοφία Προγραμματισμού, στην αντίληψη θεμάτων δυναμικού χαρακτήρα και θεμάτων που σχετίζονται με ζητήματα μνήμης, ελήφθησαν ιδιαίτερα υπόψη στη σχεδίαση της διδακτικής προσέγγισης που προτείνεται και του μαθησιακού υλικού που τη συνοδεύει. Αντιμετωπίστηκαν ως *προβλήματα προς λύση* και στις επανειλημμένες εφαρμογές και τις αξιολογήσεις της προσέγγισης που διενεργήθηκαν, εκτιμήθηκε η

επίτευξη του στόχου να επιλυθούν και ενδιάμεσως ελήφθησαν μέτρα προς αυτή την κατεύθυνση βάσει των αποτελεσμάτων που προέκυπταν από τις αξιολογήσεις.

Στοχευμένες διδακτικές παρεμβάσεις για την αντιμετώπιση τους, θα αναφερθούν στη συνέχεια στις ενότητες 3.7. και 3.8., στις οποίες παρουσιάζονται το μαθησιακό υλικό και τα μαθησιακά εργαλεία της προσέγγισης αντίστοιχα και στην ενότητα 3.4. στην οποία παρουσιάζονται οι αρχές της διδακτικής προσέγγισης.

3.3. Το διδακτικό περιεχόμενο (ή διδακτέα ύλη)

Ο καθορισμός της διδακτέας ύλης της απαραίτητης για την εισαγωγή στην Αντικειμενοστρεφή φιλοσοφία Προγραμματισμού, αποτέλεσε θέμα διερεύνησης κατά τη σχεδίαση της διδακτικής προσέγγισης. Το τι θα συμπεριλαμβάνει ένα εισαγωγικό course και το πρόβλημα της ένταξης αρκετών θεμάτων σε λίγο χρόνο έχει απασχολήσει προηγουμένως τους ερευνητές (Kölling and Rosenberg, 2001a). Πρωτεύον μέλημα και στόχος αποτέλεσε η πρόσκτηση από την πλευρά των εκπαιδευομένων, της Αντικειμενοστρεφούς φιλοσοφίας Προγραμματισμού, δηλαδή η διδασκαλία των θεμελιακών αρχών του Προγραμματιστικού αυτού παραδείγματος. Η προτεινόμενη προσέγγιση αξιοποιεί τη διάρθρωση των θεματικών ενοτήτων και των επιμέρους στόχων αυτών, που προτείνονται στον οδηγό σπουδών των IEEE/ACM (CC2001, 2001) για τον ΑΠ, ο οποίος και μελετήθηκε για να οριστικοποιηθούν τα θέματα που θα περιέχονται στη διδακτική προσέγγιση που αναπτύχθηκε.

Η τελική αναφορά (final report) για τη διδακτέα ύλη στους Υπολογιστές (Computing Curricula) αποτελεί έργο της κοινής αποστολής των ενώσεων IEEE-CS και ACM (The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery), να αναπτύξουν οδηγίες για το Πρόγραμμα Σπουδών προπτυχιακών προγραμμάτων στους Υπολογιστές. Αναφερόμαστε στο final report “Computing Curricula 2001 Computer Science, December 15, 2001” και σε Εισαγωγικά μαθήματα (Introductory courses). Η αυτόνομη μονάδα (unit) του Προγράμματος Σπουδών που ενδιαφέρει είναι η “PL6 Object-oriented programming”, που το περιεχόμενό της αναφέρεται στο CC2001 (2001), p. 115-116.

Αναπτύχθηκε μαθησιακό υλικό που καλύπτει το σύνολο των θεμάτων της ενότητας αυτής και στο οποίο επιπλέον επιχειρούνται συγκεκριμένες παρεμβάσεις με ενότητες

υλικού για την αντιμετώπιση καταγεγραμμένων μαθησιακών προβλημάτων που επανειλημμένα έχουν αναφερθεί. Οι διδακτικές ενότητες του μαθησιακού υλικού είναι οι εξής:

- **Ενότητα 1:** Φιλοσοφία Αντικειμενοστρεφούς Προγραμματισμού
- **Ενότητα 2:** Βασικές Έννοιες Αντικειμενοστρεφούς Προγραμματισμού: Αντικείμενο, Κλάση, Ιδιότητες, Μέθοδοι (object, class, attributes, methods)
- **Ενότητα 3:** Μέθοδος Κατασκευαστή (constructor), Δημιουργία Αντικειμένων – Κατανομή Μνήμης – Καταστροφή Αντικειμένων, Αναφορές σε Αντικείμενα (references)
- **Ενότητα 4:** Αφαιρετικότητα (Abstraction), Κελυφοποίηση (encapsulation) - Απόκρυψη Πληροφορίας (information hiding), Διαχωρισμός Συμπεριφοράς - Υλοποίησης (separation of behavior and implementation)
- **Ενότητα 5:** Μέθοδοι και Ιδιοχαρακτηριστικά κλάσης (static or class members)
- **Ενότητα 6:** Υπερφόρτωση Μεθόδων (methods' overloading)
- **Ενότητα 7:** Ιεραρχίες Κλάσεων (class hierarchies) - Κληρονομικότητα (inheritance) – Πολυμορφισμός (polymorphism) – Δυναμική Δέσμευση (dynamic binding)
- **Ενότητα 8:** Ολοκληρωμένο Παράδειγμα ή Μελέτη Περίπτωσης (case study)

Η λεπτομερής θεματική ανάπτυξη του διδακτικού περιεχομένου που αναπτύχθηκε και οι μαθησιακοί του στόχοι παρατίθενται στο Παράρτημα Α.

3.4. Η εκπαιδευτική φιλοσοφία της προτεινόμενης διδακτικής προσέγγισης

Η διδακτική προσέγγιση που παρουσιάζεται αποτελεί μια “objects first” προσέγγιση, όπως αυτή αναφέρεται στο IEEE/ACM Computing Curricula (CC2001, 2001) (“What distinguishes the IF (Imperatives First) approach from the objects-first model is the emphasis and ordering of the early topics.”), όπως περιγράφηκε στο δεύτερο κεφάλαιο (υποενότητα 2.4.1.3.) και όπως επίσης αναφέρεται στην ενότητα 2.6. των συστάσεων που έχουν διατυπωθεί με τη μορφή οδηγιών (guidelines) (Kölling and Rosenberg, 2001b) από τους ερευνητές που ασχολήθηκαν με την ανάπτυξη του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ. Οι λόγοι υιοθέτησης της συγκεκριμένης προσέγγισης προκύπτουν εύλογα από όσα αναφέρονται στην υποενότητα 2.4.1.3. Τα τελευταία χρόνια έχει πραγματοποιηθεί μια μετακίνηση του τρόπου διδασκαλίας του ΑΠ, υιοθετώντας αυτή την προσέγγιση. Όμως ο τρόπος υλοποίησης της OF μπορεί να ποικίλλει (Lewis, 2000).

Η στρατηγική υλοποίησης της συμφωνεί σε γενική κατεύθυνση με τη “model first” που επίσης περιγράφηκε στο δεύτερο κεφάλαιο (υποενότητα 2.4.2.3.). Συμπερασματικά στην ενότητα αυτή και στις πέντε επόμενες (3.5., 3.6., 3.7., 3.8., 3.9.) περιγράφονται

- οι βασικές αρχές της εκπαιδευτικής φιλοσοφίας και αναλύονται τα συστατικά μέρη, (μαθησιακά εργαλεία, μαθησιακό υλικό, μαθησιακό περιβάλλον, εμπλεκόμενοι στη μαθησιακή διαδικασία) μιας “objects first” προσέγγισης με μια συγκεκριμένη “model first” στρατηγική υλοποίησης της,

με τα επιπρόσθετα ιδιαίτερα χαρακτηριστικά που τις προσδίδουν συγκεκριμένες επιλογές που έχουν γίνει ως προς

- την αντιμετώπιση καταγεγραμμένων μαθησιακών προβλημάτων και δυσκολιών στη διδασκαλία του ΑΠ,
- την επιλογή των μαθησιακών εργαλείων,
- τη σχεδίαση του μαθησιακού υλικού,
- τη συνδουασμένη μαθησιακή στρατηγική (blended learning strategy) εφαρμογής της προσέγγισης με συγκεκριμένη ακολουθία μαθησιακών διαδικασιών (learning sequence),

- τους συγκεκριμένους ρόλους για τους εμπλεκόμενους στη μαθησιακή διαδικασία
- και την ενσωμάτωση χαρακτηριστικών από συγκεκριμένες παιδαγωγικές στρατηγικές (*pedagogical strategies*).

Κύριες διαφοροποιήσεις της προτεινόμενης “objects first” προσέγγισης με “model first” στρατηγική υλοποίησης από αυτήν που περιγράφηκε στο προηγούμενο κεφάλαιο (υποενότητα 2.4.2.3.), αποτελούν οι παρεμβάσεις που αυτή ενσωματώνει για την αντιμετώπιση καταγεγραμμένων μαθησιακών προβλημάτων, οι συγκεκριμένες επιλογές για τα μαθησιακά εργαλεία, η συνδυασμένη μαθησιακή στρατηγική εφαρμογής της προσέγγισης και η ενσωμάτωση σε αυτή χαρακτηριστικών της παιδαγωγικής στρατηγικής της γνωστικής μαθητείας (*cognitive apprenticeship*).

Η προτεινόμενη προσέγγιση, ουσιαστικά αποτελεί μια *τεχνολογικά υποστηριζόμενη ακολουθία μαθησιακών δραστηριοτήτων (ή διαδικασιών) (learning sequence)* που παρουσιάζονται στην επόμενη ενότητα (ενότητα 3.5.) –δηλαδή ένα μαθησιακό σενάριο, (*learning script*)– με συγκεκριμένα παιδαγωγικά χαρακτηριστικά, που μπορεί να χρησιμοποιηθεί ως ένας οδηγός υλοποίησης της προτεινόμενης “model first” στρατηγικής στη διδασκαλία του ΑΠ.

Βασικούς στόχους της προσέγγισης αποτελούν: (α) η ανάπτυξη Αντικειμενοστρεφούς σκέψης επίλυσης προβλημάτων (*development of OO problem solving skills*), (β) η εμπέδωση βασικών εννοιών-αρχών του παραδείγματος αυτού, (γ) η ανάπτυξη ικανότητας σχεδίασης, υλοποίησης, ελέγχου, αποσφαλμάτωσης απλών προγραμμάτων σε Java με τη χρήση CASE εργαλείων και προγραμματιστικών περιβαλλόντων, (δ) η αντιμετώπιση καταγεγραμμένων μαθησιακών προβλημάτων.

Η επιλογή της χρήσης της γλώσσας Java, δικαιολογείται από την επικράτηση της γλώσσας αυτής στη διδασκαλία του ΑΠ (Ragonis and Ben-Ari, 2005b; Benander et al., 2004; Biddle and Tempero, 1998; Bishop, 1997; Fleury, 2000; Kölling and Rosenberg, 2001b) και τη ζήτηση που έχει στην αγορά εργασίας. Συνεπώς προκύπτει η αναγκαιότητα οι εκπαιδευόμενοι να εξοικειωθούν με αυτή την τόσο διαδεδομένη και διαρκώς εξελισσόμενη γλώσσα προγραμματισμού κατά τη διάρκεια της εκπαίδευσής τους και να προετοιμαστούν –όσοι επαγγελματικά επιλέξουν το χώρο της ανάπτυξης λογισμικού– για τη μελλοντική επαγγελματική τους πορεία.

Επιπρόσθετα, όπως στο Kölling and Rosenberg (2001a) αναφέρεται, η Java έχει επιτύχει να γίνει ευρέως αποδεκτή ως καλύτερη γλώσσα για τη διδασκαλία του ΑΠ σε σχέση με την C++, γιατί δεν έχει την πολυπλοκότητά της. Η χρήση της C++, προσθέτει επί πλέον τις δικές της πολυπλοκότητες –που από τη φύση της παρουσιάζει– και αυτό δυσκολεύει περισσότερο την αντιμετώπιση των καθ’ εαυτών προβλημάτων της διδασκαλίας του ΑΠ.

Από τη διεξοδική βιβλιογραφική επισκόπηση που παρουσιάστηκε στο δεύτερο κεφάλαιο, συγκεντρώθηκαν αποτελεσματικές “αρχές” (“καλές πρακτικές”, best practices) στη διδασκαλία του ΑΠ, που έχουν καταγραφεί ως σύντομες οδηγίες (guidelines), ως “σχεδιαστικά πρότυπα” (design patterns) ή ως συστάσεις (Georgantaki and Retalis, 2007a). *Οι πρακτικές που υιοθετήθηκαν στην προσέγγιση και συνεπώς αποτελούν αρχές της είναι οι εξής:*

- **Ενασχόληση με τις σημαντικές έννοιες από τα αρχικά στάδια της εκπαίδευσης** (Bergin, 2006b, pattern “Early Bird”). Αυτή η αρχή, στην περίπτωση της διδασκαλίας του ΑΠ βρίσκει εφαρμογή στην “objects first” προσέγγιση όπως διατυπώνεται στο Kölling and Rosenberg (2001b) και για την υλοποίηση της, αξιοποιούνται οι δυνατότητες του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ (Kölling et al., 2003; Barnes and Kölling, 2005). Επίσης η αρχή αυτή περιλαμβάνει και την έμφαση στη διαφορετικότητα της Αντικειμενοστρεφούς φιλοσοφίας (Luker, 1994) ως προς την αντιμετώπιση προβλημάτων και την επισήμανση της ωφέλειας χρήσης της με απτά παραδείγματα.
- **Η προτεινόμενη προσέγγιση οδηγείται από τις Αντικειμενοστρεφείς έννοιες** (concepts driven) δίνοντας έμφαση στο ρόλο τους στην ανάπτυξη προγραμμάτων, χωρίς να εστιάζει στις λεπτομέρειες της γλώσσας προγραμματισμού. Παρουσιάζονται εκείνα τα στοιχεία της γλώσσας που είναι απαραίτητα για την υλοποίηση-εφαρμογή της εκάστοτε διδασκόμενης έννοιας-αρχής. Έτσι εφαρμόζονται οι απλές έννοιες σε προγράμματα ώστε να μην είναι αφηρημένες και αναδεικνύεται ο ρόλος τους στην επίλυση προβλημάτων με την ανάπτυξη προγραμμάτων οδηγούμενων από αυτές (Hadjerrouit, 1998; Hadjerrouit, 1999).

Γίνεται σταδιακή εισαγωγή των εννοιών από τις απλές στις πιο σύνθετες, που στηρίζεται στη θεμελίωση γνώσεων για τις προηγούμενες έννοιες και κατά συνέπεια βαθμιαία παρουσίαση των θεμάτων με αυξανόμενη πολυπλοκότητα. Οι διδακτικές ενότητες εστιάζουν σε ξεχωριστές έννοιες ώστε οι εκπαιδευόμενοι να αντιμετωπίσουν τις Αντικειμενοστρεφείς έννοιες-θέματα ένα-ένα κάθε φορά. Στόχο αποτελεί η οικοδόμηση στέρεας γνώσης για τις πρωταρχικές έννοιες –κλάση, αντικείμενο– και τη θέση που αυτές έχουν στην αντιμετώπιση και επίλυση ενός προβλήματος. Η επιλογή αυτή έγινε εκτιμώντας ότι είναι αναγκαίο οι εκπαιδευόμενοι να εισαχθούν βαθμιαία στην Προγραμματιστική αυτή φιλοσοφία (Bennedsen and Caspersen, 2004b), με μια προσεκτικά σχεδιασμένη ακολουθία παραδειγμάτων, δραστηριοτήτων, ασκήσεων, ερωτήσεων που επιτυγχάνουν την προοδευτική “έκθεση” τους σε συνθετότερα θέματα (scaffolding) (Schulte and Niere, 2002).

- **Προσεχτική σχεδίαση των μαθησιακών πόρων του υλικού λαμβάνοντας υπόψη καταγεγραμμένες μαθησιακές δυσκολίες** με στόχο την αντιμετώπιση τους (Holland et al., 1997; Teif and Hazzan, 2004; Anthony, 1996, Pattern “Pitfall Diagnosis and Prevention”).
- **Δίνεται ιδιαίτερη προσοχή στην ποιότητα του μαθησιακού υλικού που σχεδιάζεται** (παραδείγματα, ασκήσεις) για να αποκομίσουν οι εκπαιδευόμενοι σημαντικό όφελος από τη μελέτη και τη χρήση του (Kölling and Rosenberg, 2001b; Buck and Stucki, 2000; Fleury, 2000; Bergin, 2006a, pattern “Quality is Job One”).
- **Υλοποίηση της διαδικασίας ενεργούς δόμησης της γνώσης με τη συνεχή εμπλοκή του εκπαιδευόμενου σε κατάλληλα σχεδιασμένες δραστηριότητες ποικίλου είδους** (Hadjerrouit, 1999; Fleury, 2000; Fleury, 2001; Bergin, 2006a, pattern “Active Student”) –όπως πειραματισμός με έτοιμα παραδείγματα κώδικα, δραστηριότητες σε προγραμματιστικά περιβάλλοντα, επίλυση απλών ασκήσεων, απάντηση ερωτήσεων, εκπόνηση εκτενέστερων εργασιών– που “αναδεικνύουν” τις Αντικειμενοστρεφείς έννοιες-αρχές και αποτελούν έναυσμα για ενεργή μάθηση (active learning) (Schulte and Niere, 2002).

- **Εμπλοκή των εκπαιδευομένων στην επίλυση ασκήσεων και στη συγγραφή προγράμματος και από κενή οθόνη (from scratch).** Παράλληλα με τον πειραματισμό τους με ένα σύνολο έτοιμων προγραμμάτων και έτοιμων projects στο BlueJ, καλούνται να λύσουν μικρές-επιλεγμένες ασκήσεις σταδιακά αυξανόμενης δυσκολίας. Αυτές αφορούν στην τροποποίηση έτοιμων κλάσεων, στην επέκτασή τους και αρκετά σύντομα στη δημιουργία κλάσεων και προγραμμάτων από την αρχή.
- **Κατάλληλη σχεδίαση και χρήση αντιπροσωπευτικών παραδειγμάτων για την παρουσίαση νέων εννοιών (Fricke and Voelter, 2000, pattern “Relevant Examples”).**
- **Το μαθησιακό υλικό δε θα επιχειρεί την εξήγηση κάθε μεμονωμένης λεπτομέρειας της γλώσσας προγραμματισμού, αντίθετα θα εστιάζει στις έννοιες, την ορολογία και τη χρήση τους (Ben-Ari, 2004).**
- **Υιοθέτηση των οδηγιών “μην ξεκινάς με κενή οθόνη”, “μελέτησε κώδικα”,** που διατύπωσαν οι Kölling και Rosenberg (Kölling and Rosenberg, 2001b; Bergin, 2006a, pattern “Read before Write”). Καθοδήγηση των εκπαιδευομένων, ως το σημείο να συντάξουν κώδικα μόνοι τους, με τη χρήση πολλών έτοιμων παραδειγμάτων κώδικα που τους παρέχονται για να πειραματιστούν και μέσω αυτού του πειραματισμού να αντιληφθούν έννοιες, αλλά και να εξοικειωθούν με θέματα κώδικα, όπως σφάλματα μεταγλώττισης και εκτέλεσης (Fleury, 2000). Η τακτική αυτή σχετίζεται με το **μαθησιακό μοντέλο της εφαρμοσμένης “μαθητείας” (applied apprenticeship approach)** (Astrachan and Reed, 1995).
- **Υιοθέτηση της παρουσίασης μελετών περίπτωσης (case studies)** στις οποίες εφαρμόζονται στην πράξη οι αντικειμενοστρεφείς έννοιες που έχουν προηγουμένως παρουσιαστεί (Linn and Clancy, 1992; Fleury, 2001).
- **Δεν παραμελείται, ούτε καθυστερείται το θέμα της παρουσίασης ολοκληρωμένων προγραμμάτων** και η εμπέδωση από τους εκπαιδευόμενους της συνολικής ροής τους σε Αντικειμενοστρεφείς συνθήκες, **δηλαδή παρουσιάζονται “κλάσεις-προγράμματα” (μέθοδος main)** που κάνουν χρήση κλάσεων, δημιουργούν

αντικείμενά τους και καλούν μεθόδους επί αυτών, για την αντιμετώπιση των προβλημάτων στην κατανόηση της σφαιρικής ροής - εκτέλεσης προγραμμάτων (program flow) (Ragonis and Ben-Ari, 2005a). Από τις πρώτες απλές έννοιες, παρουσιάζονται πλήρη προγράμματα και έτσι οι εκπαιδευόμενοι έχουν μια πλήρη εικόνα (full picture), η οποία στην πορεία επαυξάνεται με αντίστοιχη αύξηση της πολυπλοκότητας των προγραμμάτων.

- **Ενσωμάτωση στην προσέγγιση της χρήσης και ενός επαγγελματικού προγραμματιστικού περιβάλλοντος**, ώστε να μην υπάρξει το κενό μεταξύ εκπαιδευτικών και μόνον εργαλείων και αδυναμίας χρήσης ενός “πραγματικού” επαγγελματικού περιβάλλοντος και να καταστεί μη προβληματική η μεταφορά σε ένα τέτοιο περιβάλλον (Xinogalos et al., 2006; Kölling et al., 2003).
- **Χρήση της τεχνολογίας και του Διαδικτύου** για την κατά το δυνατόν συχνότερη επαφή με τους εκπαιδευόμενους (Bergin, 2006a, pattern “24 by 7”).
- **Παροχή εξατομικευμένης βοήθειας στους εκπαιδευόμενους** ανάλογα με το επίπεδο του καθενός (Fleury, 2000; Bergin, 2006a, pattern “Differentiated Feedback”).
- **Υιοθέτηση της παιδαγωγικής αρχής της εκμάθησης του γνωστικού αντικείμενου του ΑΠ κοντά σε ειδικούς (experts) που αποτελεί χαρακτηριστικό της παιδαγωγικής στρατηγικής της γνωστικής μαθητείας (cognitive apprenticeship)** (Collins et al., 1989; Brown et al., 1989) και δημιουργία κατάλληλων συνθηκών για την εφαρμογή της.
- **Υιοθέτηση της συνδυασμένης μαθησιακής στρατηγικής (blended learning strategy) κατά την εφαρμογή της διδακτικής προσέγγισης**, δηλαδή παραδοσιακής “από-καθέδρας” διδασκαλίας και εμπλοκής των εκπαιδευομένων σε δραστηριότητες με τη χρήση των Διαδικτυακών τεχνολογιών (μέσω ηλεκτρονικού μαθησιακού περιβάλλοντος).
- **Καταγραφή των δυσκολιών και των λαθών των εκπαιδευομένων για να ληφθούν υπόψη στη βελτίωση της προσέγγισης** (Anthony, 1996, pattern “Pitfall Diagnosis and Prevention”).

- ο **Επανεξέταση και βελτίωση της προσέγγισης μετά από κάθε εφαρμογή της** (Anthony, 1996, pattern “Iterative Course Development”).

Ως προς το σχεδιασμό του τρόπου εφαρμογής της προσέγγισης και τις επαναλαμβανόμενες εφαρμογές της, υιοθετήθηκαν οι εξής αρχές:

Χρήση της τεχνολογίας και του Διαδικτύου για την επαφή με τους εκπαιδευόμενους και παροχή σε αυτούς εξατομικευμένης βοήθειας, υιοθέτηση της παιδαγωγικής στρατηγικής της γνωστικής μαθητείας (cognitive apprenticeship) (Collins et al., 1989; Brown et al., 1989) και της συνδυασμένης μαθησιακής στρατηγικής (blended learning strategy) κατά την εφαρμογή της διδακτικής προσέγγισης και δημιουργία κατάλληλων συνθηκών γι’ αυτό, καταγραφή των δυσκολιών και των λαθών των εκπαιδευομένων για να ληφθούν υπόψη στη βελτίωση της προσέγγισης και επανεξέταση και βελτίωση της προσέγγισης μετά από κάθε εφαρμογή της.

Το σύνολο των μαθησιακών δραστηριοτήτων έχουν σχεδιασθεί σύμφωνα με το κοινωνικό γνωστικό μοντέλο της διαδοχικής πρόσκτησης δεξιοτήτων των Zimmerman and Kitsantas (1999), το οποίο εισηγείται ότι η πρόσκτηση νέων δεξιοτήτων επιτυγχάνεται μέσω τεσσάρων διαδοχικών φάσεων: της παρατήρησης (observation), της μίμησης (emulation), του αυτοελέγχου (self-control) και της αυτορύθμισης (self-regulation).

Κατά την πρώτη φάση, δημιουργείται ένα νοητικό μοντέλο της δραστηριότητας από τις ενέργειες ενός ειδικού (του διδάσκοντα), ακούγοντας εξηγήσεις και αναλύσεις κάθε βήματος. Στη δεύτερη φάση, οι εκπαιδευόμενοι μιμούνται αυτό που βίωσαν και χρησιμοποιούν την ανατροφοδότηση από τους εκπαιδευτές και τους συνεκπαιδευόμενους τους για τη βελτίωση της απόδοσης τους, στην τρίτη συγκρίνουν τα επιτεύγματα τους με πρότυπα (standards) που απέκτησαν προηγουμένως και στην τελική φάση μαθαίνουν να προσαρμόζουν την απόδοσή τους σε αλλαγές εξωτερικών και εσωτερικών συνθηκών.

3.5. Ροή δραστηριοτήτων

Στην ενότητα αυτή παρουσιάζεται η ακολουθία των μαθησιακών δραστηριοτήτων (learning sequence) που συμβαίνουν σε κάθε φάση της μαθησιακής διαδικασίας κατά την εφαρμογή της προτεινόμενης διδακτικής προσέγγισης, δηλαδή τα βήματα εφαρμογής της. Αυτά αποτελούν ένα μαθησιακό σενάριο (learning script) που μπορεί να χρησιμοποιηθεί ως ένας οδηγός υλοποίησης της από τους εκπαιδευτές σε πραγματικές εκπαιδευτικές συνθήκες. Ένα μαθησιακό σενάριο είναι ουσιαστικά μια ακολουθία από φάσεις, το οποίο διαθέτει τα ακόλουθα πέντε χαρακτηριστικά (Dillenbourg, 2002):

- τη σειρά και την επιλογή του κατάλληλου χρόνου που συμβαίνει κάθε φάση,
- τις μαθησιακές δραστηριότητες που ο εκπαιδευτής και οι εκπαιδευόμενοι πρέπει να πραγματοποιήσουν σε κάθε φάση,
- τον τρόπο αλληλεπίδρασης μεταξύ εκπαιδευτή και εκπαιδευομένων και μεταξύ εκπαιδευομένων μεταξύ τους (με κατά πρόσωπο συναντήσεις (face-to-face), ασύγχρονα, μέσω κειμένου, μέσω φωνής, ...),
- τον τρόπο αλληλεπίδρασης των εκπαιδευομένων με τους μαθησιακούς πόρους και τα μαθησιακά εργαλεία.

Η παιδαγωγική αρχή που διέπει την εφαρμογή της διδακτικής προσέγγισης σε όλες τις φάσεις της είναι η εκμάθηση του γνωστικού αντικείμενου κοντά σε ειδικούς (experts), –χαρακτηριστικό της παιδαγωγικής στρατηγικής της γνωστικής μαθητείας (cognitive apprenticeship)– με πειραματισμό και με έμφαση στη μοντελοποίηση (modeling).

Ο ΑΠ αποτελεί κατάλληλο γνωστικό αντικείμενο για εκμάθηση κοντά σε ειδικούς, καθώς απαιτείται από τους εκπαιδευόμενους να αποκτήσουν ένα ολοκληρωμένο σύνολο γνωστικών και μεταγνωστικών δεξιοτήτων για την ανάπτυξη εφαρμογών λογισμικού η οποία αποτελεί διαδικασία επίλυσης προβλημάτων (problem solving process).

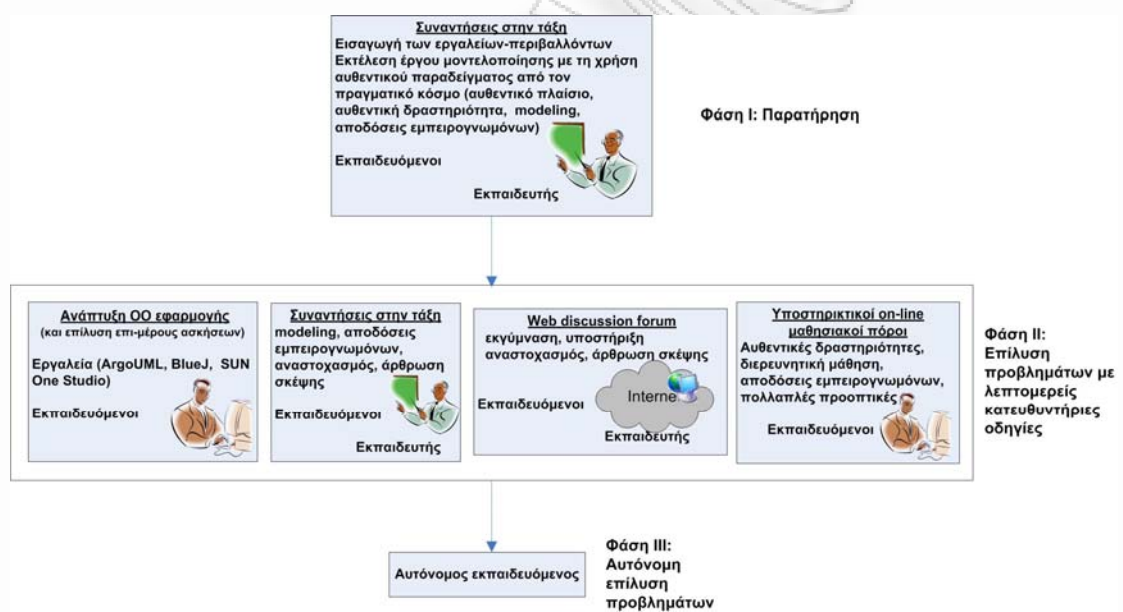
Το μαθησιακό σενάριο που θα παρουσιασθεί υποστηρίζει την εμπλοκή των εκπαιδευομένων σε μαθησιακές διαδικασίες, όπως μελέτη on-line μαθησιακών πόρων, αλληλεπίδραση με αντικειμενοστρεφή CASE εργαλεία, ασύγχρονη

επικοινωνία με τους εκπαιδευτές και τους συνεκπαιδευόμενους τους μαζί με παρακολούθηση παραδοσιακών “από-καθέδρας” διαλέξεων στην τάξη.

Η όλη διαδικασία απαρτίζεται από μαθησιακές δραστηριότητες που συμβαίνουν σε τρεις κύριες φάσεις:

- παρατήρηση της διαδικασίας ανάπτυξης μιας αντικειμενοστρεφούς εφαρμογής που υλοποιεί ένα αυθεντικό παράδειγμα από τον πραγματικό κόσμο (real world exemplar case ή real life situation), οικείο στους εκπαιδευόμενους και το οποίο παρακινεί το ενδιαφέρον τους (Collins et al., 1989; Brown et al., 1989),
- επίλυση προβλημάτων με συμβουλευτικές (κατευθυντήριες) οδηγίες και
- αυτόνομη επίλυση προβλημάτων.

Συνολικά οι φάσεις εφαρμογής της διδακτικής προσέγγισης, αποτυπώνονται στο Σχήμα 3.2.

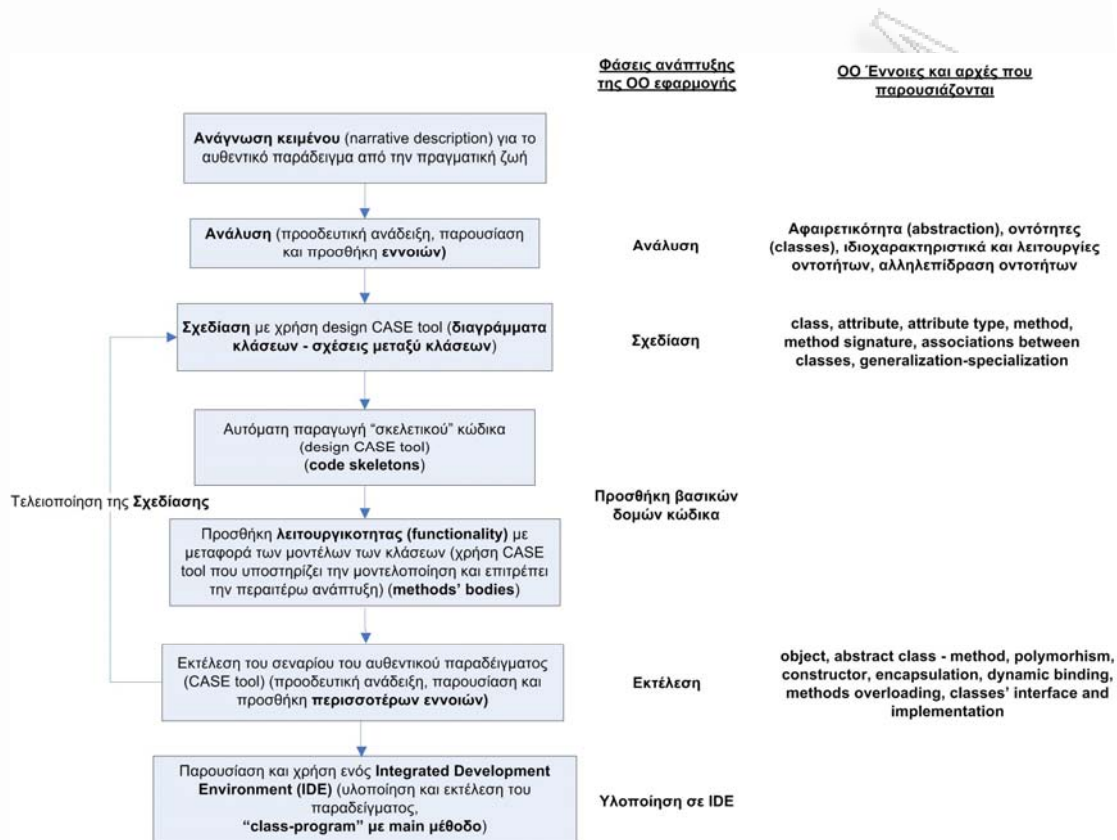


Σχήμα 3.2. Ροή δραστηριοτήτων του μαθησιακού σεναρίου

Φάση I: Παρατήρηση (observation)

Οι εκπαιδευόμενοι παρατηρούν πώς ο εκπαιδευτής –ο οποίος παίζει το ρόλο ενός ειδικού στην ανάπτυξη αντικειμενοστρεφούς λογισμικού– αναλύει, σχεδιάζει, υλοποιεί, αποσφαλματώνει και εκτελεί μια απλή αντικειμενοστρεφή εφαρμογή. Τα

βήματα της ανάπτυξης φαίνονται στο Σχήμα 3.3. Κατά τη διάρκεια κάθε βήματος, αναλύονται οι σχετικές αντικειμενοστρεφείς έννοιες και αρχές που αναδεικνύονται.



Σχήμα 3.3. Φάση παρατήρησης

Για την περίπτωση της υπό ανάπτυξη εφαρμογής γίνεται χρήση ενός αυθεντικού παραδείγματος-κατάστασης από τον πραγματικό κόσμο, οικείο στους εκπαιδευόμενους το οποίο παρακινεί το ενδιαφέρον τους (Collins et al., 1989; Brown et al., 1989) και αφορά στην περίπτωση μιας "λαϊκής αγοράς" στην οποία οι καταναλωτές επιλέγουν προϊόντα από τους πάγκους των πωλητών, οι πωλητές τα ζυγίζουν, υπολογίζουν την τιμή τους, οι καταναλωτές τα πληρώνουν και τα τοποθετούν στο καλάθι των αγορών τους. Το αυθεντικό παράδειγμα, σχεδιάζεται έτσι ώστε να συμπεριλάβει στην ανάπτυξή του όλες τις βασικές αντικειμενοστρεφείς έννοιες (κλάση, αντικείμενο, ιδιότητες, μέθοδος, αφαιρετικότητα, μέθοδος κατασκευαστή, κληρονομικότητα, πολυμορφισμός, αφηρημένες κλάσεις-μέθοδοι, διαπροσωπεία κλάσης, κελυφοποίηση, υπερφόρτωση μεθόδων, στατικές (static) κλάσεις-μέθοδοι).

Εκπαιδευτής και εκπαιδευόμενοι αρχικά, ανταλλάσσουν απόψεις στην προσπάθεια εντοπισμού των οντοτήτων (έμφυχων και άψυχων) (classes) του σεναρίου, του προσδιορισμού των χαρακτηριστικών ιδιοτήτων τους (attributes) και των λειτουργιών (operations ή methods) που εκτελούν και που είναι ουσιαστικές μέσα στο πλαίσιο του αυθεντικού παραδείγματος (abstraction) και των άλλων οντοτήτων με τις οποίες αλληλεπιδρούν. Μέσω της φάσης αυτής της ανάλυσης ο εκπαιδευτής μπορεί να εξηγήσει βασικές αντικειμενοστρεφείς έννοιες (classes, objects, attributes, κ.λ.π.).

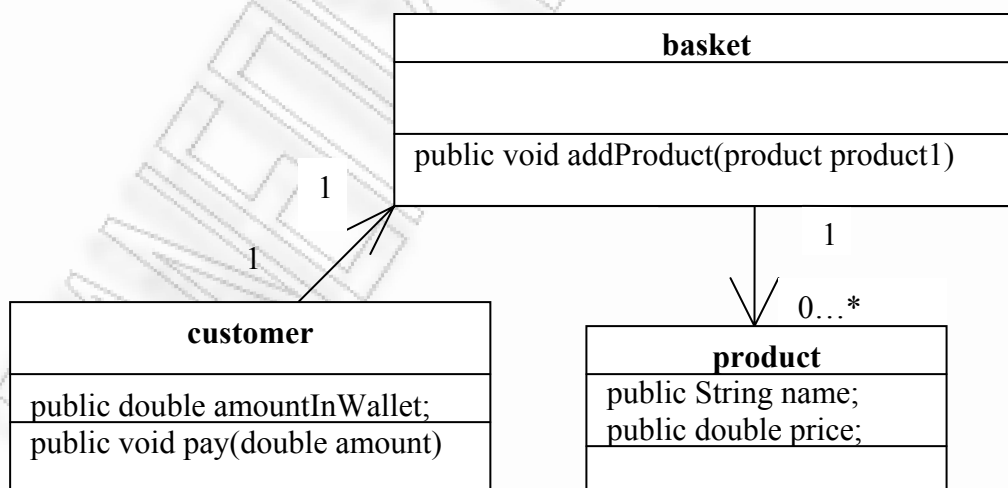
Στη συνέχεια ο εκπαιδευτής δημιουργεί το μοντέλο των κλάσεων που εντοπίστηκαν με τις ιδιότητες, τις λειτουργίες τους και τις μεταξύ τους σχέσεις (associations) (Σχήμα 3.4.) με τη χρήση ενός *design CASE tool* με δυνατότητα παραγωγής “σκελετικού” κώδικα *Java* (απαραίτητου για τη συνέχεια).

Όσα περιγράφηκαν στις παραπάνω παραγράφους αποτελούν συγκεκριμένη υλοποίηση των αφηρημένου-επιπέδου τεχνικών για τη συστηματική δημιουργία αντικειμενοστρεφών προγραμμάτων

Χώρος προβλήματος → Μοντέλο (Problem domain → model) και

Λειτουργικότητα → Κώδικας *Java* (Functionality → *Java* code) (καθορισμός attributes και μεθόδων των κλάσεων)

του Bennedsen and Caspersen (2004a) όπως αυτές αναφέρθηκαν στην υποενότητα 2.4.2.3.



Σχήμα 3.4. Μοντέλο κλάσεων με ιδιότητες, λειτουργίες και σχέσεις (ArgoUML)

Με τη χρήση του design CASE tool παράγεται “σκελετικός” κώδικας Java από τα μοντέλα των κλάσεων όπως φαίνεται στο παρακάτω Σχήμα 3.5.

<pre> public class customer { /* {src_lang=Java}*/ public double amountInWallet; /* {transient=false, volatile=false}*/ public basket mybasket; public void pay(double amount) { } } </pre>	<pre> public class seller { /* {src_lang=Java}*/ public double cashAmount; /* {transient=false, volatile=false}*/ public double weigh() { return 0.0; } public double calculateValue(product product1, double quantity) { return 0.0; } public void receivePayment(double amount) { } } </pre>
--	--

Σχήμα 3.5. “Σκελετικός” κώδικας Java (ArgoUML)

(Υλοποίηση της αφηρημένου-επιπέδου τεχνικής για τη συστηματική δημιουργία αντικειμενοστρεφών προγραμμάτων, Μοντέλο → Σκελετός κώδικα Java (Model → Java code) (Bennedsen and Caspersen, 2004a)).

Καθώς οι εκπαιδευόμενοι παρατηρούν τη διαδικασία, ο εκπαιδευτής εξηγεί έννοιες που αναδεικνύονται. Ο “σκελετικός” κώδικας στη συνέχεια εξάγεται και μεταφέρεται στο περιβάλλον ενός CASE εργαλείου το οποίο υποστηρίζει τη μοντελοποίηση –και συνεπώς παραμένει η εικόνα των μοντέλων για τους εκπαιδευόμενους–, επιτρέπει την περαιτέρω ανάπτυξη του κώδικα της εφαρμογής που έχει παραχθεί και επίσης την αλληλεπίδραση με τις κλάσεις για τη δημιουργία αντικειμένων και με τα αντικείμενα για την κλήση μεθόδων (απαραίτητο χαρακτηριστικό για την εύκολη εκτέλεση του σεναρίου του αυθεντικού παραδείγματος). Στο νέο περιβάλλον προστίθεται λειτουργικότητα στα σχεδιασμένα μοντέλα (έχει ληφθεί ιδιαίτερη προσοχή οι

λειτουργίες να υλοποιούνται κατά κανόνα εύκολα με απλές εντολές μαθηματικών πράξεων της γλώσσας προγραμματισμού, (Σχήμα 3.6.)).

<pre>public void pay(double amount) { amountInWallet -= amount; }</pre>	<pre>public double calculateValue(product product1, double quantity){ return (product1.salePrice() * quantity); }</pre>
---	---

Σχήμα 3.6. Λειτουργίες μεθόδων (BlueJ)

(Υλοποίηση της αφηρημένου-επιπέδου τεχνικής για τη συστηματική δημιουργία αντικειμενοστρεφών προγραμμάτων, Υλοποίηση των μεθόδων (Implementation of methods) (Bennedsen and Caspersen, 2004a)).

Με τη χρήση του περιβάλλοντος του CASE εργαλείου και τη δυνατότητα αλληλεπίδρασης που παρέχει, ο εκπαιδευτής εκτελεί την εξέλιξη του σεναρίου που αρχικά παρουσιάστηκε στους εκπαιδευόμενους, δημιουργώντας αντικείμενα και καλώντας επί αυτών μεθόδους. Η εκτέλεση αυτή αναδεικνύει την ανάγκη της παρουσίασης νέων εννοιών, όπως μέθοδος κατασκευαστή (constructor), κελυφοποίηση (encapsulation), διαπροσωπεία κλάσης (interface), αφηρημένη κλάση-μέθοδος (abstract class-method), υπερφόρτωση μεθόδων (methods' overloading), στατικές κλάσεις-μέθοδοι (static class-method). Όλες αυτές οι έννοιες εισάγονται σταδιακά από τον εκπαιδευτή για τη βελτίωση της αντικειμενοστρεφούς εφαρμογής που υλοποιεί το αυθεντικό παράδειγμα και γίνονται αντίστοιχες βελτιώσεις στη σχεδίαση της εφαρμογής στο design CASE tool που χρησιμοποιήθηκε αρχικά.

Στο τέλος, οι εκπαιδευόμενοι παρατηρούν την υλοποίηση του ίδιου αυθεντικού παραδείγματος σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών (IDE), στο οποίο ο εκπαιδευτής μεταφέρει όλον τον κώδικα των κλάσεων που έχει δημιουργηθεί στο CASE εργαλείο και δημιουργώντας ένα “πρόγραμμα-κλάση” (με μέθοδο main) το οποίο εκτελεί όλες τις λειτουργίες της εφαρμογής (δημιουργία αντικειμένων, κλήση μεθόδων, κατάλληλη εμφάνιση των αποτελεσμάτων της εκτέλεσης των μεθόδων).

Καθ' όλη τη διάρκεια της φάσης αυτής, ο εκπαιδευτής εξηγεί τη διαδικασία σκέψων που οδηγούν στις συγκεκριμένες επιλογές σχεδίασης και υλοποίησης της εφαρμογής κάνοντας τη σιωπηρή γνώση του φανερή και τη σκέψη του ορατή από τους εκπαιδευόμενους (Collins et al., 1991). Οι εκπαιδευόμενοι σχηματίζουν μέσω της

παρατήρησης, αλληλεπιδρώντας με έναν ειδικό και ακούγοντας τη διαδικασία της σκέψης του, ένα νοητικό μοντέλο των δραστηριοτήτων ανάπτυξης μιας αντικειμενοστρεφούς εφαρμογής από τη μοντελοποίηση έως τη δημιουργία κώδικα. Αυτό έχει ιδιαίτερη σημασία, αφού επιτελείται η πολύ σημαντική και συγχρόνως πολύ δύσκολη για μαθησιακούς λόγους διαδικασία της άρθρωσης της σιωπηρής γνώσης του εκπαιδευτή (Hislop, 2002). Η γνώση αυτή λειτουργεί ως ένα γνωστικό υπόβαθρο που ενισχύει την επίτευξη ενός έργου και περιέχει τυπικές και άτυπες πλευρές της γνώσης και των δεξιοτήτων που συχνά καλούνται “know-how”, καθώς και υποκειμενικές πεποιθήσεις, αντιλήψεις, ιδανικά, συνειδητοποιήσεις αξιών και διαισθήσεις (Reber, 1993).

Συνήθως η φάση αυτή πραγματοποιείται στην τάξη, όπου ο εκπαιδευτής αναπτύσσει τη διαδικασία της επίλυσης ενός προβλήματος και όχι απλά τη λύση του, χρησιμοποιώντας ένα κατάλληλα επιλεγμένο παράδειγμα, ένα set κατάλληλων CASE εργαλείων (Groven et al., 2003; Berge et al., 2003a) και κάνοντας αναφορές στις βασικές αντικειμενοστρεφείς έννοιες (Bennedsen and Caspersen, 2004b).

Αυτή η παρουσίαση του εκπαιδευτή μπορεί να βιντεοσκοπηθεί και να προστεθεί στους on-line μαθησιακούς πόρους που διατίθενται στους εκπαιδευόμενους, για τη μελλοντική αναφορά τους σε αυτήν.

Τα εργαλεία που επελέγησαν να χρησιμοποιηθούν είναι κατά σειρά το *UML εργαλείο σχεδίασης ArgoUML* (<http://argouml.tigris.org/>) αρχικά, στη συνέχεια το *εκπαιδευτικό προγραμματιστικό περιβάλλον BlueJ* (<http://www.bluej.org/>) και τέλος το *ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών SUN One Studio*. Εκτενέστερη αναφορά στην επιλογή τους γίνεται σε επόμενη ενότητα (ενότητα 3.8.).

Φάση II: Επίλυση προβλημάτων με λεπτομερείς συμβουλευτικές - κατευθυντήριες οδηγίες

Κατά τη διάρκεια της δεύτερης φάσης, οι εκπαιδευόμενοι αναπτύσσουν και παραδίδουν μια μικρή αντικειμενοστρεφή εφαρμογή ακολουθώντας ακριβώς τα ίδια βήματα του ειδικού, που παρατήρησαν και τους εξηγήθηκαν στην πρώτη φάση. Καθώς η παρατήρηση (observation) δεν είναι φυσικά αρκετή, προσπαθούν να ενσωματώσουν τη σιωπηρή γνώση που έχουν ακούσει. Καθ' όλη τη διάρκεια της

φάσης αυτής προσφέρεται στους εκπαιδευόμενους διαρκής υποστήριξη με ποικίλα μέσα.

Η υποστήριξη συνίσταται κυρίως σε on-line μαθησιακούς πόρους ενσωματωμένους σε ένα on-line Σύστημα Διαχείρισης της Μάθησης (Learning Management System) (LMS), όπως το σύστημα Moodle (<http://www.moodle.org>). Οι προσφερόμενοι πόροι είναι:

- η βιντεοσκοπημένη κατά βήμα παρουσίαση και εξήγηση της ανάπτυξης της αντικειμενοστρεφούς εφαρμογής του αυθεντικού παραδείγματος από τον εκπαιδευτή,
- σύντομα και περιεκτικά τμήματα θεωρίας συνοδευόμενα από αντιπροσωπευτικά παραδείγματα της καθημερινής ζωής,
- επί μέρους μικρές εργασίες (σύντομες ερωτήσεις-ασκήσεις) για περαιτέρω εξάσκηση πάνω στις αντικειμενοστρεφείς έννοιες (υποχρεωτικές ή μη ανάλογα με τις εκάστοτε συνθήκες εφαρμογής του σεναρίου) καθώς και ασκήσεις με τις λύσεις τους,
- προσεκτικά γραμμένα παραδείγματα κώδικα για την παρουσίαση εννοιών,
- μια καλά τεκμηριωμένη μελέτη περίπτωσης (case study) μιας αντικειμενοστρεφούς εφαρμογής, που περιλαμβάνει τις προδιαγραφές απαιτήσεων (requirements specification), την αντικειμενοστρεφή ανάλυση, την περιγραφή του σχεδιασμού (design decisions), τα διαγράμματα κλάσεων, τον πηγαίο κώδικα, περιπτώσεις δοκιμής (test cases) και την τεκμηρίωση (documentation) της,
- οδηγός μελέτης που περιέχεται σε κάθε διδακτική ενότητα και άλλο πληροφοριακό υλικό, όπως εγχειρίδια εγκατάστασης και χρήσης των μαθησιακών εργαλείων, σύνδεσμοι (links) σε πόρους σχετικούς με τη γλώσσα Java κ.λ.π.

Οι μαθησιακοί πόροι είναι οργανωμένοι σε διδακτικές ενότητες που αφορούν στις βασικές αντικειμενοστρεφείς έννοιες που παρουσιάστηκαν στους εκπαιδευόμενους κατά τη φάση της παρατήρησης (π.χ. διαγραμματική αναπαράσταση, κατασκευαστής, κελυφοποίηση, κληρονομικότητα κ.λ.π.). Οι ενότητες αυτές περιγράφηκαν αναλυτικά στην ενότητα του διδακτικού περιεχομένου (ενότητα 3.3.).

Επιπλέον, ως βοήθεια για την οικοδόμηση της γνώσης παρέχονται: (α) ένας διαδικτυακός τόπος ασύγχρονης συζήτησης (web asynchronous discussion forum) στον οποίο οι εκπαιδευόμενοι καταχωρούν ερωτήσεις που ο εκπαιδευτής/ειδικός ή οι

άλλοι εκπαιδευόμενοι μπορούν να απαντήσουν, σχηματίζοντας έτσι μια κοινότητα πρακτικής (community of practice) και (β) παραδοσιακές συναντήσεις στην τάξη αυτοπροσώπως με τον εκπαιδευτή, για τη συζήτηση θεμάτων και μαθησιακών δυσκολιών που προκύπτουν.

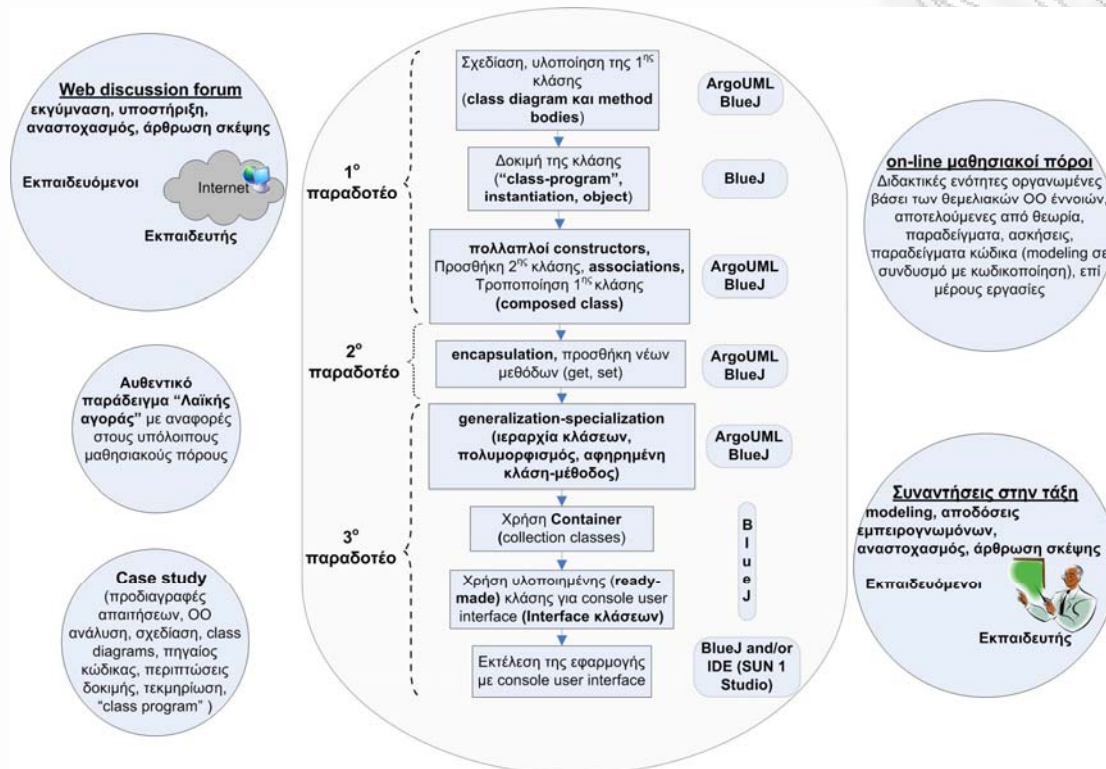
Κατά τη διάρκεια όλης αυτής της φάσης ο εκπαιδευτής υποστηρίζει τη μαθησιακή διαδικασία μέσω μεθόδων σταδιακής υποστήριξης (scaffolding) και εκγύμνασης (coaching), παρέχοντας υποδείξεις (hints), κάνοντας υπενθυμίσεις (reminders), προσφέροντας ανατροφοδότηση (feedback), συστάσεις (recommendations) και ενθαρρύνοντας εκείνες τις συζητήσεις οι οποίες αποκαλύπτουν τη διαδικασία μέσω της οποίας οι εκπαιδευόμενοι οδηγούνται στη λύση (Bennedsen and Caspersen, 2004b). Καθώς οι εκπαιδευόμενοι καθίστανται πιο αποτελεσματικοί στην προσπάθειά τους, ο υποστηρικτικός ρόλος του εκπαιδευτή μειώνεται (fading) και οι εκπαιδευόμενοι λειτουργούν πιο ενεργά ως μέλη μιας κοινότητας επαγγελματιών (community of practitioners) (Collins et al., 1989; Lave and Wenger, 1991). Μέσα σε αυτή την κοινότητα οι εκπαιδευόμενοι θέτουν ερωτήματα, επικοινωνούν και μοιράζονται τις ιδέες και τις σκέψεις τους για το πρόβλημα που έχουν να επιλύσουν και επίσης αρθρώνουν (articulation) τη σκέψη τους και στοχάζονται (reflection) πάνω στις γνώμες που οι άλλοι εκφράζουν και στις ερωτήσεις που θέτουν. Κατά τη διάρκεια αυτής της διαδικασίας, τους παρέχονται λεπτομερείς οδηγίες (scaffolds) για το σχεδιασμό και τον προγραμματισμό (Collins et al., 1989). Αυτές, υποστηρίζουν την ενεργή (active) και διερευνητική (exploratory) μάθηση, παρέχοντας στους εκπαιδευόμενους την ευκαιρία να εξερευνήσουν έννοιες από πολλαπλές προοπτικές και επίσης να δουν και να εμπεδώσουν τις λύσεις που ειδικοί δίνουν σε προβλήματα (Hadjerrouit, 1999).

Στο προτεινόμενο σενάριο οι εκπαιδευόμενοι καλούνται να υλοποιήσουν μια αντικειμενοστρεφή εφαρμογή για τη διαχείριση ενός “Βιβλιοπωλείου” και να την παραδώσουν σε τμήματα, σε έναν αριθμό παραδοτέων σε προκαθορισμένες ημερομηνίες που καθορίζονται από τις εκάστοτε συνθήκες εφαρμογής του σεναρίου. Με αυτό τον τρόπο η εφαρμογή τους αναπτύσσεται προοδευτικά, επιτρέποντας να αντιληφθούν τη συσχέτιση μεταξύ των αντικειμενοστρεφών εννοιών και την εφαρμογή τους στην ανάπτυξη της εφαρμογής.

Σημαντικό είναι ότι σε αυτή τη φάση, ζητείται από τους εκπαιδευόμενους να παραδώσουν τις εργασίες τους με αρκετή τεκμηρίωση που θα εξηγεί τη λύση που έδωσαν και τον τρόπο με τον οποίο οδηγήθηκαν σε αυτή. Τους ζητείται να

αρθρώσουν (articulate) και να αντιληφθούν και οι ίδιοι τη γνώση που αποκτούν. Με αυτό τον τρόπο ο εκπαιδευτής είναι σε θέση να αντιληφθεί πιθανές παρανοήσεις και να παρέχει βοήθεια για την αντιμετώπισή τους.

Στο Σχήμα 3.7. απεικονίζονται οι φάσεις της αναπτυσσόμενης εφαρμογής και η παρεχόμενη υποστήριξη.



Σχήμα 3.7. Φάση II - Επίλυση προβλημάτων με λεπτομερείς συμβουλευτικές - κατευθυντήριες οδηγίες

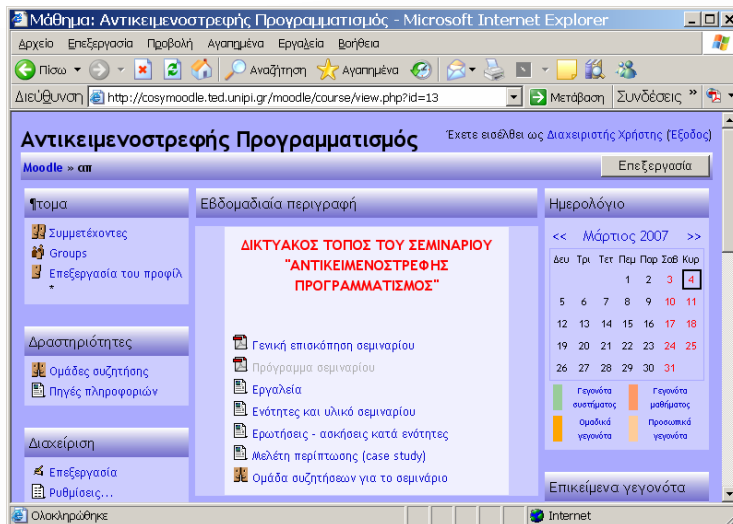
Φάση III: Αυτόνομη επίλυση προβλημάτων

Η σταδιακή απόσυρση της καθοδήγησης καθώς οι εκπαιδευόμενοι καθίστανται πιο αποτελεσματικοί, οδηγεί στην ανεξάρτητη δράση τους. Σε αυτή τη φάση, οι εκπαιδευόμενοι αναπτύσσουν μια εφαρμογή και την αποστέλλουν όχι σε τμήματα αλλά στο σύνολό της σε καθορισμένο χρόνο, χωρίς υποχρεωτικά να ακολουθήσουν τα βήματα που διδάχθηκαν κατά τις προηγούμενες φάσεις. Δεν είναι απαραίτητη η τεκμηρίωση (π.χ. διαγράμματα κλάσεων) της διαδικασίας ανάπτυξης, αλλά η έμφαση δίνεται σε αυτή τη φάση, στην ποιότητα του πηγαίου κώδικα που οι εκπαιδευόμενοι αναπτύσσουν.

3.6. Μαθησιακό περιβάλλον με αξιοποίηση των Διαδικτυακών τεχνολογιών

Στην υλοποίηση της προτεινόμενης διδακτικής προσέγγισης αξιοποιείται το Διαδίκτυο (Internet) παρέχοντας ένα ολοκληρωμένο on-line μαθησιακό περιβάλλον, για την υποστήριξη της παραδοσιακής “από-καθέδρας” διδασκαλίας. Το περιβάλλον αυτό, χρησιμοποιείται ως μέσο εγγραφής των εκπαιδευομένων, ως μέσο διανομής/διάθεσης του μαθησιακού υλικού, ως μέσο πληροφόρησης και ανάρτησης ανακοινώσεων κατά τη διάρκεια της εφαρμογής της προσέγγισης στην πράξη για τη διαδικασία του μαθήματος (επικείμενα γεγονότα, παραδόσεις εργασιών (προθεσμίες) και ό,τι άλλο χρειαστεί) και ως μέσο επικοινωνίας μέσω ασύγχρονων συζητήσεων (asynchronous discussion fora) μεταξύ εκπαιδευτή-εκπαιδευομένων και εκπαιδευομένων μεταξύ τους. Δημιουργείται έτσι ένα ολοκληρωμένο μαθησιακό περιβάλλον (learning environment) που καλύπτει θέματα διανομής πληροφοριακού και μαθησιακού υλικού, επικοινωνίας, διαχείρισης τάξης κ.ά.

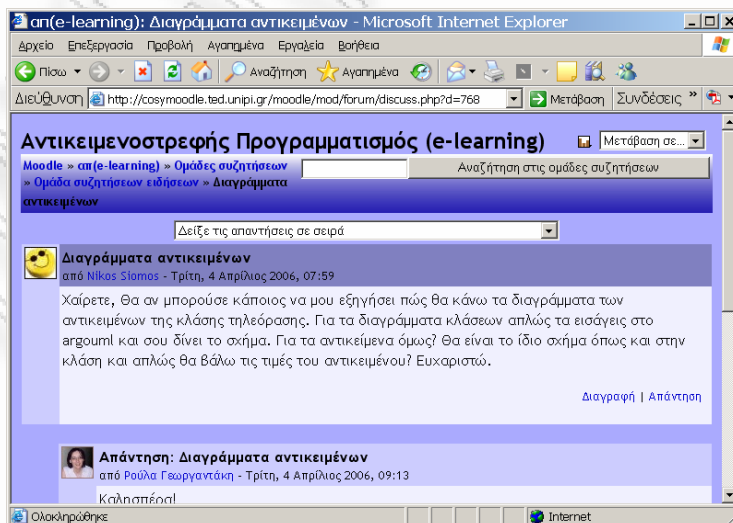
Συγκεκριμένα κατά τις εφαρμογές της διδακτικής προσέγγισης, δημιουργήθηκαν δικτυακοί τόποι στο Moodle LMS (<http://www.moodle.org>) του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά (Σχήματα 3.8., 3.9., 3.10., 3.11.) (<http://cosymoodle.ted.unipi.gr/>) και στο περιβάλλον της ηλεκτρονικής πύλης μαθημάτων e-class του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά (Σχήμα 3.12.) (<http://orfeas.ted.unipi.gr/elearn/index.php>), στους οποίους, εκτός του μαθησιακού υλικού περιελήφθησαν και οι διαφάνειες των διαλέξεων, όλες οι απαραίτητες πληροφορίες για την εγκατάσταση και χρήση των εργαλείων, πηγές χρήσιμου υλικού σχετικού με τη γλώσσα Java και ένας οδηγός μελέτης. Τα περιβάλλοντα αυτά, εκτός από μέσο διανομής μαθησιακού υλικού, χρησιμοποιήθηκαν για να πραγματοποιούνται ασύγχρονες συζητήσεις σχετικά με θέματα του γνωστικού αντικείμενου κατά τη διάρκεια μελέτης και εκπόνησης των εργασιών από τους εκπαιδευόμενους και για να στέλνονται οι εργασίες που οι εκπαιδευόμενοι παρέδιδαν.



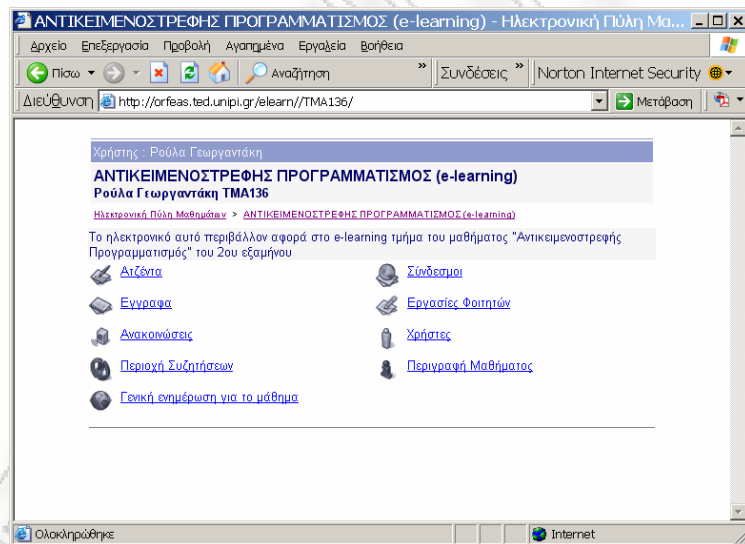
Σχήμα 3.8. Δικτυακός τόπος εφαρμογής της προτεινόμενης προσέγγισης στο Moodle LMS



Σχήμα 3.9. Ασύγχρονο forum συζητήσεων

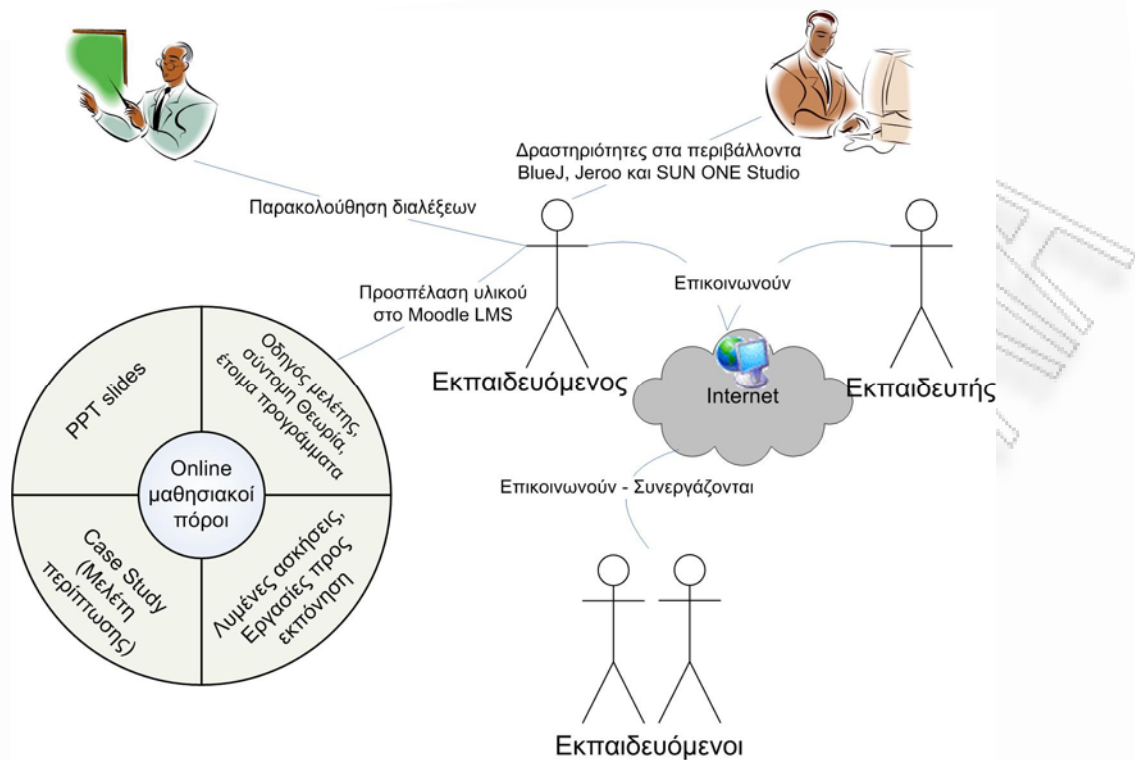


Σχήμα 3.10. Υποβολή ερώτησης από εκπαιδευόμενο, στο forum συζητήσεων



Σχήμα 3.12. Δικτυακός τόπος στο περιβάλλον της ηλεκτρονικής πύλης μαθημάτων e-class του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά

Στο Σχήμα 3.13. απεικονίζεται συνολικά το Διαδικτυακά υποστηριζόμενο μαθησιακό περιβάλλον.



Σχήμα 3.13. Σφαιρική παρουσίαση του Διαδικτυακά υποστηριζόμενου μαθησιακού περιβάλλοντος

3.7. Μαθησιακό υλικό

Το μαθησιακό υλικό είναι οργανωμένο ως σημειώσεις γύρω από τις θεμελιακές Αντικειμενοστρεφείς έννοιες και αρχές και όχι όπως ένα βιβλίο της γλώσσας Προγραμματισμού Java, επιβεβαιώνοντας έτσι την “εννοιο-οδηγούμενη” (concepts driven) φιλοσοφία της προτεινόμενης προσέγγισης.

Συνολικά η διαμόρφωσή του, είναι τέτοια ώστε να παρουσιάζει σαφώς τους μαθησιακούς στόχους, να δίνει εναύσματα για εφαρμογή και εμβάθυνση σε έννοιες της κάθε διδακτικής ενότητας, να προσφέρει στοιχεία ώστε οι εκπαιδευόμενοι να βρίσκονται σε σχέση αλληλεπίδρασης με το αντικείμενο που μελετούν και παραδείγματα που θα βοηθούν τους εκπαιδευόμενους να πραγματοποιήσουν μαθησιακές δραστηριότητες.

Αποτελείται από το υλικό που αφορά στο αυθεντικό παράδειγμα από τον πραγματικό κόσμο (“λαϊκή αγορά”) σε μορφή σεναρίου (περιγραφικής διήγησης) ή βίντεο και από τους πόρους εκείνους που αποτελούν υποστηρικτικό υλικό στη βήμα προς βήμα

ανάπτυξη μιας αντικειμενοστρεφούς εφαρμογής και απαρτίζονται από τις διδακτικές ενότητες που αναφέρθηκαν στην ενότητα 3.3. του Διδακτικού περιεχομένου.

Στην αρχή κάθε διδακτικής ενότητας παρουσιάζονται τα θέματα που καλύπτει, οι εκπαιδευτικές δραστηριότητες που οι εκπαιδευόμενοι καλούνται να πραγματοποιήσουν, διατυπώνονται οι διδακτικοί της στόχοι, τι αναμένεται από τους εκπαιδευόμενους να επιτευχθεί με όρους κατανόησης εννοιών-αρχών και εφαρμογής τους σε προγράμματα με τη χρήση των εκπαιδευτικών και άλλων εργαλείων που χρησιμοποιούνται. Σε κάθε ενότητα επίσης παρέχεται οδηγός μελέτης της που περιλαμβάνει συμβουλές για το πώς πρέπει να μελετηθεί το υλικό.

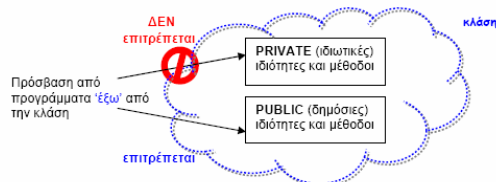
Το μαθησιακό υλικό που συνοδεύει την προτεινόμενη διδακτική προσέγγιση, έχει ενοποιηθεί σε “πακέτο” σύμφωνα με την προδιαγραφή IMS Content Packaging specification (IMS, 2003). Έτσι είναι δυνατή η επαναχρησιμοποίηση του (re-usability) για την υποστήριξη μαθημάτων ΑΠ και η ενσωμάτωση του σε LMSs.

3.7.1. Πόροι μαθησιακού υλικού

Το μαθησιακό υλικό περιλαμβάνει πόρους διαφόρων ειδών οι οποίοι περιγράφονται στη συνέχεια.

- *Σύντομα και περιεκτικά τμήματα θεωρίας, όπως:*
 - στο Σχήμα 3.14. που αφορά στην έννοια της διαπροσωπείας (interface) της κλάσης και του διαχωρισμού της από την υλοποίηση της (implementation),
 - στο Σχήμα 3.15. που αφορά στην έννοια της υπερίσχυσης μεθόδου (method's overriding),

4.2.4. ΔΙΑΠΡΟΣΩΠΕΙΑ (INTERFACE) – ΥΛΟΠΟΙΗΣΗ (IMPLEMENTATION)
Διαπροσωπεία (interface) του αντικείμενου είναι οι δημόσιες ιδιότητες και μέθοδοι του (μόνον οι υπογραφές τους) και Υλοποίηση (implementation) του αντικείμενου είναι οι ιδιωτικές ιδιότητες και μέθοδοι του και ο κώδικας όλων των μεθόδων του, ιδιωτικών και δημόσιων.



Σχήμα 3.14. Διαπροσωπεία-υλοποίηση κλάσης

7.1.2. ΕΝΑ ΠΑΡΑΔΕΙΓΜΑ ΥΠΕΡΙΣΧΥΣΗΣ ΜΕΘΟΔΟΥ (OVERRIDING), ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕ ΠΟΛΥ ΣΥΧΝΑ

Όλες οι κλάσεις είναι απόγονοι της κλάσης Object της Java. Η κλάση αυτή διαθέτει toString() μέθοδο που επιστρέφει ένα String object το οποίο περιέχει το όνομα της κλάσης στην οποία ανήκει το αντικείμενο και έναν κωδικό του αντικείμενου που λέγεται hash code. Όλες οι κλάσεις, θα θέλουν να ξαναυλοποιήσουν (overriding) την μέθοδο που κληρονομούν και να εμφανίζουν κάτι που θα έχει νόημα για αυτή την κλάση (και όχι έναν κωδικό που δεν είναι κατανοητός). Ήδη έχουμε συναντήσει αρκετές φορές την μέθοδο toString(). Σε όλες αυτές τις περιπτώσεις η μέθοδος toString() που υλοποιούσαμε ήταν overridden μέθοδος της υπερκλάσης Object.

Σχήμα 3.15. Υπερίσχυση μεθόδων

στο Σχήμα 3.16. που αφορά στις μορφές πολυμορφισμού (parametric and subtype polymorphism),

στο Σχήμα 3.17. που αφορά στη διάκριση μεταξύ υπερφόρτωσης και υπερίσχυσης μεθόδων (overriding versus overloading).



Σχήμα 3.16. Μορφές πολυμορφισμού

7.4. ΥΠΕΡΙΣΧΥΣΗ ΣΕ ΑΝΤΙΔΙΑΣΤΟΛΗ ΜΕ ΥΠΕΡΦΟΡΤΩΣΗ ΜΕΘΟΔΩΝ (Overriding vs Overloading)

7.4.1. ΘΕΩΡΙΑ

• Η υπερίσχυση (overriding) είναι ο ορισμός μιας μεθόδου στιγμιότυπου σε μια υποκλάση, για να υπερκαλύψει μια μέθοδο που διαφορετικά θα κληρονομούσε όπως ήταν από την υπερκλάση της. Οι δύο μέθοδοι έχουν ακριβώς τον ίδιο τύπο, σειρά και αριθμό παραμέτρων και τον ίδιο τύπο επιστροφής

Σε κάθε περίπτωση, η Java επιλέγει ποια μέθοδος θα εκτελεστεί, ανάλογα με το ποια είναι πλησιέστερη στην ιεραρχία στο object για το οποίο έχει κληθεί

• Η υπερφόρτωση (overloading) είναι ο ορισμός μεθόδου με το ίδιο όνομα αλλά διαφορετικό αριθμό ή/και τύπο παραμέτρων

Η Java επιλέγει σαν μέθοδο που θα εκτελεστεί, αυτή που η formal σειρά και τύπος παραμέτρων της, ταιριάζει με τις actual παραμέτρους στην εντολή κλήσης της μεθόδου. Δεν υπάρχει η έννοια της ιεραρχίας στην υπερφόρτωση μεθόδων

Είναι χαρακτηριστικό ότι ενώ η υπερφόρτωση μεθόδων επιλέγεται κατά την μεταγλώττιση του προγράμματος, η επίλυση της υπερίσχυσης μεθόδων γίνεται δυναμικά στο χρόνο εκτέλεσης.

Methods overloading = compile-time polymorphism

Methods overriding = run-time polymorphism

Σχήμα 3.17. Overriding versus Overloading

- Αντιπροσωπευτικά και κατάλληλα επιλεγμένα παραδείγματα κώδικα για την παρουσίαση εννοιών, όπως:
 - το παράδειγμα του Σχήματος 3.18. που παρουσιάζει την έννοια της στατικής (static) μεταβλητής κλάσης (static member) (από την 5^η διδακτική ενότητα του υλικού),
 - το παράδειγμα του Σχήματος 3.19. που παρουσιάζει τη δυναμική δέσμευση (dynamic binding).


```

class Counter {
// attributes του μετρητή
private int value = 0; // τιμή μετρητή
private static int limit = 100; // ανώτατο όριο του μετρητή

// methods του μετρητή
public void inc() { // αύξηση τιμής μετρητή
value++;
if (value >= limit)
value = 0; }

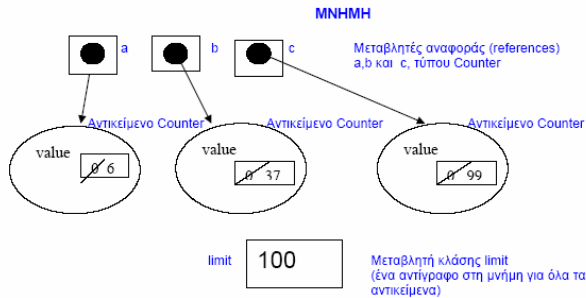
public int get() { return value; } // προσπέλαση τιμής μετρητή

public void set (int newvalue) { // καθορισμός τιμής μετρητή
if (newvalue >= 0 && newvalue <= limit)
value = newvalue;
else
{System.out.println("value out of limits");
value=0;}
}
}

```

COUNTER
- int value
- int limit
+ void inc()
+ void set(int newvalue)
+ int get()

και η κατάσταση στη μνήμη –αν φανταστούμε να εκτελείται το πρόγραμμα που αναφέρθηκε παραπάνω– διαμορφώνεται τώρα ως εξής:



Σχήμα 3.18. Στατική μεταβλητή κλάσης

7.3.2.2. ΠΑΡΑΔΕΙΓΜΑΤΑ

```

1.
class first
{
void print ()
{
System.out.println ("This is the print part of the first class");
}
}
class second extends first
{
void print () // override print
{
System.out.println ("This is the print part of the second class");
}
}

class DynamicBind
{
public static void main (String args[])
{
first ref = new second (); // reference of type first, new object of type second
// Ask yourself: for which object type, method print() is called, for first or
// second?
// and the answer indicates which method is called
ref.print();
}
}

```

Έξοδος του προγράμματος:
This is the print part of the second class

Σχήμα 3.19. Dynamic binding

- Λυμένες ασκήσεις, όπως:

η λυμένη άσκηση του Σχήματος 3.20. για την εμβάθυνση στην έννοια των τύπων αναφοράς (reference types) με τη χρήση objects' diagrams και σχηματικής παράστασης της μνήμης,

η λυμένη άσκηση του Σχήματος 3.21. για την αντιδιαστολή overloaded και overridden μεθόδων (από την 7^η διδακτική ενότητα του υλικού).

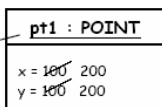
3.3.3. ΑΣΚΗΣΕΙΣ

1. Δίνεται το παρακάτω πρόγραμμα. Τι θα εκτυπωθεί στην κύρια έξοδο; Πώς διαμορφώνεται η κατάσταση στην μνήμη κατά την εκτέλεσή του;

```
import java.awt.Point; // εισαγωγή της έτοιμης Java κλάσης Point από το πακέτο (package)
// java.awt

class testReferences {
public static void main(String [] args) {
    Point pt1, pt2; // Δήλωση pt1, pt2: αναφορές (references) τύπου Point
                    // δηλ. μπορούν να γίνουν assignment σε αντικείμενα τύπου Point
    pt1 = new Point(100,100); // δημιουργία αντικειμένου Point το οποίο γίνεται
                            // assignment στην αναφορά pt1
    pt2 = pt1; // και οι δύο αναφορές, αναφέρονται ('δείχνουν') στο ίδιο αντικείμενο
    pt1.x = 200;
    pt1.y = 200;
    System.out.println("Point 1 : "+pt1.x+", "+pt1.y);
    System.out.println("Point 2 : "+pt2.x+", "+pt2.y);
}
}
```

ΔΙΑΓΡΑΜΜΑ ΑΝΤΙΚΕΙΜΕΝΟΥ



ή pt2 (δύο «ονόματα» για το ίδιο αντικείμενο)

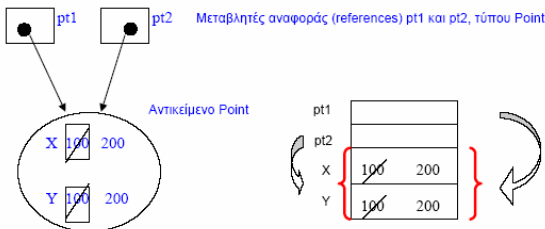
Επερώτηση:

Η Java διαθέτει ένα μεγάλο πλήθος υλοποιημένων κλάσεων τις οποίες μπορούμε να χρησιμοποιούμε. Οι κλάσεις αυτές είναι οργανωμένες σε "πακέτα" (packages), δηλ. κάτι σαν κατηγορίες ανάλογα με την χρήση τους. Έτσι κλάσεις που έχουν υλοποιηθεί για να εξυπηρετούν παρόμοιους σκοπούς (π.χ. κλάσεις για την δημιουργία Γραφικών Διεπαφών Χρήστη (Graphical User Interface, GUI), ή κλάσεις για τον χειρισμό ήχου και) ανήκουν στο ίδιο "πακέτο". Ένα από τα "πακέτα" της Java είναι το java.awt και μια από τις κλάσεις που περιέχει είναι η Point. Για να "γίνει" διαθέσιμη στο πρόγραμμά μας η κλάση αυτή, αρκεί σαν πρώτη εντολή να γράψουμε την import java.awt.Point;

Λύση

Point 1 : 200,200
Point 2 : 200,200

MNΗMH



Ερώτηση: Αν προστεθεί στο τέλος του παραπάνω προγράμματος η εντολή pt2 = null; θα είναι το αντικείμενο κατάλληλο για "καταστροφή" στο επόμενο garbage collection που θα συμβεί αυτόματα;

Απάντηση: Ένα αντικείμενο "καταστρέφεται" δηλ. αποδεσμεύεται ο χώρος μνήμης που καταλαμβάνουν τα ιδιοαρκητήρια του, όταν δεν υπάρχει καμία αναφορά (reference) συνδεδεμένη με αυτό. Με την εντολή pt2 = null; θα καταργηθεί η "σύνδεση" της pt2 αναφοράς με το υπάγονο αντικείμενο, αλλά θα εξακολουθεί να υπάρχει η "σύνδεση" της pt1 αναφοράς με αυτό. Άρα δεν θα "καταστραφεί" στο επόμενο garbage collection που θα συμβεί.

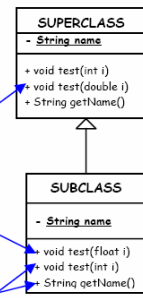
Σχήμα 3.20. Reference types

ΑΣΚΗΣΗ: Στο παρακάτω πρόγραμμα ποιες μέθοδοι είναι overridden και ποιες overloaded;

```
class Superclass {
private static String name = "Superclass";
public void test(final int i) {
    System.out.println("Superclass test(int): " + i);
}
public void test(final double i) {
    System.out.println("Superclass test(double): " + i);
}
public String getName() {
    return name;
}
}

class Subclass extends Superclass {
private static String name = "Subclass";
public void test(float i) {
    System.out.println("Subclass test(float): " + i);
}
public void test(final int i) {
    System.out.println("Subclass test(int): " + i);
}
public String getName() {
    return name;
}
}

public class Call1 {
public static void main(String args[]) {
    .....
}
}
```



ΛΥΣΗ:

Στην κλάση Superclass η μέθοδος void test(final double i) είναι overloaded διότι στην ίδια κλάση υπάρχει και η void test(final int i) δηλ. μια μέθοδος με ίδιο όνομα και διαφορετικό τύπο παραμέτρων.

Στην κλάση Subclass οι μέθοδοι void test(float i) και String getName() είναι overridden, διότι είναι 2 μέθοδοι που υπάρχουν με ίδιο όνομα και ακριβώς ίδια υπογραφή και στην υπερωκλήση της κλάσης στην οποία δηλώνονται, συνεπώς 2 μέθοδοι που κληρονομούνται και ορίζονται ξανά στην υποκλάση. Η μέθοδος void test(float i) είναι overloaded διότι στην ίδια κλάση υπάρχει και η void test(final int i) δηλ. μια μέθοδος με ίδιο όνομα και διαφορετικό τύπο παραμέτρων.

Σχήμα 3.21. Αντιδιαστολή overloaded-overridden μεθόδων

- Ερωτήσεις με απαντήσεις, όπως:

η ερώτηση του Σχήματος 3.22. που στοχεύει στο να ενισχυθεί η κατανόηση της "κρυφής" λειτουργίας της δημιουργίας αντικειμένων από μια κλάση (instantiation).

3.1. ΜΕΘΟΔΟΣ ΚΑΤΑΣΚΕΥΑΣΤΗ (CONSTRUCTOR)

Ερώτηση: Γιατί απαιτείται η δημιουργία των αντικειμένων (instantiation) μιας κλάσης; Δε μας αρκεί η κλάση;

Για παράδειγμα για μια δοσμένη κλάση Cat με ιδιοχαρακτηριστικό age δεν μπορούμε να πούμε

```
Cat.age = 5;
```

αλλά πρέπει να πούμε

```
Cat Frisky = new Cat(); // δημιουργία αντικειμένου Cat στο οποίο αναφέρεται  
// η μεταβλητή αναφοράς Frisky τύπου Cat
```

```
Frisky.age = 5;
```

Απάντηση:

Για να καταλάβουμε το γιατί ας δούμε τα παρακάτω.

Εντελώς αντίστοιχα αν σκεφτούμε τον τύπο των ακεραίων σαν μια κλάση, δεν μπορούμε να πούμε

```
int = 5;
```

αλλά λέμε

```
int x;  
x = 5;
```

Πρώτα δηλώνουμε μια μεταβλητή τύπου int και μετά την χειριζόμαστε.

Δηλαδή το ίδιο κάνουμε και για τους βασικούς (primitive) τύπους δεδομένων. Δηλώνουμε μεταβλητές του συγκεκριμένου τύπου (τόσες όσες απαιτεί το πρόβλημά μας).

Αντίστοιχα, στις κλάσεις δημιουργούμε αντικείμενά τους (με την χρήση του operator new) κι αυτά χειριζόμαστε.

Σημείωση:

Για να είμαστε ακόμα πιο σαφείς και συγκεκριμένοι στις κλάσεις γίνονται δυο ενέργειες: Δηλώνεται μια μεταβλητή (που παρακάτω θα δούμε ότι λέγεται μεταβλητή αναφοράς ή τύπος αναφοράς (reference type)) που έχει ως τύπο τον τύπο της κλάσης και δημιουργείται (με την χρήση του operator new) αντικείμενο τι κλάσης που το «όνομα του» (μπορούμε να πούμε) είναι το όνομα της μεταβλητής που δηλώθηκε. Χρησιμοποιώντας το όνομα αυτής της μεταβλητής χειριζόμαστε το αντικείμενο. Οι δυο αυτές ενέργειες μπορούν να γίνουν με έναν απ τους δυο παρακάτω ισοδύναμους τρόπους:

```
A) Cat Frisky; // μεταβλητή με όνομα Frisky τύπου Cat  
   Frisky = new Cat(); // δημιουργία ενός αντικειμένου τύπου Cat με «όνομα» Frisky  
B) Cat Frisky = new Cat(); // δήλωση μεταβλητής Frisky τύπου Cat και δημιουργία  
   // αντικειμένου στο οποίο αναφέρεται
```

Δηλαδή δεν αρκεί -όπως στους βασικούς τύπους δεδομένων- μόνο η δήλωση, χρειάζεται επιπλέον και η δημιουργία. Το παρακάτω δεν θα λειτουργήσει

```
Cat Frisky;  
Frisky.age = 3;
```

Επίσης χωρίς την δημιουργία αντικειμένων πως θα χειριζόμασταν την περίπτωση προβλημάτων στα οποία απαιτούνται περισσότερα από ένα αντικείμενα της κλάσης Cat;

Σε αντιστοιχία στους ακεραίους λέμε

```
int x;  
int y;  
x=4;  
y=7;
```

Σχήμα 3.22. Δημιουργία αντικειμένων

- Εκπαιδευτικές δραστηριότητες στα προγραμματιστικά περιβάλλοντα που χρησιμοποιούνται στην προσέγγιση, με περιγραφή του σκοπού τους, όπως: η δραστηριότητα του Σχήματος 3.23. στο περιβάλλον BlueJ, για την εμπέδωση των εννοιών διαπροσωπεία-υλοποίηση κλάσης, η δραστηριότητα του Σχήματος 3.24. στο περιβάλλον ArgoUML, για την αντίληψη της reverse engineering δυνατότητας του εργαλείου.

4.5. ΔΙΑΠΡΟΣΩΠΕΙΑ ΚΛΑΣΗΣ: ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ BLUEJ

Σκοπός της δραστηριότητας είναι να εμπεδώσετε τις έννοιες υλοποίηση και διαπρωσιμεία για μια κλάση, χρησιμοποιώντας υλοποιημένες κλάσεις (τις οποίες θα υποβάλουμε και σε τροποποιήσεις) και εκμεταλλευόμενοι το περιβάλλον BlueJ, στο οποίο υπάρχει η δυνατότητα να κινούμεθα ανάμεσα στις δύο όψεις, implementation και interface.

Ανοίξτε το περιβάλλον BlueJ. Ανοίξτε το project Rectangle.

1. Με right click πάνω στο σχήμα της κλάσης επιλέξτε Open Editor και πάνω δεξιά επιλέξτε Interface, για να δείτε ποια είναι η διαπρωσιμεία της κλάσης. Παρατηρήστε τη περιλαμβάνει. Επιλέγοντας από το ίδιο σημείο Implementation επιστρέψτε στον κώδικα της κλάσης. Κάντε συγκεκριμένες αλλαγές (τις οποίες θα σημειώσετε) στους προσδιοριστές ορατότητας των ιδιοτήτων και των μεθόδων της. Μετά τις αλλαγές ερευνήστε εκ νέου το Interface της κλάσης. Καταλαβαίνετε τις αλλαγές που επήλθαν; Παραμορφώστε κίνησης και άλλες αλλαγές παραμορφώστε τις επιπτώσεις τους στο Interface της κλάσης.

7.5.6. ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ArgoUML

Στην παρακάτω δραστηριότητα, θα εισάγετε τον έτοιμο κώδικα πολλών κλάσεων στο εργαλείο ArgoUML και θα παρατηρήσετε τα class diagrams που θα παραχθούν και τις μεταξύ τους σχέσεις.

Στο αρχείο κώδικα Java του παραπάνω παραδείγματος (αρχείο abstractclass.java) συμπεριλαμβάνεται ο κώδικας πέντε συνολικά κλάσεων. Πώς φαντάζεστε τα class diagrams των κλάσεων αυτών και τις σχέσεις τους; Σχεδιάστε το σε χαρτί. Ανοίξτε τώρα το ArgoUML και εισάγετε τον κώδικα. (File → Import Sources... και στο παράθυρο διαλόγου που θα εμφανισθεί πατήστε OK). Στο πάνω αριστερό τμήμα του παραθύρου, αναπτύξτε το Model για να εμφανισθούν τα συστατικά του και κάντε click πάνω στο Class Diagram με όνομα _classes. Έτσι στην περιοχή των Class Diagrams εμφανίζονται σε μίνι μέγεθος οι κλάσεις. Αναπτύξτε κάθε μια με right click και Show → Show All Compartments. Παρατηρήστε το αποτέλεσμα. Το είχατε σχεδιάσει έτσι;

Σχήμα 3.23. Δραστηριότητα στο BlueJ

Σχήμα 3.24. Δραστηριότητα στο ArgoUML

- Στο τέλος κάθε ενότητας, υπάρχουν ειδικά σχεδιασμένες ερωτήσεις-ασκήσεις που στοχεύουν στην εκτίμηση των γνώσεων που αποκτήθηκαν και των δεξιοτήτων που αναπτύχθηκαν, τις οποίες οι εκπαιδευόμενοι εκπονούν και παραδίδουν σε καθορισμένο χρόνο. Όπως:
η ερώτηση του Σχήματος 3.25. για τη διάκριση των εννοιών κλάση και αντικείμενο (από τη 2^η διδακτική ενότητα του υλικού),
οι ερωτήσεις του Σχήματος 3.26. για τη δημιουργία αντικειμένων (από την 3^η διδακτική ενότητα του υλικού),

2. Να κατατάξετε κάθε μια από τις παρακάτω λέξεις στην κατηγορία που πιστεύετε ότι ανήκει (αντικείμενο (object) ή κλάση (class))
Δέντρο
Ο τάδε φοιτητής του Τμήματος Δ.Τ.Ψ. Σ.
Κύκλος
Αεροπλάνο
Σχήμα
Σημείο συντεταγμένων (0,10)
Ο Σκύλος μας
Φοιτητό
Ο καθηγητής μου κ. Ταδόπουλος
Ανθρωπος
Το Τμήμα Α3 του 3ου Λυκ. Πειραιά
Παιδί
Το Τμήμα Διδ. Τεχν. Ψηφ. Συστημ.
Το παγωτό που έφαγα σήμερα
Η μοτοσυκλέτα μου

Σχήμα 3.25. Ερώτηση για τη διάκριση κλάσης-αντικειμένου

ΔΗΜΙΟΥΡΓΙΑ ΑΝΤΙΚΕΙΜΕΝΩΝ – ΜΕΘΟΔΟΣ ΚΑΤΑΣΚΕΥΑΣΤΗ

1. Σε ποιες από τις παρακάτω φάσεις υπάρχουν αντικείμενα (objects) στην μνήμη;
α) Μετά την συγγραφή ενός προγράμματος
β) Κατά την μεταγλώττιση του προγράμματος
γ) Μετά την μεταγλώττιση του προγράμματος
δ) Κατά την διάρκεια της εκτέλεσης του προγράμματος
ε) Μετά την εκτέλεση του προγράμματος
2. Τι είδους και πόσα αντικείμενα δημιουργούνται από το παρακάτω τμήμα προγράμματος;

```
class Point
{
    private double x;
    private double y;
}

public class testconstr {
    public static void main(String [] args) {
        Point a;
        new Point();
        .....
    }
}
```
3. Δίνεται το παρακάτω τμήμα προγράμματος. Αποδώστε σχηματικά την κατάσταση που δημιουργείται στη μνήμη κατά την εκτέλεσή του. Τι θα τυπωθεί στην κύρια έξοδο;

```
class Point
{
    private double x;
    private double y;
}

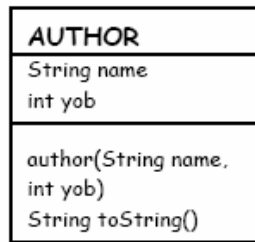
public Point () {
    x = 0;
    y = 0;
}

public Point (double x, double y) {
    this.x = x;
    this.y = y;
}

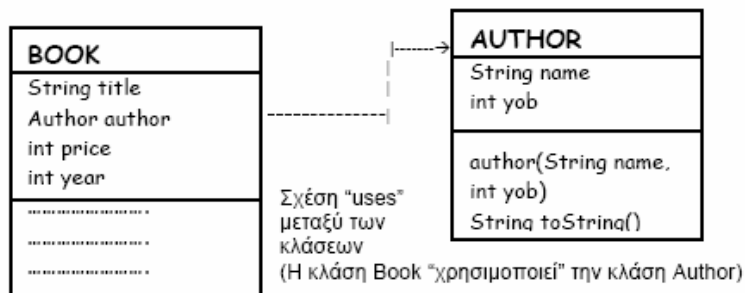
public void show () {
    System.out.println("tetmimeni:"+x);
    System.out.println("tetagmimeni:"+y);
}
.....
Point p1= new Point();
Point p2 = new Point(3,2);
p1.show();
p2.show();
.....
```

Σχήμα 3.26. Δημιουργία αντικειμένων

η άσκηση του Σχήματος 3.27. για την υλοποίηση συνθετιμένης (composed) κλάσης (από την 3^η διδακτική ενότητα του υλικού),



Τροποποιήστε την υλοποίηση της κλάσης Book, χρησιμοποιώντας την νέα κλάση Author. Το διάγραμμα κλάσεων που παριστάνει την νέα κατάσταση είναι:



Σχήμα 3.27. Υλοποίηση composed κλάσης

η ερώτηση του Σχήματος 3.28. για τη διάκριση των σχέσεων ιεραρχίας, όλου-μέρους, κλάσης-αντικειμένου (από την 7^η διδακτική ενότητα του υλικού).

3. Σε ποια από τα παρακάτω ζεύγη υπάρχει σχέση ιεραρχίας; (Δηλαδή σχέση υπερκλάσης υποκλάσης)

- | | |
|--|---------------------------------|
| person → man | Σάντουιτς → Τυρί |
| Γάτα → η Γάτα μας | Όχημα → Αυτοκίνητο |
| Ζώο → Ελέφαντας | person → εγώ |
| Αυτοκίνητο → Πόρτα | Εργαζόμενος → Τραπεζ. υπάλληλος |
| Κυκλικό σχήμα → η ρόδα του ποδηλάτου μου | Φρούτο → μήλο |
| Θηλαστικό → Γάτα | Δέντρο → Αχλαδιά |
| Μοτοσικλέτα → Τιμόνι | Αεροπλάνο → πτερύγια |
| Σχήμα → Κύκλος | |
| Εργαζόμενος → Ο εργαζόμενος κ.Τάδε στην Τράπεζα ABCD | |
| Κύκλος → ο Κύκλος κέντρου (0,0) και ακτίνας 5 cm | |

Σχήμα 3.28. Ερώτηση για τη διάκριση σχέσεων

- Στο υλικό, περιέχονται και όλα τα αρχεία πηγαίου (source) κώδικα Java των παραδειγμάτων που παρουσιάζονται και όλα τα αρχεία τα υλοποιημένα στα εκπαιδευτικά προγραμματιστικά περιβάλλοντα που χρησιμοποιούνται στην

προσέγγιση και τα οποία αρχεία χρησιμοποιούνται κατά την ανάπτυξη των θεμάτων.

- Οι διδακτικές ενότητες συμπληρώνονται με ένα ολοκληρωμένο παράδειγμα (μια μελέτη περίπτωσης, *case study*) που περιλαμβάνει τις προδιαγραφές απαιτήσεων (requirements specification), την περιγραφή του σχεδιασμού (design decisions), την τεκμηρίωση (documentation), τη δοκιμή (testing) (Σχήμα 3.29) και τον κώδικα μιας εφαρμογής στην οποία εφαρμόζονται και “επιδεικνύονται” οι αρχές του ΑΠ που έχουν παρουσιασθεί προηγουμένως. Η παιδαγωγική σημασία των μελετών περίπτωσης είναι επαρκώς τεκμηριωμένη (Linn and Clancy, 1992) και επίσης συστάσεις διδασκαλίας περιλαμβάνουν την ύπαρξη μελετών περίπτωσης, ώστε να προσφέρεται μια συνολική εικόνα για την οργάνωση των προγραμμάτων (Fleury, 2000).

Περιπτώσεις Δοκιμής

Περίπτωση 1.

Εργασία: Δημιουργία Τράπεζας

Δημιουργία αντικειμένου της κλάσης Bank

ΚΛΑΣΗ	ΜΕΘΟΔΟΣ	ΕΙΣΑΓΩΓΗ ΠΑΡΑΜΕΤΡΩΝ (ΔΕΔΟΜΕΝΑ)	ΤΙΜΗ ΕΠΙΣΤΡΟΦΗΣ (ΑΠΟΤΕΛΕΣΜΑΤΑ)
Bank	Κατασκευαστή	"JavaBank"	Αντικείμενο bank1

Περίπτωση 2.

Εργασία: Δημιουργία Πελάτη

Δημιουργία αντικειμένου της κλάσης Customer

ΚΛΑΣΗ	ΜΕΘΟΔΟΣ	ΕΙΣΑΓΩΓΗ ΠΑΡΑΜΕΤΡΩΝ (ΔΕΔΟΜΕΝΑ)	ΤΙΜΗ ΕΠΙΣΤΡΟΦΗΣ (ΑΠΟΤΕΛΕΣΜΑΤΑ)
Customer	Κατασκευαστή	"ΑΝΑΓΝΩΣΤΟΠΟΥΛΟΣ ΘΩΜΑΣ" "ΤΕΙΡΕΣΙΑ 10 12345 ΑΘΗΝΑ" "210 3425678"	Αντικείμενο customer1

Περίπτωση 3.

Εργασία: Δημιουργία Λογαριασμών «ταμιευτηρίου» και «τρεχούμενου» για τον πελάτη ΑΝΑΓΝΩΣΤΟΠΟΥΛΟ ΘΩΜΑ.

(Κλήση της μεθόδου openAccount από το αντικείμενο bank1 που έχουμε δημιουργήσει)

ΚΛΑΣΗ	ΜΕΘΟΔΟΣ	ΕΙΣΑΓΩΓΗ ΠΑΡΑΜΕΤΡΩΝ (ΔΕΔΟΜΕΝΑ)	ΤΙΜΗ ΕΠΙΣΤΡΟΦΗΣ (ΑΠΟΤΕΛΕΣΜΑΤΑ)
Bank- CurrentAccount Account	openAccount Κατασκευαστή getAccountNumber	customer1 "CurrentAccount"	Αριθμός Λογαριασμού:1

Bank SavingsAccount Account	openAccount Κατασκευαστή getAccountNumber	customer1 "SavingsAccount"	Αριθμός Λογαριασμού:2
-----------------------------------	---	-------------------------------	--------------------------

Σχήμα 3.29. Περιπτώσεις δοκιμής (test cases) της μελέτης περίπτωσης

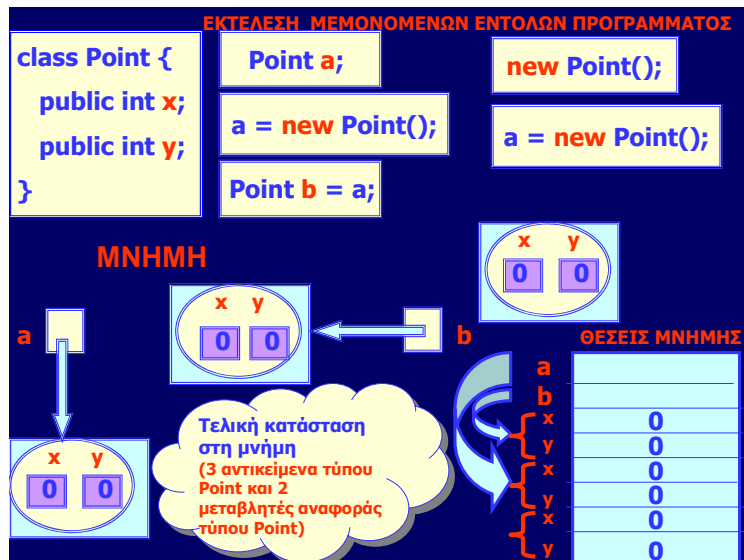
3.7.2. Σχεδιασμός μαθησιακού υλικού και μαθησιακές δυσκολίες

Στο σχεδιασμό όλου του υλικού, ιδιαίτερη προσοχή δόθηκε στο να καλυφθούν δυσκολίες και παρανοήσεις, αξιοποιώντας συστάσεις διδασκαλίας που έχουν διατυπωθεί. Τα παραδείγματα που σχεδιάστηκαν, δόθηκε μεγάλη προσοχή να μην είναι σε αντίθεση με τους γενικούς κανόνες και τις αρχές που θέλουμε να διδαχθούν οι εκπαιδευόμενοι, π.χ. κελυφοποίηση, δηλαδή η ανάπτυξη του υλικού και η προσέγγιση που ακολουθήθηκε είχε μια συνέπεια ως προς το σχεδιασμό της (Buck and Stucki, 2000) και ήταν σταθερά προσανατολισμένη στις βασικές αρχές-έννοιες της Αντικειμενοστρεφούς φιλοσοφίας Προγραμματισμού. Προσοχή δόθηκε στην ύπαρξη σχολίων μέσα στα παραδείγματα κώδικα, ώστε οι εκπαιδευόμενοι να έρχονται από πολύ νωρίς σε επαφή με κώδικα προσεκτικά σχεδιασμένο και σχολιασμένο (Kölling and Rosenberg, 2001b).

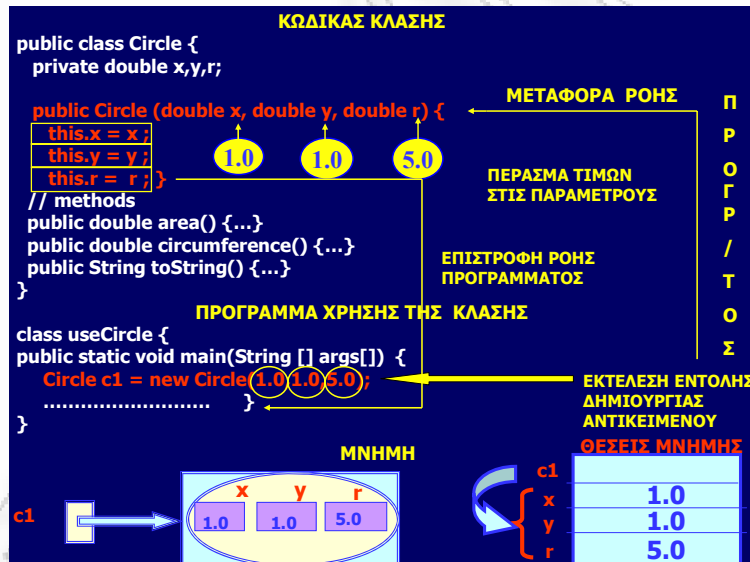
- Για την αντιμετώπιση συγχύσεων-παρανοήσεων-ταυτίσεων εννοιών ΑΠ:
 - ◆ Σχεδιάστηκαν παραδείγματα κλάσεων με τις ιδιότητες και τις μεθόδους τους και παραδείγματα αντικειμένων αυτών των κλάσεων, για την αποφυγή της ταύτισης των δύο εννοιών, την ενίσχυση της διάκρισης μεταξύ τους, την αποφυγή σύγχυσης των ιδιοτήτων ενός αντικειμένου με τα μέρη από τα οποία αποτελείται και τις ενέργειες που μπορεί να εκτελέσει.
 - ◆ Στα παραδείγματα και τις ασκήσεις που χρησιμοποιήθηκαν, οι κλάσεις διέθεταν περισσότερες από μία ιδιότητες και κάποιες διαφορετικού τύπου (int, String, double, κ.λ.π.) για την αποφυγή της ταύτισης των εννοιών αντικείμενο-μεταβλητή.
 - ◆ Δόθηκε ιδιαίτερη προσοχή στην παρουσίαση συνεχώς κλάσεων με σωστό σχεδιασμό για την εμπέδωση της αρχής της κελυφοποίησης από τους εκπαιδευόμενους και στην παρουσίαση παραδειγμάτων και ασκήσεων με περισσότερα του ενός αντικείμενα προερχόμενα από μια κλάση για να αποφευχθεί η σύγχυση κλάσης-αντικειμένου.
 - ◆ Αρχικά παρουσιάστηκαν μόνον οι έννοιες κλάση-αντικείμενο και καθυστέρησε η παρουσίαση της έννοιας της κληρονομικότητας. Στόχος αυτής της παρέμβασης ήταν η οικοδόμηση στέρεας γνώσης για τις πρωταρχικές αυτές έννοιες (σε συνδυασμό και με τις λειτουργίες δημιουργίας και καταστροφής

αντικειμένων και στη συνέχεια των εννοιών κελυφοποίηση – απόκρυψη πληροφορίας) και η διάκριση των σχέσεων κλάση-αντικείμενο από τις σχέσεις ιεραρχίας, κάτι μαθησιακά δύσκολο να επιτευχθεί από την αρχή. Η επιλογή αυτή διαφοροποιεί την προτεινόμενη προσέγγιση που περιγράφεται και προτείνεται από την “objects first” όπως παρουσιάζεται στο CC2001 (2001), που υπαγορεύει την παρουσίαση των εννοιών κλάση, αντικείμενο, κληρονομικότητα από την αρχή (CC2001, 2001, p. 30: “The first course in either sequence begins immediately with the notions of objects and inheritance, giving students early exposure to these ideas.”). Στο επόμενο κεφάλαιο των μελετών περίπτωσης (εφαρμογές της προσέγγισης) αξιολογείται η αποτελεσματικότητα αυτής της επιλογής.

- Για την αντιμετώπιση προβλημάτων σχετικά με τη δημιουργία στιγμιότυπων και τη μέθοδο του κατασκευαστή, σχεδιάστηκαν και αναλύθηκαν παραδείγματα, εξαντλώντας όλες τις δυνατές διαφορετικές περιπτώσεις κατασκευαστή (εξ’ ορισμού και άμεση δήλωση) και τρόπων αρχικοποίησης των στιγμιότυπων.
- Λειτουργίες που σχετίζονται με τη μνήμη και άρα είναι “εσωτερικές” και “αόρατες”, συμβαίνουν κυρίως κατά τη δημιουργία αντικειμένων, κατά τον ορισμό των τύπων αναφοράς (reference types), κατά τη “σύνδεση” και “αποσύνδεση” των τύπων αναφοράς με και από τα αντικείμενα, κατά την καταστροφή των αντικειμένων (destruction) και κατά την έμμεση κλήση του κατασκευαστή μέσω του τελεστή new. Ο κύριος λόγος της δυσκολίας είναι ότι οι εκπαιδευόμενοι δεν μπορούν να “δουν” και να “αλληλεπιδράσουν” με αυτές τις λειτουργίες. Αυτό μας οδήγησε στην ενσωμάτωση μιας απόπειρας δυναμικής “οπτικοποίησης” (Ben-Ari et al., 2002) αυτού που συμβαίνει στη μνήμη σε όλες τις παραπάνω περιπτώσεις, με τη χρήση λογισμικού προσομοίωσης (PowerPoint) για ένα προσεκτικά επιλεγμένο σύνολο εντολών (Σχήμα 3.30.). Σε συνδυασμό, χρησιμοποιήθηκε και η στατική σχηματική παράσταση της κατάστασης στη μνήμη (βλ. Σχήματα 3.30., 3.31.). Η παρέμβαση αυτή, στόχευε να αντιμετωπίσει και τη σύγχυση ταυτότητας και ιδιότητας ενός αντικειμένου (με όνομα name ή παρόμοιο), με τη χρήση πολλαπλών αναθέσεων και αναθέσεων εκ νέου τύπων αναφοράς σε αντικείμενα.



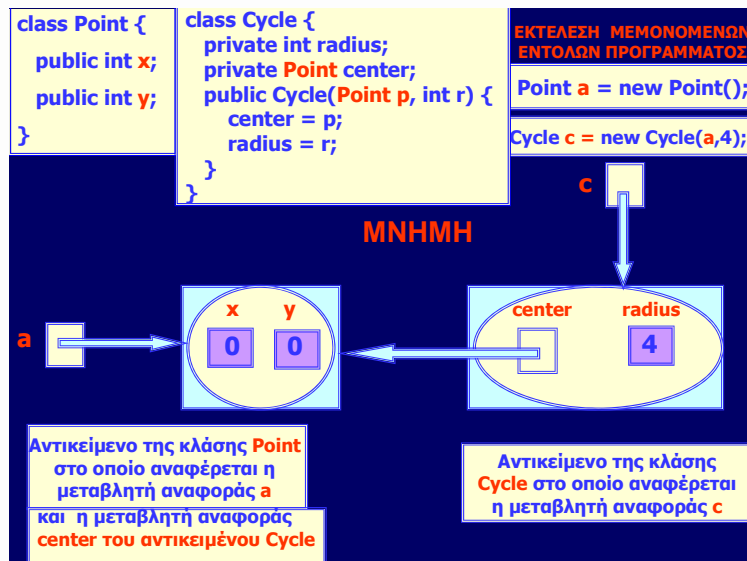
Σχήμα 3.30. Δυναμική οπτικοποίηση κατάστασης μνήμης



Σχήμα 3.31. Δυναμική οπτικοποίηση κατάστασης μνήμης και ροή προγράμματος

- Θεωρήθηκε η έννοια του “τύπου αναφοράς” πολύ σημαντική για την κατανόηση άλλων θεμάτων, όπως ο πολυμορφισμός και η δυναμική δέσμευση. Για την αντίληψη θεμάτων όπως η ανάθεση ενός τύπου αναφοράς του τύπου μιας υπερκλάσης σε ένα αντικείμενο του τύπου μιας υποκλάσης – η πιο ουσιαστική μορφή πολυμορφισμού – απαιτείται η οικοδόμηση στέρεας γνώσης για τους τύπους αναφοράς. Κατά συνέπεια, δόθηκε ιδιαίτερη έμφαση και προσοχή στη διδασκαλία τους και στην αντιπαράθεση τους με τους βασικούς τύπους δεδομένων (primitive types), με τη χρήση

οπτικοποίησης (Σχήματα 3.32., 3.33.), σχηματικών παραστάσεων, δραστηριοτήτων, λυμένων ασκήσεων.



Σχήμα 3.32. Reference types – composed κλάσεις



Σχήμα 3.33. Κατανομή μνήμης – primitive types

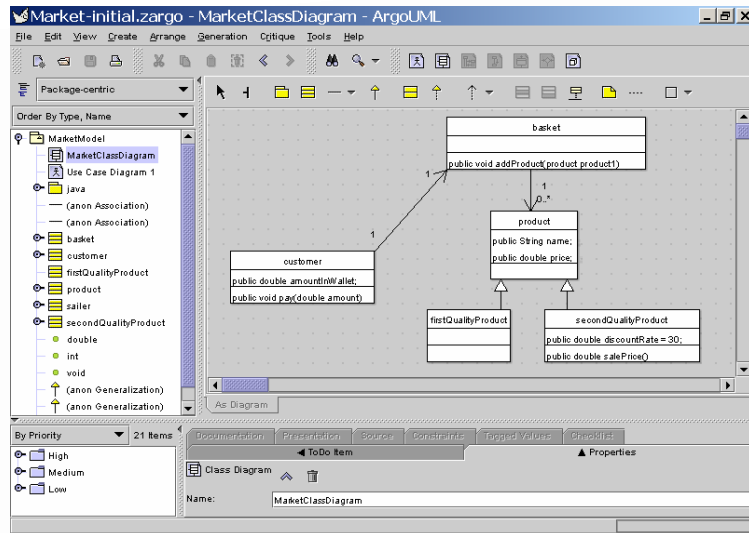
3.8. Μαθησιακά εργαλεία

Οι ανάγκες της εκπαιδευτικής φιλοσοφίας της προτεινόμενης διδακτικής προσέγγισης απαιτούν:

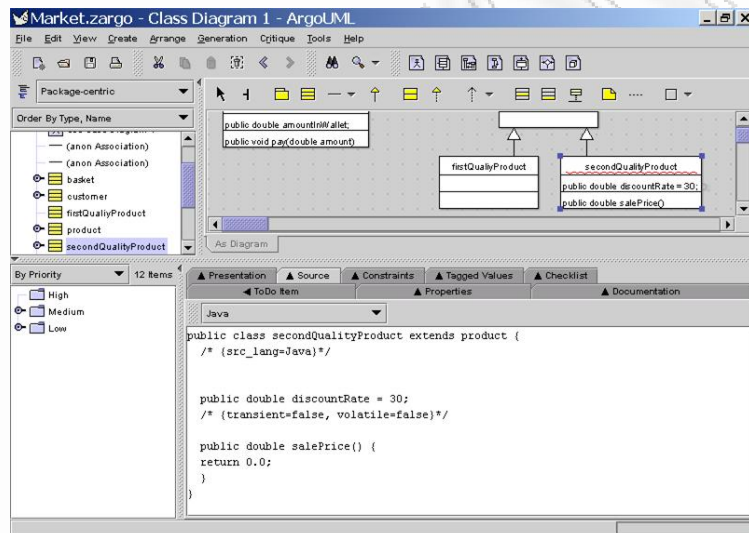
- ένα εργαλείο για την υποστήριξη της “model first” “πλευράς” της προσέγγισης που θα έχει τη δυνατότητα παραγωγής “σκελετικού” κώδικα Java,
- ένα προγραμματιστικό περιβάλλον που θα “επιβάλλει” με τα χαρακτηριστικά του τη μοντελοποίηση, θα υποστηρίζει την οπτικοποίηση εννοιών και σχέσεων, θα είναι εκπαιδευτικό αλλά και συγχρόνως θα μπορεί να αποτελέσει ένα πλήρες περιβάλλον ανάπτυξης εφαρμογών και θα υποστηρίζει τη γλώσσα Προγραμματισμού Java,
- ένα επαγγελματικό ολοκληρωμένο προγραμματιστικό περιβάλλον ανάπτυξης εφαρμογών Java.

Τα προγραμματιστικά περιβάλλοντα και εργαλεία για τα οποία σχεδιάστηκαν δραστηριότητες που ενσωματώθηκαν σε συγκεκριμένα σημεία του μαθησιακού υλικού, και τα οποία χρησιμοποιούν οι εκπαιδευόμενοι κατά την εκπόνηση των εργασιών, είναι:

- Το *UML εργαλείο ArgoUML* (<http://argouml.tigris.org/>). Η UML είναι “γραφική” γλώσσα σχεδίασης συστημάτων Λογισμικού. Για την υποστήριξη της μοντελοποίησης στη διδακτική προσέγγιση χρησιμοποιούνται κυρίως τα διαγράμματα κλάσης (class diagrams) από αυτή τη γλώσσα. Το εργαλείο ArgoUML είναι από τα απλούστερα UML εργαλεία, “ελαφρύ”, open source και παρέχει τη δυνατότητα παραγωγής “σκελετικού” κώδικα Java από τα διαγράμματα και αντίστροφα τη δυνατότητα παραγωγής διαγραμμάτων από τον κώδικα. Χρησιμοποιείται για τη σχεδίαση του αυθεντικού παραδείγματος της “λαϊκής αγοράς” (Σχήματα 3.34., 3.35.) στην πρώτη φάση του μαθησιακού σεναρίου, για την πραγματοποίηση εκπαιδευτικών δραστηριοτήτων από τους εκπαιδευόμενους οι οποίες περιέχονται στο μαθησιακό υλικό, και κατά την εκπόνηση των εργασιών.



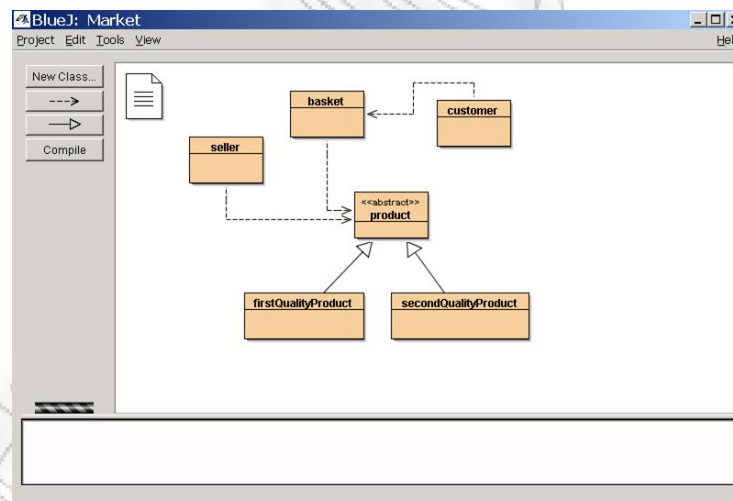
Σχήμα 3.34. Ένα στιγμιότυπο των διαγραμμάτων κλάσεων του αυθεντικού παραδείγματος της “λαϊκής αγοράς” (ArgoUML)



Σχήμα 3.35. Ο παραγόμενος “σκελετικός” κώδικας (ArgoUML)

- Το εκπαιδευτικό προγραμματιστικό περιβάλλον *BlueJ* (<http://www.bluej.org/>) (Kölling and Rosenberg, 2001a; Kölling et al., 2003). Αρχικά χρησιμοποιείται για τη μεταφορά του “σκελετικού” κώδικα που παράγει το εργαλείο ArgoUML και την περαιτέρω ανάπτυξη του αυθεντικού παραδείγματος της “λαϊκής αγοράς” (Σχήμα 3.36.). Επιλέχθηκε γιατί υποστηρίζει πλήρως την OF προσέγγιση “αντανακλώντας” το αντικειμενοστρεφές παράδειγμα προγραμματισμού, είναι εύκολο στη χρήση του (*simplicity*), αποκτούν σε αυτό οπτική υπόσταση οι κλάσεις και τα αντικείμενα (*visualization*) και επίσης επιτρέπει αλληλεπίδραση με τις κλάσεις δημιουργώντας αντικείμενα και με τα αντικείμενα καλώντας μεθόδους και ελέγχοντας την

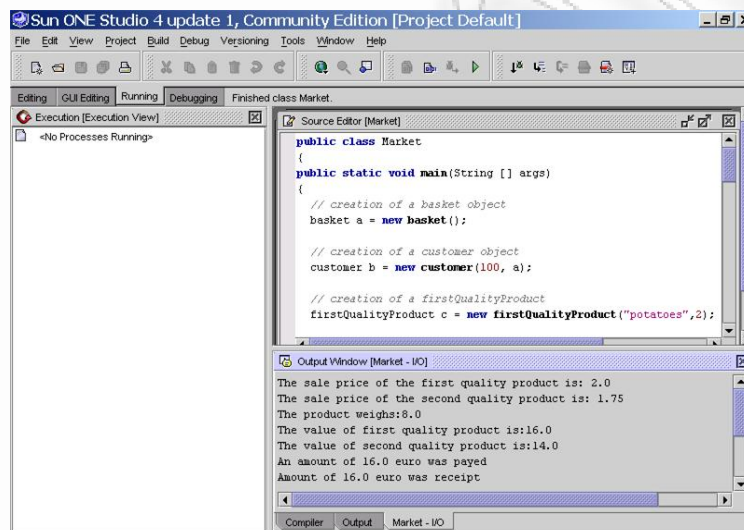
κατάσταση τους (*interaction*), δηλαδή επιτρέπει την αλληλεπίδραση και τον πειραματισμό κατά τη μαθησιακή διαδικασία. Επίσης είναι ένα εργαλείο το οποίο επιβάλλει το “modeling” (Berge et al., 2003a; Borge and Kaasbøll, 2003) και υποστηρίζει πλήρως την ανάπτυξη προγραμμάτων με κώδικα Java με εύκολο πέρασμα από το “οπτικό” περιβάλλον στο “κειμενικό” του κώδικα. Στην κατεύθυνση αντιμετώπισης διαπιστωμένων συγχύσεων-παρανοήσεων, αξιοποιείται με (α) τη δημιουργία πολλών αντικειμένων από μια κλάση και μελέτη-παρατήρηση της γραφικής τους αναπαράστασης, για την αποφυγή της σύγχυσης κλάσης-αντικειμένου και (β) τη δημιουργία αντικειμένων με ακριβώς την ίδια κατάσταση και έλεγχο των τιμών των ιδιοτήτων τους μέσω της δυνατότητας “inspection” για την αποφυγή της παρανόησης ότι αντικείμενα με ίδια κατάσταση είναι το ίδιο “ένα” αντικείμενο. Αξιοποιείται επίσης για την αντιμετώπιση του προβλήματος αντίληψης της συνολικής ροής-εκτέλεσης των προγραμμάτων χρησιμοποιώντας ολοκληρωμένα προγράμματα (μέθοδος main).



Σχήμα 3.36. Μεταφορά του αυθεντικού παραδείγματος στο BlueJ

- Ο προγραμματιστικός μικρόκοσμος *Jeroo* (<http://www.jeroo.org/>) (Sanders and Dorn, 2003a; b). Η χρήση των προγραμματιστικών μικρόκοσμων έχει προταθεί για την αντιμετώπιση προβλημάτων. Επιλέχτηκε η χρήση του *Jeroo* (υποεπότητα 2.5.1.) για την υποστηρικτική προσέγγιση των εννοιών “αντικείμενο” και “μέθοδος” στις οποίες από κατασκευής το περιβάλλον εστιάζει και επίσης για τη συνάφεια της γλώσσας του με τη Java και την ευκολία στη σύνταξη της.
- Το ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών Java, *SUN One Studio*. Η διδασκαλία της διαδικασίας υλοποίησης του αυθεντικού παραδείγματος της “λαϊκής

αγοράς” στο τελικό στάδιο, υποστηρίχθηκε από τη χρήση αυτού του περιβάλλοντος (Σχήμα 3.37.). Χρησιμοποιήθηκε ως το κατάλληλο εκπαιδευτικό μέσο για την επίδειξη της δημιουργίας, μεταγλώττισης και εκτέλεσης της “ολοκληρωμένης” αντικειμενοστρεφούς εφαρμογής του αυθεντικού παραδείγματος (με “main” μέθοδο). Επιλέχτηκε για την ανάπτυξη ικανότητας χρήσης ενός επαγγελματικού περιβάλλοντος και την αντιμετώπιση του προβλήματος αντίληψης της συνολικής ροής-εκτέλεσης των προγραμμάτων (Ragonis and Ben-Ari, 2005a). Παρουσιάστηκαν ολοκληρωμένα προγράμματα –αλλά και σχεδιάστηκαν σχετικές δραστηριότητες– που κάνουν χρήση κλάσεων, δημιουργούν αντικείμενα τους και καλούν μεθόδους επί αυτών με τη χρήση κώδικα, για τη δημιουργία μιας πλήρους εικόνας των Αντικειμενοστρεφών προγραμμάτων και της δυναμικής της εκτέλεσής τους. Αποτελεί ένα απλό και “ελαφρύ” ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών Java.



Σχήμα 3.37. Στιγμιότυπο από το περιβάλλον SUN One Studio, μετά την εκτέλεση του προγράμματος με main μέθοδο που υλοποιεί το αυθεντικό παράδειγμα

3.9. Ρόλος εμπλεκόμενων στη μαθησιακή διαδικασία

Κύρια συνιστώσα στην προτεινόμενη διδακτική προσέγγιση αποτελεί το εμπλεκόμενο ανθρώπινο δυναμικό που καλείται να επιτελέσει συγκεκριμένους ρόλους ανά φάση της προτεινόμενης ροής μαθησιακών δραστηριοτήτων. Προβλέπονται οι ακόλουθοι ρόλοι:

- Εκπαιδευτής

- Εκπαιδευόμενος/οι
- Προσωπικό τεχνικής υποστήριξης

Ο εκπαιδευτής ενδέχεται να μην είναι ένας ή να υπάρχει και βοηθός εκπαιδευτή. Στον Πίνακα 3.1. φαίνονται οι ενδεικτικοί ρόλοι του ανθρώπινου δυναμικού για την εφαρμογή της διδακτικής προσέγγισης και τα καθήκοντα του καθενός.

Πίνακας 3.1. Ενδεικτικοί ρόλοι του ανθρώπινου δυναμικού

Ρόλος	Καθήκοντα
Εκπαιδευτής	<ul style="list-style-type: none"> • Δημιουργεί στο ηλεκτρονικό περιβάλλον μάθησης (LMS) το δικτυακό τόπο στον οποίο θα έχουν πρόσβαση οι εκπαιδευόμενοι κατά τη διάρκεια της μαθησιακής διαδικασίας • Ενημερώνει στην πρώτη δια ζώσης διδακτική συνάντηση τους εκπαιδευόμενους για τον τρόπο διεξαγωγής του μαθήματος, τον τρόπο επικοινωνίας (URL του ηλεκτρονικού περιβάλλοντος μάθησης, e-mail, chat, forum, κ.λ.π.), τη δομή του μαθησιακού υλικού, τα μαθησιακά εργαλεία, το περιεχόμενο και τον τρόπο αξιοποίησης του μαθησιακού υλικού και των επιπλέον μαθησιακών πηγών, τη διαδικασία αξιολόγησης και για κάποιες σημαντικές αρχικές ημερομηνίες • Οργανώνει και διεκπεραιώνει τις ομαδικές διδακτικές συναντήσεις με τους εκπαιδευόμενους και φροντίζει για τη διαθεσιμότητα των απαραίτητων σε αυτές εποπτικών μέσων διδασκαλίας • Ενημερώνει το μαθησιακό υλικό του δικτυακού τόπου • Αναρτά έγκαιρα στο δικτυακό τόπο οποιαδήποτε πληροφορία ή ανακοίνωση κατά τη διάρκεια της εφαρμογής της προσέγγισης για τη διαδικασία του μαθήματος (επικείμενα γεγονότα όπως διδακτικές συναντήσεις στην τάξη, προθεσμίες παράδοσης εργασιών κ.λ.π.) • Πληροφορεί και καθοδηγεί τους εκπαιδευόμενους, όποτε χρειασθεί, για τον τρόπο μελέτης του υλικού • Προτείνει δραστηριότητες εμπέδωσης και πρακτικής των αντικειμενοστρεφών εννοιών • Δίνει θέματα για συζήτηση και προβληματισμό στο διαδικτυακό τόπο ασύγχρονης συζήτησης • Απαντά σε ερωτήσεις, επιλύει απορίες, αποσαφηνίζει έννοιες του γνωστικού αντικειμένου και δίνει ανατροφοδότηση στους εκπαιδευόμενους τόσο στις δια ζώσης συναντήσεις όσο και μέσω του ηλεκτρονικού περιβάλλοντος μάθησης • Δίνει συμβουλές μέσω e-mail ή στο forum ασύγχρονης συζήτησης • Παρακολουθεί τη διεξαγωγή συζητήσεων και επεμβαίνει για

<p>Εκπαιδευτής</p>	<p>τη διευθέτηση θεμάτων ή υποκινεί τους εκπαιδευόμενους σε πιο ενεργή συμμετοχή</p> <ul style="list-style-type: none"> • Στηρίζει ψυχολογικά τους εκπαιδευόμενους καθ' όλη τη διάρκεια της μαθησιακής διαδικασίας • Παρέχει εξατομικευμένη βοήθεια στους εκπαιδευόμενους • Συνεργάζεται με τους άλλους εκπαιδευτές (στην περίπτωση περισσοτέρων του ενός) και με το βοηθό εκπαιδευτή για θέματα που αφορούν στη διεξαγωγή της εφαρμογής της διδακτικής προσέγγισης και τον καταμερισμό εργασιών (ανά διδακτική ενότητα ή/και ανά καθήκον) αλλά και επί του περιεχομένου της μαθησιακής διαδικασίας • Εμπλουτίζει το μαθησιακό υλικό, αν το επιτρέπει η χρονική εξέλιξη της εφαρμογής και το επίπεδο των εκπαιδευομένων • Παρακολουθεί την ποιότητα της εργασίας του βοηθού εκπαιδευτή • Μεριμνά για την απρόσκοπτη παρακολούθηση των εκπαιδευομένων και τη συμμετοχή τους στις δια ζώσης συναντήσεις, συνεργαζόμενος με τους αρμόδιους για τις απαιτήσεις της διεξαγωγής των ομαδικών συναντήσεων • Παρακινεί τους εκπαιδευόμενους να συμμετέχουν στις ασύγχρονες συζητήσεις • Δίνει επιπρόσθετες οδηγίες στους εκπαιδευόμενους σχετικά με τις εργασίες προς εκπόνηση και τη διαδικασία αξιολόγησης • Παραλαμβάνει τις εργασίες που οι εκπαιδευόμενοι αποστέλλουν μέσω του ηλεκτρονικού περιβάλλοντος μάθησης και τους ενημερώνει για τυχόν προβλήματα με τις εργασίες τους • Σχολιάζει τις εργασίες των εκπαιδευομένων (εξατομικευμένα) προς την κατεύθυνση να “θεραπεύσει” δυσκολίες και παρανοήσεις τους • Φροντίζει να μην παραβιάζονται οι προθεσμίες των γεγονότων • Διορθώνει τις εργασίες των εκπαιδευομένων • Ανακοινώνει το βαθμό επίδοσης κάθε εκπαιδευόμενου σύμφωνα με τα κριτήρια αξιολόγησης που έχουν τεθεί στην αρχή της εφαρμογής • Σημειώνει τυχόν προβλήματα και δυσλειτουργίες κατά τη διάρκεια της διεξαγωγής της εφαρμογής για να ληφθούν υπόψη σε επόμενη εφαρμογή • Σημειώνει δυσκολίες και παρανοήσεις που διαπιστώνει να παρουσιάζουν οι εκπαιδευόμενοι σχετικά με το γνωστικό αντικείμενο, ώστε να ληφθούν υπόψη για τη βελτίωση των κατάλληλων συνιστωσών της διδακτικής προσέγγισης
<p>Εκπαιδευόμενος</p>	<ul style="list-style-type: none"> • Συμμετέχει στις ομαδικές διδακτικές συναντήσεις

Εκπαιδευόμενος	<ul style="list-style-type: none"> • Εγκαθιστά τα προβλεπόμενα από την προσέγγιση εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα • Μελετά το υλικό και πραγματοποιεί τις προτεινόμενες από το πρόγραμμα και τον εκπαιδευτή μαθησιακές δραστηριότητες • Θέτει απορίες και θέματα προς συζήτηση στο ηλεκτρονικό περιβάλλον μάθησης και στις δια ζώσης συναντήσεις • Απαντά σύγχρονα ή ασύγχρονα σε θέματα προς συζήτηση που τέθηκαν από άλλους εκπαιδευόμενους ή/και τον εκπαιδευτή ή το βοηθό • Επικοινωνεί σύγχρονα ή ασύγχρονα με συνεκπαιδευόμενους ή/και σε προσωπικό επίπεδο με τον εκπαιδευτή και τους βοηθούς • Εκπονεί και παραδίδει έγκαιρα τις προβλεπόμενες εργασίες
Τεχνικό προσωπικό	<ul style="list-style-type: none"> • Απαντά ασύγχρονα σε τεχνικά θέματα που τέθηκαν από εκπαιδευόμενους ή εκπαιδευτές • Επιλύει τεχνικά προβλήματα • Διασφαλίζει την απρόσκοπτη λειτουργία της διαδικασίας ηλεκτρονικής μάθησης (LMS) σε τεχνικό επίπεδο • Προασπίζει την ασφάλεια της διαδικασίας ηλεκτρονικής μάθησης (LMS) σε τεχνικό επίπεδο

3.10. Σύνοψη

Στο κεφάλαιο αυτό παρουσιάστηκαν και αναπτύχθηκαν αναλυτικά τα θεμέλια και τα δομικά στοιχεία της διδακτικής προσέγγισης που προτείνεται για την εισαγωγή στον ΑΠ. Η διδασκαλία του ΑΠ, στην ανάπτυξη της προσέγγισης αντιμετωπίστηκε ως διδακτικό πρόβλημα προς λύση και ο σχεδιασμός της ως συστημική προσέγγιση για την επίλυσή του με ανάπτυξη συγκεκριμένου παιδαγωγικού και τεχνολογικού πλαισίου και τρόπου εφαρμογής.

Η ανάπτυξη του κεφαλαίου παρουσίασε το πρόβλημα, τη διδακτέα ύλη, την εκπαιδευτική φιλοσοφία της προσέγγισης (μακροσκοπική περιγραφή) και τις δομικές συνιστώσες της προσέγγισης που αναπτύχθηκε (περιγραφή σε μικροεπίπεδο) που είναι το μαθησιακό υλικό της, τα μαθησιακά εργαλεία που επελέγησαν να χρησιμοποιηθούν και το ανθρώπινο δυναμικό που εμπλέκεται στην εφαρμογή της στη διδακτική πράξη με τα καθήκοντα του καθενός.

Στο επόμενο κεφάλαιο παρουσιάζονται οι περιπτώσεις εφαρμογής της προσέγγισης στη διδακτική πράξη (case studies), οι μελέτες αξιολόγησής τους με τα ευρήματά τους και οι βελτιώσεις της προσέγγισης μετά από κάθε εφαρμογή της.

РАНЕЕ НЕ ПЕРПА

ΚΕΦΑΛΑΙΟ 4

Εφαρμογή και Αξιολόγηση της Προτεινόμενης Διδακτικής Προσέγγισης – Μελέτες Περίπτωσης (Case Studies)

4.1. Εισαγωγή

Η προτεινόμενη διδακτική προσέγγιση που παρουσιάστηκε στο προηγούμενο κεφάλαιο είναι απόρροια διαδοχικών εφαρμογών σε αυθεντικά εκπαιδευτικά περιβάλλοντα στην Τριτοβάθμια εκπαίδευση. Οι εφαρμογές αυτές, αποσκοπούσαν στη διερεύνηση της μαθησιακής αποτελεσματικότητας της προτεινόμενης μεθόδου και στην ανακάλυψη σημείων που έχρηζαν βελτίωσης.

Στο κεφάλαιο αυτό, περιγράφονται αναλυτικά τρεις εφαρμογές της προτεινόμενης διδακτικής προσέγγισης (μελέτες περίπτωσης). Περιγράφονται με λεπτομέρεια το εκπαιδευτικό σκηνικό (χώρος, χρόνος, συνθήκες) και τα αποτελέσματα αξιολόγησης. Μέσα από τα συμπεράσματα και τις βελτιώσεις που προτεινόταν σε κάθε μελέτη περίπτωσης, περιγράφεται η εξελικτική πορεία της ανάπτυξης της διδακτικής προσέγγισης για τον ΑΠ και αιτιολογείται το πώς δημιουργήθηκε η τελικά προτεινόμενη μορφή της διδακτικής προσέγγισης.

Συγκεκριμένα μέσα σε διάστημα δύο ακαδημαϊκών ετών πραγματοποιήθηκαν τρεις μελέτες περίπτωσης σε αυθεντικά περιβάλλοντα Τριτοβάθμιας Εκπαίδευσης σε προπτυχιακό και μεταπτυχιακό επίπεδο σπουδών και οι αντίστοιχες συστηματικές μελέτες αξιολόγησής τους.

Η ανατροφοδότηση από κάθε μια μελέτη αξιολόγησης, λειτουργούσε ως δεδομένο ενίσχυσης και βελτίωσης της προτεινόμενης διδακτικής προσέγγισης.

Οι μελέτες περίπτωσης πραγματοποιήθηκαν:

- Σε μακράς διάρκειας σεμινάριο κατά το εαρινό εξάμηνο του 2005 (ενότητα 4.4.) από το Μάρτιο του 2005 έως το Μάιο του 2005, με τη μορφή φροντιστηριακού μαθήματος για τους προπτυχιακούς φοιτητές του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά.
- Με τη μορφή e-learning μαθήματος, ως μέρος του υποχρεωτικού μαθήματος “Αντικειμενοστρεφής Προγραμματισμός” στο Τμήμα ΔΤΨΣ του Πανεπιστημίου

Πειραιά, αλλά ανεξάρτητο αυτού και υποστηρικτικό αυτού (ενότητα 4.5.). Η εφαρμογή αυτή έγινε κατά το εαρινό εξάμηνο του 2006 (Φεβρουάριος 2006 – Ιούνιος 2006).

- Σε σεμιναριακό e-learning μάθημα από το Μάρτιο του 2006 έως το Μάιο του 2006 (ενότητα 4.6.), στο Τ.Ε.Ι. Πειραιά, στο πλαίσιο του ΠΜΣ “Information Technology”, που διεξάγεται σε συνεργασία με το Paisley University of Scotland. Στο κεφάλαιο αυτό, αρχικά θα παρουσιασθεί μια συνοπτική περιγραφή της εξέλιξης της προτεινόμενης μεθόδου. Στη συνέχεια θα δοθούν στοιχεία για το εννοιολογικό πλαίσιο των μελετών αξιολόγησης που πραγματοποιήθηκαν και θα ακολουθήσει η παρουσίαση των τριών πιλοτικών εφαρμογών της προτεινόμενης διδακτικής προσέγγισης και των ευρημάτων των μελετών αξιολόγησής τους.

4.2. Η εξελικτική ανάπτυξη της προσέγγισης μέσω επανειλημμένων εφαρμογών και αξιολογήσεων

Στις ενότητες 4.4., 4.5. και 4.6. περιγράφονται οι τρεις μεγάλης διάρκειας εφαρμογές της προτεινόμενης διδακτικής προσέγγισης.

Στην πρώτη από τις παραπάνω εφαρμογές, ενδιέφερε κυρίως, να αξιολογηθεί η ποιότητα του μαθησιακού υλικού και των πόρων από τους οποίους αποτελείται (ενότητα 3.7.), η αποδοχή του τρόπου διεξαγωγής της (on-line μαθησιακό περιβάλλον (ενότητα 3.6.)) και η αξιοποίηση των μαθησιακών εργαλείων (ενότητα 3.8.).

Συγχρόνως, η πρώτη αυτή εφαρμογή απέφερε σημαντικό υλικό για την περαιτέρω βελτίωση και ενίσχυση της διδακτικής προσέγγισης. Προέκυψε η ανάγκη να γίνει επίδειξη στους εκπαιδευόμενους συνολικά της ανάπτυξης μιας αντικειμενοστρεφούς εφαρμογής με έμφαση στη μοντελοποίηση, δηλαδή να υπάρξει στη μαθησιακή διαδικασία η φάση της παρατήρησης (Φάση I, ενότητα 3.5.), να δοθεί έμφαση στη δημιουργία ενεργών on-line κοινοτήτων μάθησης μεταξύ των εκπαιδευομένων και προστέθηκε η εκπόνηση από μέρους των εκπαιδευομένων πέραν των μικρών επί μέρους εργασιών, ενός ενιαίου project με συνεκτικότητα που αναπτύχθηκε βήμα-βήμα (Φάση II, ενότητα 3.5.). Επίσης προέκυψε η αναγκαιότητα αλλαγής σειράς στην παρουσίαση κάποιων εννοιών-θεμάτων και συνολικά επήλθαν διορθώσεις και

βελτιώσεις του μαθησιακού υλικού, βάσει παρατηρήσεων και διαπιστώσεων κατά την πρώτη αυτή εφαρμογή.

Στη δεύτερη εφαρμογή, υλοποιήθηκε η τρίτη φάση της ροής δραστηριοτήτων (Φάση III, ενότητα 3.5.) οδηγώντας τους εκπαιδευόμενους σε τελικές εξετάσεις για το μάθημα “Αντικειμενοστρεφής Προγραμματισμός”. Από τη δεύτερη αυτή εφαρμογή, προέκυψε η σχετική δυσκολία της χρήσης διαγραμματικής τεχνικής χωρίς τη χρήση κάποιου ειδικού εργαλείου σχεδίασης, αλλά και η δυσκολία της παρουσίασης της φάσης της σχεδίασης μιας αντικειμενοστρεφούς εφαρμογής (class diagrams) χωρίς αυτό. Έτσι, ενσωματώθηκε στη διδακτική προσέγγιση η χρήση ενός σχεδιαστικού CASE εργαλείου όπως είναι το ArgoUML (ενότητα 3.8.).

Κατά την τρίτη εφαρμογή της προτεινόμενης διδακτικής προσέγγισης, η δεύτερη φάση της ροής δραστηριοτήτων, λόγω αυξημένου φόρτου των συμμετεχόντων, περιελάμβανε την εκπόνηση του ενιαίου project χωρίς τις μικρές επί μέρους εργασίες και επίσης η τρίτη φάση αφορούσε στην ανάπτυξη ενός τελικού project.

Η διδακτική προσέγγιση στην αρχική της μορφή και πριν ξεκινήσουν οι μεγάλης διάρκειας εφαρμογές της, εφαρμόστηκε δοκιμαστικά και ανιχνευτικά για τη συγκέντρωση μιας πρώτης γενικής ανατροφοδότησης, σε σύντομης διάρκειας σεμινάριο από 28/2/2005 έως 14/3/2005, στο ΠΜΣ του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά (Γεωργαντάκη και Ρετάλης, 2005). Οι συμμετέχοντες φοιτητές (28 τον αριθμό) προέρχονταν και από τις τρεις (3) κατευθύνσεις του ΠΜΣ (Ηλεκτρονική Μάθηση, Ψηφιακές Επικοινωνίες και Δίκτυα, Δικτυοκεντρικά Συστήματα). Η εφαρμογή περιελάμβανε 4 τρίωρα θεωρητικής παράδοσης και διεξήχθη σε εργαστηριακό χώρο, όπου οι φοιτητές είχαν στη διάθεση τους εγκαταστημένα τα περιβάλλοντα που χρησιμοποιήθηκαν και όλο το μαθησιακό υλικό. Μετά την παρουσίαση εννοιών-θεμάτων με τη χρήση παραδειγμάτων, δινόταν η ευκαιρία στους φοιτητές να τα δοκιμάσουν, καθώς τα αρχεία κώδικα Java, τα Bluej projects και τα αρχεία Jeroo, συμπεριλαμβάνονταν στο υλικό.

Αυτή η τόσο σύντομη εφαρμογή δεν υπολογίζεται ως δοκιμή της προτεινόμενης διδακτικής προσέγγισης αυτής καθεαυτής, αλλά ως δοκιμή ρύθμισης του μαθησιακού υλικού και των μαθησιακών εργαλείων, αφού η προσέγγιση που προτείνεται, αποσκοπεί στην επίτευξη των στόχων που αναφέρονται στην ενότητα 3.4. μέσω μιας

μαθησιακής διαδικασίας η οποία περιλαμβάνει μια ακολουθία μαθησιακών δραστηριοτήτων (learning sequence) που συμβαίνουν σε κάθε φάση της (ενότητα 3.5.) και η ολοκλήρωση των φάσεων επιτυγχάνεται σε διάστημα μερικών μηνών (2-3), είτε με αυξημένη φυσική παρουσία του διδάσκοντα (ενότητα 4.4.), είτε δίνοντας έμφαση στην αξιοποίηση των διαδικτυακών τεχνολογιών (ενότητες 4.5., 4.6.).

Από τα σχόλια και την κριτική των φοιτητών σε αυτό το σύντομο σεμινάριο, προέκυψε θετική αξιολόγηση του τρόπου διδασκαλίας, του μαθησιακού υλικού και των μαθησιακών εργαλείων.

Κατά τη διάρκεια του τελευταίου μαθήματος ένας φοιτητής ρώτησε: “είχατε σχεδιάσει το υλικό με κάποιο εκπαιδευτικό-παιδαγωγικό τρόπο; Γιατί εγώ βλέπω ότι από την αρχή μέχρι το τέλος, είχε μια συνοχή που εμένα με βοήθησε. Όταν διαβάζεις ένα βιβλίο δε συμβαίνει κάτι τέτοιο”. Ένας υποψήφιος Διδάκτορας που παρακολουθούσε γιατί στο πλαίσιο του διδακτορικού του χρειαζόταν να χρησιμοποιήσει Java, αρκετά αργότερα από το τέλος του σεμιναρίου είπε ότι δε θα μπορούσε να είχε κάνει αυτό που του χρειαζόταν αν δεν είχε παρακολουθήσει και αν δεν είχε χρησιμοποιήσει το υλικό του σεμιναρίου. Επίσης αρκετοί φοιτητές και αφού παρήλθε διάστημα από τη λήξη του σεμιναρίου, εκτίμησαν ότι βοηθήθηκαν ιδιαίτερα παρακολουθώντας το. Αυτό συνέβη και για φοιτητές οι οποίοι δε διέθεταν καμία απολύτως προηγούμενη γνώση Προγραμματισμού.

Η γνώση και η εμπειρία που αποκτήθηκε από τη διεξαγωγή του σεμιναρίου αυτού, χρησιμοποιήθηκε για να γίνουν συμπληρώσεις, διορθώσεις και βελτιώσεις του μαθησιακού υλικού.

4.3. Εννοιολογικό πλαίσιο αξιολόγησης

Το πλαίσιο αξιολόγησης των μελετών περίπτωσης στηρίζεται στη μέθοδο αξιολόγησης CADMOS-E (Retalis et al., 1998; Psaromiligkos and Retalis, 2003), η οποία αποτελεί μια σε βήματα μικτή μέθοδο που υποστηρίζεται από ειδικά σχεδιασμένα ερωτηματολόγια προ-αξιολόγησης (pre-test) και μετά-αξιολόγησης (post-test). Τα ερωτηματολόγια παρέχουν δεδομένα για ποσοτική και ποιοτική ανάλυση.

Η αξιολόγηση, εστιάζει στη μαθησιακή αποτελεσματικότητα της προτεινόμενης διδακτικής προσέγγισης. Η αποτελεσματικότητα αυτή σχετίζεται και προκύπτει από

πολλαπλούς μετρήσιμους δείκτες, που συνίστανται τόσο σε γνωστικά αποτελέσματα όσο και σε αποτελέσματα τάσεων και διαθέσεων (Makrakis et al., 1998) (π.χ. αντίληψη των βασικών αντικειμενοστρεφών εννοιών, ανάπτυξη δεξιοτήτων προγραμματισμού με αντικειμενοστρεφή λογική, αυτοεκτίμηση των συμμετεχόντων για το επίπεδο της γνώσης τους στον ΑΠ, προτιμήσεις και στάσεις τους σε σχέση με τον ΑΠ).

Οι μελέτες αξιολόγησης πραγματοποιούνται, αναλύοντας κυρίως, τις απαντήσεις των συμμετεχόντων στις εφαρμογές της διδακτικής προσέγγισης στα δύο ερωτηματολόγια που προαναφέρθηκαν και τις εργασίες που εκπονούν και παραδίδουν. Επιπρόσθετα στοιχεία αξιολόγησης προκύπτουν, μέσω άμεσης παρατήρησης και σημειώσεων κατά τη διάρκεια των εφαρμογών, από στοιχεία συζητήσεων μέσα στην τάξη και συνεντεύξεων σε τυχαίο δείγμα συμμετεχόντων σε κάποιες περιπτώσεις.

Τις πρώτες μέρες διεξαγωγής της εφαρμογής –και αφού δίνονται οι απαραίτητες εξηγήσεις στους συμμετέχοντες– συμπληρώνεται το ερωτηματολόγιο προ-αξιολόγησης (*pre-test*). Μετά το τέλος της εφαρμογής συμπληρώνεται το ερωτηματολόγιο μετά-αξιολόγησης (*post-test*). Στο “pre-test” περιέχονται ερωτήσεις σχετικές με δημογραφικά δεδομένα των συμμετεχόντων, το υπόβαθρό τους στο γνωστικό αντικείμενο του ΑΠ, την αυτοεκτίμησή τους για το επίπεδό τους στον ΑΠ, τις προτιμήσεις εκμάθησης, τη στάση τους απέναντι στον ΑΠ και επιπλέον, μερικές ερωτήσεις που αναφέρονται στις προσδοκίες των συμμετεχόντων από την εφαρμογή της διδακτικής προσέγγισης, την εκπαιδευτική της φιλοσοφία, τον τρόπο διεξαγωγής της.

Στο “post-test” ερωτηματολόγιο, αρκετές ερωτήσεις του αρχικού ερωτηματολογίου επαναλαμβάνονται, για να διαπιστωθεί η επίδραση της διδακτικής προσέγγισης που εφαρμόστηκε και επιπλέον, περιέχονται σε αυτό ερωτήσεις κλειστού τύπου για αξιολόγηση της ποιότητας του μαθησιακού υλικού και της μαθησιακής αποτελεσματικότητας των εκπαιδευτικών περιβαλλόντων και της διδακτικής προσέγγισης. Υπάρχει και ένα σύνολο ερωτήσεων ανοικτού τύπου, στις οποίες οι συμμετέχοντες διατυπώνουν τι τους άρεσε και τι όχι σε σχέση με το μαθησιακό υλικό, τις αδυναμίες του υλικού και της προσέγγισης, καθώς και προτάσεις για τη βελτίωσή τους.

Ποσοτικά στοιχεία προκύπτουν από τις ερωτήσεις κλειστού τύπου στις οποίες η απάντηση δίνεται με τη χρήση της πεντάβαθμης κλίμακας Likert και

πραγματοποιείται ποιοτική ανάλυση περιεχομένου του δεύτερου μέρους του ερωτηματολογίου μετά-αξιολόγησης (ερωτήσεις ανοικτού τύπου).

Στην αξιολόγηση της προτεινόμενης διδακτικής προσέγγισης, επιδιώξαμε:

- Την καταγραφή της γνώμης των συμμετεχόντων με τη χρήση των δύο ερωτηματολογίων τα οποία συμπλήρωσαν και υπέβαλλαν ηλεκτρονικά. Τα ερωτηματολόγια όλων των εφαρμογών παρατίθενται στα Παραρτήματα Β, Γ, Δ, Ε, ΣΤ και Ζ.

Πραγματοποιήθηκε συγκριτική στατιστική ανάλυση των δεδομένων που συγκεντρώθηκαν από τα “pre-test” και “post-test” ερωτηματολόγια.

Η ανάλυση αφορούσε στην εξαγωγή ποσοτικών στοιχείων για τα εξής:

- ◆ Αυτοεκτίμηση των συμμετεχόντων για το επίπεδό τους στο γνωστικό αντικείμενο του ΑΠ πριν και μετά την εφαρμογή της διδακτικής προσέγγισης.
- ◆ Επίδραση της προσέγγισης στη στάση των συμμετεχόντων απέναντι στον ΑΠ.
- ◆ Συμβολή στη μαθησιακή διαδικασία, παραγόντων σχετικών με το μαθησιακό υλικό και τη διεξαγωγή του σεμιναρίου (ύπαρξη διαλέξεων στο Moodle, δυνατότητα συζητήσεων, επίλυση ασκήσεων, εκπόνηση εργασιών, ενασχόληση με περιβάλλοντα, υποστήριξη από τους διδάσκοντες κ.λ.π.).
- ◆ Αξιολόγηση της ποιότητας και της μαθησιακής αποτελεσματικότητας του μαθησιακού υλικού συνολικά και μεμονωμένων μαθησιακών του πόρων (παραδείγματα, θεωρία, λυμένες ασκήσεις, μελέτη περίπτωσης, αρχεία κώδικα κ.λ.π.).
- ◆ Αξιολόγηση της αποτελεσματικότητας των μαθησιακών εργαλείων που χρησιμοποιούνται.
- ◆ Επίσης ποιοτικά στοιχεία εξήχθησαν από τις ερωτήσεις ανοικτού τύπου.
- Την αποτίμηση των μαθησιακών αποτελεσμάτων της προσέγγισης, από τις εργασίες τις οποίες παρέδωσαν οι συμμετέχοντες, από την επίδοσή τους στις τελικές εξετάσεις –όπου αυτό συνέβη–, μέσω άμεσης παρατήρησης και σημειώσεων κατά τη διάρκεια του σεμιναρίου και από συζητήσεις μέσα στην τάξη. Όλα τα παραπάνω δεδομένα, μελετήθηκαν για να εξαχθούν συμπεράσματα για την εμπέδωση εννοιών, την ικανότητα υλοποίησης προγραμμάτων με Αντικειμενοστρεφή φιλοσοφία, την αντιμετώπιση καταγεγραμμένων δυσκολιών-παρανοήσεων και την ανάδειξη νέων μη καταγεγραμμένων.

4.4. Μελέτη περίπτωσης 1 (case study-1)

Στην ενότητα αυτή παρουσιάζεται η πρώτη πιλοτική εφαρμογή της προτεινόμενης διδακτικής προσέγγισης.

Όπως αναφέρθηκε στην ενότητα 4.2. σε αυτή την εφαρμογή *ενδιέφερε κυρίως, να αξιολογηθεί η ποιότητα του μαθησιακού υλικού και των πόρων από τους οποίους αποτελείται (ενότητα 3.7.), η αποδοχή του τρόπου διεξαγωγής της προσέγγισης (on-line μαθησιακό περιβάλλον (ενότητα 3.6.)) και η αξιοποίηση των μαθησιακών εργαλείων (ενότητα 3.8.).*

4.4.1. Χώρος εφαρμογής

Η διδακτική προσέγγιση που σχεδιάστηκε, εφαρμόστηκε κατά το εαρινό εξάμηνο του 2005 σε μακράς διάρκειας σεμινάριο (10 εβδομάδες, από 30/3/2005 έως 8/6/2005), που διεξήχθη για τους προπτυχιακούς φοιτητές όλων των εξαμήνων του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά με τη μορφή φροντιστηριακού μαθήματος (Georgantaki and Retalis, 2007a; Γεωργαντάκη και Ρετάλης, 2006). Η συμμετοχή ήταν προαιρετική και περιελάμβανε διαλέξεις στο αμφιθέατρο χωρίς να συνοδεύεται από εργαστήριο.

4.4.2. Συμμετέχοντες

Δήλωσαν συμμετοχή στο μάθημα είκοσι (20) φοιτητές ηλικίας από 18 έως 23 ετών. Κατά τις πρώτες 12 διαλέξεις παρακολούθησαν όλοι και κατά τις υπόλοιπες 5, περίπου 10 άτομα. Η μειούμενη παρουσία οφείλεται κατά κύριο λόγο στις υποχρεώσεις των φοιτητών και από ένα χρονικό σημείο και μετά, στην προετοιμασία που έκαναν ενόψει της εξεταστικής περιόδου. Κάποιοι φοιτητές που δεν παρακολούθησαν όλες τις διαλέξεις αξιοποίησαν τις άλλες δυνατότητες που τους παρέχονταν (on-line μαθησιακό περιβάλλον, ασύγχρονες συζητήσεις).

Όλοι οι φοιτητές ενδιαφέρονταν για το αντικείμενο του ΑΠ, αφού η γνώση του αποτελεί μεγάλη βοήθεια για μια σειρά σχετικών μαθημάτων του Προγράμματος

Σπουδών του Τμήματος και είχαν τουλάχιστον διδαχθεί προηγουμένως το διαδικασιακό παράδειγμα Προγραμματισμού με τη χρήση της γλώσσας C. Το κίνητρο των φοιτητών για τη συμμετοχή τους στο φροντιστηριακό μάθημα, ήταν να βελτιώσουν τις γνώσεις του και τις δεξιότητές τους στον ΑΠ. Μόνον τρεις φοιτητές, από αυτούς που το παρακολούθησαν το έκαναν και για να προετοιμαστούν καλύτερα για τις εξετάσεις του αντίστοιχου υποχρεωτικού μαθήματος “Αντικειμενοστρεφής Προγραμματισμός” του Προγράμματος Σπουδών. Συγκεκριμένα οι λόγοι που έκαναν αρχικά τους φοιτητές να παρακολουθήσουν το μάθημα φαίνονται στον Πίνακα 4.1, όπως προέκυψε από αντίστοιχη ερώτηση του ερωτηματολογίου μετά-αξιολόγησης.

Πίνακας 4.1. Κίνητρα παρακολούθησης (case study-1)

Ερώτηση	
Για ποιο/ούς λόγους αρχικά είχατε αποφασίσει να παρακολουθήσετε το σεμινάριο;	Αριθμός φοιτητών που επέλεξαν την απάντηση (max 13)
Επειδή με ενδιαφέρει το αντικείμενο.	11
Δεν είχα κατανοήσει το αντικείμενο παρακολουθώντας το/α μάθημα/τα στο Τμήμα.	8
Με δυσκολεύει το αντικείμενο.	7
Δεν έχω περάσει το/α σχετικό/α μάθημα/τα.	6
Ήθελα κάτι περισσότερο από ότι έχω διδαχθεί.	5
Για άλλο λόγο.	3

Οι άλλοι λόγοι που οι τρεις φοιτητές επικαλέστηκαν είναι:

- “Δεν είχα παρακολουθήσει τα σχετικά μαθήματα κατά τη διάρκεια των σπουδών και θεώρησα μειονέκτημα να μην έχω έστω και τις βασικές γνώσεις του αντικείμενου αυτού τελειώνοντας αυτή τη σχολή. Επίσης θεωρώ ότι η γλώσσα αυτή ζητείται και από την αγορά εργασίας σε μεγάλο βαθμό.”
- “Ήθελα να δημιουργήσω καλύτερες βάσεις με το συγκεκριμένο αντικείμενο καθώς και να ταξινομήσω καλύτερα τα πράγματα που είχα μάθει στο μυαλό μου.”
- “Για το λόγο ότι η Java είναι μια πολύ ισχυρή γλώσσα σήμερα και ότι ήταν πολύ καλή ευκαιρία το σεμινάριο.”

4.4.3. Υλοποίηση της εφαρμογής

Κατά τη διάρκεια του μαθήματος πραγματοποιήθηκαν δέκα επτά (17) δίωρες διαλέξεις δύο φορές την εβδομάδα, στις οποίες αναπτύχθηκαν τα θέματα των εννοιών που περιγράφηκαν στην ενότητα 3.3., δόθηκε έμφαση στην παρουσίαση και εξήγηση παραδειγμάτων και ασκήσεων και όχι στη θεωρητική και μόνο εισαγωγή των Αντικειμενοστρεφών εννοιών-αρχών και παρουσιάζονταν διαρκώς τα προγραμματιστικά περιβάλλοντα (BlueJ, Jeroo, SUN One Studio) και τα παραδείγματα κώδικα, με τη χρήση φορητού υπολογιστή και βιντεοπροβολέα, ακολουθώντας μια “live coding” προσέγγιση στις διαλέξεις (Hyland and Clynych, 2002). Μέσω των εργασιών που οι φοιτητές παρέδιδαν σε καθορισμένο χρόνο, αξιολογήθηκαν τα μαθησιακά αποτελέσματα της προσέγγισης.

Το on-line μαθησιακό περιβάλλον (ενότητα 3.6.) αυτής της μελέτης περίπτωσης, υλοποιήθηκε με τη δημιουργία δικτυακού τόπου του σεμιναρίου στο περιβάλλον του LMS Moodle (<http://www.moodle.org>).

4.4.4. Αξιολόγηση της εφαρμογής

Οι φοιτητές συμπλήρωσαν τα δύο ερωτηματολόγια (pre-test και post-test) –που παρατίθενται στα Παραρτήματα Β και Γ αντίστοιχα– στην αρχή και μετά το τέλος του σεμιναρίου.

Συνολικά οι φοιτητές που συμμετείχαν στην αξιολόγηση αυτής της εφαρμογής της διδακτικής προσέγγισης, συμπλήρωσαν τα δύο ερωτηματολόγια αξιολόγησης και παρέδωσαν τις εργασίες ήταν δεκατρείς (13), 8 γυναίκες και 5 άνδρες που προέρχονταν από όλα τα εξάμηνα. Η πλειοψηφία τους (8 στους 13 = 61.5%), χρησιμοποίησε μόνον το υλικό που διατέθηκε στο μάθημα και αυτό προσθέτει αξία στην αξιολόγηση τους και τα σχόλια τους, καθώς η διαδικασία δεν επηρεάστηκε από άλλους μαθησιακούς πόρους.

4.4.4.1. Επεξεργασία ερωτηματολογίων

Ποσοτικά Αποτελέσματα

Από την επεξεργασία των απαντήσεων στα δύο ερωτηματολόγια προέκυψαν τα ποσοτικά στοιχεία που παρουσιάζονται στους παρακάτω πίνακες.

Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στον ΑΠ

Πίνακας 4.2. Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στο γνωστικό αντικείμενο του ΑΠ (case study-1)

Ερώτηση	Μ.Ο. (Από το “pre-test”)	Μ.Ο. (Από το “post-test”)
Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον ΑΠ;	1.9*	
Πώς εκτιμάτε μετά τη διεξαγωγή του σεμιναρίου, την ικανότητά σας να γράψετε προγράμματα σε Java;		3.4**

* Κωδικοποίηση απαντήσεων:

5 = Επαγγελματίας... 1 = Αρχάριος

**Κωδικοποίηση απαντήσεων:

5 = Πολύ ικανός/ή... 1 = Καθόλου ικανός/ή

Η αυτοεκτίμηση των φοιτητών έχει σημαντικά αυξηθεί μετά την παρακολούθηση του σεμιναρίου και τη χρήση του μαθησιακού υλικού. Ενώ πριν οι φοιτητές κατέτασσαν κατά μέσο όρο τους εαυτούς τους μεταξύ του επιπέδου “Αρχάριος/α” και “Λίγο έμπειρος/η”, μετά την ολοκλήρωση του σεμιναρίου η κατάταξη που έκαναν κατά μέσο όρο, όσον αφορά την ικανότητά τους να γράψουν προγράμματα σε Java, ήταν μεταξύ του “Μέτρια ικανός/η” και του “Αρκετά ικανός/η”.

Στάση των φοιτητών απέναντι στο αντικείμενο του ΑΠ

Πίνακας 4.3. Προτιμήσεις και στάσεις των φοιτητών σχετικά με το αντικείμενο του ΑΠ (case study-1)

Ερώτηση	Μ.Ο. (Από το “pre-test”)	Μ.Ο. (Από το “post-test”)
Σας αρέσει το αντικείμενο του ΑΠ;	4.1	4.1
Δυσκολεύεστε με τον ΑΠ;	3.5	3.3
Αντιμετωπίζετε γενικά με φόβο τον ΑΠ;	2.5	2.5
Θεωρείτε ότι ο ΑΠ απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;	3.5	3.8
Θεωρείτε ότι μάθατε ΑΠ με αυτά που έχετε διδαχθεί έως τώρα; *	2.1	3.8

Κωδικοποίηση απαντήσεων:

5 = Πολύ... 1 = Καθόλου

*Κωδικοποίηση απαντήσεων:

4 = Έμαθα καλά... 1 = Δεν έμαθα καθόλου

Από τον Πίνακα 4.3. μπορούν να εξαχθούν τα εξής συμπεράσματα:

- Οι δυσκολίες των φοιτητών μειώθηκαν μετά το σεμινάριο.
- Οι φόβοι τους για τον ΑΠ παρέμειναν οι ίδιοι, χωρίς να είναι ιδιαίτερα υψηλοί.

Αυτό το στοιχείο θα συζητηθεί στη συνέχεια σε συνδυασμό και με επόμενα ευρήματα.

- Έκριναν ότι η γνώση που έχουν αποκτήσει σχετικά με τον ΑΠ είναι πολύ καλύτερη μετά το σεμινάριο (από 2.1 αυξήθηκε σε 3.8 κατά μέσο όρο).
- Μετά το τέλος του σεμιναρίου θεωρούν “ότι ο ΑΠ απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια” σε ελαφρά υψηλότερο βαθμό από ότι πριν. Αυτό εξηγείται από το γεγονός ότι αρκετοί φοιτητές απέκτησαν ουσιαστική επαφή με τον ΑΠ κατά τη διάρκεια του σεμιναρίου (παρακολουθώντας διαλέξεις, πραγματοποιώντας εκπαιδευτικές δραστηριότητες, εκπονώντας εργασίες) και σχημάτισαν σαφέστερη εικόνα για αυτό το αδιαμφισβητήτως απαιτητικό αντικείμενο.

Και στα δύο ερωτηματολόγια περιέχονταν οι εξής γενικές ερωτήσεις ανοικτού τύπου:

- “Τι είναι αυτό που σας άρεσε περισσότερο στον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;”

Στο “*pre-test*” κυριάρχησαν οι παρακάτω απαντήσεις: “με τον Προγραμματισμό έχεις την αίσθηση της δημιουργίας”, “βρίσκεις λύσεις”, “δημιουργείς εφαρμογές για να λύσεις προβλήματα και να αυτοματοποιήσεις λειτουργίες”, “οξύνεις το πνεύμα σου”, “αποκτάς βαθύτερη γνώση της λειτουργίας των Υπολογιστών” και ότι “ο Προγραμματισμός απαιτεί κριτική σκέψη και όχι απομνημόνευση”.

Στο “*post-test*” ένα μεγάλο ποσοστό απαντήσεων επαναλάμβανε τις ίδιες απόψεις και επιπρόσθετα οι φοιτητές σημείωσαν τα εξής: “ο Προγραμματισμός κάνοντας χρήση του περιβάλλοντος BlueJ”, “η εύρεση καλών λύσεων σε περίπλοκα προβλήματα” και επίσης ότι “ο Προγραμματισμός είναι ένα ευκολότερο αντικείμενο από ότι πριν θεωρούσαν”.

- “Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;”

Οι απαντήσεις στο “*pre-test*” περιελάμβαναν: “η μη αποδοτική μέθοδος διδασκαλίας”, “η έλλειψη πρακτικής εξάσκησης”, “το γεγονός ότι ο Προγραμματισμός απαιτεί μεγάλη προσπάθεια”, “η δυσκολία του να κωδικοποιήσεις

μα λύση σε μια γλώσσα προγραμματισμού”, “η αποτυχία να κατανοηθεί η θεωρία και να εφαρμοστεί στην πράξη”.

Στο “*post-test*” ανέφεραν: “Είναι δύσκολο να κατανοήσεις τη φιλοσοφία του Προγραμματισμού”, “είναι δύσκολο να σκέφτεσαι προγραμματιστικά”, “είναι πολύπλοκος ο προγραμματιστικός τρόπος σκέψης”, “είναι πολύπλοκες οι γλώσσες Προγραμματισμού” και ότι “είναι δύσκολη η αποσφαλμάτωση των προγραμμάτων”.

Όπως μπορεί να συναχθεί, μετά το σεμινάριο, οι απαντήσεις των φοιτητών έγιναν πιο επακριβείς για τις δυσκολίες που αντιμετωπίζουν. Αυτό δείχνει ότι το σεμινάριο τους βοήθησε να αντιληφθούν καλύτερα τις διάφορες δύσκολες “πλευρές” του Προγραμματισμού και ότι απέκτησαν πληρέστερη εικόνα για το γνωστικό αυτό αντικείμενο.

Συμβολή παραγόντων του τρόπου εφαρμογής της διδακτικής προσέγγισης στη μαθησιακή διαδικασία

Πίνακας 4.4. Συμβολή παραγόντων της διδακτικής προσέγγισης στη βελτίωση γνώσεων και δεξιοτήτων (case study-1)

Ερώτηση	M.O. (Από το “pre- test”)	M.O. (Από το “post- test”)
Αξιολογείτε τη σημασία που θεωρείτε ότι (θα έχουν) είχαν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας.		
Ύπαρξη των διαλέξεων στο WWW.	4.6	4.8
Ύπαρξη του σχετικού με το μάθημα υλικού στο WWW.	4.7	4.7
Ύπαρξη μελετών περίπτωσης (case studies) (δηλαδή ολοκληρωμένων παραδειγμάτων) στο μαθησιακό υλικό.	4.2	4.6
Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του περιβάλλοντος Moodle.	3.8	3.8
Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του περιβάλλοντος Moodle.	4.4	4.5
Το υλικό που προήλθε από συζητήσεις μεταξύ συμφοιτητών μου μέσω του περιβάλλοντος Moodle (απλή ανάγνωση μηνυμάτων κι όχι ίδια συμμετοχή στη συζήτηση).	*	3.8
Το υλικό που προήλθε από συζητήσεις των συμφοιτητών μου με τους διδάσκοντες μέσω του περιβάλλοντος Moodle (απλή ανάγνωση μηνυμάτων κι όχι ίδια συμμετοχή στη συζήτηση).	*	4.3
Οι διορθώσεις/παρατηρήσεις από τους διδάσκοντες στις	*	4.9

εργασίες/ασκήσεις που παρέδιδα.		
Η εκπόνηση εργασιών/ασκήσεων στο πλαίσιο του μαθήματος.	4.7	4.9
Η ενασχόληση με τα περιβάλλοντα που ήταν διαθέσιμα στο μάθημα.	4.5	4.6

Κωδικοποίηση απαντήσεων: 5 = Πολύ σημαντικός... 1 = Πολύ ασήμαντος

* ερώτηση μη διαθέσιμη στο “pre-test”

Από τον Πίνακα 4.4. συνάγονται τα εξής:

- Τα σχόλια και οι διορθώσεις των διδασκόντων στις εργασίες που οι φοιτητές παρέδιδαν, αποτέλεσε πολύ σημαντικό παράγοντα της διδακτικής προσέγγισης στη βελτίωση των γνώσεων και των δεξιοτήτων των φοιτητών. Αυτό καταδεικνύει πόσο ευεργετική είναι για τους φοιτητές η επικοινωνία με τους διδάσκοντες, η αλληλεπίδραση μαζί τους και η ανατροφοδότηση που παίρνουν για τις απορίες τους. Αυτό το είδος της εξατομικευμένης βοήθειας εστιάζει στο επίπεδο γνώσεων του κάθε φοιτητή, κάτι που αποβαίνει ύψιστης σημασίας γεγονός για τη βελτίωση των γνώσεων και των δεξιοτήτων τους. Επιβεβαιώνονται με αυτά τα ευρήματα, η υιοθέτηση της αρχής (ενότητα 3.4.) περί **“παροχής εξατομικευμένης βοήθειας στους εκπαιδευόμενους”** (Fleury, 2000; Bergin, 2006a, pattern “Differentiated Feedback”) και της **παιδαγωγικής αρχής της εκμάθησης του γνωστικού αντικειμένου του ΑΠ κοντά σε ειδικούς (experts) (cognitive apprenticeship)** (Collins et al., 1989; Brown et al., 1989)
- Η εκπόνηση εργασιών, συνέβαλε σε πολύ μεγάλο βαθμό στην αύξηση της γνώσης των φοιτητών, δικαιώνοντας την αρχή (ενότητα 3.4.) **“Υλοποίηση της διαδικασίας ενεργούς δόμησης της γνώσης με τη συνεχή εμπλοκή του εκπαιδευόμενου σε κατάλληλα σχεδιασμένες δραστηριότητες ποικίλου είδους”** (Hadjerrouit, 1999; Fleury, 2000; Fleury, 2001; Bergin, 2006a, pattern “Active Student”).
- Ήταν πολύ σημαντικός παράγων για τους φοιτητές, να έχουν πρόσβαση στις διαφάνειες των διαλέξεων και στο on-line υλικό αναφοράς μέσω του Moodle LMS.
- Η μελέτη περίπτωσης (case study) συνέβαλε κατά πολύ, αν και αρχικά αξιολόγησαν αυτόν τον παράγοντα χαμηλά σε σχέση με άλλους. Αυτό εξηγείται πιθανόν, από το γεγονός ότι οι φοιτητές δεν ήταν εξοικειωμένοι αρκετά με κάτι τέτοιο. Στο “post-test” όμως εκτίμησαν ιδιαίτερα αυτό τον παράγοντα επιβεβαιώνοντας έτσι την υιοθέτηση της σύστασης (ενότητα 3.4.) για **χρήση case**

studies (Linn and Clancy, 1992; Fleury, 2001).

- Η ενασχόληση με τα εκπαιδευτικά εργαλεία εκτιμήθηκε το ίδιο υψηλά δικαιώνοντας επίσης την αρχή (ενότητα 3.4.), “**Υλοποίηση της διαδικασίας ενεργούς δόμησης της γνώσης με τη συνεχή εμπλοκή του εκπαιδευόμενου σε κατάλληλα σχεδιασμένες δραστηριότητες ποικίλου είδους**”.
- Η χαμηλότερη αξιολόγηση (3.8) (σε σχέση με τους άλλους παράγοντες), έγινε για τη “δυνατότητα συζητήσεων με συμφοιτητές μέσω του περιβάλλοντος Moodle” και παρέμεινε η ίδια και στο “post-test”. Αυτό μπορεί να εξηγηθεί από το γεγονός ότι οι φοιτητές δεν ήταν εξοικειωμένοι με τέτοια μέσα και τρόπους. Ίσως το γεγονός ότι προέρχονταν από διαφορετικά εξάμηνα δεν ενίσχυσε τη δημιουργία μιας “κοινότητας μάθησης”. Το ίδιο συνέβη και με τον παράγοντα “Το υλικό που προήλθε από συζητήσεις μεταξύ συμφοιτητών μου μέσω του περιβάλλοντος Moodle”.
- Αξιοπρόσεκτο είναι ότι οι συζητήσεις με τους διδάσκοντες (είτε άμεσα είτε έμμεσα μέσω ανάγνωσης συζητήσεων άλλων φοιτητών με τους διδάσκοντες), αξιολογήθηκε λίγο χαμηλότερα από άλλους παράγοντες (4.5 και 4.3), αλλά πολύ υψηλότερα από τον παράγοντα “συζητήσεις με συμφοιτητές” (3.8). Αυτό δείχνει τη σημαντικότητα και την ανάγκη επικοινωνίας των φοιτητών με τους διδάσκοντες.

Συνολικά από τον Πίνακα 4.4. συνάγεται, ότι οι αρχικές προσδοκίες των φοιτητών ικανοποιήθηκαν σε μεγάλο βαθμό.

Συμβολή μαθησιακού υλικού και διδακτικής προσέγγισης

Πίνακας 4.5. Αξιολόγηση των μαθησιακών πόρων του υλικού μεμονωμένα (case study-1)

Ερώτηση	M.O. (Από το “post-test”)
Αξιολογήστε το βαθμό με τον οποίο οι παρακάτω μαθησιακοί πόροι συνέβαλαν στη βελτίωση των γνώσεων και των δεξιοτήτων σας.	
Λυμένες ασκήσεις, παραδείγματα.	4.8
Θεωρία.	4.5
Ερωτήσεις, ασκήσεις.	4.4
Τα αρχεία κώδικα Java που περιλαμβάνονται στο υλικό.	4.5
Τα υλοποιημένα projects στο περιβάλλον BlueJ.	4.2
Οι λυμένες ασκήσεις στο περιβάλλον Jeroo.	3.4
Η μελέτη περίπτωσης (case study).	4.7

Κωδικοποίηση απαντήσεων: 5 = Πολύ σημαντική... 1 = Πολύ ασήμαντη

Από τον παραπάνω Πίνακα μπορούν να εξαχθούν τα ακόλουθα συμπεράσματα:

- Οι φοιτητές επιδοκίμασαν ιδιαίτερα τις λυμένες ασκήσεις και τα παραδείγματα του υλικού. Αυτό αποδεικνύει την ποιότητα και την καταλληλότητά τους και δικαιώνει την υιοθέτηση της αρχής (ενότητα 3.4.) **“Κατάλληλη σχεδίαση και χρήση αντιπροσωπευτικών παραδειγμάτων για την παρουσίαση νέων εννοιών”** (Fricke and Voelter, 2000, pattern “Relevant Examples”).
- Ο επόμενος σημαντικότερος μαθησιακός πόρος είναι η μελέτη περίπτωσης (case study). Αυτό επιβεβαιώνει ότι ολοκληρωμένα παραδείγματα-εφαρμογές στα οποία εφαρμόζονται στην πράξη οι αρχές ενός αντικειμένου, έχουν μεγάλη σημασία για τους φοιτητές.
- Επόμενος σε σημασία πόρος, είναι η θεωρία και ο πηγαίος κώδικας των παραδειγμάτων και των ασκήσεων που περιέχεται στο υλικό, επιβεβαιώνοντας την υιοθέτηση της οδηγίας (ενότητα 3.4.) **“μελέτησε κώδικα” (Read code)** (Kölling and Rosenberg, 2001b; Ala-Mutka; Bergin, 2006a, pattern “Read before Write”). Η θεωρία αποτελείται κυρίως από σύντομα και περιεκτικά τμήματα (υποενότητα 3.7.1.) αλλά –όπως προκύπτει– χωρίς να μειώνεται η ποιότητα και η αποτελεσματικότητά της.
- Σημαντικοί παράγοντες είναι οι ερωτήσεις-ασκήσεις (εργασίες) που οι φοιτητές εκπόνησαν και παρέδωσαν.
- Τα υλοποιημένα projects στο περιβάλλον BlueJ, ακολουθούν σε συμβολή στη βελτίωση των γνώσεων και των δεξιοτήτων.
- Ο τελευταίος σε συμβολή πόρος είναι οι λυμένες ασκήσεις στο περιβάλλον Jeroo. Αυτό το στοιχείο θα συζητηθεί στη συνέχεια σε συνδυασμό και με άλλα σχετικά.

Πίνακας 4.6. Αξιολόγηση του μαθησιακού υλικού στο σύνολό του και της διδακτικής προσέγγισης (case study-1)

Ερώτηση	M.O. (Από το “post- test”)
Το σεμινάριο με βοήθησε να ξεπεράσω δυσκολίες που αντιμετώπιζα σε σχέση με τον ΑΠ.	4.2
Το σεμινάριο με βοήθησε να ξεπεράσω αδυναμίες μου σε σχέση με τον ΑΠ.	4.1
Το σεμινάριο με βοήθησε να ξεπεράσω φοβίες που είχα σε σχέση με τον ΑΠ.	4.0

Πώς θα χαρακτηρίζατε το περιεχόμενο του μαθησιακού υλικού; *	4.5
Το περιεχόμενο του μαθησιακού υλικού επιτυγχάνει τους στόχους που αναφέρονται στην αρχή κάθε διδακτικής ενότητας.	4.7
Οι ερωτήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους στόχους εκμάθησης.	4.5
Οι ασκήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους στόχους εκμάθησης.	4.4
Ικανοποιήθηκαν οι λόγοι για τους οποίους παρακολουθήσατε; **	4.7

Κωδικοποίηση απαντήσεων: 5 = Συμφωνώ απόλυτα... 1 = Διαφωνώ απόλυτα

*Κωδικοποίηση απαντήσεων: 5 = Πολύ ενδιαφέρον, 4 = Ενδιαφέρον, 3 = Αδιάφορο, 2 = Κουραστικό, 1 = Δυσνόητο

**Κωδικοποίηση απαντήσεων: 5 = Πολύ... 1 = Καθόλου

Από τον Πίνακα 4.6. συμπεραίνονται τα εξής:

- Η διδακτική προσέγγιση που υιοθετήθηκε κατά τη διάρκεια του σεμιναρίου και το προσφερόμενο μαθησιακό υλικό, παρείχε μεγάλη βοήθεια στους φοιτητές για να ξεπεράσουν δυσκολίες, αδυναμίες και φόβους τους σχετικά με τον ΑΠ. Παρότι στον Πίνακα 4.3. φαίνεται ότι οι φόβοι των φοιτητών σχετικά με το αντικείμενο του ΑΠ παρέμειναν οι ίδιοι μετά το τέλος του σεμιναρίου (2.5), εδώ φαίνεται ότι εκτιμούν ιδιαίτερα υψηλά τη συμβολή του σεμιναρίου στο να ξεπεράσουν φοβίες σε σχέση με τον ΑΠ (4.0).
- Η ποιότητα του μαθησιακού υλικού αξιολογήθηκε πολύ υψηλά (4.5 κατά Μ.Ο. στην κλίμακα από 1 έως 5). Η καταλληλότητα του μαθησιακού υλικού και των ερωτήσεων-ασκήσεων για την επίτευξη των μαθησιακών στόχων, κρίθηκαν επίσης πολύ θετικά. Αυτό το στοιχείο αποδεικνύει την πολύ επαρκή ικανοποίηση της αρχής που υιοθετήθηκε στην προτεινόμενη διδακτική προσέγγιση (ενότητα 3.4.), “**ιδιαίτερη προσοχή στην ποιότητα του μαθησιακού υλικού που σχεδιάζεται**” (Kölling and Rosenberg, 2001b; Buck and Stucki, 2000; Fleury, 2000; Bergin, 2006a, pattern “Quality is Job One”).
- Το επίπεδο ικανοποίησης από το σεμινάριο ήταν πολύ υψηλό.

Μαθησιακά εργαλεία

Ζητήθηκε από τους φοιτητές να αναφέρουν τα περισσότερο σημαντικά γι’ αυτούς χαρακτηριστικά του περιβάλλοντος BlueJ. Οι συγκεντρωμένες απαντήσεις τους φαίνονται στον παρακάτω Πίνακα.

Πίνακας 4.7. Αξιολόγηση της ευχρηστίας του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ (case study-1)

Ερώτηση	
Ποια από τα παρακάτω χαρακτηριστικά του BlueJ σας βοήθησαν στη μαθησιακή διαδικασία; Επιλέξτε αυτά που νομίζετε.	Αριθμός φοιτητών που επέλεξαν το χαρακτηριστικό (max 13)
Ευκολία χρήσης του περιβάλλοντος.	11
Οι κλάσεις και τα αντικείμενα αποκτούν οπτική υπόσταση.	10
Επιτρέπεται να αλληλεπιδράς απευθείας με τις κλάσεις και τα αντικείμενα.	8
Εύκολη δημιουργία αντικειμένων και οπτικοποίησης τους.	9
Εύκολη εκτέλεση των μεθόδων και έλεγχός τους (εύκολο πέρασμα παραμέτρων και επιστροφή αποτελέσματος), χωρίς να χρειάζεται να γράφεις κώδικα γι' αυτό το σκοπό.	9
Εύκολος έλεγχος των τιμών των ιδιοχαρακτηριστικών των αντικειμένων.	7
Εύκολο compilation με επισήμανση των συντακτικών λαθών απευθείας στο παράθυρο του editor με μήνυμα και επισήμανση της γραμμής κώδικα με το λάθος.	9
Διαθέτει όψη “implementation” και όψη “interface” για τις κλάσεις.	6
Διαθέτει debugger εύκολο στη χρήση.	4
Εύκολος πειραματισμός με άμεσο αποτέλεσμα (feedback).	5

Οι φοιτητές εκτίμησαν ιδιαίτερα την ευκολία χρήσης του περιβάλλοντος. Σε συζητήσεις στην τάξη, αρκετοί σημείωναν ότι το περιβάλλον είναι τόσο φιλικό και εύχρηστο που τους παρακινούσε να το χρησιμοποιούν για την ανάπτυξη των προγραμμάτων τους. Όπως αναμενόταν, τα “δυνατά” χαρακτηριστικά του BlueJ, ήταν η δυνατότητα αλληλεπίδρασης, η οπτικοποίηση και η απλότητά του (Kölling et al., 2003; Kölling and Rosenberg, 2001b).

- Στην ερώτηση “Σε τι κυρίως πιστεύετε ότι σας βοήθησε το περιβάλλον BlueJ; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ότι άλλο εσείς νομίζετε)”

Οι φοιτητές απάντησαν:

- ◆ Κλήση μεθόδων. Σχέσεις μεταξύ κλάσεων.

- ◆ Το BlueJ τελικά βοήθησε σημαντικά στην αποσαφήνιση των εννοιών όπως οι κλάσεις, οι μέθοδοι αυτών αλλά και ενεργειών όπως η δημιουργία αντικειμένων κτλ μέσω κυρίως της οπτικής αναπαράστασης που παρέχει.
- ◆ Σίγουρα στην έννοια των αντικειμένων.
- ◆ Η δυνατότητα να κάνεις compile κάθε συνάρτηση μόνη της για να βρίσκεις τα λάθη στον κώδικα. Η ευκολία χρήσης του περιβάλλοντος. Η δυνατότητα να βλέπεις σε σχήματα τις κλάσεις που δημιουργείς, τα βέλη που σχηματίζονται και να καταλαβαίνεις τις έννοιες όπως κλήση μεθόδου, δημιουργία αντικειμένου, κληρονομικότητα.
- ◆ Με βοήθησε γιατί μπορώ να δω τι γίνεται όταν γράφω τον κώδικα και πως συνδέονται οι κλάσεις με το πρόγραμμα.
- ◆ Στο συγκεκριμένο περιβάλλον είναι εφικτή η γραφική αναπαράσταση των κλάσεων, της κληρονομικότητας και γενικά των προγραμμάτων που φτιάχνουμε. Θεωρώ πως αυτή η δυνατότητα του BlueJ με βοήθησε γενικά στην κατανόηση του Αντικειμενοστρεφή Προγραμματισμού.
- ◆ Ήταν πολύ απλό στη χρήση του και εξοικειώθηκα γρήγορα για τη συγγραφή προγραμμάτων.
- ◆ Τι είναι αντικείμενο, τι είναι κλάση, τι είναι αντικειμενοστρέφεια. Στις αναφορές και όλα όσα μάθαμε, αυτά λόγω της σχηματικής παρουσίασης. Βοηθάει όμως παράλληλα και στη συγγραφή προγραμμάτων. Το θεωρώ απόλυτα ολοκληρωμένο περιβάλλον για τα όσα ασχοληθήκαμε. Ακριβώς η οπτικοποίηση –όπως αναφέρεται παρακάτω– είναι που δίνει το πλεονέκτημα στο περιβάλλον.
- ◆ Το BlueJ με βοήθησε σημαντικά στη συνειδητοποίηση του ΤΙ γίνεται μέσα εκεί σε ένα πρόγραμμα, αλλά και στο πως σχετίζονται οι κλάσεις μεταξύ τους, αλλά και με τα αντικείμενα.

Ζητήθηκε από τους φοιτητές να αναφέρουν τα περισσότερο σημαντικά γι' αυτούς χαρακτηριστικά και για το περιβάλλον Jeroo. Οι απαντήσεις τους φαίνονται στον παρακάτω Πίνακα.

Πίνακας 4.8. Αξιολόγηση της ευχρηστίας του προγραμματιστικού μικρόκοσμου Jeroo (case study-1)

Ερώτηση	Αριθμός φοιτητών που επέλεξαν το χαρακτηριστικό (max 13)
Ποια από τα παρακάτω χαρακτηριστικά του Jeroo σας βοήθησαν στη μαθησιακή διαδικασία; Επιλέξτε αυτά που νομίζετε.	
Οπτικοποίηση της εκτέλεσης του προγράμματος στο Island τμήμα του περιβάλλοντος, με κίνηση των αντικειμένων.	6
Είναι απλό και εστιάζει στις έννοιες object και method.	6
Δυνατότητα βήμα προς βήμα εκτέλεσης.	5
Όλα τα συστατικά του περιβάλλοντος είναι ορατά σε <u>ένα</u> παράθυρο <u>διαρκώς</u> .	4
Επισήμανση με διαφορετικό χρώμα της γραμμής κώδικα που εκτελείται κάθε στιγμή.	4
Ευκολία σύνταξης της γλώσσας του περιβάλλοντος.	4
Συνάφεια της γλώσσας του περιβάλλοντος με την Java.	4
Δυνατότητα επεξεργασίας προβλημάτων με ενδιαφέρον και όχι προβλημάτων επεξεργασίας συμβόλων και αριθμών.	4
Δυνατότητα να δημιουργήσεις με ευκολία πολλά διαφορετικά Island layouts για να ελέγξεις τα προγράμματά σου.	2

Σε αυτή την ερώτηση απάντησε μικρός αριθμός φοιτητών (6) και εκτίμησαν χαρακτηριστικά του περιβάλλοντος όπως ο Πίνακας 4.8. δείχνει (την οπτικοποιημένη με κίνηση των αντικειμένων και κατά βήμα εκτέλεση του προγράμματος, την απλότητά του και την έμφασή του στις έννοιες “αντικείμενο” και “μέθοδος”, την επισήμανση με διαφορετικό χρώμα της γραμμής κώδικα που εκτελείται κάθε στιγμή κ.ά.).

- Στην ερώτηση “Σε τι κυρίως πιστεύετε ότι σας βοήθησε το περιβάλλον Jeroo; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ότι άλλο εσείς νομίζετε)”

Τέσσερις φοιτητές απάντησαν:

- ◆ Βοήθησε στο να δω τον προγραμματισμό περισσότερο σαν παιχνίδι. Ήταν σημαντικό που ήρθαμε σε επαφή με το Jeroo στην αρχή του σεμιναρίου καθώς βοήθησε στο να φύγει κάπως η φοβία προς τον αντικειμενοστρεφή προγραμματισμό.

- ◆ Με βοήθησε γιατί όταν έγραφα ένα πρόγραμμα έβλεπα την εκτέλεσή του με κίνηση και εικόνα και όχι ένα παράθυρο με αποτελέσματα και μπορώ εύκολα να δω και να διορθώσω τα λάθη μου.
- ◆ Με βοήθησε να μάθω να σκέφτομαι προγραμματιστικά.
- ◆ Το Jeroo με βοήθησε αρκετά στην κατανόηση των επαναληπτικών βρόχων, στους ελέγχους αλήθειας των καταστάσεων και στην εικονική βέβαια πραγμάτωσή τους. Είναι αρκετά παιδαγωγικό, πιο ξεκούραστο, σε βοηθάει να πλησιάσεις τον προγραμματισμό.

Έξι άλλοι φοιτητές απαντώντας στην ίδια ερώτηση έγραψαν:

- ◆ σε τίποτα απολύτως...
- ◆ Πέρα από το να μας εισάγει απλά στην έννοια της αντικειμενοστρέφειας του προγραμματισμού, δε βρήκα άλλη χρησιμότητα, συγκρίνοντας το κιόλας με το BlueJ.
- ◆ Δεν το χρησιμοποίησα δυστυχώς...
- ◆ Δεν έχω ασχοληθεί ιδιαίτερα με το περιβάλλον και δε γνωρίζω
- ◆ Δε δούλεψα πολύ με το Jeroo
- ◆ Δεν το χρησιμοποίησα.

Όπως προκύπτει από τα παραπάνω μόνον τέσσερις από τους συμμετέχοντες φοιτητές χρησιμοποίησαν το περιβάλλον Jeroo και σημείωσαν θετικά χαρακτηριστικά του και τη βοήθεια που τους προσέφερε, τέσσερις φοιτητές δε χρησιμοποίησαν καθόλου το περιβάλλον Jeroo και άλλοι δύο ισχυρίστηκαν ότι δεν τους χρησίμευσε σχεδόν καθόλου. Επίσης στην αντίστοιχη ερώτηση του Πίνακα 4.5., οι λυμένες ασκήσεις στο περιβάλλον Jeroo αναδείχθηκαν ο τελευταίος σε συμβολή μαθησιακός πόρος στη βελτίωση των γνώσεων και των δεξιοτήτων των φοιτητών. Σε συζητήσεις στην τάξη κάποιοι φοιτητές αποφάνθηκαν ότι το περιβάλλον τους φάνηκε “παιδικό” και αυτό δεν αποτελεί κίνητρο στο να το χρησιμοποιήσουν και ότι εκτιμούν ότι η χρήση του δε θα τους προσθέσει σε γνώση.

Όλα τα παραπάνω οδήγησαν στην παρέμβαση της αφαίρεσης του προγραμματιστικού μικρόκοσμου Jeroo και όλων των ασκήσεων-δραστηριοτήτων των σχετικών με αυτόν, από την προτεινόμενη διδακτική προσέγγιση στις επόμενες εφαρμογές της. Αυτό έγινε

εκτιμώντας σε συνδυασμό την αξιολόγηση και των δύο εκπαιδευτικών περιβαλλόντων που χρησιμοποιήθηκαν στην πρώτη αυτή πιλοτική εφαρμογή της προσέγγισης.

Ποιοτικά αποτελέσματα

Στις ανοικτού τύπου ερωτήσεις του ερωτηματολογίου μετά-αξιολόγησης δόθηκαν απαντήσεις που παρατίθενται στη συνέχεια.

- **Ερώτηση:** Τι σας άρεσε περισσότερο στο μαθησιακό υλικό;
 - ◆ Η θεωρία όπως ήταν διατυπωμένη.
 - ◆ Τα πολλά παραδείγματα και οι διαφορετικού τύπου ασκήσεις. Πραγματικά με βοήθησαν πάρα πολύ.
 - ◆ Παραδείγματα και ασκήσεις.
 - ◆ Η απλότητα με την οποία περιγράφονταν οι έννοιες αλλά παράλληλα και η εκτενής ανάλυσή τους. Το γεγονός ότι επαναλαμβάνονταν συχνά σημαντικά σημεία εκεί που πραγματικά χρειαζόταν χωρίς να γίνεται κουραστική η ανάγνωση του υλικού.
 - ◆ Η ποικιλία και τα πολλά λυμένα παραδείγματα. Οι ασκήσεις και επεξηγήσεις τους.
 - ◆ Τα πολλά παραδείγματα και η επικοινωνία με την καθηγήτρια.
 - ◆ Η θεωρία και οι λυμένες ασκήσεις.
 - ◆ Ότι το μάθημα ξεκίνησε από αρχάριο επίπεδο.
 - ◆ Η άμεση επαφή με τα προγράμματα καθώς και η ύπαρξη του διαρκώς ενημερωμένου δικτυακού τόπου.
 - ◆ Ο τρόπος που διδάσκονταν, διαφάνειες και πολλά παραδείγματα κατά τη διάρκεια του μαθήματος.
 - ◆ Ήταν το πιο κατάλληλο από όσα έχω ασχοληθεί για να ξεκινήσει κανείς με Java. Το γεγονός ότι η καθηγήτρια ασχολιόταν πάρα πολύ με τον κάθε μαθητή ξεχωριστά, πολύ μεγάλο βάρος στις απορίες. Το ότι εγώ κατάλαβα τη φιλοσοφία του αντικειμενοστρεφούς προγραμματισμού –επιτέλους–. Το υλικό ήταν απλό –χωρίς να μειώνει το επίπεδο αυτό– και καλά δομημένο.
 - ◆ Η ξεκούραστη και ομαλή διάρθρωση του κειμένου-κώδικα ώστε να διαβάζεται και να είναι ευκολότερα κατανοητά. Στοιχίση, χρώματα, κλπ.

- **Ερώτηση:** Τι δε σας άρεσε στο μαθησιακό υλικό;

- ◆ Το Jeroo.
 - ◆ Η ποσότητα και σε μερικά ήταν λίγο βαρετά διατυπωμένα.
 - ◆ Οι ασκήσεις επαναλαμβάνονταν πολύ. Συναντούσα πολύ συχνά όμοιες εκφωνήσεις. Ήταν λίγο κουραστικό.
 - ◆ Το ότι ήταν κάπως παθητικό το κείμενο. Δε μας ενεργοποιούσε λίγο πριν τη θεωρία για να φτιάξουμε ένα πρόγραμμα που θα κάνει κάτι... Άρα χρειαζόμαστε αυτό... Που το λέμε έτσι και κάνει αυτά. Δηλαδή πιο ενεργό και ανακαλυπτικό κείμενο.
- **Ερώτηση:** Τι θα μπορούσε να σας βοηθήσει στην κατανόηση του γνωστικού αντικειμένου και δεν υπάρχει στο υλικό ή στην οργάνωση του σεμιναρίου;
 - ◆ Η πρακτική εξάσκηση ταυτόχρονα με την παράδοση του μαθήματος.
 - ◆ Πρακτική εξάσκηση σε H/Y.
 - ◆ Θα με βοηθούσε αν είχαμε περισσότερες ώρες να ασχοληθούμε με το να γράφουμε κώδικα στην αίθουσα μόνοι μας και μετά να το συζητάμε όπως κάναμε στο τέλος. Αυτό με βοήθησε να καταλάβω πως φτιάχνουμε ένα πρόγραμμα από την αρχή ως το τέλος.
 - ◆ Κάποιο εργαστήριο.
 - ◆ Θα μπορούσε να με βοηθήσει η περιγραφή του σκοπού της κάθε νέας έννοιας που δέχομαι. Και όχι να τη δέχομαι μηχανικά με αποτέλεσμα να ψάχνω εγώ μετά με απορίες ποιος είναι ο ρόλος και ο σκοπός της. Π.χ. static δηλώσεις μεθόδων. Γιατί γίνεται αυτό; Θέλω πρώτα να μου δημιουργείται η ανάγκη και μετά να αναζητήσω την έννοια.

Προκύπτει, από τις συστάσεις των φοιτητών στις δύο προηγούμενες ερωτήσεις, η ανάγκη καλύτερης διατύπωσης των ασκήσεων-ερωτήσεων και η προσεκτικότερη παρουσίαση εννοιών ώστε πριν να έχει αναδειχθεί η ανάγκη τους.

Φυσικά, αναδείχθηκε η αδιαμφισβήτητη ανάγκη τα μαθήματα διδασκαλίας Προγραμματισμού να συνοδεύονται από εργαστήριο που λόγοι υποδομής δε μας επέτρεψαν να γίνει στο σεμινάριο και οι ίδιοι λόγοι επιβάλλουν τη διδασκαλία αντίστοιχων υποχρεωτικών μαθημάτων στο Τμήμα ΔΤΨΣ, μόνο μέσω διαλέξεων.

- **Ερώτηση:** Ποιοι μαθησιακοί πόροι νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;

- ◆ Ίσως η παράθεση κάποιων sites που έχουν αρκετό υλικό σχετικά με το αντικείμενο να λειτουργούσε σαν μια επιπλέον βοήθεια.
- ◆ Έπρεπε να ασχοληθούμε περισσότερο με πιο πολύπλοκα προβλήματα.
- ◆ Οι ασκήσεις να είναι πιο δημιουργικές, ας είναι περισσότερες. Δηλαδή όπως είναι αυτές που είχαμε αλλά και κάποιες πιο απλές όπου καθοδηγούμαστε να φτιάξουμε κάτι πολύ απλό αλλά από μόνοι μας και να το βλέπουμε.

Από τα παραπάνω, προκύπτει η αναγκαιότητα ενσωμάτωσης στη διδακτική προσέγγιση πολύπλοκότερων εργασιών-εφαρμογών οι οποίες θα παρακινούν περισσότερο τους φοιτητές στην υλοποίηση τους και η παράθεση επιπλέον πληροφοριακού υλικού στο οποίο θα μπορούν να ανατρέξουν για το αντικείμενο του ΑΠ και τη Java.

- **Ερώτηση:** Ποιες ενότητες/τμήματα του υλικού νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;
 - ◆ Κελυφοποίηση και Πολυμορφισμός
 - ◆ Κελυφοποίηση-απόκρυψη πληροφορίας (encapsulation-information hiding) - Διαχωρισμός συμπεριφοράς-υλοποίησης.
 - ◆ Οι ενότητες με τα interfaces.
 - ◆ Οι ασκήσεις.

Οι φοιτητές προτείνουν βελτίωση της 3^{ης} διδακτικής ενότητας του μαθησιακού υλικού και ενός τμήματος της 7^{ης} διδακτικής ενότητας.

Επιπλέον από συζητήσεις στην τάξη, προέκυψε η δυσκολία των φοιτητών να αντεπεξέλθουν στις ανάγκες του σεμιναρίου έως ότου τους παρουσιάστηκε η 4^η σε σειρά διδακτική ενότητα που αφορούσε στη μέθοδο κατασκευαστή, στη δημιουργία αντικειμένων και στους τύπους αναφοράς. Να σημειωθεί ότι σε αυτήν την εφαρμογή της διδακτικής προσέγγισης (case study-1) η σειρά διδασκαλίας της 3^{ης} και 4^{ης} διδακτικής ενότητας ήταν αντίστροφη από αυτή που εμφανίζεται στο Παράρτημα Α, η οποία αποτελεί κα την τελική σειρά παρουσίασης των διδακτικών ενότητων.

4.4.4.2. Επεξεργασία εργασιών

Μαθησιακά αποτελέσματα

Καταρχάς, τρεις από τους φοιτητές που παρακολούθησαν το σεμινάριο συμμετείχαν

στις εξετάσεις του υποχρεωτικού μαθήματος “Αντικειμενοστρεφής Προγραμματισμός” και όλοι πέτυχαν θετικό αποτέλεσμα, ένας με βαθμό 10, ο άλλος με 8 και ο τρίτος με 6.

Στη συνέχεια, παρατίθενται “ενδείξεις” που αποδεικνύουν την κατανόηση εννοιών-αρχών και άρα την αποτελεσματικότητα της προτεινόμενης διδακτικής προσέγγισης στην αντιμετώπιση σημαντικών ζητημάτων στη διδασκαλία του ΑΠ (ενότητα 3.2.), καθώς και δυσκολίες και παρανοήσεις που εντοπίσαμε όπως προέκυψαν από τη μελέτη των σημειώσεων που κρατήθηκαν κατά τη διάρκεια των διαλέξεων και των εργασιών που οι φοιτητές παρέδωσαν.

- Οι φοιτητές διακρίνουν καλά μεταξύ τους τις έννοιες κλάση-αντικείμενο. Αυτό προέκυψε από την ορθή κατάταξη ενός συνόλου λέξεων σε μια από τις δύο κατηγορίες.
 - Κανένας δε θεώρησε ένα σύνολο από αντικείμενα π.χ. το τμήμα A_3 του $3^{ου}$ Λυκείου Πειραιά, ως κλάση (Teif and Hazzan, 2004).
 - Κανένας δε συγγέει τη σχέση όλου-μέρους με τη σχέση κλάση-αντικειμένου ή τη σχέση ιεραρχίας (Teif and Hazzan, 2004).
 - Πριν τη διδασκαλία της 7^{ης} διδακτικής ενότητας οι περισσότεροι συνέχισαν τη σχέση κλάση-αντικειμένου με τη σχέση ιεραρχίας (Teif and Hazzan, 2004). Μετά την ολοκλήρωση και της 7^{ης} ενότητας οι περισσότεροι φοιτητές απάντησαν σωστά στις ίδιες ερωτήσεις.
 - Μια παρανόηση που παρατηρήθηκε είναι ότι μερικοί φοιτητές αποδίδουν ως ιδιότητες κλάσεων και αυτές που αποτελούν συνάρτηση άλλων ήδη δηλωμένων ιδιοτήτων π.χ.
 - ◆ κλάση Κύκλος: ιδιότητες: ακτίνα, διάμετρος, εμβαδόν, περίμετρος
 - ◆ κλάση Ορθογώνιο: ιδιότητες: μήκος, πλάτος, μήκος διαγωνίου
- Σα βελτίωση, για την επόμενη εφαρμογή της διδακτικής προσέγγισης συστήνεται για την αντιμετώπιση αυτής της παρανόησης, ο πειραματισμός των φοιτητών με την απόδοση ιδιοτήτων σε ειδικά επιλεγμένες κλάσεις που είναι αρκετά πιθανό να τους δημιουργήσουν την παραπάνω παρανόηση και η συζήτηση της προσπάθειάς τους στην τάξη ή στο forum συζητήσεων.
- Σε μικρό σχετικά βαθμό παρατηρήθηκε η ήδη διαπιστωμένη δυσκολία να διακρίνουν μεταξύ λειτουργιών και ιδιοτήτων μιας κλάσης π.χ. ομιλία, ακοή, όραση, σκέψη για την κλάση “Άνθρωπος” και μεταξύ των μερών από τα οποία αποτελείται και ιδιοτήτων π.χ.

- ◆ ρίζες, κορμός για την κλάση “Δέντρο”
- ◆ πόδια, χέρια, κεφάλι για την κλάση “Άνθρωπος” (Teif and Hazzan, 2004)
- Σε μικρό βαθμό *δυσκολεύονται* να αποδώσουν ως ιδιότητες οντοτήτων λογικές (boolean) μεταβλητές, ή να κατατάξουν στην κατηγορία “ιδιότητα κλάσης” εκφράσεις που μεταφράζονται σε boolean μεταβλητές π.χ. δε θεωρούν το “πεινάει ή όχι” ιδιότητα των κλάσεων “Άνθρωπος” και “Σκύλος” αλλά αντίθετα το θεωρούν μέθοδο.

Η παρανόηση αυτή μπορεί να αποδοθεί στη χρήση ρήματος που παραπέμπει σε ενέργεια (συμπεριφορά και επομένως μέθοδο) και προτείνεται η χρήση του “είναι πεινασμένος ή όχι” αντί του “πεινάει ή όχι”. Επίσης για παρόμοιο λόγο εκτιμάται ότι θεωρούν τα: “ρύθμιση έντασης ήχου” και “ρύθμιση εικόνας” για την κλάση “Τηλεοπτική συσκευή” και τα: “ομιλία”, “σκέψη”, “ακοή”, “όραση” για την κλάση “Άνθρωπος”, ιδιότητες και όχι μεθόδους, εξαιτίας της χρήσης ουσιαστικού και όχι ρήματος για την περιγραφή τους.

Γενικά ο *ορθός προσδιορισμός των ιδιοτήτων κλάσεων, αναδεικνύεται δύσκολο σημείο* για τους φοιτητές. Ως μέθοδος αντιμετώπισης θα μπορούσε να χρησιμοποιηθεί η χρήση ειδικά σχεδιασμένων παραδειγμάτων στα οποία έχουν αποδοθεί λανθασμένα αλλά και σωστά ιδιότητες σε κλάσεις και η συζήτησή τους στην τάξη ή στο forum συζητήσεων. Τα παραδείγματα θα πρέπει να σχεδιάζονται αφού ληφθούν υπόψη όλες οι προαναφερθείσες δυσκολίες.

- Ως προς τη σημασία των προσδιοριστών ορατότητας private, public και τη χρήση τους στην υλοποίηση κλάσεων, η πλειοψηφία των φοιτητών αντιλαμβάνεται την αρχή της κελυφοποίησης-απόκρυψης πληροφορίας (encapsulation-information hiding) και δημιουργούν μεθόδους (accessors) για την προσπέλαση των ιδιωτικών (private) ιδιοτήτων.
- Όσον αφορά στην υλοποίηση κλάσεων το κύριο *πρόβλημα* που αναδείχθηκε είναι, ότι μερικοί φοιτητές όταν χρειάζονταν κάποια ενδιάμεση μεταβλητή, τη δήλωναν ως ιδιότητα της κλάσης, αντί για τοπική μεταβλητή στην κατάλληλη μέθοδο.

Αυτό επιβεβαιώνει και τα προηγούμενα ευρήματα για τη σημασία που έχει η κατανόηση του τι αποτελεί ιδιότητα κλάσης και επίσης μπορεί να θεωρηθεί απόρροια της εμπειρίας των φοιτητών από το διαδικασιακό Προγραμματισμό, όπου όλες οι αναγκαίες μεταβλητές συνηθίζεται να δηλώνονται στην αρχή του

προγράμματος και αυτό είναι συνήθως το σημείο που στον ΑΠ δηλώνονται οι ιδιότητες μιας κλάσης.

- Μόνο λίγοι φοιτητές, μεταξύ μιας υλοποίησης κλάσης με private ιδιότητες και μεθόδους get, set για αυτές τις ιδιότητες και της ίδιας κλάσης με public ιδιότητες, θεωρούν ευκολότερη στην κατανόηση τη δεύτερη (Fleury, 2001) και ταυτόχρονα αυτοί οι φοιτητές αναφέρουν ότι ο σχεδιασμός αυτός παραβιάζει την αρχή της κελυφοποίησης. Ένας φοιτητής διατύπωσε σαφώς την εξής σημαντική άποψη: “ο σχεδιασμός της πρώτης κλάσης είναι προτιμότερος αφού χρησιμοποιούνται σωστά οι προσδιοριστές ορατότητας στις ιδιότητες και τις μεθόδους. Με πρώτη ματιά, ο δεύτερος σχεδιασμός φαίνεται ευκολότερος στην κατανόηση αφού δεν έχουμε μεθόδους για την πρόσβαση στις ιδιότητες. Αλλά είναι καλύτερα να δυσκολευτούμε λίγο στην κατανόηση των προγραμμάτων, παρά να χρησιμοποιούμε “ευάλωτα” προγράμματα (δηλαδή με public ιδιότητες) όπως στη δεύτερη περίπτωση”.
- Μερικοί φοιτητές συμπεριλαμβάνουν λανθασμένα στη διαπροσωπεία (interface) των κλάσεων και τον κώδικα των μεθόδων τους εκτός από την “υπογραφή” τους.
- Πριν τη διδασκαλία των θεμάτων “δημιουργία αντικειμένων-κατασκευαστής-τύποι αναφοράς” (4^η διδακτική ενότητα), παρατηρήθηκε σε μικρό ποσοστό η εντύπωση του “τίποτα” ή “καθόλου τιμή” ή “κενό” ως τιμή των ιδιοτήτων, όταν δημιουργείται αντικείμενο χωρίς να υπάρχει μέθοδος κατασκευαστή δηλωμένη στην κλάση (Ragonis and Ben-Agi, 2002). Μετά την 4^η διδακτική ενότητα οι παρανοήσεις αυτές δεν παρατηρήθηκαν και οι φοιτητές αντιλήφθηκαν καλά την κατανομή μνήμης που συμβαίνει όταν δημιουργούνται αντικείμενα, όλες τις περιπτώσεις αρχικοποίησης των ιδιοτήτων (Ragonis and Ben-Agi, 2002) και επίσης ότι είναι αναγκαία η εκτέλεση εντολής με τον τελεστή new για να δημιουργηθεί αντικείμενο.
- Παρατηρήθηκε σύγχυση στην κατανομή μνήμης για τους βασικούς τύπους δεδομένων. Μερικοί φοιτητές θεώρησαν και τους βασικούς τύπους (π.χ. double x) ως αναφορά σε μια θέση μνήμης που περιέχει την τιμή της μεταβλητής. Αυτό επιβεβαιώνει τη δυσκολία που αντιμετωπίζουν σε θέματα που σχετίζονται με τη μνήμη (Milne and Rowe, 2002). Ως βελτίωση, για την επόμενη εφαρμογή της διδακτικής προσέγγισης συστήνεται να δοθεί έμφαση στη δυναμική “οπτικοποίηση” του τι συμβαίνει “εσωτερικά” στη μνήμη και για την περίπτωση των βασικών τύπων δεδομένων.

- Παρατηρήθηκε *δυσκολία* στις σχηματικές παραστάσεις της κατάστασης στη μνήμη στην περίπτωση των *composed* κλάσεων –δηλαδή των κλάσεων που διαθέτουν ιδιότητες του τύπου μιας άλλης κλάσης–, στους τύπους αναφοράς που “συνδέονται” σε αντικείμενα αυτών των κλάσεων και στην καταστροφή αντικειμένων τους.
Θα πρέπει και εδώ να αξιοποιηθεί η παρέμβαση της “οπτικοποίησης”.
- Δεν παρατηρήθηκε η παρανόηση της θεώρησης δύο αντικειμένων με ακριβώς ίδιες τιμές στις ιδιότητές τους ως ένα αντικείμενο (Holland et al., 1997).
- Δεν παρατηρήθηκε η παρανόηση ότι η ύπαρξη και μόνον της μεθόδου του κατασκευαστή στον κώδικα της κλάσης, αρκεί για τη δημιουργία αντικειμένων (Ragonis and Ben-Ari, 2002).
- Σε μικρό ποσοστό παρατηρήθηκε η *παρανόηση* ότι δεν μπορούμε να μεταβάλλουμε την τιμή μιας μεταβλητής κλάσης (static variable).
Η εξήγηση είναι ότι οι φοιτητές ερμηνεύουν τη λέξη static που χαρακτηρίζει τις μεταβλητές κλάσης, ως σταθερά (constant).
- Αντιλαμβάνονται τη μοναδικότητα των μεταβλητών κλάσεων για όλα τα αντικείμενα μιας κλάσης, όπως διαπιστώθηκε με ερωτήσεις του τύπου “τι θα εκτυπωθεί;” και ερωτήσεις που ζητούσαν τη σχηματική παράσταση της μνήμης.
- Υλοποιούν σωστά την κλήση μεθόδων κλάσης και αναφέρονται επίσης σωστά στις μεταβλητές κλάσης και στιγμιοτύπου από μεθόδους κλάσης και στιγμιοτύπου.
- Σε μικρό βαθμό, παρατηρήθηκε η *παρανόηση*, ότι σε μια μεταβλητή κλάσης μπορεί να γίνει αναφορά μόνο μέσα από μια μέθοδο κλάσης. Δηλαδή “συσχετίζουν” λανθασμένα τις μεθόδους και τις μεταβλητές κλάσης μεταξύ τους.
- Αντιλαμβάνονται σωστά την έννοια της υπερφόρτωσης (overloading), διακρίνουν τις αποδεκτές μορφές της και μπορούν να “επιλύσουν” το ποια μέθοδος εκτελείται, μεταξύ υπερφορτωμένων μεθόδων.
- Οι φοιτητές είναι σε θέση να γενικεύουν και να εξειδικεύουν μεταξύ οντοτήτων και επίσης υλοποιούν σωστά ιεραρχίες κλάσεων.
- Αντιλαμβάνονται σωστά τι και πώς κληρονομείται σε όλη την έκταση μιας ιεραρχίας κλάσεων με *εξάιρεση* ότι θεωρούν ότι κληρονομείται και η μέθοδος του κατασκευαστή.

- “Επιλύουν” σωστά ποια μέθοδος εκτελείται στην περίπτωση της υπερίσχυσης (overriding) μεθόδων, όπως προέκυψε από ερωτήσεις του τύπου “τι θα εκτυπωθεί;”.
- Διακρίνουν σωστά μεταξύ τους τις overloaded και τις overridden μεθόδους και συνεπώς τα δύο αυτά διαφορετικά θέματα.

4.4.5. Συμπεράσματα – Βελτιώσεις

Από την πρώτη πιλοτική εφαρμογή της διδακτικής προσέγγισης, προέκυψαν θετικά συμπεράσματα ως προς την υιοθέτηση των αρχών που διέπουν την εκπαιδευτική φιλοσοφία της (ενότητα 3.4.), την ποιότητα και τη μαθησιακή αποτελεσματικότητα του μαθησιακού υλικού που τη συνοδεύει (ενότητα 3.7.), τη μαθησιακή αποτελεσματικότητα του τρόπου εφαρμογής της (on-line μαθησιακό περιβάλλον, forum, διαλέξεις) (ενότητες 3.5., 3.6.), την αξιοποίηση του εκπαιδευτικού περιβάλλοντος BlueJ (ενότητα 3.8.), την αντιμετώπιση καταγεγραμμένων δυσκολιών στη διδασκαλία του ΑΠ (ενότητα 3.2.). Συγχρόνως, η πρώτη αυτή εφαρμογή απέφερε σημαντικό υλικό για την περαιτέρω βελτίωση και ενίσχυση της διδακτικής προσέγγισης. Συγκεκριμένα:

- Προκύπτει η ανάγκη να παρουσιάζονται στους εκπαιδευόμενους, συνολικά όλες οι φάσεις ανάπτυξης μιας αντικειμενοστρεφούς εφαρμογής (ανάλυση, σχεδιασμός, υλοποίηση, έλεγχος, αποσφαλμάτωση, εκτέλεση), επαρκώς πολύπλοκης για το επίπεδο τους και ενδιαφέρουσας.
- Στις εργασίες που οι εκπαιδευόμενοι εκπονούν και παραδίδουν, κρίνεται σκόπιμο να γίνει μια σημαντική αλλαγή που αφορά στην παράδοση σε τμήματα ενός μεσαίου μεγέθους project, το οποίο θα αναπτύσσεται σταδιακά ακολουθώντας την παρουσίαση που θα τους έχει γίνει και υποβοηθούμενοι από το μαθησιακό υλικό και το on-line μαθησιακό περιβάλλον στο οποίο εντάσσεται η διδακτική προσέγγιση και τις δυνατότητες που αυτό προσφέρει. Ανάλογα με τις συνθήκες της εφαρμογής της προσέγγισης και το φόρτο των εκπαιδευομένων, μπορεί να τους ζητείται και η εκπόνηση των επί μέρους εργασιών μαζί με το project ή μόνον του project και οι επί μέρους εργασίες να μην είναι υποχρεωτικές. Η αλλαγή αυτή θα δώσει στους εκπαιδευόμενους τη δυνατότητα της αποδοτικότερης αντίληψης

του πώς συσχετίζονται μεταξύ τους οι αντικειμενοστρεφείς έννοιες και βρίσκουν εφαρμογή στην ανάπτυξη ενός έργου.

- Το μαθησιακό εργαλείο του προγραμματιστικού μικρόκοσμου Jeroo, δεν προέκυψε ότι ενισχύει την αποτελεσματικότητα της προσέγγισης και γι' αυτό παραλείφθηκε η χρήση του από τις επόμενες εφαρμογές της διδακτικής προσέγγισης.
- Έγινε φανερή η ανάγκη τόσο από συζητήσεις στην τάξη όσο και από την επεξεργασία των εργασιών που οι φοιτητές παρέδωσαν, της αλλαγής της σειράς μεταξύ της 3^{ης} (αφαιρετικότητα, κελυφοποίηση-απόκρυψη πληροφορίας, διαχωρισμός συμπεριφοράς-υλοποίησης) και της 4^{ης} διδακτικής ενότητας (κατασκευαστής, δημιουργία αντικειμένων, κατανομή μνήμης, τύποι αναφοράς), επειδή η δημιουργία αντικειμένων και η μέθοδος του κατασκευαστή αποτελούν πολύ σημαντικά θέματα και θα πρέπει να παρουσιάζονται πριν από οποιαδήποτε άλλη έννοια και αμέσως μετά την εισαγωγή των θεμελιωδών αντικειμενοστρεφών εννοιών.
- Στην 2^η διδακτική ενότητα που αφορά στις βασικές έννοιες κλάση, αντικείμενο, ιδιότητες, μέθοδοι, έγιναν παρεμβάσεις στην κατεύθυνση αντιμετώπισης των δυσκολιών που περιγράφηκαν ως προς τον ορθό προσδιορισμό των ιδιοχαρακτηριστικών μιας κλάσης. Επίσης προσοχή δόθηκε στη χρήση ρημάτων αντί ουσιαστικών για τις μεθόδους των κλάσεων.
- Στην 3^η διδακτική ενότητα (κατασκευαστής κ.λ.π.) η δυναμική “οπτικοποίηση” (υποενότητα 3.7.2.) χρησιμοποιήθηκε και για τους βασικούς τύπους δεδομένων (primitive data types) και για τις συντεθειμένες (composed) κλάσεις.
- Έγιναν επεμβάσεις σε πόρους του μαθησιακού υλικού ώστε να διατυπώνονται καλύτερα οι ερωτήσεις-ασκήσεις και ελήφθησαν υπόψη παρατηρήσεις που οι φοιτητές έκαναν για παρανοήσεις που τους δημιουργούσαν συγκεκριμένα σημεία στο υλικό, ώστε να αποφευχθούν στο μέλλον. Επίσης προσοχή δόθηκε, στην πολύ σημαντική παρατήρηση ενός φοιτητή της παρουσίασης των εννοιών αφού πρώτα αναδειχθεί η ανάγκη τους. Π.χ. στην 5^η διδακτική ενότητα, η παρουσίαση της έννοιας μεταβλητή κλάσης (static variable).
- Ενισχύθηκε το υλικό της 4^{ης} (κελυφοποίηση κ.λ.π.) και της 7^{ης} διδακτικής ενότητας με επιπλέον θεωρία-παραδείγματα-ασκήσεις και προστέθηκε στην 7^η ενότητα η παρουσίαση κάποιων εννοιών (όπως τελικές κλάσεις - μέθοδοι).

- Ενσωματώθηκε στην αρχική παρουσίαση της προσέγγισης, περισσότερο πληροφοριακό υλικό.
- Αδιαμφισβήτητα οι φοιτητές θα πρέπει να εξασκούνται κάνοντας χρήση Η/Υ και αυτό θα πρέπει να ενισχύεται κατά το δυνατόν από τον τρόπο εφαρμογής της προσέγγισης (πραγματοποίηση προγραμματιστικών έργων, συζήτησή τους και παράδοσή τους) ή/και να εξασφαλίζεται η απαραίτητη υποδομή για τη διενέργεια εργαστηρίων.
- Στις επόμενες εφαρμογές της προσέγγισης, κρίνεται σκόπιμο να δοθεί έμφαση στην πραγματοποίησή τους με τρόπο που να ενισχύει τη δημιουργία ενεργών online κοινοτήτων μάθησης μεταξύ των εκπαιδευομένων και όχι μόνο μεταξύ εκπαιδευομένων και εκπαιδευτών.

4.5. Μελέτη περίπτωσης 2 (case study-2)

Στην ενότητα αυτή παρουσιάζεται η δεύτερη πιλοτική εφαρμογή της προτεινόμενης διδακτικής προσέγγισης, στην οποία ακολουθήθηκαν οι φάσεις που περιγράφηκαν στην ενότητα 3.5. με την παρουσίαση στους φοιτητές μιας ενδιαφέρουσας και επαρκώς πολύπλοκης για το επίπεδό τους αντικειμενοστρεφούς εφαρμογής και των φάσεων ανάπτυξής της, κάνοντας χρήση των μαθησιακών εργαλείων BlueJ και SUN One Studio και συστηματική χρήση της διαγραμματικής τεχνικής (διαγράμματα κλάσεων και αντικειμένων). Η εφαρμογή σε αυτή τη δεύτερη μελέτη περίπτωσης έγινε χωρίς ακόμα να έχει ενσωματωθεί στη διδακτική προσέγγιση η χρήση κάποιου design CASE tool. Επίσης στην προσέγγιση επήλθαν οι βελτιώσεις – αλλαγές – προσθήκες που περιγράφηκαν στην υποενότητα 4.4.5. σε σχέση με την πρώτη πιλοτική εφαρμογή της.

4.5.1. Χώρος εφαρμογής

Η διδακτική προσέγγιση εφαρμόστηκε κατά το εαρινό εξάμηνο του 2006 με τη μορφή e-learning μαθήματος, ως μέρος του υποχρεωτικού μαθήματος “Αντικειμενοστρεφής Προγραμματισμός” στο Τμήμα ΔΤΨΣ του Πανεπιστημίου Πειραιά, αλλά ανεξάρτητο αυτού και υποστηρικτικό αυτού. Οι φοιτητές είχαν τη δυνατότητα να συμμετέχουν στο e-learning τμήμα του μαθήματος –συμμετέχοντας

στις δια ζώσης συναντήσεις στην τάξη που περιελάμβανε, μελετώντας το μαθησιακό υλικό που τους παρέχόταν, πραγματοποιώντας εκπαιδευτικές δραστηριότητες, συμμετέχοντας και αξιοποιώντας τις δυνατότητες του on-line μαθησιακού περιβάλλοντος και εκπονώντας και παραδίδοντας εργασίες– και να παρακολουθούν το μάθημα που διεξαγόταν με τον παραδοσιακό τρόπο διδασκαλίας (από-καθέδρας). Η τελική τους επίδοση στο μάθημα προέκυπτε κατά 20% από τη βαθμολογία των εργασιών που παρέδιδαν για το e-learning τμήμα του μαθήματος και κατά το υπόλοιπο 80% από την τελική γραπτή εξέταση του μαθήματος. Συνεπώς προκύπτει εύλογα το κίνητρο συμμετοχής των φοιτητών και η διαφορετικότητα του σε σύγκριση με τα κίνητρα των φοιτητών στην προηγούμενη εφαρμογή (case study-1) της προτεινόμενης διδακτικής προσέγγισης, χωρίς φυσικά να αποκλείεται το ενδιαφέρον των φοιτητών για κάτι περισσότερο του υποχρεωτικού μαθήματος και ενισχυτικό αυτού.

4.5.2. Συμμετέχοντες

Δήλωσαν συμμετοχή στο e-learning τμήμα του μαθήματος εξήντα έξι (66) φοιτητές, 38 άνδρες και 28 γυναίκες, ηλικίας από 18 έως 24 ετών. Προέρχονταν από όλα τα εξάμηνα, με την πλειοψηφία τους από το 2^ο εξάμηνο του οποίου μάθημα αποτελεί ο “Αντικειμενοστρεφής Προγραμματισμός”. Συγκεκριμένα η κατανομή των συμμετεχόντων φοιτητών ανά εξάμηνο είναι η εξής:

2^ο εξάμηνο: 45

4^ο εξάμηνο: 10

6^ο εξάμηνο: 4

8^ο εξάμηνο: 3

Πτυχίο: 4

Όλοι είχαν προηγουμένως διδαχθεί κατά τη διάρκεια του 1^{ου} εξαμήνου των σπουδών τους διαδικασιακό προγραμματισμό με τη χρήση της γλώσσας προγραμματισμού C.

4.5.3. Υλοποίηση της εφαρμογής

Κατά τη διάρκεια του e-learning μαθήματος πραγματοποιήθηκαν τέσσερις (4) συνεδρίες συντονισμού και επίλυσης αποριών και επίσης σε αυτές έγινε η

παρουσίαση των φάσεων υλοποίησης της αντικειμενοστρεφούς εφαρμογής με τη χρήση φορητού υπολογιστή και βιντεοπροβολέα και κάνοντας χρήση των προγραμματιστικών περιβαλλόντων BlueJ και SUN One Studio.

Η διεξαγωγή του e-learning μαθήματος ακολούθησε τη ροή δραστηριοτήτων που περιγράφηκε στην ενότητα 3.5. Το on-line μαθησιακό περιβάλλον (ενότητα 3.6.) αυτής της μελέτης περίπτωσης, υλοποιήθηκε με τη δημιουργία δικτυακού τόπου στο περιβάλλον της ηλεκτρονικής πύλης μαθημάτων e-class του Τμήματος ΔΤΨΣ του Πανεπιστημίου Πειραιά (<http://orfeas.ted.unipi.gr/elearn/index.php>).

4.5.4. Αξιολόγηση της εφαρμογής

Οι φοιτητές συμπλήρωσαν τα δύο ερωτηματολόγια (pre-test και post-test) –που παρατίθενται στα Παραρτήματα Δ και Ε αντίστοιχα– στην αρχή και μετά το τέλος του e-learning μαθήματος.

Στις εργασίες τις οποίες εκπόνησαν, περιλαμβανόταν εκτός των επί μέρους μικρών εργασιών και μια μεσαίου μεγέθους εφαρμογή (υποενότητα 4.4.5.) την οποία παρέδωσαν κατά τμήματα, καθώς προχωρούσαν τη μελέτη των διδακτικών ενοτήτων του μαθησιακού υλικού και εξελισσόταν αντίστοιχα η παρουσίαση στις δια ζώσης συναντήσεις στην τάξη. Οι εργασίες όπως αναφέρθηκε, βαθμολογήθηκαν και συνεισέφεραν κατά 20% στην τελική επίδοση του μαθήματος.

Η τρίτη φάση (Φάση ΙΙΙ: Αυτόνομη επίλυση προβλημάτων, από τις φάσεις που περιγράφηκαν στην ενότητα 3.5.) στη δεύτερη αυτή εφαρμογή της διδακτικής προσέγγισης, δεν περιείχε αυτόνομη επίλυση με ξεχωριστό project αλλά οι φοιτητές οδηγήθηκαν στις εξετάσεις του μαθήματος.

Συνολικά οι φοιτητές που συμμετείχαν στην αξιολόγηση αυτής της εφαρμογής της διδακτικής προσέγγισης ήταν εξήντα έξι (66) ως προς τα δύο ερωτηματολόγια, εξήντα τέσσερις (64) ως προς την παράδοση των εργασιών (28 γυναίκες και 36 άνδρες) και εξήντα (60) ως προς τη συμμετοχή τους στις τελικές εξετάσεις του μαθήματος (28 γυναίκες και 32 άνδρες). Η πλειοψηφία τους και σε αυτή τη μελέτη περίπτωσης (44 στους 66 = 66.66%), χρησιμοποίησε μόνον το υλικό που διατέθηκε στο e-learning τμήμα του μαθήματος.

4.5.4.1. Επεξεργασία ερωτηματολογίων

Ποσοτικά Αποτελέσματα

Από την επεξεργασία των απαντήσεων στα δύο ερωτηματολόγια προέκυψαν τα ποσοτικά στοιχεία που παρουσιάζονται στους παρακάτω πίνακες. Να σημειωθεί ότι επειδή αναμενόταν –λόγω του συνδυασμού του e-learning μαθήματος με το υποχρεωτικό μάθημα του 2^{ου} εξαμήνου του Τμήματος ΔΤΨΣ– αρκετοί φοιτητές να μην έχουν προηγούμενη εμπειρία στο αντικείμενο του ΑΠ, σχετικές ερωτήσεις του ερωτηματολογίου προ-αξιολόγησης αναφέρονται στον Προγραμματισμό γενικά και όχι στον ΑΠ.

Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στον ΑΠ

Πίνακας 4.9. Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στο γνωστικό αντικείμενο του ΑΠ (case study-2)

Ερώτηση	M.O. (Από το “pre-test”)	M.O. (Από το “post-test”)
Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον Προγραμματισμό;	2.30	*
Ποιο εκτιμάτε ότι είναι το επίπεδό σας στο διαδικασιακό προγραμματισμό;	*	2.80
Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον ΑΠ;	*	2.53
Πώς εκτιμάτε μετά τη διεξαγωγή του e-learning μαθήματος, την ικανότητά σας να γράψετε προγράμματα σε Java;	*	3.14**

Κωδικοποίηση απαντήσεων:

5 = Επαγγελματίας... 1 = Αρχάριος

**Κωδικοποίηση απαντήσεων:

5 = Πολύ ικανός/ή... 1 = Καθόλου ικανός/ή

* ερώτηση μη διαθέσιμη στο “pre-test” ή το “post-test”

Η αυτοεκτίμηση των φοιτητών έχει σημαντικά αυξηθεί μετά την παρακολούθηση του e-learning μαθήματος, ακόμα και για το επίπεδό τους στο αντικείμενο του διαδικασιακού προγραμματισμού (από 2.3 σε 2.8), γεγονός που υποδηλώνει την πολλαπλά θετική επίδραση του. Αξιολογούν, ακόμα υψηλότερα (3.14) την ικανότητά τους να γράψουν προγράμματα χρησιμοποιώντας τη γλώσσα προγραμματισμού Java (μεταξύ του “Μέτρια ικανός/η” και του “Αρκετά ικανός/η”).

Στάση των φοιτητών απέναντι στο αντικείμενο του ΑΠ

Πίνακας 4.10. Προτιμήσεις και στάσεις των φοιτητών σχετικά με το αντικείμενο του ΑΠ (case study-2)

Ερώτηση	M.O. (Από το “pre-test”)	M.O. (Από το “post-test”)
Σας αρέσει το αντικείμενο του Προγραμματισμού (ΑΠ στο “post-test”);	4.27	3.97
Δυσκολεύεστε με τον Προγραμματισμό (ΑΠ στο “post-test”);	2.98	3.27
Αντιμετωπίζετε γενικά με φόβο τον Προγραμματισμό (ΑΠ στο “post-test”);	2.20	2.56
Θεωρείτε ότι ο Προγραμματισμός (ΑΠ στο “post-test”) απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;	3.47	3.36
Θεωρείτε ότι μάθατε Προγραμματισμό (ΑΠ στο “post-test”) με αυτά που έχετε διδαχθεί έως τώρα;	2.68	3.20
Πόσο σίγουροι/ες αισθάνεστε ότι θα επιτύχετε στις εξετάσεις του μαθήματος;	*	3.32**

Κωδικοποίηση απαντήσεων:

5 = Πολύ... 1 = Καθόλου

**Κωδικοποίηση απαντήσεων:

5 = Πολύ σίγουρος/η... 1 = Καθόλου σίγουρος/η

* ερώτηση μη διαθέσιμη στο “pre-test”

Από τον Πίνακα 4.10. μπορούν να εξαχθούν τα εξής συμπεράσματα:

- Οι δυσκολίες και οι φόβοι των φοιτητών για τον ΑΠ, φαίνεται να έχουν ελαφρά αυξηθεί μετά τη διεξαγωγή του e-learning μαθήματος. Αυτό το στοιχείο θα συζητηθεί στη συνέχεια σε συνδυασμό και με επόμενα ευρήματα.
- Μετά το τέλος του e-learning μαθήματος θεωρούν “ότι ο ΑΠ απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια” σε ελαφρά μικρότερο βαθμό από ότι πριν (από 3.47 σε 3.36). Πάντως εξακολουθούν να έχουν αυτή τη γνώμη σε ένα υψηλό ποσοστό (M.O. 3.36 στην κλίμακα από 1 έως 5). Εξήγηση γι’ αυτό αποτελεί το ότι μέσω του e-learning μαθήματος σχημάτισαν σαφή εικόνα για αυτό το αδιαμφισβητήτως απαιτητικό αντικείμενο.
- Στον ίδιο λόγο μπορεί να αποδοθεί και η μικρή πτώση (από 4.27 σε 3.97) στην ερώτηση “σας αρέσει το αντικείμενο του Προγραμματισμού (ή ΑΠ στο post-test)” καθώς επίσης και στο ότι τα 2/3 των συμμετεχόντων φοιτητών (οι προερχόμενοι από το 2^ο εξάμηνο σπουδών) ήρθαν για πρώτη φορά σε επαφή με τον ΑΠ κατά τη διάρκεια του e-learning μαθήματος. Παρ’ όλα αυτά η απάντηση ότι τους αρέσει αυτό το απαιτητικό αντικείμενο με M.O. 3.97 (στην κλίμακα από 1 έως 5) με την

πρώτη επαφή τους μαζί του, είναι πολύ θετική.

- Έκριναν ότι η γνώση που έχουν αποκτήσει σχετικά με τον ΑΠ είναι πολύ καλύτερη μετά το e-learning μάθημα (από 2.68 σε 3.20).

Το αποτέλεσμα αυτό συγκρινόμενο με το αντίστοιχο του case study-1 (αύξηση από 2.1 σε 3.8), μπορεί να δείχνει υποδεέστερο, αλλά θα πρέπει να συνεκτιμηθούν και οι διαφορετικοί παράγοντες υλοποίησης των δύο case-studies (1 και 2). Στην πρώτη μελέτη περίπτωσης είχαμε πολύ μεγάλο αριθμό διαλέξεων, ενώ στη δεύτερη η διεξαγωγή έγινε με τη μορφή e-learning μαθήματος.

- Ο Μ.Ο. 3.32, από τις απαντήσεις των φοιτητών στην ερώτηση του “post-test” “Πόσο σίγουροι/ες αισθάνεστε ότι θα επιτύχετε στις εξετάσεις του μαθήματος;” δηλώνει ένα αρκετά καλό επίπεδο αυτοπεποίθησης για τις γνώσεις που απέκτησαν και τη σιγουριά τους να αντεπεξέλθουν στις εξετάσεις.

Στις γενικές ερωτήσεις ανοικτού τύπου:

- “Τι είναι αυτό που σας άρεσε περισσότερο στον Προγραμματισμό (ΑΠ στο post-test) από τη μέχρι τώρα επαφή σας με το αντικείμενο;”

Στο “pre-test” κυριάρχησαν οι παρακάτω απαντήσεις: “η ακρίβεια που απαιτεί όταν προσπαθείς να λύσεις ένα πρόβλημα”, “ο τρόπος σκέψης που θα πρέπει να αναπτύξει κανείς είναι εντελώς διαφορετικός από τον τρόπο σκέψης για την επίλυση προβλημάτων σε οποιοδήποτε άλλο μάθημα,” “δίνει τη δυνατότητα επίλυσης προβλημάτων με δικό μου τρόπο”, “είναι μια δημιουργική διαδικασία”, “κυρίως η αίσθηση της δημιουργίας”, “η αίσθηση της ικανοποίησης που ακολουθεί την επίλυση ενός προβλήματος”, “η κατανόηση του τρόπου σκέψης του Η/Υ”, “η κατανόηση της λογικής του πως δημιουργούνται τα προγράμματα και κατ' επέκταση τα λογισμικά”, “ότι θεωρητικά οτιδήποτε μπορεί να πραγματοποιηθεί, από την πρόσθεση δύο ακεραίων μέχρι το σχεδιασμό προχωρημένων video games γραφικών”, “το αλγοριθμικό μέρος, δηλαδή το σκεπτικό επίλυσης που πρέπει να γίνεται βήμα βήμα και τίποτα να μη θεωρείται δεδομένο”, “τα συναισθήματα που έχω όταν καταφέρνω να ολοκληρώσω κυρίως ένα δύσκολο πρόγραμμα”, “το αποτέλεσμα είναι πολύ ικανοποιητικό όταν καταφέρνεις να επιλύσεις ένα πρόβλημα γεγονός που αυξάνει την αυτοπεποίθησή σου και σε κάνει να νιώθεις περήφανος για τον εαυτό σου”, “είναι μεγάλη ηθική ικανοποίηση για κάποιον το να λειτουργεί το πρόγραμμα το οποίο το έχει φτιάξει μόνος του”, “ο κατακεραματισμός του βασικού προβλήματος σε άλλα

μικρότερα διευκολύνει την κατανόηση και τελικά την επίλυση του βασικού προβλήματος”, “ότι συνδέει το χρήστη στενότερα με τον υπολογιστή”, “μου αρέσει περισσότερο που ο προγραμματισμός ξεφεύγει από τις τυπικές διαδικασίες που περιλαμβάνουν άλλα μαθήματα και σου αφήνει περισσότερα περιθώρια να σκεφτείς, να προσπαθήσεις αρκετά και να πρωτοτυπήσεις”.

Και σε αυτή τη μελέτη περίπτωσης, οι φοιτητές μιλούν αρκετά για τη χαρά της δημιουργίας που τους προσφέρει ο Προγραμματισμός και την ικανοποίηση που νιώθουν όταν υλοποιήσουν ένα πρόγραμμα που επιλύει ένα πρόβλημα.

Στο “post-test” δόθηκαν απαντήσεις όπως: “η αντιμετώπιση των προβλημάτων ως αντικείμενα νομίζω ότι είναι κάτι το οποίο βοηθάει αν μπορέσει κάποιος να μπει στη φιλοσοφία του αντικειμενοστρεφούς προγραμματισμού”, “ο διαφορετικός τρόπος σκέψης σε σχέση με το διαδικασιακό προγραμματισμό”, “πιστεύω ότι η δομή του Αντικειμενοστρεφούς Προγραμματισμού σε συνδυασμό με τη γενικότερη συνοχή της γλώσσας ήταν αυτό που μου άρεσε περισσότερο”, “η αντιστοίχιση οντοτήτων του πραγματικού κόσμου με οντότητες λογισμικού”, “το γεγονός ότι μπορείς κάποιος να φτιάξει πολύ εύχρηστα και χρήσιμα προγράμματα αρκετά εύκολα”, “η δόμηση του, δηλαδή η δυνατότητα δημιουργίας ξέχωρων κλάσεων και προγραμμάτων”, “το ότι έμαθα ένα διαφορετικό είδος προγραμματισμού το οποίο είναι πιο ρεαλιστικό”, “δίνει περισσότερες δυνατότητες από ότι ο δομημένος”, “η δυνατότητα δημιουργίας κλάσεων που κάνει πιο εύκολη τη δημιουργία ενός προγράμματος ενώ και η ύπαρξη της κληρονομικότητας που βοηθάει στην ταχύτερη ανάπτυξη του χωρίς να χρειάζεται η συγγραφή όμοιων κομματιών κώδικα πολλές φορές”, “η εντελώς διαφορετική σκοπιά αντιμετώπισης σύγχρονων προβλημάτων σε σύγκριση με τις διαδικαστικές γλώσσες, όπως είναι η C”, “Ο τρόπος με τον οποίο είναι δομημένος ο αντικειμενοστρεφής προγραμματισμός και ειδικά η μορφή της γλώσσας Java. Είναι πολύ ζωντανή και πιο προσιτή από άλλες γλώσσες προγραμματισμού, όχι αντικειμενοστρεφείς, όπως η C”, “Το γεγονός ότι δεν εξαρτάται από τις διαδικασίες αλλά από τα ίδια τα αντικείμενα. Οι κλάσεις απλοποιούν κατά πολύ τη δουλειά του προγραμματιστή και οι έτοιμες μέθοδοι της Java παρέχουν τρομερά εργαλεία. Ο πολυμορφισμός και η κληρονομικότητα μπορούν να αλλάξουν τις έτοιμες κλάσεις της java και να τις φέρουν στα μέτρα της εφαρμογής που θέλουμε να αναπτύξουμε (π.χ threads)”, “μου άρεσε ο τρόπος σκέψης που ακολουθούσαμε, δηλαδή από μια

γενική ιδέα μπορούσαμε στη συνέχεια να επεκταθούμε σε πιο ειδικά θέματα”, “η δυνατότητα να κατηγοριοποιείς τις οντότητες σε γενικότερες και ειδικότερες, χωρίς να χρειάζεται να επαναλαμβάνεις τις μεθόδους”, “Σου παρέχει πολλές ευκολίες σε σχέση με το διαδικασιακό προγραμματισμό. Είναι πιο δύσκολο στην εκμάθηση αλλά σε μεγάλα προβλήματα αυτό το είδος προγραμματισμού είναι αρκετά καλύτερο σε σχέση με το διαδικασιακό”, “ότι το πρόγραμμα μπορούσε να δημιουργηθεί πιο εύκολα από τη γλώσσα C”, “η λογική που ο αντικειμενοστρεφής προγραμματισμός μπορεί και χειρίζεται τα διάφορα προβλήματα (η χρησιμοποίηση αντικειμένων με μεθόδους που απλουστεύουν τη δημιουργία του κώδικα σε αντίθεση με τον κλασικό προγραμματισμό)”, “το ότι η γραφή των διαφόρων προγραμμάτων στη Java είναι απλή και κατανοητή”.

Οι παραπάνω απαντήσεις δείχνουν έντονα ότι οι φοιτητές έχουν αντιληφθεί σε πολύ μεγάλο βαθμό τη φιλοσοφία του ΑΠ, τη διαφορετικότητα του σε σχέση με το διαδικασιακό προγραμματισμό στην επίλυση προβλημάτων και τα οφέλη χρήσης του και συνεπώς ότι έχει επιτευχθεί ο σχετικός στόχος της προτεινόμενης διδακτικής προσέγγισης που αναφέρθηκε στην ενότητα 3.4., της ανάπτυξης Αντικειμενοστρεφούς σκέψης επίλυσης προβλημάτων.

- “Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Προγραμματισμό (ΑΠ στο post-test) από τη μέχρι τώρα επαφή σας με το αντικείμενο;”

Οι απαντήσεις στο “pre-test” περιελάμβαναν: “τα μικρά λάθη που είναι δύσκολο να βρεθούν, ώστε να γίνει compile και να τρέξει το πρόγραμμα, π.χ. το να ξεχάσεις να βάλεις αρχική τιμή σε μια μεταβλητή”, “η κατασκευή κώδικα”, “το ότι δεν έχουμε εργαστήρια”, “σε δυσκολότερα προβλήματα είναι αρκετά πιο δύσκολο να κατανοήσεις τον τρόπο με τον οποίο πρέπει να λυθεί το πρόβλημα, ή να βρεις τον κατάλληλο τρόπο να το λύσεις καθώς υπάρχουν πολλοί” “δεν μπορούσα πάντα να αντιστοιχίσω τη θεωρία με την πράξη”, “η μεταβίβαση από τον αλγόριθμο στην εγγραφή κώδικα...οι εντολές και οι ιδιοτροπίες της C”, “η δυσκολία να αποτυπώσω τις σκέψεις μου στο χαρτί, δηλαδή η ολοκληρωμένη συγγραφή του κώδικα που συνθέτει ένα πρόγραμμα”, “Θα προτιμούσα τα μαθήματα να γίνονται σε εργαστήρια διότι βαριέμαι πολύ εύκολα μια θεωρητική παράδοση. Όταν άκουσα για το e-learning χάρηκα διότι είναι έμμεσα ένα real-time περιβάλλον εκμάθησης αυτού του τόσο σημαντικού κομματιού των σπουδών μου που ονομάζεται προγραμματισμός”, “Ο

τρόπος σκέψης”, “η εκμάθηση της γλώσσας”, “Μάλλον ο τρόπος διδασκαλίας! Προσωπικά ένιωθα ότι η διδασκαλία προχωρούσε με άλματα και όχι με βήματα”, “ο ιδιόρρυθμος τρόπος σκέψης που απαιτεί”, “η έλλειψη πρακτικής εξάσκησης και η απουσία διδασκαλίας των μαθημάτων πάνω σε πραγματικούς υπολογιστές”.

Το πνεύμα των απαντήσεων και εδώ είναι ανάλογο του case study-1 (ο τρόπος διδασκαλίας, η έλλειψη εργαστηρίων, η δυσκολία κωδικοποίησης, η εφαρμογή της θεωρίας).

Στο “*post-test*” ανέφεραν: “Βρήκα δυσκολίες στην αρχή μέχρι να κατανοήσω τον τρόπο σκέψης, επειδή είναι αρκετά διαφορετικός από τη C”, “Στην κατανόηση κάποιων δύσκολων εννοιών όπως είναι η υπερφόρτωση μεθόδων”, “κληρονομικότητα”, “στα τελευταία κεφάλαια, μου δημιούργησε δυσκολίες το overloading, ο πολυμορφισμός και η κληρονομικότητα”, “ότι ήταν κάτι καινούργιο και άγνωστο”, “στην αρχή ήταν οι κατασκευαστές αλλά στη συνέχεια δεν είχα πρόβλημα. πιστεύω πιο πολύ το κάλεσμα των κλάσεων μέσα από άλλες κλάσεις όπου μετά είδα πως λύνεται και κατάλαβα και τέλος οι κλάσεις με την κληρονομικότητα με δυσκόλεψαν αρκετά”, “Η ίδια η φιλοσοφία του αντικειμενοστρεφή προγραμματισμού”, “το γεγονός ότι διαθέτει αρκετές σύνθετες έννοιες και απαιτεί μεγάλη προσπάθεια για την κατανόηση τους”, “η μη δυνατότητα μαθημάτων σε εργαστήρια”, “να συνηθίσω την καινούργια γλώσσα και οι προθεσμίες που είχαμε, παρόλο που μας έβαζαν σε ένα πρόγραμμα ώστε να παραδοθούν όλες οι εργασίες”, “Το ότι άργησα να μπω στη φιλοσοφία του e-learning, όπως φαντάζομαι και οι περισσότεροι, λόγω έλλειψης προηγούμενης πείρας”, “η ενότητα 7”, “Αντιμετώπισα ορισμένες δυσκολίες στην αρχική μου προσπάθεια να κατανοήσω κάποιες έννοιες του πολυμορφισμού όπως η δυναμική δέσμευση”, “Στην αρχή λίγο οι έννοιες κλάση-αντικείμενο”, “Η δημιουργία προγραμμάτων σε Java”, “Τα τελευταία κεφαλαία 6 & 7 ομολογώ δεν τα κατανόησα πλήρως και μάλλον πως με μπέρδεψαν λίγο...”, “Η κληρονομικότητα μου αρέσει πολύ σαν τεχνική αλλά επίσης είναι και αρκετά δύσκολη”, “Το σκεπτικό του”, “Οι δυσκολίες με τον αντικειμενοστρεφή προγραμματισμό ήταν μέχρι να κατανοήσουμε τις διάφορες έννοιες που συναντήσαμε”, “Το να ξεφύγω από την “κλασική”(διαδικασιακή) έννοια του προγραμματισμού. Θα μπορούσα να πω πως ήταν γενικά η αντικειμενοστρεφής θεώρηση”, “Στον τομέα των εργασιών που ήταν αρκετά μεγάλες, αν και περιείχαν

σημαντικό και εμπειριστατωμένο υλικό”, “Ο πολυμορφισμός”, “στην αρχή με δυσκόλεψε ο τρόπος σκέψης δημιουργίας ενός προγράμματος που είναι κάπως διαφορετικός σε σχέση με την C που μάθαμε στο πρώτο εξάμηνο, αλλά αυτό το πρόβλημα λύθηκε στην πορεία και πιστεύω ότι χωρίς την ύπαρξη αυτής της τεραστίας βοήθειας που μας παρείχε το e-learning μάθημα θα είχα αρκετά προβλήματα στην κατανόηση εννοιών και αρχών”, “ αρχικά μέχρι να ενστερνιστώ τη φιλοσοφία και τη χρήση του δυσκολεύτηκα”, “οι σχέσεις μεταξύ των κλάσεων”, “ Η όλη φιλοσοφία του αντικειμενοστρεφή προγραμματισμού σε σχέση με τη φιλοσοφία του δομημένου προγραμματισμού”, “Ο τρόπος σκέψης που έπρεπε να συλλάβω ώστε να με βοηθήσει στην επίλυση των προβλημάτων”.

Όπως μπορεί να συναχθεί, κυριαρχούν απαντήσεις που έχουν να κάνουν με τη διαφορετικότητα της φιλοσοφίας του ΑΠ (που όμως οι απαντήσεις στην παραπάνω ερώτηση του post-test “Τι είναι αυτό που σας άρεσε περισσότερο στον ΑΠ από τη μέχρι τώρα επαφή σας με το αντικείμενο;” δείχνουν μάλλον ότι πρόκειται για κάτι που τους δυσκολεύει μεν, αλλά το αντιλήφθηκαν δε), την έλλειψη πρακτικής εξάσκησης, με δυσκολία στις έννοιες των τελευταίων διδακτικών ενοτήτων, αλλά και απαντήσεις που δηλώνουν δυσκολίες που υπήρξαν αρχικά αλλά ξεπεράστηκαν στην πορεία του e-learning μαθήματος.

Συμβολή παραγόντων του τρόπου εφαρμογής της διδακτικής προσέγγισης στη μαθησιακή διαδικασία

Πίνακας 4.11. Συμβολή παραγόντων της διδακτικής προσέγγισης στη βελτίωση γνώσεων και δεξιοτήτων (case study-2)

Ερώτηση	M.O. (Από το “pre- test”)	M.O. (Από το “post- test”)
Αξιολογείστε τη σημασία που θεωρείτε ότι (θα έχουν) είχαν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας.		
Ύπαρξη των διαλέξεων στο WWW.	4.59	*
Ύπαρξη του σχετικού με το μάθημα υλικού στο WWW (Σύνδεσμος “Εγγραφα” του e-class).	4.80	4.32
Ύπαρξη μελετών περίπτωσης (case studies) (δηλαδή ολοκληρωμένων παραδειγμάτων) στο μαθησιακό υλικό.	4.79	4.68
Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.	4.10	4.10

Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.	4.50	4.49
Η επίλυση επιλεγμένων ασκήσεων στο πλαίσιο του μαθήματος.	4.74	4.42
Η εκπόνηση εκτενών εργασιών στο πλαίσιο του μαθήματος.	4.59	4.35
Η ενασχόληση με τα περιβάλλοντα που θα είναι διαθέσιμα στο e-learning μάθημα.	4.52	*
Η δομή του e-learning μαθήματος, έτσι ώστε να σε οδηγεί “από το απλό στο σύνθετο” δηλ. από απλούστερες έννοιες να σε εισάγει σταδιακά (“βήμα-βήμα”) σε δυσκολότερες.	*	4.41
Η οργάνωση της ύλης κάθε διδακτικής ενότητας του e-learning μαθήματος (με σύντομη θεωρία, παραδείγματα με κώδικα, λυμένες ασκήσεις, δραστηριότητες σε εκπαιδευτικά περιβάλλοντα).	*	4.44
Η ανάρτηση στο ηλεκτρονικό μαθησιακό περιβάλλον, προτεινόμενων λύσεων των ασκήσεων που περιέχονταν στις εργασίες που παρέδιδα.	*	4.46

Κωδικοποίηση απαντήσεων: 5 = Πολύ σημαντική... 1 = Πολύ ασήμαντη
 * ερώτηση μη διαθέσιμη στο “pre-test” ή το “post-test”

Από τον Πίνακα 4.11. συνάγονται τα εξής:

- Όλοι ο παράγοντες εκτιμήθηκαν από τους φοιτητές ότι συνέβαλαν κατά πολύ (>4.10 στην κλίμακα από 1 έως 5) στη βελτίωση των γνώσεων και των δεξιοτήτων τους. Αυτά τα αποτελέσματα δικαιώνουν και σε αυτή τη μελέτη περίπτωσης, την υιοθέτηση των συγκεκριμένων αρχών (ενότητα 3.4.) που αναφέρθηκαν στην υποενότητα 4.4.4.1., στην εκπαιδευτική φιλοσοφία της προτεινόμενης διδακτικής προσέγγισης.
- Κάποια μικρή κάμψη όπου παρατηρείται στους Μ.Ο. από το “post-test”, εξηγείται εύκολα από το γεγονός ότι οι φοιτητές αρχικά έχουν πολύ μεγάλες προσδοκίες συνήθως, ιδίως από ένα μάθημα που διεξάγεται με τρόπο διαφορετικό από τον παραδοσιακό. Παρ’ όλα αυτά οι Μ.Ο. από το “post-test” παρέμειναν πολύ υψηλοί.
- Επιπλέον σε αυτή τη μελέτη περίπτωσης, **αξιολογήσαμε** με κατάλληλες ερωτήσεις στο “post-test”, **την επιλογή της σταδιακής εισαγωγής των εννοιών “από το απλό στο σύνθετο” και τη χρήση ποικιλίας μαθησιακών πόρων για την παρουσίαση εννοιών.** Και οι δύο επιλογές αξιολογήθηκαν πολύ υψηλά (4.41 και 4.44 αντίστοιχα).
- Η αξιολόγηση για τη “δυνατότητα συζητήσεων με συμφοιτητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος” (4.1) η οποία παραμένει και σταθερή στο “post-test”, είναι υψηλότερη από αυτήν της πρώτης μελέτης περίπτωσης (case

study-1) (3.8). Αποδεικνύεται έτσι ότι οι φοιτητές σε αυτή την περίπτωση δημιούργησαν ενεργές κοινότητες μάθησης κάνοντας χρήση των διαδικτυακών τεχνολογιών και ότι επετεύχθη ο στόχος που είχε τεθεί για τις επόμενες –μετά την πρώτη– εφαρμογές της προσέγγισης και διατυπώθηκε στην υποενότητα 4.4.5. Η συνοχή και ομοιομορφία των φοιτητών, στη δεύτερη αυτή μελέτη περίπτωσης, ήταν σαφώς μεγαλύτερη (μεγάλος αριθμός φοιτητών προερχόταν από το ίδιο εξάμηνο σπουδών).

Συμβολή μαθησιακού υλικού και διδακτικής προσέγγισης

Πίνακας 4.12. Αξιολόγηση των μαθησιακών πόρων του υλικού μεμονωμένα (case study-2)

Ερώτηση	
Αξιολογήστε το βαθμό με τον οποίο οι παρακάτω μαθησιακοί πόροι συνέβαλαν στη βελτίωση των γνώσεων και των δεξιοτήτων σας.	M.O. (Από το “post-test”)
Λυμένες ασκήσεις, παραδείγματα.	4.80
Οι εργασίες που παραδίδαμε.	4.56
Η μελέτη περίπτωσης (case study).	4.36
Θεωρία.	4.33
Οι δραστηριότητες στο περιβάλλον BlueJ (που περιέχονταν στο υλικό και πραγματοποιούσα κατά τη μελέτη μου).	4.20
Τα αρχεία κώδικα Java που περιλαμβάνονται στο υλικό.	4.12
Το ολοκληρωμένο παράδειγμα που παρέδωσα (εφαρμογή βιβλιοπωλείου).	4.12
Τα υλοποιημένα projects στο περιβάλλον BlueJ.	4.11
Οι οδηγίες μελέτης της κάθε διδακτικής ενότητας.	3.89
Το υλικό που προήλθε από τις συζητήσεις στην “Περιοχή συζητήσεων” (forum) του e-learning μαθήματος.	3.73
Οι συναντήσεις του e-learning μαθήματος και οι διαλέξεις σε αυτές τις συναντήσεις.	3.30

Κωδικοποίηση απαντήσεων: 5 = Πολύ σημαντικός... 1 = Πολύ ασήμαντος

Από τον παραπάνω Πίνακα μπορούν να εξαχθούν τα ακόλουθα συμπεράσματα:

- Οι φοιτητές επιδοκίμασαν ιδιαίτερα τις λυμένες ασκήσεις και τα παραδείγματα του υλικού (ο μαθησιακός αυτός πόρος παραμένει πρώτος στην αξιολόγηση και σε αυτή τη μελέτη περίπτωσης).
- Σημαντικός παράγων είναι οι ερωτήσεις-ασκήσεις (εργασίες) που οι φοιτητές εκπόνησαν και παρέδωσαν δικαιώνοντας πάλι την αρχή (ενότητα 3.4.) **“Υλοποίηση της διαδικασίας ενεργούς δόμησης της γνώσης με τη συνεχή εμπλοκή του εκπαιδευόμενου σε κατάλληλα σχεδιασμένες δραστηριότητες**

ποικίλου είδους” (Hadjerrouit, 1999; Fleury, 2000; Fleury, 2001; Bergin, 2006a, pattern “Active Student”).

- Οι επόμενοι σημαντικότεροι μαθησιακοί πόροι είναι η μελέτη περίπτωσης (case study), η θεωρία, οι δραστηριότητες στο περιβάλλον BlueJ, ο πηγαίος κώδικας των παραδειγμάτων και των ασκήσεων που περιέχεται στο υλικό, η μεσαίου μεγέθους εφαρμογή που εκπόνησαν και παρέδωσαν (υψηλή αξιολόγηση με Μ.Ο. 4.12 που δικαιώνει την απόφαση να ενσωματωθεί στην προτεινόμενη διδακτική προσέγγιση μια τέτοιου είδους εργασία) και τα υλοποιημένα projects στο περιβάλλον BlueJ ακολουθούν σε συμβολή στη βελτίωση των γνώσεων και των δεξιοτήτων των φοιτητών.
- Ακολουθούν ο οδηγός μελέτης κάθε διδακτικής ενότητας, το υλικό των ασύγχρονων συζητήσεων στο on-line μαθησιακό περιβάλλον του μαθήματος και τελευταίος σε συμβολή πόρος είναι οι δια ζώσης συναντήσεις του e-learning μαθήματος.

Συμπερασματικά όλοι οι μαθησιακοί πόροι αξιολογήθηκαν αρκετά υψηλά, ιδιαίτερα αυτοί καθαυτοί οι πόροι του μαθησιακού υλικού και σε μικρότερο βαθμό πόροι που αφορούσαν στον τρόπο εφαρμογής της προσέγγισης (π.χ. συζητήσεις, δια ζώσης συναντήσεις).

Πίνακας 4.13. Αξιολόγηση του μαθησιακού υλικού στο σύνολό του και της διδακτικής προσέγγισης (case study-2)

Ερώτηση	Μ.Ο. (Από το “post-test”)
Το σεμινάριο με βοήθησε να ξεπεράσω αδυναμίες μου σε σχέση με τον ΑΠ.	3.92
Το σεμινάριο με βοήθησε να ξεπεράσω φοβίες που είχα σε σχέση με τον ΑΠ.	3.60
Πώς θα χαρακτηρίζατε το περιεχόμενο του μαθησιακού υλικού; *	4.08
Το περιεχόμενο του μαθησιακού υλικού επιτυγχάνει τους στόχους που αναφέρονται στην αρχή κάθε διδακτικής ενότητας.	4.02
Οι ερωτήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους στόχους εκμάθησης.	4.14
Οι ασκήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους στόχους εκμάθησης.	4.12

Κωδικοποίηση απαντήσεων: 5 = Συμφωνώ απόλυτα... 1 = Διαφωνώ απόλυτα

*Κωδικοποίηση απαντήσεων: 5 = Πολύ ενδιαφέρον, 4 = Ενδιαφέρον, 3 = Αδιάφορο, 2 = Κουραστικό, 1 = Δυσνόητο

Από τον Πίνακα 4.13. συμπεραίνονται τα εξής:

- Η διδακτική προσέγγιση που υιοθετήθηκε κατά τη διάρκεια του e-learning μαθήματος και το προσφερόμενο μαθησιακό υλικό, παρείχε σημαντική βοήθεια στους φοιτητές για να ξεπεράσουν αδυναμίες και φόβους τους σχετικά με τον ΑΠ. Παρότι στον Πίνακα 4.10. φαίνεται ότι οι φόβοι των φοιτητών σχετικά με το αντικείμενο του ΑΠ αυξήθηκαν ελαφρά μετά το τέλος του e-learning μαθήματος (από 2.20 σε 2.56), εδώ φαίνεται ότι εκτιμούν αρκετά τη συμβολή του e-learning μαθήματος στο να ξεπεράσουν φοβίες σε σχέση με τον ΑΠ (3.60).
- Η ποιότητα του μαθησιακού υλικού αξιολογήθηκε πολύ υψηλά (4.08 κατά Μ.Ο. στην κλίμακα από 1 έως 5). Η καταλληλότητα του μαθησιακού υλικού και των ερωτήσεων-ασκήσεων για την επίτευξη των μαθησιακών στόχων, κρίθηκαν επίσης πολύ θετικά, αποδεικνύοντας και σε αυτή τη μελέτη περίπτωσης την πολύ επαρκή ικανοποίηση της αρχής που υιοθετήθηκε στην προτεινόμενη διδακτική προσέγγιση (ενότητα 3.4.), “**ιδιαίτερη προσοχή στην ποιότητα του μαθησιακού υλικού που σχεδιάζεται**” (Kölling and Rosenberg, 2001b; Buck and Stucki, 2000; Fleury, 2000; Bergin, 2006a, pattern “Quality is Job One”).

Μαθησιακά εργαλεία

Οι απαντήσεις των φοιτητών σχετικά με τα σημαντικά χαρακτηριστικά του περιβάλλοντος BlueJ, φαίνονται στον παρακάτω Πίνακα.

Πίνακας 4.14. Αξιολόγηση της ευχρηστίας του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ (case study-2)

Ερώτηση	Αριθμός φοιτητών που επέλεξαν το χαρακτηριστικό (max 66)
Ποια από τα παρακάτω χαρακτηριστικά του BlueJ σας βοήθησαν στη μαθησιακή διαδικασία; Επιλέξτε αυτά που νομίζετε.	
Οι κλάσεις και τα αντικείμενα αποκτούν οπτική υπόσταση.	55
Εύκολη δημιουργία αντικειμένων και οπτικοποίησης τους.	54
Ευκολία χρήσης του περιβάλλοντος.	51
Εύκολος έλεγχος των τιμών των ιδιοχαρακτηριστικών των αντικειμένων.	51
Εύκολος πειραματισμός με άμεσο αποτέλεσμα (feedback).	47
Επιτρέπεται να αλληλεπιδράς απευθείας με τις κλάσεις και τα αντικείμενα.	46
Εύκολη εκτέλεση των μεθόδων και έλεγχός τους (εύκολο πέρασμα παραμέτρων και επιστροφή αποτελέσματος), χωρίς να	44

χρειάζεται να γράφεις κώδικα γι' αυτό το σκοπό.	
Εύκολο compilation με επισήμανση των συντακτικών λαθών απευθείας στο παράθυρο του editor με μήνυμα και επισήμανση της γραμμής κώδικα με το λάθος.	42
Διαθέτει όψη “implementation” και όψη “interface” για τις κλάσεις.	25
Διαθέτει debugger εύκολο στη χρήση.	21

Και σε αυτή τη μελέτη περίπτωσης, τα “δυνατά” χαρακτηριστικά του BlueJ, ήταν η οπτικοποίηση των κλάσεων και των αντικειμένων, η απλότητά του και η δυνατότητα αλληλεπίδρασης που προσφέρει (Kölling et al., 2003; Kölling and Rosenberg, 2001b).

- Στην ερώτηση “Σε τι κυρίως πιστεύετε ότι σας βοήθησε το περιβάλλον BlueJ; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ότι άλλο εσείς νομίζετε)”

Οι φοιτητές απάντησαν:

- ◆ Στην κατανόηση των σχέσεων μεταξύ αντικειμένων και κλάσεων κυρίως.
- ◆ Με βοήθησε να κατανοήσω τις έννοιες: κλάση, υπερκλάση, αντικείμενα, μέθοδοι και προσδιοριστές ορατότητας.
- ◆ Να κατανοήσουμε στην πράξη έννοιες όπως: δημιουργία αντικειμένου – κλάσης, ιεραρχία κλάσεων και κληρονομικότητα.
- ◆ Με βοήθησε στο να κατανοήσω τις overloading και overriding μεθόδους, την έννοια της κλάσης και του αντικείμενου.
- ◆ Αποσαφήνισα έννοιες όπως η κληρονομικότητα.
- ◆ Το περιβάλλον BlueJ βοήθησε πολύ στην κατανόηση της σχέσης κλάσεων-αντικειμένων, της κληρονομικότητας, των κατασκευαστών, του πολυμορφισμού.
- ◆ Κυρίως βοήθησε στον πιο ξεκάθαρο διαχωρισμό των εννοιών implementation και interface, καθώς και τις σχέσεις ανάμεσα στις κλάσεις. Επίσης, στην κατανόηση της κατάστασης ενός αντικειμένου κάθε στιγμή.
- ◆ Το λάτρεψα γιατί είναι πολύ εύκολο στη χρήση και παράλληλα δε σου στερεί δυνατότητες. Κυρίως βοηθάει στην κατανόηση των σχέσεων μεταξύ των κλάσεων.

- ◆ Την έννοια του "is a" και την έννοια uses. Επίσης η γραφική απεικόνιση των κλάσεων που παρέχει το BlueJ βοήθησε πάρα πολύ στη συνολική αντίληψη μίας εφαρμογής με πολλές κλάσεις (εφαρμογή που παραδώσαμε).
- ◆ Να κατανοήσω περισσότερο την εκτέλεση προγραμμάτων και την έξοδο που έχει κάθε πρόγραμμα.
- ◆ Θεωρώ το BlueJ από τους καλύτερους compiler για Java.
- ◆ Σε όλα, αφού διευκόλυνε στον εντοπισμό των λαθών σε κάθε τμήμα κώδικα.
- ◆ Με την προσπάθεια μεταγλώττισης παρουσίαζε ποιο ήταν το λάθος στο πρόγραμμα και πού βρισκόταν.
- ◆ Η χρήση του compiler βοήθησε πολύ στο να κατανοήσουμε τη λειτουργία της Java μέσα από τα λάθη μας.
- ◆ Το BlueJ βοήθησε στην κατανόηση αρχικά των αντικειμένων και των μεθόδων, αλλά και στη συνέχεια σε κάθε απορία σε οποιοδήποτε θέμα που προέκυπτε μέσα από τη θεωρία, μπορούσε να λυθεί από το BlueJ. Επίσης ήταν ιδιαίτερα χρήσιμο επειδή μπορούσες να δημιουργήσεις προγράμματα χωρίς μέθοδο main.
- ◆ Με βοήθησε να καταλάβω ποια είναι η ουσιαστική λειτουργία της Java και να την κάνω στην πράξη.
- ◆ Στο να πειραματιστούμε με προγράμματα. Αυτό το περιβάλλον είναι πολύ απλό και κατανοητό.
- ◆ Στο να κατανοήσω τη μετατροπή θεωρίας σε πράξη.
- ◆ Στην αρχή διαβάζοντας τη θεωρία δεν καταλάβαινα και πολλά. Με το περιβάλλον BlueJ –χάρη στην ευκολία χρήσης του– κατάλαβα ακριβώς τι ήταν εκείνα που διάβαζα και με βοήθησε να ξεκινήσω καλά.
- ◆ Μας βοήθησε στην κατανόηση του πώς υλοποιούνται τα προγράμματα. Είναι ένα εύκολο και κατανοητό περιβάλλον.

Αξιολόγηση δυσκολίας των αντικειμενοστρεφών εννοιών – αρχών

Πίνακας 4.15. Αξιολόγηση της δυσκολίας αντικειμενοστρεφών εννοιών – αρχών (case study-2)

Ερώτηση	
Αξιολογήστε σε τι επίπεδο σας δυσκόλεψαν οι παρακάτω έννοιες/θέματα του ΑΠ.	M.O. (Από το "post-test")
Κλάση - Αντικείμενο - Ιδιότητες -Μέθοδοι	1.27

Προσδιοριστές ορατότητας (public, private, protected)	1.58
Δημιουργία αντικειμένων - Κατανομή μνήμης	1.62
Αναφορές σε αντικείμενα (references)	1.77
Μεταβλητές και μέθοδοι κλάσης (static members)	1.86
Μέθοδος Κατασκευαστή (constructor)	1.91
Σχέση “είναι” (“is a”)	2.00
Υπερφόρτωση μεθόδων (methods overloading)	2.02
Δημιουργία ιεραρχιών κλάσεων	2.05
Κελυφοποίηση - απόκρυψη πληροφορίας (encapsulation– information hiding) – Διαχωρισμός συμπεριφοράς - υλοποίησης	2.06
Κληρονόμηση ιδιοτήτων – μεθόδων στις Ιεραρχίες κλάσεων (inheritance)	2.06
Υπερίσχυση μεθόδων (methods overriding)	2.20
Πολυμορφισμός (Polymorphism)	2.44
Δυναμική κλήση μεθόδου ή Δυναμική δέσμευση (Dynamic Binding)	2.58

Κωδικοποίηση απαντήσεων: 3 = Μεγάλη δυσκολία... 1 = Καθόλου δυσκολία

Οι περισσότερο δύσκολες έννοιες-αρχές για τους φοιτητές, όπως προκύπτει από τον παραπάνω πίνακα είναι αυτές που περιέχονται στην 7^η και τελευταία διδακτική ενότητα της προτεινόμενης διδακτικής προσέγγισης, εύρημα που συμφωνεί και με τις απαντήσεις κάποιων φοιτητών στην ανοικτού τύπου ερώτηση του “post-test”, “Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον ΑΠ από τη μέχρι τώρα επαφή σας με το αντικείμενο;” που αναφέρθηκε παραπάνω. Κατά πόσον αντεπεξήλθαν στη δυσκολία αυτών των εννοιών θα συζητηθεί στη συνέχεια, στην ανάλυση των εργασιών που παρέδωσαν.

Ποιοτικά αποτελέσματα

Στις ανοικτού τύπου ερωτήσεις του ερωτηματολογίου μετά-αξιολόγησης δόθηκαν απαντήσεις που παρατίθενται στη συνέχεια.

- **Ερώτηση:** Τι σας άρεσε περισσότερο στο μαθησιακό υλικό;
 - ♦ Η θεωρία, τα λυμένα παραδείγματα και οι πολλές εφαρμογές μέσω των εργασιών που με βοήθησαν στην κατανόηση του μαθήματος.
 - ♦ Που έχει συγκεκριμένα παραδείγματα και υλοποιημένα προγράμματα. Έτσι είναι πιο εύκολο να κατανοήσουμε τη θεωρία.
 - ♦ Η θεωρία στην οποία αναφέρονταν πολύ καλά τα βασικά κάθε ενότητας.

- ◆ Εφάρμοσε στην πράξη κάτι που ως τώρα ακουγόταν μόνο στη θεωρία και επίσης βοήθησε στην καλύτερη οργάνωση της μελέτης για την Java.
- ◆ Τα κομμάτια της θεωρίας που είχαν και προγράμματα ως παραδείγματα και τελικά έδιναν και την έξοδο του προγράμματος. Επιπλέον, τα αρχεία Power Point.
- ◆ Η διαδικασία με την οποία γινόταν το μάθημα. Ήταν κάτι πρωτότυπο.
- ◆ Η οργάνωση και η αναλυτική καθοδήγηση του υλικού.
- ◆ Το γεγονός ότι οδηγεί από το απλό στο σύνθετο.
- ◆ Η σχέση ανάμεσα στη θεωρία και τις ασκήσεις, καθώς και η σειρά με την οποία δινόταν η ύλη, οι οποίοι δύο παράγοντες οδήγησαν σε σταδιακό χτίσιμο των γνώσεων.
- ◆ Ο τρόπος οργάνωσης του μαθήματος.
- ◆ Οι ασκήσεις όπου έπρεπε να βρούμε το λάθος σε ένα κομμάτι κώδικα.
- ◆ Ότι ήταν αρκετά πλήρες και οργανωμένο.
- ◆ Περισσότερο μου άρεσε η ύπαρξη παραδειγμάτων –και το ολοκληρωμένο παράδειγμα της ενότητας 8– και το γεγονός πως οι ερωτήσεις και οι ασκήσεις ήταν πολύ κοντά στη θεωρία και σε βοηθούσαν βήμα προς βήμα να καταλάβεις και τα πιο δύσκολα κομμάτια.
- ◆ Το ότι συνδύαζε θεωρία και πράξη.
- ◆ Η μελέτη έτοιμων προγραμμάτων.
- ◆ Το πλούσιο υλικό τόσο σε θεωρία, όσο και σε ασκήσεις.
- ◆ Η δυνατότητα που μας δινόταν ως σπουδαστές, να επικοινωνούμε ή να ενημερωνόμαστε κ.ά. για το μάθημα ακόμα κι αν χρειαζόταν την ώρα που το θέλαμε, να βρισκόμαστε χιλιόμετρα μακριά από το περιβάλλον του πανεπιστημίου.
- ◆ Μου αρέσει περισσότερο η δυνατότητα που είχαμε όποια στιγμή θέλαμε και επικοινωνούσαμε είτε με την υπεύθυνο του e-learning είτε με τους συμφοιτητές μας για τυχόν απορίες.
 - ◆ Ο τρόπος με τον οποίο ήταν δομημένο το υλικό.
 - ◆ Το case study.
 - ◆ Το κεφάλαιο που αναφερόταν στην τεχνική της κληρονομικότητας.
 - ◆ Οι σημειώσεις για τη θεωρία κάθε ενότητας γιατί ήταν πολύ κατανοητές.

- ◆ Τα απτά παραδείγματα (π.χ. γάτα, σκύλος), Ο φιλικός τρόπος γραφής, Η συγκεκριμένη δομή (τόρα μαθαίνουμε αυτό...), Το project bookstore και η δυνατότητα που μας έδωσε να συνθέσουμε τις γνώσεις μας.
- ◆ Η δομή του που ακολουθούσε εξελικτική πορεία.
- ◆ Ήταν όλα συγκεντρωμένα και αυτό διευκόλυνε πολύ. Σε αντίθετη περίπτωση θα χρειαζόταν περισσότερος χρόνος για την εκμάθηση της ίδιας ποσότητας ύλης.
- ◆ Η εισαγωγή στις έννοιες της Java ξεκινώντας από πολύ απλά πράγματα κ με σχετικά αργό ρυθμό ώστε να τα κατανοήσουμε σε βάθος.
- ◆ Τα παραδείγματα προγραμμάτων τα οποία κατά την άποψη μου σε βάζουν μέσα στη φιλοσοφία και σε εξοικειώνουν με το μάθημα.
- ◆ Ο τρόπος παρουσίασης του. Υπήρχαν αρχεία κειμένου αλλά και power point που βοηθούσαν οπτικά.
- ◆ Το γεγονός ότι οι διδακτικές ενότητες ξεκινούσαν από εύκολα θέματα και βήμα-βήμα πήγαιναν σε δυσκολότερα.
- ◆ Μου άρεσε που το μάθημα γινόταν μέσω του διαδικτύου και των συνεδριών.
- ◆ Η δομή του που από την αρχή, βήμα-βήμα παρείχε τη νέα γνώση.
- ◆ Το θεωρητικό μέρος το οποίο δεν είχε "άχρηστες" πληροφορίες, οι οδηγίες για το πως θα μελετήσω και τα πολλά παραδείγματα.
- ◆ Μου άρεσε το γεγονός ότι υπήρχαν πολλά λυμένα προβλήματα τα οποία μας καθοδηγούσαν προς τη λύση των ασκήσεων που είχαμε και μας βοηθούσαν να κατανοήσουμε περισσότερο, καλύτερα αυτά που διαβάζαμε στη θεωρία. Ήταν ευκολονόητα γραμμένα και για μένα που δεν ήξερα τίποτα για το αντικείμενο έμαθα πάρα πολλά.
- ◆ Το ότι είναι αρκετά σαφείς οι σημειώσεις.

Να σημειωθεί εδώ, η αναφορά των φοιτητών στο τμήμα του υλικού στο οποίο έγινε χρήση δυναμικής "οπτικοποίησης" με λογισμικό προσομοίωσης (PowerPoint), που επελέγη για την αντιμετώπιση συγκεκριμένων μαθησιακών δυσκολιών (υποενότητα 3.7.2.).

- **Ερώτηση:** Τι δε σας άρεσε;
 - ◆ Μου ήταν δύσκολο, επειδή δεν έχω internet, να μπαίνω καθημερινά στο site να βλέπω αν υπάρχουν ανακοινώσεις.

- ◆ Δε μου άρεσε η επικοινωνία μας στο e-learning γιατί λίγες φορές μπόρεσα να καταλάβω τις απορίες και τις απαντήσεις που δόθηκαν. Δικές μου απορίες ίσως να μην κατάφερα να λύσω.
- ◆ Για κάποιες από τις εργασίες ήταν μικρό το χρονικό διάστημα παράδοσης.
- ◆ Πιστεύω ότι εκτός από τις συνεδρίες που γίνονταν θα έπρεπε να υπάρχουν και κάποιες ώρες εργαστήρια έτσι ώστε να γίνεται καλύτερη κατανόηση του αντικειμένου.
- ◆ Νομίζω ότι σε κάποια σημεία ζητούνταν πράγματα που δεν μπορούσαμε να αντιμετωπίσουμε από μόνοι μας, αν και αυτά λύθηκαν από την επικοινωνία στο forum.
- ◆ Η ασάφεια σε μερικά κεφάλαια στη θεωρία.
- ◆ Θα ήθελα περισσότερα παραδείγματα στη θεωρία. Οι ασκήσεις θα ήθελα να έχουν μεγαλύτερη ποικιλία και το χρονικό περιθώριο για την παράδοση των εργασιών να ήταν μεγαλύτερο.
- ◆ Το γεγονός ότι σε πολλά σημεία η θεωρία είναι γραμμένη συνοπτικά, εντάσσει το φοιτητή απότομα στο γνωστικό αντικείμενο του κεφαλαίου καθώς και το γεγονός ότι τα παραδείγματα πολλές φορές δεν επαρκούν για την κατανόηση της θεωρίας.
- ◆ Δε μου άρεσαν οι τελευταίες ενότητες διότι αντιμετώπισα δυσκολία στην κατανόηση τους (Dynamic Binding-Polymorphism).
- ◆ Το ότι δεν υπήρχε συγχρονισμός του e-learning με το μάθημα της τάξης.
- ◆ Το άγχος της τελευταίας στιγμής...παράδοση εργασιών.
- ◆ Στην αρχή μου ξένισε σαν ιδέα και πιστεύω πως δεν μπορεί να αντικαταστήσει τα εργαστήρια που δεν υπάρχουν.
- ◆ Η κατασκευή διαγραμμάτων σε πολλές από τις ερωτήσεις και τις ασκήσεις.
- ◆ Το ότι δε με έμαθε να γράφω προγράμματα. Θεωρώ πως παραμένει πολύ θεωρητικό όπως και οι διαλέξεις του καθηγητή.
- ◆ Προσωπικά πιστεύω, πως χρειαζόταν μεγαλύτερη ποικιλία κυρίως ως προς τα παραδείγματα, τα οποία παίζουν πολύ σημαντικό ρόλο σε για εμένα προσωπικά.
- ◆ Δε μου άρεσαν οι συγκεντρώσεις που έγιναν κατά τη διάρκεια του μαθήματος. Δεν υπήρχαν ουσιώδεις ερωτήσεις και γενικά ενδιαφέρον από τους συμμαθητές μου, βάζοντας και τον εαυτό μου μέσα σ' αυτούς.

- ◆ Η μεγάλη δυσκολία από τη μετάβαση από το ένα κομμάτι στο άλλο. Η απότομη εισαγωγή σε νέες έννοιες.
- ◆ Πολύ συμπυκνωμένη γνώση, θα μπορούσαν οι εργασίες να περιλάμβαναν μία ενότητα κάθε φορά και στην οποία θα μπορούσαμε να εμβαθύνουμε περισσότερο.
- ◆ Το ότι δε συμπεριλήφθηκε στις διδακτικές ενότητες και μια η οποία να είχε ως αντικείμενο την έννοια των εξαιρέσεων.
- ◆ Οι παραπομπές στο java documentation για πράγματα τα οποία δεν υπήρχαν στη θεωρία.
- ◆ Ορισμένα σημεία της θεωρίας θεωρώ ότι ήταν ελλιπή καθώς χρειαζόταν να διαβάσω πολλές φορές κάποιο κομμάτι και να ανατρέξω στα παραδείγματα παρακάτω για να το αντιληφθώ.
- ◆ Το γεγονός ότι θεωρητικά θα ήταν ικανό να μην είχαμε συναντήσει ποτέ τον ή την καθηγήτρια που μας έκανε το μάθημα. τα άτομα που δε διαθέτουν πρόσβαση στο Διαδίκτυο δεν έχουν την ικανότητα "ζωντανής" παρακολούθησης του, καθώς επίσης και την παράδοση των εργασιών. Το μάθημα είναι εντελώς απρόσωπο πράγμα που δεν πρέπει να συμβαίνει σε κανένα μάθημα του πανεπιστήμιου εφόσον αυτά πρέπει να μας καλλιεργούν τη συνεργασία και την αμοιβαία βοήθεια μεταξύ καθηγητών και φοιτητών.
- ◆ Κάπου στο τέλος προχώρησε κάπως απότομα χωρίς λεπτομερή παραδείγματα.
- ◆ Ίσως το μέγεθος των ασκήσεων που είχαμε για παράδοση.
- ◆ Δε μου άρεσε που δεν υπήρχαν πολλά παραδείγματα με εικόνες από το πρόγραμμα BlueJ.
- ◆ Το ότι δεν είχαμε πρόσωπο με πρόσωπο πολύ συχνές συναντήσεις με το διδάσκοντα καθηγητή.

Προκύπτει από τα παραπάνω, η ανάγκη για πρακτική εξάσκηση σε εργαστήρια, η πρόταση για καλύτερη αξιοποίηση του *forum* και των συναντήσεων στην τάξη (που οι ίδιοι οι φοιτητές ομολογούν ότι δεν αξιοποίησαν αρκετά), η ανάγκη των φοιτητών για “ζωντανή” επικοινωνία με τους διδάσκοντες, η πρόταση να συμπεριληφθούν στο υλικό οι εξαιρέσεις (*exceptions*), η δυσκολία κατασκευής των διαγραμμάτων (όπως αναφέρθηκε στη δεύτερη αυτή μελέτη περίπτωσης δε χρησιμοποιήθηκε κάποιο *design CASE tool* γι’ αυτό το σκοπό) και η απαίτηση κάποιων φοιτητών για μεγαλύτερη άνεση χρόνου για τη μελέτη του υλικού και την παράδοση εργασιών.

- **Ερώτηση:** Τι θα μπορούσε να σας βοηθήσει στην κατανόηση του γνωστικού αντικειμένου και δεν υπάρχει στο υλικό ή στην οργάνωση του e-learning μαθήματος;
 - ◆ Περισσότερα λυμένα παραδείγματα.
 - ◆ Δεν πιστεύω ότι υπάρχει κάτι περαιτέρω εκτός από τα εργαστήρια που προαναφέρθηκαν για λίγο περισσότερη πρακτική εξάσκηση.
 - ◆ Πιο σύνθετα παραδείγματα
 - ◆ Θα ήθελα σε κάθε ενότητα του μαθήματος να υπάρχουν παραδείγματα προγραμμάτων, για τα οποία να εξηγείται κάθε εντολή τους, τι κάνει, γραμμή προς γραμμή.
 - ◆ Θα προτιμούσα να είχαμε περισσότερες συνεδρίες κατά τη διάρκεια του εξαμήνου.
 - ◆ Η ύπαρξη αναλυτικής θεωρίας και επαρκών παραδειγμάτων για τη σωστή κατανόηση του αντικειμένου κάθε κεφαλαίου.
 - ◆ Να εξηγούνται καλύτερα στην αρχή οι έννοιες και οι εντολές της Java.
 - ◆ Εργαστήρια.
 - ◆ Μια αναλυτική εισαγωγή στη Java, τόσο στη δημιουργία προγράμματος όσο και στο χειρισμό των περιβαλλόντων που χρησιμοποιούνται.
 - ◆ Η άμεση επαφή και επικοινωνία.
 - ◆ Περισσότερη πρακτική εξάσκηση.
 - ◆ Περισσότερα αρχεία Java.
 - ◆ Ίσως η μελέτη έτοιμου κώδικα από σύνθετα προγράμματα.
 - ◆ Μπορώ να πω ότι με κάλυψε. Πάντως αν θα πρότεινα και κάτι άλλο θα ήταν στο τέλος, σαν μια ξεχωριστή ενότητα, ένα σετ προτεινομένων ασκήσεων για την εξάσκηση μας, οι οποίες θα ήταν από όλες τις διδακτικές ενότητες και θα είχαν ως γενικότερο σκοπό την εφαρμογή όσων μάθαμε.
 - ◆ Όλα όσα βρίσκονταν στο υλικό ήταν πολύ χρήσιμα και με βοήθησαν παρά πολύ. Χρήσιμο θα ήταν για όσους ενδιαφέρονταν να είχατε συμπεριλάβει κάποια e-books σε μορφή pdf σχετικά με την Java για περαιτέρω διάβασμα έστω και στην αγγλική γλώσσα.
 - ◆ Κάποια από τις εργασίες, καλό θα ήταν να είναι ομαδική.

- ◆ Πιστεύω θα έπρεπε να υπάρχουν περισσότερα παραδείγματα ολοκληρωμένων προγραμμάτων τα οποία θα είχαν κάποιο αυξημένο επίπεδο δυσκολίας από τα άλλα.
- ◆ Ζωντανά παραδείγματα τύπου βίντεο και multimedia τα οποία θα μας εξηγούσαν "ζωντανά" βήμα-βήμα τι θα έπρεπε να κάνουμε.
- ◆ Παραδείγματα με εικόνες. Όπως θα γινόταν αν κάναμε μάθημα μέσα στην τάξη και με τη χρήση του βιντεοπροβολέα θα βλέπαμε όλα αυτά τα οποία λέτε μέσα στη θεωρία με τα αρχεία pdf.
- ◆ Περισσότερα ολοκληρωμένα παραδείγματα. Πιστεύω επιπλέον ότι ομαδικές εργασίες 2-3 ατόμων σε συνδυασμό με την περιοχή συζητήσεων θα μπορούσε να έχει καλύτερα αποτελέσματα ως προς την κατανόηση του μαθησιακού υλικού.

Προκύπτει από τα παραπάνω η ανάγκη, για πρακτική εξάσκηση σε εργαστήρια (για άλλη μια φορά), για *εξήγηση λεπτομερώς των εντολών του κώδικα και της χρήσης των περιβαλλόντων* (κάτι που δεν παρατηρήθηκε στην πρώτη μελέτη περίπτωσης, αφού εκεί υπήρχαν πολλές διαλέξεις) και ξανά η ανάγκη των φοιτητών για “ζωντανή” επικοινωνία με τους διδάσκοντες. Προτείνεται η ενσωμάτωση και άλλων ασκήσεων στο μαθησιακό υλικό για περαιτέρω εξάσκηση, ομαδικών εργασιών (*οι φοιτητές φαίνεται να εκτιμούν τη σημασία της δημιουργίας κοινότητας πρακτικών (community of practices)* (ενότητα 3.5.)) και συνθετότερων προγραμμάτων.

- **Ερώτηση:** Ποιοι μαθησιακοί πόροι νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;
 - ◆ Λυμένες ασκήσεις.
 - ◆ Η θεωρία πρέπει να είναι πιο συνοπτική και περιεκτική.
 - ◆ Θεωρία, παραδείγματα.
 - ◆ Το αντίστοιχο βιβλίο του πανεπιστημίου.
 - ◆ Χρειάζεται σαφέστερη διατύπωση των ζητούμενων στις εργασίες.
 - ◆ Η θεωρία πρέπει να ελαττωθεί και να υπάρχει μόνο για υποστήριξη των παραδειγμάτων.
 - ◆ Οι σημειώσεις θα μπορούσαν να ήταν πιο απλές και αναλυτικές.
 - ◆ Τα παραδείγματα και οι λυμένες ασκήσεις, οι συναντήσεις του e-learning μαθήματος.

- ◆ Τα τελευταία κεφάλαια που ήταν λίγο πιο δύσκολα και μερικά πράγματα δεν μπορούσα να τα καταλάβω πολύ εύκολα.
- ◆ Το φόρουμ να λειτουργεί πιο αποτελεσματικά.
- ◆ Το προτελευταίο κομμάτι του μαθησιακού υλικού, δηλαδή το 7ο.
- ◆ Στο θέμα των ασκήσεων. Θα πρέπει να είναι ευκολότερες και ο βαθμός δυσκολίας τους να αυξάνεται με πολύ αργό ρυθμό.
- ◆ Το καλύτερο θα ήταν να γινόταν αυτό το μάθημα στην τάξη και ακόμα καλύτερο θα ήταν το τμήμα αυτό να είχε εργαστήρια όπως σε άλλες σχολές που κάθονται όλοι οι μαθητές μαζί με τον καθηγητή μπροστά στους υπολογιστές και προγραμματίζουν.
- ◆ Η 8η ενότητα.
- ◆ Τα υλοποιημένα projects στο περιβάλλον BlueJ.
- ◆ Οι εργασίες και λίγο το θεωρητικό μέρος.

Να σημειωθεί, ότι ο φοιτητής απαντούν επηρεασμένοι από το φόρτο της παράδοσης των εργασιών του e-learning μαθήματος. Φαίνεται να υπάρχει κάποια δυσκολία στην 7^η διδακτική ενότητα και αυτό συμφωνεί και με ευρήματα που αναφέρθηκαν προηγουμένως (αξιολόγηση δυσκολίας αντικειμενοστρεφών εννοιών-αρχών)

- **Ερώτηση:** Ποιες ενότητες/τμήματα του υλικού νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;
 - ◆ Δυναμική κλήση μεθόδου ή Δυναμική δέσμευση (Dynamic Binding)
 - Πολυμορφισμός (Polymorphism)
 - Υπερίσχυση μεθόδων (methods overriding)
 - Υπερφόρτωση μεθόδων (methods overloading)
 - Κελυφοποίηση - απόκρυψη πληροφορίας (encapsulation-information hiding)
 - Διαχωρισμός συμπεριφοράς - υλοποίησης
 - ◆ Μέθοδος Κατασκευαστή.
 - ◆ Ολοκληρωμένο παράδειγμα (case study).
 - ◆ Ένα τμήμα του υλικού που χρειάζεται ίσως κάποια μικρή βελτίωση είναι: αφηρημένες και τελικές κλάσεις και μέθοδοι – οι διασυνδέσεις ή διεπαφές.
 - ◆ Ίσως οι εισαγωγικές στον αντικειμενοστρεφή προγραμματισμό. Πιστεύω δε γίνεται αρκετά σαφές ότι πρόκειται για μια τελείως διαφορετική προσέγγιση των προβλημάτων προς επίλυση - άλλη φιλοσοφία. Γι αυτό και πολλοί

δυσκολεύονται αλλά και προσπαθούν να σκεφτούν πρώτα σε C και μετά να αποδώσουν σε Java.

- ◆ Το θέμα του packaging.
- ◆ Πιστεύω ότι το τμήμα του υλικού σε σχέση με τους πίνακες θα μπορούσε να περιέχει περισσότερα στοιχεία και μερικά παραδείγματα ακόμα.
- ◆ Όλες θα γίνονταν καλύτερα αν υπήρχαν εικόνες.

Οι έννοιες-αρχές που δυσκόλεψαν τους φοιτητές, καθορίζουν και την απάντησή τους στην παραπάνω ερώτηση. Προκύπτει η πρόταση για βελτίωση του case study, του τμήματος της 7^{ης} διδακτικής ενότητας περί αφηρημένων και τελικών κλάσεων και μεθόδων και περί διασυνδέσεων και η ανάγκη να παρουσιασθούν περισσότερα περί των πακέτων κλάσεων (packages) και των πινάκων (arrays). Να σημειωθεί ότι οι περισσότερες από αυτές τις παρατηρήσεις δεν υπήρξαν στην πρώτη μελέτη περίπτωσης, που διενεργήθηκε με πολλές διαλέξεις στην τάξη.

- **Ερώτηση:** Σε τι κυρίως πιστεύετε ότι σας βοήθησε η συστηματική χρήση της διαγραμματικής τεχνικής (διαγράμματα κλάσεων και αντικειμένων) στο μαθησιακό υλικό; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ό,τι άλλο εσείς νομίζετε)
 - ◆ Κυρίως στην κατανόηση των σχέσεων μεταξύ αντικειμένων και κλάσεων.
 - ◆ Τα διαγράμματα βοηθούν πολύ έτσι ώστε να δει και να κατανοήσει κανείς αμέσως τις μεθόδους που περιλαμβάνει μία κλάση δηλαδή τι δυνατότητες έχει να κάνει κάποια πράγματα και τα χαρακτηριστικά των αντικειμένων. Όλα αυτά φαίνονται απευθείας στα διαγράμματα οπότε βοηθούν στην κατανόηση.
 - ◆ Μας βοήθησε να κατανοήσουμε τις έννοιες του αντικειμένου και της κλάσης καθώς και την αλληλεπίδραση των κλάσεων μεταξύ τους.
 - ◆ Κυρίως στην τελευταία ενότητα του πολυμορφισμού.
 - ◆ Στο να αντιληφθώ πλήρως τη σχέση μεταξύ κλάσεων, ιδιοχαρακτηριστικών, μεθόδων και υπερκλάσεων.
 - ◆ Βοήθησε στο διαχωρισμό των εννοιών ιδιοχαρακτηριστικά-μέθοδοι.
 - ◆ Είναι πολύ σημαντική καθώς με βοήθησε να καταλάβω τις έννοιες: κλάση, υπερκλάση, αντικείμενα, μέθοδοι και προσδιοριστές ορατότητας και υπερφόρτωση μεθόδων.

- ◆ Τα διαγράμματα με βοήθησαν στην εις βάθος κατανόηση των κλάσεων και των αντικειμένων.
- ◆ Στη κατανόηση διαφοροποίησης των public και private.
- ◆ Καλύτερη κατανόηση των εννοιών κλάση-αντικείμενο, τις ιεραρχίες...
- ◆ Στη δημιουργία του κώδικα.
- ◆ Η διαγραμματική τεχνική πιστεύω ότι χρησιμεύει μέχρι την κατανόηση του τι είναι κλάση. Θα έλεγα μέχρι την 1η 2η ενότητα. Μετά μάλλον είναι κουραστική. Ειδικά όταν πρέπει να σχεδιάσουμε οι ίδιοι διαγράμματα. Το θεωρώ περιττό.
- ◆ Τα διαγράμματα κλάσεων βοήθησαν στην αντίληψη της λειτουργίας την κληρονομικότητας (ποιες μέθοδοι και ιδιοχαρακτηριστικά κληρονομούνται και τι σημαίνει αυτό για τις υποκλάσεις).
- ◆ Με έκανε να καταλάβω πολύ καλύτερα τη χρήση της κλάσης και του αντικειμένου και να αντιστοιχίζω σωστά τα ιδιοχαρακτηριστικά και τους προσδιοριστές ορατότητας.
- ◆ Η συστηματική χρήση της διαγραμματικής τεχνικής με βοήθησε αρκετά αλλά πιστεύω ότι μου ήταν αρκετά χρονοβόρα η διαδικασία σχεδιασμού των διαγραμμάτων κλάσεων στην εκπόνηση των εργασιών και θα μπορούσαμε να την αποφύγουμε.
- ◆ Με βοήθησε να καταλάβω καλύτερα τις έννοιες του αντικειμένου και της κλάσης και πως τα διάφορα ιδιοχαρακτηριστικά και μέθοδοι αναφέρονται σε αυτά.
- ◆ Ίσως την ιεραρχία των κλάσεων αλλά γενικά δε μου αρέσει η χρήση διαγραμματικών τεχνικών... δεν τη θεωρώ απαραίτητη.
- ◆ Η χρήση της διαγραμματικής τεχνικής με βοήθησε στο να καταλαβαίνω τι πρέπει να εμπεριέχω στη κάθε κλάση και να βλέπω τη σχέση της με κάποια άλλη.
- ◆ Ήταν εύκολο για εμένα. Από τη στιγμή που το είχα καταλάβει δε με βοήθησε πουθενά.
- ◆ Στο διαχωρισμό της έννοιας του ιδιοχαρακτηριστικού από τις μεθόδους.
- ◆ Μας βοήθησε να καταλάβουμε πως λειτουργούν οι κλάσεις και αυτό γιατί τα διαγράμματα κλάσεων και αντικειμένων ήταν εύκολα στην κατανόηση εκεί όπου δεν καταλάβαινες και τόσο καλά τη θεωρία.
- ◆ Δε νομίζω ότι η συγκεκριμένη τεχνική με βοήθησε κάπου.

Από τα παραπάνω προκύπτει ότι οι φοιτητές βοηθήθηκαν από τη χρήση της διαγραμματικής τεχνικής (class diagrams) και εντόπισαν πολύ καλά πώς και σε τι και συγχρόνως ότι η διαδικασία σχεδιασμού κάποιες φορές τους κούρασε και ήταν χρονοβόρα γι' αυτούς. Προκύπτει η ανάγκη της ενσωμάτωσης στην προτεινόμενη διδακτική προσέγγιση της χρήσης ενός εργαλείου σχεδίασης που θα απλοποιεί τη διαδικασία και επιπλέον θα ενισχύει και τη φάση της κωδικοποίησης με την παραγωγή “σκελετικού” κώδικα από τα διαγράμματα κλάσεων.

Οι επόμενες απαντήσεις των φοιτητών στην ίδια ερώτηση, από παρανόησή τους, αναφέρονται στη βοήθεια που τους προσέφερε η δυναμική “οπτικοποίηση” για θέματα δημιουργίας αντικειμένων – κατανομής μνήμης και η στατική σχηματική παράσταση της κατάστασης στη μνήμη (υποενότητα 3.7.2.).

- ◆ Να κατανοήσω περισσότερο τα προγράμματα στην εκτέλεσή τους.
- ◆ Στην πλήρη κατανόηση αυτών που βλέπαμε στη θεωρία.
- ◆ Η διαγραμματική τεχνική με βοήθησε πολύ στο να καταλάβω την κατανομή της μνήμης, ιδιαίτερα σε περιπτώσεις ύπαρξης μεταβλητών αναφοράς και κατάργηση αυτών.
- ◆ Βοήθησε κυρίως στην κατανόηση του τρόπου λειτουργίας των προγραμμάτων και των διαφορών μεταξύ αντικειμένων και κλάσεων.
- ◆ Η χρήση της διαγραμματικής τεχνικής με βοήθησε στην κατανόηση των αναφορών σε αντικείμενα.
- ◆ Στο να καταλάβω τις έννοιες βάζοντας τις στο μυαλό μου σε μια τάξη.
- ◆ Στο να κατανοήσω πως διαμορφώνεται η κατάσταση στη μνήμη.
- ◆ Με βοήθησε να καταλάβω πιο καλά τη θεωρία και να διαχωρίσω τις έννοιες μέσα στο μυαλό μου.
- ◆ Να κατανοήσω καλύτερα τις διαδικασίες που γίνονται και πώς αυτές καταλήγουν στο εκάστοτε αποτέλεσμα.
- ◆ Με βοήθησε περισσότερο πιστεύω να κατανοήσω την κατάσταση στη μνήμη κατά τη διάρκεια εκτέλεσης των προγραμμάτων, την κληρονομηση ιδιοτήτων και την ιεραρχία κλάσεων. Οι πολλές όμως όμοιες ασκήσεις καταντούσαν βαρετές.
- ◆ Με βοήθησε στο να καταλάβω το πως ταξινομείται ένα πρόγραμμα και την έννοια κάποιων εντολών στο πρόγραμμα το πώς δηλαδή λειτουργούν μέσα

στο πρόγραμμα και το λόγο ύπαρξης τους σ' αυτό. Σε γενικές γραμμές καθοδήγησε των τρόπο σκέψης μου όσον αφορά τη λύση των συγκεκριμένων προβλημάτων που αφορούν το μάθημα.

Από τις παραπάνω απαντήσεις, προκύπτει η ουσιαστική βοήθεια που τους προσέφεραν αυτές οι παρεμβάσεις στη σχεδίαση του μαθησιακού υλικού για την αντιμετώπιση καταγεγραμμένων μαθησιακών δυσκολιών (υποενότητα 3.7.2.).

4.5.4.2. Επεξεργασία εργασιών

Μαθησιακά αποτελέσματα

Οι εργασίες που οι φοιτητές εκπόνησαν και παρέδωσαν κατά τη διάρκεια του e-learning μαθήματος, ήταν σημαντικές σε όγκο και όπως αναφέρθηκε εκτός των επί μέρους μικρών εργασιών, περιελάμβαναν και ένα μεσαίου μεγέθους project το οποίο υλοποίησαν και παρέδωσαν τμηματικά μαζί με τμήματα των υπολοίπων μικρότερων εργασιών.

Οι φοιτητές σχεδίασαν και υλοποίησαν σε μεγάλο ποσοστό ορθά (βλ. παρακάτω “Βαθμολογία εργασιών”), την εφαρμογή ενός “Βιβλιοπωλείου”. Η εφαρμογή απαιτούσε ανάλυση, σχεδίαση και υλοποίηση κώδικα σε Java και περιελάμβανε την εφαρμογή στην πράξη όλων των βασικών αντικειμενοστρεφών εννοιών και αρχών (σχεδίαση κλάσης, σχεδίαση συντεθειμένης κλάσης, κελυφοποίηση, χρήση πολλαπλών κατασκευαστών, ιεραρχία κλάσεων, κληρονομικότητα, χρήση συλλογών (collection classes) και iterators για την προσπέλαση των στοιχείων τους, πολυμορφισμό, δυναμική δέσμευση, αφηρημένες κλάσεις και μεθόδους, χρήση μεταβλητών και μεθόδων κλάσης (static members), τη χρήση ready-made κλάσης (interface κλάσης) για το user interface).

Παρά τη δυσκολία που οι φοιτητές αναφέρουν ότι αντιμετώπισαν σε αντικειμενοστρεφείς έννοιες και αρχές (Πίνακας 4.15.), αντεπεξήλθαν αρκετά ικανοποιητικά στο να εφαρμόσουν αυτές τις δύσκολες –κατά την αξιολόγησή τους– έννοιες.

Κατά την εκπόνηση των εργασιών, φαίνεται να έχουν δώσει μεγαλύτερη βάση σε υλοποίηση προγραμμάτων και ασκήσεων και όχι σε πολύ μικρές επί μέρους εργασίες, πράγμα λογικό αφού στόχος τους αποτελούσε η επίτευξη καλύτερης βαθμολογίας για τις εργασίες τους.

Κάποιες δυσκολίες που παρατηρήθηκαν στην πρώτη μελέτη περίπτωσης δείχνουν να παραμένουν και σε αυτή την περίπτωση. Συγκεκριμένα:

- Μερικοί φοιτητές αποδίδουν ως ιδιότητες κλάσεων και αυτές που αποτελούν συνάρτηση άλλων ήδη δηλωμένων ιδιοτήτων π.χ. κλάση Κύκλος: ιδιότητες: ακτίνα, διάμετρος. Κατά τη διάρκεια όμως των συναντήσεων του e-learning μαθήματος, φάνηκε οι φοιτητές να έχουν αντιληφθεί σωστά το θέμα, διευκρινίζοντας ότι όταν γράφουν ως ιδιότητες την ακτίνα και τη διάμετρο στην κλάση Κύκλος δεν εννοούν και τις δύο μαζί, αλλά ότι τις αναφέρουν αφού μπορούν να αποτελέσουν ιδιότητες της οντότητας “Κύκλος”. *Η συζήτηση του θέματος συνεπώς, διευκρίνισε κάτι που αρχικά είχε θεωρηθεί παρανόηση.*
- Παρατηρήθηκε η ήδη διαπιστωμένη *δυσκολία* να διακρίνουν μεταξύ λειτουργιών και ιδιοτήτων μιας κλάσης π.χ. ομιλία, ακοή, όραση, σκέψη για την κλάση “Άνθρωπος” και μεταξύ των μερών από τα οποία αποτελείται και ιδιοτήτων π.χ. πόδια, χέρια, κεφάλι για την κλάση “Άνθρωπος” (Teif and Hazzan, 2004). Παραμένει συνολικά –σε μικρότερο βαθμό–, η *δυσκολία του ορθού προσδιορισμού των ιδιοτήτων κλάσεων* και θα πρέπει να αποτελεί σημείο προσοχής στη διδασκαλία.
- Κάποιοι φοιτητές, για να αποκτήσει ένα πρόγραμμα πρόσβαση σε μεταβλητή κλάσης, αλλάζουν τον προσδιοριστή ορατότητας σε *public* και δε σκέφτηκαν να υλοποιήσουν μεθόδους *get* και *set* γι’ αυτήν τη μεταβλητή. Συγχρόνως όμως σε άλλα σημεία των εργασιών, δείχνουν να έχουν αντιληφθεί την αρχή της κελυφοποίησης και τη διατυπώνουν μάλιστα, όπως προκύπτει από τις παρακάτω απαντήσεις τους.

Απάντηση φοιτητή στην παραπάνω εργασία:

*“Η πρώτη αλλαγή θα ήταν να κάναμε τις μεταβλητές της κλάσης δημόσιες. Αυτό όμως δεν προτείνεται γιατί θα καταστρατηγούταν η αρχή της ενθυλάκωσης. Έτσι προτείνω να βάλουμε στην τάξη *Tree* μια μέθοδο η οποία θα μπορεί να θέσει την τιμή του *height*.”*

Απαντήσεις φοιτητών σε εργασία που είχαν να σχολιάσουν το σχεδιασμό μιας κλάσης με δύο διαφορετικούς τρόπους:

*“Καταλληλότερος τρόπος είναι να δηλωθούν οι ιδιότητες *private* γιατί παρέχει περισσότερη ασφάλεια.”*

“Καταλληλότερος σχεδιασμός είναι αυτός με τη δήλωση των ιδιοτήτων ως *private* γιατί εξασφαλίζει την προσπέλαση των x και y με τους *setters* και ακολουθεί τη σχεδίαση βάσει διαπροσωπείας και υλοποίησης.”

“Θεωρώ σωστότερο τον πρώτο τρόπο σχεδιασμού με τις ιδιότητες *private*. Μπορεί να είναι πιο “μπελαλίδικος” γιατί χρειάζεται τουλάχιστον τη δημιουργία δύο τουλάχιστον μεθόδων για κάθε ιδιοχαρακτηριστικό (*get* και *set*) αλλά τον θεωρώ τον πλέον σωστό.”

“Σε γενικές γραμμές μια σωστή τακτική θα ήταν να έχουμε πρόσβαση στα *attributes* μιας κλάσης μέσω των μεθόδων της γιατί έτσι μας δίνεται η δυνατότητα και να περιορίζουμε το πεδίο τιμών που μπορούμε να βάλουμε σε ένα *attribute* αλλά και να δημιουργούμε ένα καλό και εύχρηστο *Interface*.”

“Ο πρώτος τρόπος σχεδιασμού είναι ο καταλληλότερος, γιατί διασφαλίζεται η ακεραιότητα των ιδιοχαρακτηριστικών και επιτρέπεται η προσπέλαση τους μόνο μέσω κατάλληλων ορισμένων από τον προγραμματιστή μεθόδων.”

“Πιστεύω ότι η πιο σωστή μέθοδος γραφής της κλάσης είναι η πρώτη. Είναι σωστότερη από την πλευρά του ότι τα ιδιοχαρακτηριστικά της κλάσης δεν μπορούν να προσπελαστούν από μόνα τους και ότι χρειάζονται ειδικές μέθοδοι για να γίνει η προσπέλαση. Επίσης στην πρώτη περίπτωση δεν μπορώ να αλλάξω τις αυθεντικές τιμές αλλά μόνο να τις χρησιμοποιήσω. Στη δεύτερη μπορεί να έχω κατά λάθος παραβίαση τους.”

“Το πρόγραμμα a είναι καταλληλότερο από το b . Ο λόγος είναι γιατί με τις μεθόδους προσπέλασης *get* μπορούμε να προσπελάσουμε ιδιωτικές μεταβλητές και έτσι να τηρείται και η αρχή απόκρυψης πληροφορίας της κλάσης από το υπόλοιπο πρόγραμμα. Αντίθετα στο πρόγραμμα b , τα ιδιοχαρακτηριστικά της κλάσης δεν έχουν δηλωθεί *private* κάτι που σημαίνει ότι το πρόγραμμα μπορεί να εκχωρεί τιμές σε αυτά χωρίς να διασφαλίζεται η απόκρυψη πληροφορίας.”

“Θεωρώ καταλληλότερο σχεδιασμό τον πρώτο καθώς μπορεί να καλύψει μεγαλύτερο φάσμα προγραμμάτων με επιτυχία ενώ ο δεύτερος αναφέρεται μόνο σε συγκεκριμένα προγράμματα στα οποία δεν υπάρχει πρόβλημα οι μεταβλητές τους να είναι *public* (όπως π.χ. στην περίπτωση της κλάσης *Point* που τα x και y μπορούν να πάρουν οποιαδήποτε τιμή).”

“Στον πρώτο τρόπο ο δημιουργός των κλάσεων έχει ακολουθήσει σωστή πρακτική, ενώ στο δεύτερο όχι. Στην περίπτωση του πρώτου προγράμματος, προσπελούνται πεδία μέσω των μεθόδων *getx* & *gety*. Έτσι αποκρύπτονται οι λεπτομέρειες του

προβλήματος και οποιαδήποτε αλλαγή στην υλοποίηση της κλάσης δε συνεπάγεται και αλλαγές στη διαπροσωπεία της. Επιπλέον, μπορείς να επιβάλλεις περιορισμούς στις τιμές των πεδίων της κλάσης και έτσι τα αποτελέσματα του προγράμματος θα είναι πιο ακριβή και πιο κοντά στην πραγματικότητα. Αντίθετα, στην περίπτωση του δεύτερου προγράμματος, τα πεδία προσπελούνται απ' ευθείας. Δεν υπάρχουν λεπτομέρειες μη ορατές. Δηλαδή, δεν υπάρχει διαχωρισμός στη συμπεριφορά και στον τρόπο υλοποίησης, πράγμα που όμως συμβαίνει στο πρώτο πρόγραμμα. Αυτό έχει ως αποτέλεσμα, να μπορεί να τεθεί οποιαδήποτε τιμή στις μεταβλητές x & y από το πρόγραμμα που τις χρησιμοποιεί και όπως καταλαβαίνουμε δεν υπάρχει κανένας περιορισμός, πράγμα που συνήθως δεν έχει τα επιθυμητά αποτελέσματα.”

“Καταλληλότερος σχεδιασμός είναι ο πρώτος γιατί συμφωνεί με τη γενική αρχή που θέλει να είναι οι μεταβλητές *private* και οι μέθοδοι *public* έτσι ώστε ο χρήστης να ξέρει τι κάνει το πρόγραμμα χωρίς να είναι αναγκασμένος να γνωρίζει πώς το κάνει.”

- Θεωρούν ότι κληρονομούνται και οι *constructors* σε μια ιεραρχία κλάσεων.

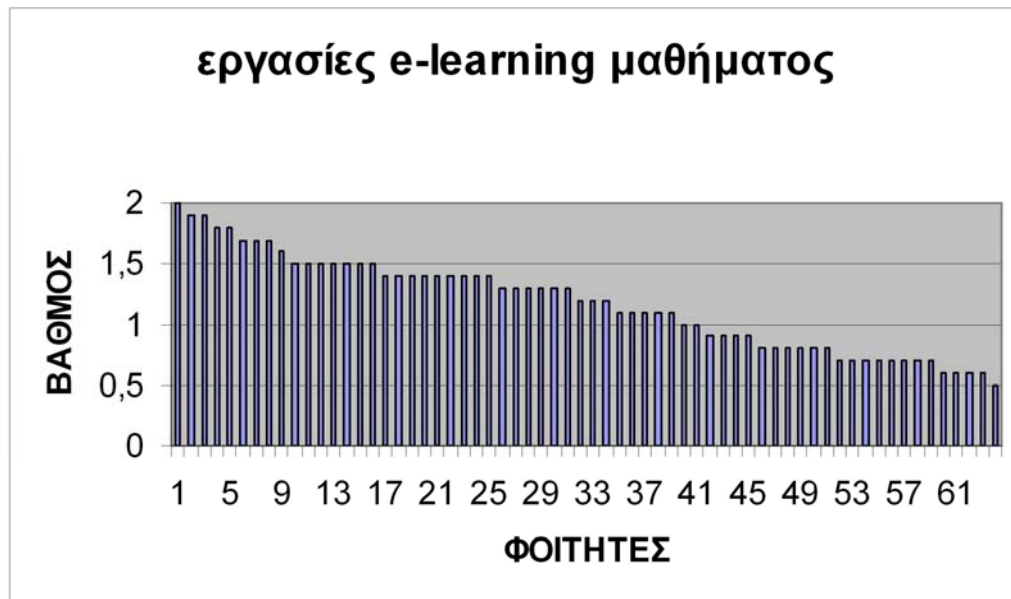
Οι παρεμβάσεις που έγιναν για την αποφυγή παρανοήσεων μετά την πρώτη εφαρμογή της προτεινόμενης διδακτικής προσέγγισης, απέφεραν αποτελέσματα. Συγκεκριμένα:

- Η αποφυγή χρήσης ρήματος για ιδιότητες μιας κλάσης π.χ. “είναι πεινασμένος ή όχι” αντί του “πεινάει ή όχι”, φαίνεται να λειτούργησε θετικά στην απόδοση σωστά ιδιοτήτων σε κλάσεις.
- Δεν παρατηρήθηκε, παρά σε πολύ μικρό βαθμό, στην υλοποίηση κλάσεων να δηλώνουν ως ιδιότητα κλάσης κάποια ενδιάμεση μεταβλητή που τους χρειαζόταν στη διαδικασία.
Αυτό δηλώνει, ότι ο επηρεασμός από το διαδικασιακό προγραμματισμό φαίνεται να έχει ξεπεραστεί και ότι οι φοιτητές λειτουργούν “αντικειμενοστρεφώς”.
- Παρατηρήθηκε σε πολύ μικρότερο βαθμό σύγχυση στην κατανομή μνήμης για τους βασικούς τύπους δεδομένων. Ελάχιστοι φοιτητές θεώρησαν και τους βασικούς τύπους (π.χ. *double x*) ως αναφορά σε μια θέση μνήμης που περιέχει την τιμή της μεταβλητής.

Βαθμολογία εργασιών

Ο Μ.Ο. της βαθμολογίας των εργασιών που παρέδωσαν οι φοιτητές για το e-learning

τμήμα του μαθήματος ήταν 1.17 με τυπική απόκλιση 0.39 (κλίμακα βαθμολογίας από 0 έως 2, με 41 στους 64 φοιτητές (ποσοστό 64%) με βαθμολογία πάνω από τη βάση). Η γραφική αναπαράσταση των αποτελεσμάτων αυτών φαίνονται στο Σχήμα 4.1.



Σχήμα 4.1. Βαθμολογία των εργασιών του e-learning μαθήματος (case study-2)

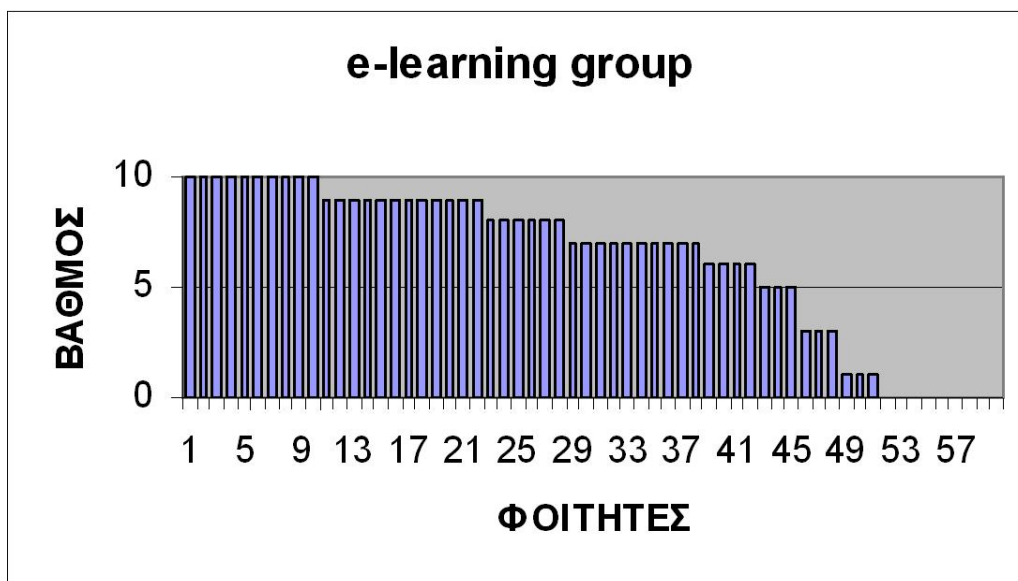
4.5.4.3. Επεξεργασία αποτελεσμάτων εξετάσεων

Από τους φοιτητές που συμμετείχαν στο e-learning μάθημα, 60 συμμετείχαν και στις τελικές εξετάσεις του μαθήματος.

Οι εξετάσεις περιείχαν ερωτήσεις και ασκήσεις που:

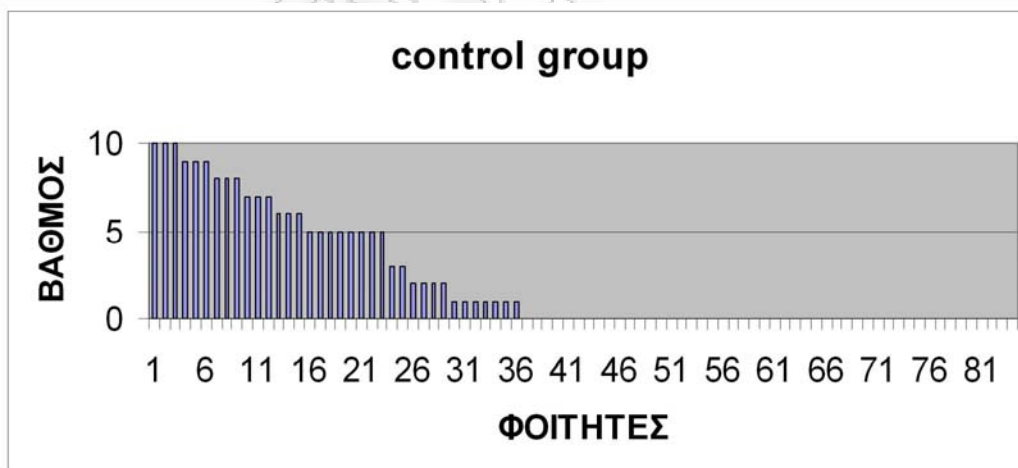
- Είχαν ως στόχο να ελέγξουν την κατανόηση των αντικειμενοστρεφών εννοιών από τη μεριά των φοιτητών και την ικανότητα εφαρμογής τους σε προγράμματα Java.
- Ζητούσαν από τους φοιτητές να αναλύσουν προβλήματα με αντικειμενοστρεφή τρόπο σκέψης και να υλοποιήσουν μικρά προγράμματα σε Java για την επίλυσή τους.
- Είχαν ως στόχο να ελέγξουν την ικανότητα των φοιτητών να γράψουν προγράμματα σε Java, τα οποία υλοποιούν ένα δοσμένο σχεδιασμό κλάσεων (class diagram design).

Ο Μ.Ο. όρος της βαθμολογίας τους ήταν 6.28 με τυπική απόκλιση 3.50 (κλίμακα βαθμολογίας από 0 έως 10) και το ποσοστό επιτυχίας τους ήταν 75% (45 στους 60) (Σχήμα 4.2.).



Σχήμα 4.2. Βαθμολογία στις τελικές εξετάσεις, των φοιτητών που παρακολούθησαν το e-learning μάθημα (case study-2)

Αντίστοιχα, οι φοιτητές που δε συμμετείχαν στο e-learning τμήμα του μαθήματος αλλά παρακολούθησαν το μάθημα με τον παραδοσιακό τρόπο διδασκαλίας και συμμετείχαν στις τελικές εξετάσεις του μαθήματος (ομάδα ελέγχου – control group) ήταν 84 στον αριθμό και ο Μ.Ο. όρος βαθμολογίας τους ήταν 2.15 με τυπική απόκλιση 3.18 και ποσοστό επιτυχίας 27.38% (23 από τους 84 φοιτητές πέτυχαν θετικό αποτέλεσμα) (Σχήμα 4.3.).



Σχήμα 4.3. Βαθμολογία στις τελικές εξετάσεις, των φοιτητών που δεν παρακολούθησαν το e-learning μάθημα (case study-2)

Μετά τις τελικές εξετάσεις του μαθήματος 10 φοιτητές, τυχαία επιλεγμένοι, ρωτήθηκαν να εκφράσουν γενικά τη γνώμη τους για το e-learning μάθημα σε μια σύντομη συνέντευξη. Τα σχόλια που έγιναν παρατίθενται στη συνέχεια: “Μας βοήθησε πολύ, αν είχε γίνει και σε άλλα μαθήματα κάτι τέτοιο π.χ. Αλγόριθμοι και Δομές δεδομένων, θα είχαμε βοηθηθεί”, “Βοήθησε πολύ το e-learning και οι ασκήσεις του”, “Ήταν πολύ καλό για μας αυτό που έγινε”, “Βοήθησε πολύ αυτό που κάνατε”, “Δεν πέρασα στις εξετάσεις, γιατί δεν προετοιμάστηκα, αλλά το e-learning ήταν κάτι πολύ καλό”, “Μόνον το e-learning παρακολούθησαμε και ήταν πολύ αρκετό για να μάθουμε”, “Σας ευχαριστούμε πολύ γι’ αυτό που κάνατε”, “Εγώ δυσκολευόμουν πολύ στην αρχή να γράψω πρόγραμμα. Αλλά τελικά με τη βοήθεια του e-learning έχω φτάσει σε πολύ καλό επίπεδο και πέρασα με 9 τις εξετάσεις”, “Το e-learning βοήθησε πολύ, μακάρι να ήταν όλα τα μαθήματα έτσι”, “Μόνο το e-learning διάβασα, τίποτε άλλο. Από αυτό έμαθα ΑΠ και Java”.

Όλα τα παραπάνω αποτελέσματα και σχόλια είναι ιδιαίτερος ενθαρρυντικός, σχετικά με την αποτελεσματικότητα του e-learning μαθήματος σε σχέση με τον παραδοσιακό τρόπο διδασκαλίας.

4.5.4.4. Στατιστική επεξεργασία ερωτηματολογίων για τη μέτρηση της μαθησιακής αποτελεσματικότητας της προτεινόμενης διδακτικής προσέγγισης και τον προσδιορισμό των παραγόντων που συμβάλλουν σε αυτήν

Στη δεύτερη αυτή μελέτη περίπτωσης, είχαμε το σημαντικότερο δείγμα συμμετεχόντων (66 άτομα που απήντησαν στα pre-test και post-test ερωτηματολόγια σε σύνολο 66 εγγεγραμμένων εκ των οποίων 42% γυναίκες και 58% άνδρες). Το δείγμα αυτό επέτρεπε λεπτομερή στατιστική επεξεργασία των δεδομένων των ερωτηματολογίων σε δεύτερο επίπεδο και έτσι πραγματοποιήθηκε περαιτέρω αξιολόγηση αυτής της μελέτης περίπτωσης με τη χρήση του στατιστικού πακέτου SPSS. Ειδικότερα, η αξιολόγηση αυτή, έγινε με επεξεργασία μερικών διάσπαρτων στα ερωτηματολόγια pre-test και post-test ερωτήσεων, ειδικά σχεδιασμένων για να μετρήσουν τη μαθησιακή αποτελεσματικότητα και να προσδιορίσουν τους παράγοντες που την ερμηνεύουν.

Πιο συγκεκριμένα, η αξιολόγηση επικεντρώθηκε στα: α) κατά πόσο η προτεινόμενη διδακτική προσέγγιση είναι αποτελεσματική και διευκολύνει τη μάθηση και β) ποιοι παράγοντες συμβάλλουν στην αποτελεσματικότητά της.

Πραγματοποιήθηκε η ανάλυση της Πολλαπλής Γραμμικής Παλινδρόμησης (Regression Analysis).

Η εξαρτημένη μεταβλητή στην ανάλυση, είναι αυτή που ονομάζουμε “Μαθησιακή αποτελεσματικότητα” του e-learning μαθήματος.

Αυτή θεωρούμε ότι επηρεάζεται από τις παρακάτω μεταβλητές-παράγοντες που τις θεωρούμε ανεξάρτητες:

- το μαθησιακό στυλ (The learner’s learning style), σύνθετη μεταβλητή για το μαθησιακό στυλ των φοιτητών (κατά Gregorg (Gregorg, 1979)),
- την προηγούμενη εμπειρία στον προγραμματισμό (The learner’s previous experience in programming), απλή μεταβλητή για την προηγούμενη εμπειρία των φοιτητών στον προγραμματισμό,
- το διδακτικό σχεδιασμό του μαθήματος (Contribution of the Instructional Design), σύνθετη μεταβλητή που αποτιμά τη διδακτική προσέγγισή του e-learning μαθήματος (“model first” approach),
- την ποιότητα των μαθησιακών πόρων (Quality of the learning resources), σύνθετη μεταβλητή για την ποιότητα των μαθησιακών πόρων.

Κάθε σύνθετη βασική μεταβλητή (ανεξάρτητη ή εξαρτημένη) λειτουργικοποιήθηκε από ένα πολλαπλό σύνολο στοιχείων το οποίο αποτυπώθηκε και μετρήθηκε από αντίστοιχες διαφορετικές ερωτήσεις των ερωτηματολογίων pre-test και post-test. Κάθε μια ερώτηση εστίαζε και σε μια ελαφρά διαφορετική σκοπιά της βασικής μεταβλητής.

Αρχικά, οι υποθέσεις που έγιναν ως προς τις ερωτήσεις που συνθέταν μια μεταβλητή της μελέτης αξιολόγησης, εγκυροποιήθηκαν στατιστικά και στη συνέχεια έγινε ανάλυση της συνεισφοράς της κάθε σύνθετης μεταβλητής στην αποτελεσματικότητα της προτεινόμενης διδακτικής προσέγγισης που εφαρμόστηκε στο e-learning μάθημα.

Εξαρτημένη μεταβλητή “Μαθησιακή αποτελεσματικότητα”

Η μεταβλητή αυτή που μέτρησε την αποτελεσματικότητα του e-learning μαθήματος, συντίθεται από επιμέρους στοιχεία-μεταβλητές που αντιστοιχούν στις εξής ερωτήσεις του post-test ερωτηματολογίου (Παράρτημα Ε): 29 έως 33, 36 έως 39, 41 έως 43, 46 και 47 (συνολικά 14 επιμέρους στοιχεία-μεταβλητές).

Οι παραπάνω 14 μεταβλητές ελέγχθηκαν ως προς την αξιοπιστία τους με το Cronbach’s Alpha reliability analysis τεστ.

Με τη μέθοδο αυτή, πριν οριστικοποιηθεί ο ακριβής αριθμός των στοιχείων (επιμέρους μεταβλητών) που θα συνθέσουν τελικά τη σύνθετη μεταβλητή μας, εφαρμόζουμε το Cronbach's Alpha reliability analysis τεστ και παρατηρούμε την τιμή του Alpha που προκύπτει. Κάθε στοιχείο-μεταβλητή που αυξάνει σημαντικά την τιμή του Alpha συντελεστή (coefficient) εξαιρείται και πραγματοποιείται νέα ανάλυση, μέχρις ότου επιτευχθεί η υψηλότερη δυνατή τιμή αξιοπιστίας. Έτσι βάζοντας όλες τις παραπάνω μεταβλητές στο τεστ, σε πρώτη φάση εξάγονται τα αποτελέσματα που φαίνονται στον Πίνακα 4.16.

Πίνακας 4.16. Reliability Analysis – Scale (Alpha) (a) (case study–2)

RELIABILITY ANALYSIS - SCALE (ALPHA)

		Mean	Std Dev	Cases
1.	29	3,9697	,7640	66,0
2.	30	3,2727	,9206	66,0
3.	31	2,5606	1,2543	66,0
4.	32	3,4242	1,0086	66,0
5.	33	3,1970	,9801	66,0
6.	36B	2,5303	,8633	66,0
7.	37	3,9242	,8998	66,0
8.	38	3,6061	1,0058	66,0
9.	39	4,0758	,9815	66,0
10.	41	4,0152	,7941	66,0
11.	42	4,1364	,5788	66,0
12.	43	4,1212	,7749	66,0
13.	46	3,1061	,8793	66,0
14.	47	3,3182	,9141	66,0

Item-total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Alpha if Item Deleted
29	45,2879	32,9466	,3914	,7182
30	45,9848	40,8152	-,3995	,7956
31	46,6970	41,8452	-,3978	,8215
32	45,8333	38,1103	-,1738	,7800
33	46,0606	28,7040	,6950	,6785
36B	46,7273	30,2014	,6341	,6909
37	45,3333	28,9026	,7495	,6756

38	45,6515	28,2305	,7224	,6737
39	45,1818	29,7510	,5836	,6929
41	45,2424	30,8019	,6275	,6945
42	45,1212	32,6312	,6045	,7065
43	45,1364	30,4888	,6864	,6894
46	46,1515	30,8690	,5446	,7003
47	45,9394	30,9501	,5090	,7035

Reliability Coefficients

N of Cases = 66,0 N of Items = 14

Alpha = ,7364

Στον παραπάνω Πίνακα, φαίνεται ότι στη στήλη “Alpha if Item Deleted” οι μεταβλητές-ερωτήσεις 30,31 και 32 οδηγούν σε μεγαλύτερη τιμή του Alpha αν τις εξαιρέσουμε. Καταρχάς εξαιρέθηκε η μεταβλητή 31 που οδηγεί στη μεγαλύτερη τιμή και προέκυψε νέος πίνακας. Τελικά εξαιρώντας και τις τρεις μεταβλητές, προκύπτει ο επόμενος Πίνακας 4.17. με πάρα πολύ υψηλή τιμή αξιοπιστίας (92%).

Πίνακας 4.17. Reliability Analysis – Scale (Alpha) (b) (case study–2)

RELIABILITY ANALYSIS - SCALE (ALPHA)

		Mean	Std Dev	Cases
1.	29	3,9697	,7640	66,0
2.	33	3,1970	,9801	66,0
3.	36B	2,5303	,8633	66,0
4.	37	3,9242	,8998	66,0
5.	38	3,6061	1,0058	66,0
6.	39	4,0758	,9815	66,0
7.	41	4,0152	,7941	66,0
8.	42	4,1364	,5788	66,0
9.	43	4,1212	,7749	66,0
10.	46	3,1061	,8793	66,0
11.	47	3,3182	,9141	66,0

N of Cases = 66,0

Item-total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item- Total Correlation	Squared Multiple Correlation	Alpha if Item Deleted
29	36,0303	45,3529	,4696	,3357	,9233
33	36,8030	40,6837	,7298	,7551	,9121
36B	37,4697	42,3760	,6805	,7460	,9144
37	36,0758	40,3788	,8379	,7701	,9064
38	36,3939	39,9040	,7754	,6727	,9096
39	35,9242	40,9019	,7092	,6240	,9132
41	35,9848	43,4613	,6378	,5932	,9163
42	35,8636	45,3811	,6480	,6440	,9173
43	35,8788	42,4159	,7681	,7148	,9108
46	36,8939	42,2193	,6806	,6466	,9144
47	36,6818	42,0972	,6606	,6548	,9154

RELIABILITY ANALYSIS - SCALE (ALPHA)

Reliability Coefficients 11 items

Alpha = ,9213 Standardized item alpha = ,9222

Συνεπώς οι 11 τελικά μεταβλητές, αγγίζουν ένα πάρα πολύ καλό ποσοστό αξιοπιστίας, της τάξης του 92%.

Οι τρεις μεταβλητές που το Cronbach's Alpha reliability analysis τεστ εξείρεσε (Δυσκολεύεστε με τον Αντικειμενοστρεφή Προγραμματισμό; Αντιμετωπίζετε γενικά με φόβο τον Αντικειμενοστρεφή Προγραμματισμό;. Θεωρείτε ότι ο Αντικειμενοστρεφής Προγραμματισμός απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;) ως μη αξιόπιστες (ή λιγότερο αξιόπιστες σε σχέση με τις 11 τελικές) για τη μέτρηση της μαθησιακής αποτελεσματικότητας, γίνονται εύλογα αποδεκτές ως λιγότερο κατάλληλες γι' αυτή τη μέτρηση.

Ανεξάρτητες μεταβλητές

- Η μεταβλητή που μέτρησε το μαθησιακό στυλ (The learner's learning style) των φοιτητών, συντίθεται από επιμέρους στοιχεία-μεταβλητές που αντιστοιχούν στις

ερωτήσεις 11 έως 19 του post-test ερωτηματολογίου (Παράρτημα Ε) (συνολικά 9 επιμέρους στοιχεία-μεταβλητές).

Οι μεταβλητές αυτές δε χρειάστηκε να ελεγχθούν ως προς την αξιοπιστία τους γιατί έχουμε εγκυρότητα περιεχομένου από ειδικό και επομένως έγιναν αποδεκτές ως έχουν. Συγκεκριμένα το μαθησιακό στυλ μετρήθηκε με έναν αριθμό μεταβλητών που αντιπροσωπεύουν τις τέσσερις μαθησιακές κατηγορίες (“Concrete Sequential,” “Abstract Random,” “Abstract Sequential,” and “Concrete Random”), που προσδιορίζονται στο Gregorc (1979). Συγκεκριμένα οι κατηγορίες αυτές βασίζονται στον τρόπο που αντιλαμβανόμαστε (**perceiving**) και κατατάσσουμε (**ordering**) την πληροφορία. Η αντίληψη είναι ο τρόπος που προσλαμβάνουμε την πληροφορία και διακρίνεται σε **συγκεκριμένο (Concrete)** και **αφηρημένο (Abstract)**. Η κατάταξη είναι ο τρόπος που χειριζόμαστε την πληροφορία που προσλαμβάνουμε και διακρίνεται σε **ακολουθιακό (Sequential)** και **τυχαίο (Random)**.

Άτομα που διαθέτουν αφηρημένη αντίληψη (abstract perception) μπορούν να κατανοήσουν και οπτικοποιούν την πληροφορία χωρίς να χρησιμοποιήσουν τις πέντε αισθήσεις τους, συλλαμβάνουν ιδέες που δεν είναι στην πραγματικότητα “ορατές”. Αντίθετα, η αντίληψη με συγκεκριμένο τρόπο (concrete perception), απαιτεί πληροφορία που είναι ορατή στον πραγματικό, φυσικό κόσμο και η οποία μπορεί να κατανοηθεί από τις φυσικές αισθήσεις.

Τώρα, οι ακολουθιακοί μαθητευόμενοι (sequential learners) χρησιμοποιούν έναν γραμμικό, βήμα προς βήμα, μεθοδικό τρόπο για την επεξεργασία και οργάνωση της πληροφορίας. Στην άλλη πλευρά η τυχειότητα (randomness), είναι ο τρόπος με τον οποία τα άτομα αφομοιώνουν την πληροφορία με τη σειρά που τους παρουσιάζεται και την επεξεργάζονται χωρίς κανέναν προκαθορισμένο τρόπο.

Ο συνδυασμός των διαφορετικών τρόπων αντίληψης και κατάταξης της πληροφορίας οδηγεί στις τέσσερις μαθησιακές κατηγορίες (learning styles) που είναι οι: Concrete Sequential, Abstract Random, Abstract Sequential και Concrete Random.

- Η μεταβλητή που μέτρησε την προηγούμενη εμπειρία των φοιτητών στον προγραμματισμό (The learner’s previous experience in programming), είναι απλή μεταβλητή και προκύπτει από την ερώτηση 9 του pre-test ερωτηματολογίου (Παράρτημα Δ).
- Η μεταβλητή που μέτρησε το διδακτικό σχεδιασμό του μαθήματος (Contribution of the Instructional Design), συντίθεται από επιμέρους στοιχεία-μεταβλητές που

αντιστοιχούν στις ερωτήσεις 21 έως 28 του post-test ερωτηματολογίου (Παράρτημα Ε) (συνολικά 8 επιμέρους στοιχεία-μεταβλητές).

Και αυτές οι μεταβλητές δε χρειάστηκε να ελεγχθούν ως προς την αξιοπιστία τους, μιας και αποτιμώνται τα στοιχεία που συνθέτουν το διδακτικό σχεδιασμό της μεθοδολογίας που ακολουθήθηκε και κατά συνέπεια έγιναν αποδεκτές ως έχουν.

- Η σύνθετη μεταβλητή που μέτρησε την ποιότητα των μαθησιακών πόρων (Quality of the learning resources), συντίθεται από επιμέρους στοιχεία-μεταβλητές που αντιστοιχούν στις υποερωτήσεις της ερώτησης 40 του post-test ερωτηματολογίου (Παράρτημα Ε) (συνολικά 11 επιμέρους στοιχεία-μεταβλητές).

Και εδώ δεν απαιτείται το τεστ αξιοπιστίας, μιας και αποτιμώνται τα στοιχεία που συνθέτουν τους μαθησιακούς πόρους της προσέγγισης και κατά συνέπεια έγιναν αποδεκτά ως έχουν.

Έλεγχος προϋποθέσεων εφαρμογής της ανάλυσης της Πολλαπλής Γραμμικής Παλινδρόμησης (Regression Analysis)

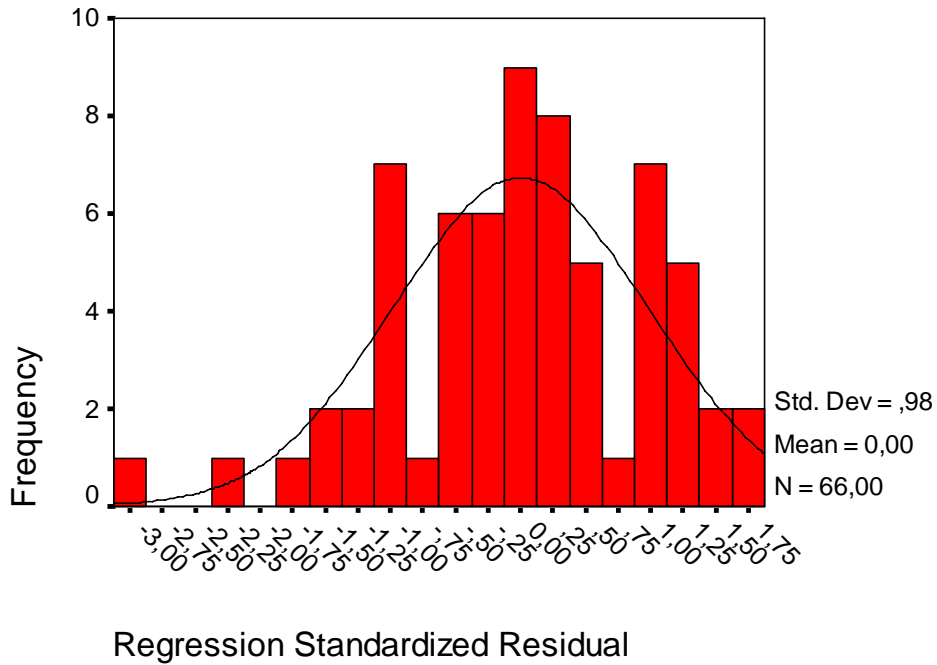
Για την εφαρμογή της μεθόδου της Πολλαπλής Γραμμικής Παλινδρόμησης, θα πρέπει να ισχύουν οι εξής πέντε προϋποθέσεις:

- Το δείγμα να έχει συγκροτηθεί με βάση τις αρχές της τυχαίας δειγματοληψίας.
- Προϋπόθεση η οποία ικανοποιείται από τον τρόπο επιλογής των συμμετεχόντων στη μελέτη περίπτωσης και την καθολική συμμετοχή τους στη μελέτη αξιολόγησης.
- Οι ανεξάρτητες μεταβλητές να είναι κανονικά κατανομημένες.

Για τον έλεγχο της κανονικότητας ελέγχουμε τα υπόλοιπα (residuals). Αν η σχέση είναι γραμμική μεταξύ της εξαρτημένης και των ανεξάρτητων και η εξαρτημένη μεταβλητή κατανέμεται κανονικά για κάθε τιμή των ανεξάρτητων μεταβλητών τότε η κατανομή των υπολοίπων πρέπει να είναι κανονική. Όπως βλέπουμε στα παρακάτω διαγράμματα η κατανομή των υπολοίπων (residuals) πλησιάζει πάρα πολύ την κανονική κατανομή.

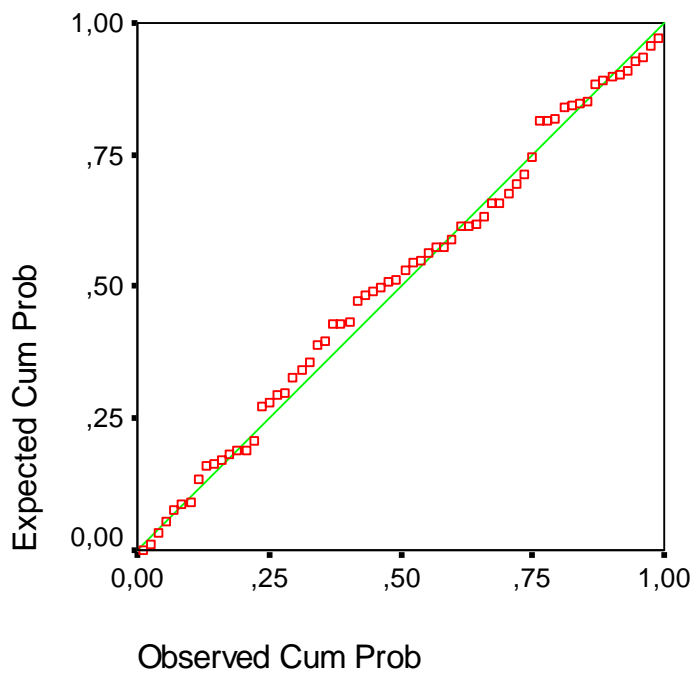
Histogram

Dependent Variable: Effectiveness



Normal P-P Plot of Regression Star

Dependent Variable: Effectiveness



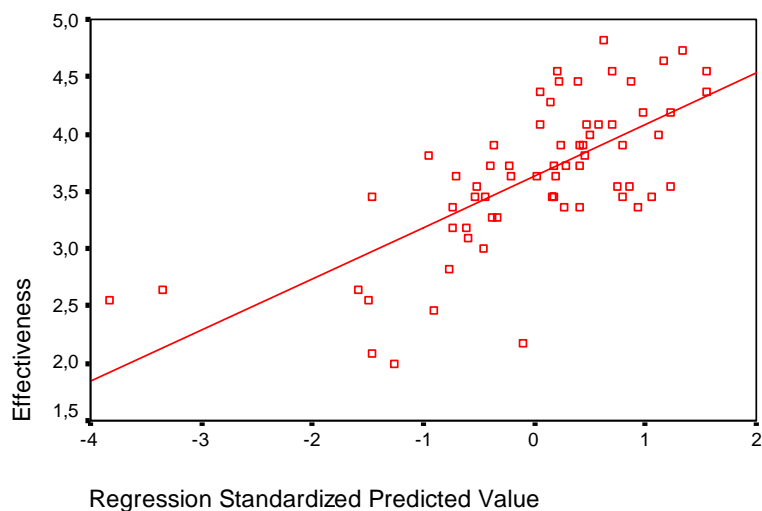
Σχήμα 4.4. Έλεγχος κανονικότητας (case study-2)

- Να έχουμε ομοσκεδαστικότητα (Homogeneity of variance (homoscedasticity)).

Για να ελέγξουμε αν η διακύμανση είναι σταθερή (Homogeneity of variance), δημιουργούμε τα παρακάτω διαγράμματα των υπολοίπων σε σχέση με τις προβλεφθείσες τιμές, καθώς και σε σχέση με τις τιμές των ανεξάρτητων μεταβλητών που εισήλθαν στο παλινδρομικό μοντέλο. Αυτό που ψάχνουμε για να έχουμε ομοσκεδαστικότητα είναι να έχουμε ομοιόμορφη διασπορά των τιμών της εξαρτημένης μεταβλητής πάνω και κάτω από τη γραμμή παλινδρόμησης. Όπως βλέπουμε από τα παρακάτω διαγράμματα αυτό πράγματι συμβαίνει.

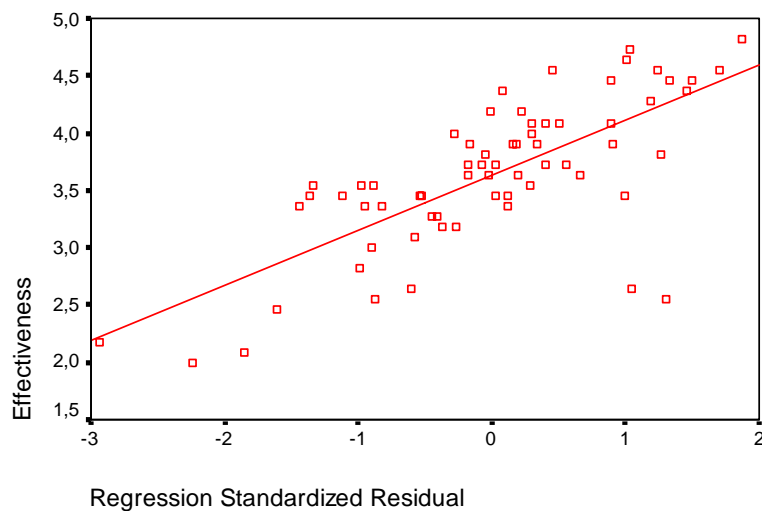
Scatterplot

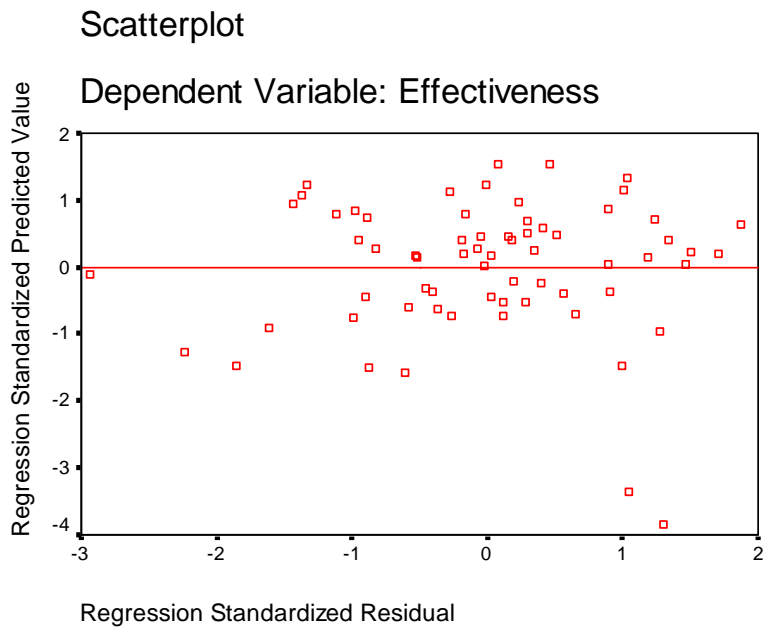
Dependent Variable: Effectiveness



Scatterplot

Dependent Variable: Effectiveness

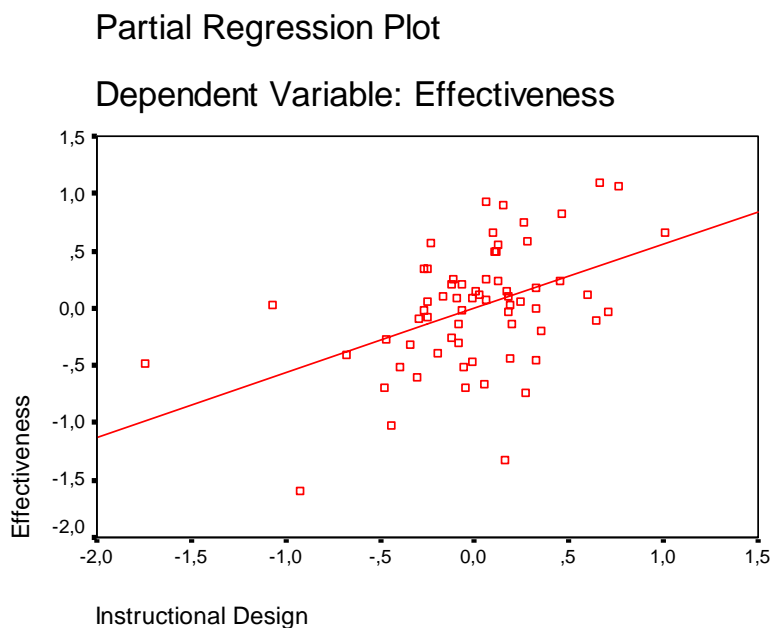




Σχήμα 4.5. Έλεγχος ομοσκεδαστικότητας (case study–2)

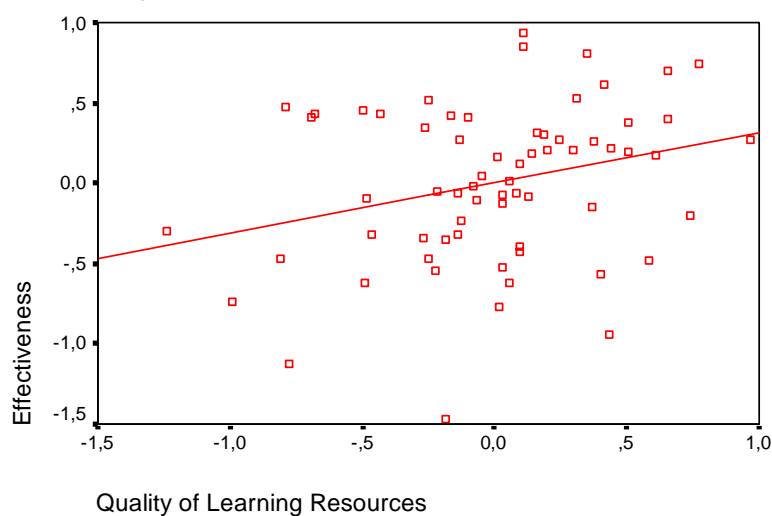
- Η σχέση μεταξύ των ανεξάρτητων μεταβλητών και της εξαρτημένης να είναι γραμμική.

Για να ελέγξουμε την προϋπόθεση της γραμμικότητας δημιουργούμε τα παρακάτω διαγράμματα μεταξύ της εξαρτημένης και της κάθε ανεξάρτητης μεταβλητής. Όσο πιο κοντά στη γραμμή παλινδρόμησης συγκεντρώνονται οι τιμές, τόσο μεγαλύτερη είναι η γραμμική σχέση μεταξύ τους.



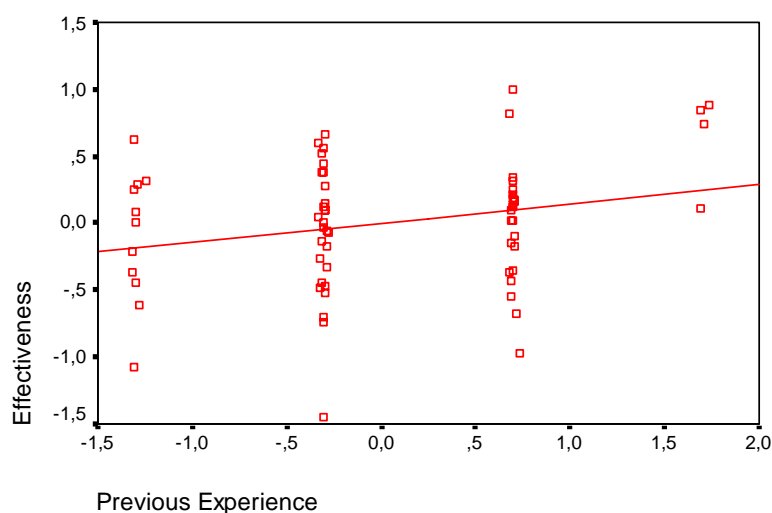
Partial Regression Plot

Dependent Variable: Effectiveness



Partial Regression Plot

Dependent Variable: Effectiveness



Σχήμα 4.6. Έλεγχος γραμμικότητας (case study–2)

Το τελευταίο (Previous Experience) δείχνει ότι έχει τη λιγότερο γραμμική σχέση από όλα τα προηγούμενα (γι' αυτό και όπως προκύπτει παρακάτω στην παρουσίαση των αποτελεσμάτων της αξιολόγησης, είναι και η ανεξάρτητη μεταβλητή που συνεισέφερε λιγότερο στην “Μαθησιακή αποτελεσματικότητα”).

- Οι ανεξάρτητες μεταβλητές να μη συσχετίζονται υψηλά μεταξύ τους (έλεγχος πολυσυγγραμμικότητας – multicollinearity).

Οι τεχνικές της παλινδρομικής ανάλυσης & της συσχέτισης, προϋποθέτουν ότι οι ανεξάρτητες μεταβλητές δε συσχετίζονται υψηλά μεταξύ τους. Τέτοια πολυσυγγραμμικότητα υπάρχει, όταν μια από τις ανεξάρτητες μεταβλητές σε μια εξίσωση παλινδρόμησης συνδέεται γραμμικά και τέλεια με μια ή περισσότερες από τις άλλες ανεξάρτητες μεταβλητές στην εξίσωση.

Η πιο συχνή διαδικασία ελέγχου είναι η εξέταση της μήτρας των διμεταβλητών συσχετίσεων. Συνήθως αν καμία συσχέτιση δεν ξεπερνά κάποια προκαθορισμένη τιμή (συνήθως είναι το 0.80) τότε δεν υπάρχει πρόβλημα πολυσυγγραμμικότητας.

Με βάση των παρακάτω πίνακα συσχετίσεων δε βλέπουμε να υπάρχει κάποια τέτοια ισχυρή σχέση.

Πίνακας 4.18. Έλεγχος πολυσυγγραμμικότητας – a (case study–2)

Correlations

		Effectiveness	Learning Style	Instructional Design	Quality of Learning Resources	Previous Experience
Effectiveness	Pearson Correlation	1,000	,265	,631	,549	,192
	Sig. (2-tailed)		,031	,000	,000	,123
	N	66	66	66	66	66
Learning Style	Pearson Correlation	,265	1,000	,262	,198	,042
	Sig. (2-tailed)	,031		,034	,111	,737
	N	66	66	66	66	66
Instructional Design	Pearson Correlation	,631	,262	1,000	,604	,015
	Sig. (2-tailed)	,000	,034		,000	,903
	N	66	66	66	66	66
Quality of Learning Resources	Pearson Correlation	,549	,198	,604	1,000	,001
	Sig. (2-tailed)	,000	,111	,000		,992
	N	66	66	66	66	66
Previous Experience	Pearson Correlation	,192	,042	,015	,001	1,000
	Sig. (2-tailed)	,123	,737	,903	,992	
	N	66	66	66	66	66

* Correlation is significant at the 0.05 level (2-tailed).

** Correlation is significant at the 0.01 level (2-tailed).

Επίσης για τον έλεγχο της πολυσυγγραμμικότητας, μπορεί να χρησιμοποιηθεί και ο παρακάτω Πίνακας 4.19., ο οποίος είναι ο Πίνακας “Coefficients” που παρατίθεται και εξηγείται στη συνέχεια (Πίνακας 4.20.), με επιπλέον στατιστικά στοιχεία για πολυσυγγραμμικότητα.

Από τη στήλη του Πίνακα “Collinearity Statistics” οι τιμές του Tolerance δεν είναι κάτω από το 0.10 και οι τιμές του VIF (variance inflation factor= $1/\text{Tolerance}$) δεν είναι πάνω από 10. Οι τιμές αυτές, εξασφαλίζουν ότι δεν έχουμε πρόβλημα πολυσυγγραμμικότητας.

Πίνακας 4.19. Έλεγχος πολυσυγγραμμικότητας – b (case study–2)

Coefficients

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95% Confidence Interval for B		Correlations			Collinearity Statistics	
		B	Std. Error	Beta			Lower Bound	Upper Bound	Zero-order	Partial	Part	Tolerance	VIF
1	(Constant)	,272	,521		,521	,604	-,770	1,313					
	Instructional Design	,760	,117	,631	6,500	,000	,526	,993	,631	,631	,631	1,000	1,000
2	(Constant)	-,154	,540		-,284	,777	-1,233	,926					
	Instructional Design	,567	,142	,471	3,987	,000	,283	,851	,631	,449	,375	,635	1,574
	Quality of Learning Resources	,309	,138	,265	2,242	,028	,034	,584	,549	,272	,211	,635	1,574
3	(Constant)	-,477	,552		-,864	,391	-1,579	,626					
	Instructional Design	,562	,139	,466	4,044	,000	,284	,840	,631	,457	,372	,635	1,575
	Quality of Learning Resources	,312	,135	,267	2,315	,024	,043	,581	,549	,282	,213	,635	1,575
	Previous Experience	,145	,072	,184	2,004	,049	,000	,290	,192	,247	,184	1,000	1,000

a Dependent Variable: Effectiveness

Αποτελέσματα αξιολόγησης

Η Πολλαπλή Γραμμική Παλινδρόμηση (Regression Analysis) έδωσε τα παρακάτω αποτελέσματα:

Πίνακας 4.20. Regression Analysis – SPSS output (case study–2)

Variables Entered/Removed^d

Model	Variables Entered	Variables Removed	Method
1	Instructional Design		Stepwise (Criteria: Probability-of-F-to-enter <= ,050, Probability-of-F-to-remove >= ,100).
2	Quality of Learning Resources		Stepwise (Criteria: Probability-of-F-to-enter <= ,050, Probability-of-F-to-remove >= ,100).
3	Previous Experience		Stepwise (Criteria: Probability-of-F-to-enter <= ,050, Probability-of-F-to-remove >= ,100).

a. Dependent Variable: Effectiveness

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	,631 ^a	,398	,388	,5066	,398	42,253	1	64	,000
2	,665 ^b	,442	,424	,4914	,045	5,028	1	63	,028
3	,690 ^c	,476	,451	,4800	,034	4,016	1	62	,049

a. Predictors: (Constant), Instructional Design

b. Predictors: (Constant), Instructional Design, Quality of Learning Resources

c. Predictors: (Constant), Instructional Design, Quality of Learning Resources, Previous Experience

ANOVA^d

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	10,845	1	10,845	42,253	,000 ^a
	Residual	16,427	64	,257		
	Total	27,273	65			
2	Regression	12,059	2	6,030	24,970	,000 ^b
	Residual	15,213	63	,241		
	Total	27,273	65			
3	Regression	12,985	3	4,328	18,782	,000 ^c
	Residual	14,288	62	,230		
	Total	27,273	65			

a. Predictors: (Constant), Instructional Design

b. Predictors: (Constant), Instructional Design, Quality of Learning Resources

c. Predictors: (Constant), Instructional Design, Quality of Learning Resources, Previous Experience

d. Dependent Variable: Effectiveness

Descriptive Statistics

	Mean	Std. Deviation	N
Effectiveness	3,6364	,6478	66
Previous Experience	2,30	,82	66
Learning Style	3,6835	,3444	66
Instructional Design	4,4280	,5376	66
Quality of Learning Resources	4,1391	,5552	66

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,272	,521		,521	,604
	Instructional Design	,760	,117	,631	6,500	,000
2	(Constant)	-,154	,540		-,284	,777
	Instructional Design	,567	,142	,471	3,987	,000
	Quality of Learning Resources	,309	,138	,265	2,242	,028
3	(Constant)	-,477	,552		-,864	,391
	Instructional Design	,562	,139	,466	4,044	,000
	Quality of Learning Resources	,312	,135	,267	2,315	,024
	Previous Experience	,145	,072	,184	2,004	,049

a. Dependent Variable: Effectiveness

Excluded Variables^ξ

Model		Beta In	t	Sig.	Partial Correlation	Collinearity Statistics
						Tolerance
1	Previous Experience	,182 ^a	1,916	,060	,235	1,000
	Learning Style	,108 ^a	1,073	,287	,134	,932
	Quality of Learning Resources	,265 ^a	2,242	,028	,272	,635
2	Previous Experience	,184 ^b	2,004	,049	,247	1,000
	Learning Style	,097 ^b	,990	,326	,125	,929
3	Learning Style	,089 ^c	,933	,355	,119	,928

a. Predictors in the Model: (Constant), Instructional Design

b. Predictors in the Model: (Constant), Instructional Design, Quality of Learning Resources

c. Predictors in the Model: (Constant), Instructional Design, Quality of Learning Resources, Previous Experience

d. Dependent Variable: Effectiveness

Στον πρώτο πίνακα (variables entered/removed) φαίνονται οι μεταβλητές που επηρεάζουν τελικά την εξαρτημένη μας μεταβλητή “Μαθησιακή αποτελεσματικότητα” (Learning Effectiveness) και είναι οι:

- Διδακτικός σχεδιασμός (Instructional Design)

- Ποιότητα μαθησιακών πόρων (Quality of Learning Resources)
- Προηγούμενη εμπειρία στον προγραμματισμό (Previous Experience in Programming)

Ο επόμενος πίνακας (Model Summary) είναι ο πιο σημαντικός γιατί μπορούμε να δούμε το ποσοστό ερμηνείας της εξαρτημένης μεταβλητής από τις ανεξάρτητες, δηλ. τη συνεισφορά της κάθε ανεξάρτητης μεταβλητής στη μαθησιακή αποτελεσματικότητα της προτεινόμενης διδακτικής προσέγγισης. Έτσι βλέπουμε ότι:

- η ανεξάρτητη μεταβλητή “Διδακτικός σχεδιασμός” (Instructional Design) ερμηνεύει το 39.8% της “Μαθησιακής αποτελεσματικότητας” (Learning Effectiveness)
- η ανεξάρτητη μεταβλητή “Ποιότητα μαθησιακών πόρων” (Quality of Learning Resources) ερμηνεύει το 4.5% της “Μαθησιακής αποτελεσματικότητας” (Learning Effectiveness) και
- τέλος, η ανεξάρτητη μεταβλητή “Προηγούμενη εμπειρία στον προγραμματισμό” (Previous Experience) ερμηνεύει το 3.4% της “Μαθησιακής αποτελεσματικότητας” (Learning Effectiveness)

Το ποσοστό 39.8% ή καλύτερα το 40% αποτελεί ένα πάρα πολύ καλό ποσοστό για τον παράγοντα “Διδακτικός σχεδιασμός”. Από αυτό προκύπτει ότι η διδακτική προσέγγιση “model first” που ακολουθήθηκε στο e-learning μάθημα, είχε σημαντική συνεισφορά στη μαθησιακή αποτελεσματικότητα του μαθήματος, πράγμα που επιβεβαιώνεται (α) από τα τελικά αποτελέσματα επιτυχόντων στο μάθημα και (β) από την ποιοτική ανάλυση των ερωτηματολογίων (πολύ θετικά σχόλια για τον τρόπο διδασκαλίας). Σημαντικό είναι να ληφθεί υπόψη ότι η ανάλυση έγινε συμπεριλαμβανομένων και των αποτυχόντων(!) που ήσαν λίγοι.

Επίσης, οι παράγοντες “Ποιότητα μαθησιακών πόρων” (Quality of Learning Resources) και “Προηγούμενη εμπειρία στον προγραμματισμό” (Previous Experience) ερμηνεύουν μαζί ένα 8% που είναι επίσης σημαντικό. Από την ποιοτική ανάλυση των ερωτηματολογίων η συμμετοχή της μεταβλητής “ποιότητα των μαθησιακών πόρων” επαληθεύεται εύκολα, μιας και οι περισσότεροι φοιτητές εκφράστηκαν πολύ θετικά για την ποιότητα των μαθησιακών πόρων με ιδιαίτερη έμφαση στα παραδείγματα κάτω από το περιβάλλον BlueJ. Όσον αφορά τη

μεταβλητή “Προηγούμενη εμπειρία στον προγραμματισμό”, από το pre-test ερωτηματολόγιο, παρατηρείται ότι το 66% των φοιτητών προέρχονταν από την Τεχνολογική Κατεύθυνση και είχαν διδαχθεί το μάθημα “Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον” σε αντίθεση με το υπόλοιπο 34% που δεν είχε διδαχθεί κάποιο αντίστοιχο προγραμματιστικό μάθημα. Το γεγονός αυτό δίνει αυτό το μικρό ποσοστό ερμηνείας της μαθησιακής αποτελεσματικότητας της προσέγγισης.

Στον επόμενο πίνακα φαίνεται ο αντίστοιχος πίνακας (Model Summary) της Πολλαπλής Γραμμικής Παλινδρόμησης μόνο για τους φοιτητές που είχαν εμπειρία στον Προγραμματισμό.

Πίνακας 4.21. Regression Analysis – Model Summary (case study–2, φοιτητές με εμπειρία στον Προγραμματισμό)

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	,609 ^a	,371	,356	,5133	,371	24,776	1	42	,000

a. Predictors: (Constant), Instructional Design

Στην τελευταία περίπτωση, φαίνεται να ερμηνεύει σε ποσοστό 37.1% την “Μαθησιακή αποτελεσματικότητα” (Learning Effectiveness) του συγκεκριμένου δείγματος μόνο ο παράγοντας “Διδακτικός σχεδιασμός” (Instructional Design), ένα εξίσου σημαντικό ποσοστό.

Συνολικά επομένως, ο “Διδακτικός σχεδιασμός“ (είτε στην περίπτωση όλων των φοιτητών είτε στην περίπτωση αυτών με κάποια προηγούμενη εμπειρία στον Προγραμματισμό), είναι ο παράγοντας που επηρέασε περισσότερο τη μαθησιακή αποτελεσματικότητα και μάλιστα κατά ένα πολύ αξιολόγο ποσοστό. Αυτό αποτελεί πολύ σημαντικό στοιχείο αξιολόγησης για την προτεινόμενη διδακτική προσέγγιση.

Ο παράγοντας “μαθησιακό στυλ” (Learning Style) των φοιτητών δεν έδωσε κάποιο σημαντικό ποσοστό ερμηνείας της μαθησιακής αποτελεσματικότητας, πράγμα που σημαίνει ότι τόσο ο διδακτικός σχεδιασμός όσο και η ποιότητα των μαθησιακών

πόρων ήταν κατάλληλες για τους διάφορους μαθησιακούς τύπους των φοιτητών του δείγματος. Αυτό που διαπιστώθηκε από την ποιοτική ανάλυση των ερωτηματολογίων ήταν ότι ο τρόπος διδασκαλίας είχε μια θετική επίδραση στον τρόπο διαβάσματος-προσέγγισης του μαθήματος σε όλους τους φοιτητές.

Η ποιότητα των μαθησιακών πόρων ως παράγοντας που συντελεί στην αύξηση της μαθησιακής αποτελεσματικότητας έχει αναφερθεί και σε άλλες μελέτες (Psaromiligkos and Retalis, 2003). Όσον αφορά τη διδακτική προσέγγιση “model first” που ακολουθήθηκε στο e-learning μάθημα, η παρούσα μελέτη αξιολόγησης αποτελεί και την πρώτη ολοκληρωμένη μελέτη.

Τέλος, όσον αφορά τη μέτρηση της “Μαθησιακής αποτελεσματικότητας” της προτεινόμενης διδακτικής προσέγγισης, από τον Πίνακα “Coefficients” ο τύπος της γραμμικής παλινδρόμησης για την εξαρτημένη μεταβλητή μας “Μαθησιακή αποτελεσματικότητα” (Effectiveness) προκύπτει να είναι ο εξής:

$$y = -0.477 + 0.562 * \text{Instructional Design} + 0.312 * \text{Quality of Learning Resources} + 0.145 * \text{Previous Experience}$$

όπου $y = \text{Effectiveness}$.

Αντικαθιστώντας στην παραπάνω εξίσωση, τις μέσες τιμές των ανεξάρτητων μεταβλητών, όπως φαίνονται στον Πίνακα “Descriptive Statistics”, προκύπτει η πάρα πολύ καλή τιμή για το “Learning Effectiveness” **3,6364** (στην πεντάβαθμη κλίμακα Likert που χρησιμοποιήθηκε στα ερωτηματολόγια).

4.5.5. Συμπεράσματα – Βελτιώσεις

Αποφάσεις για τη βελτίωση της προτεινόμενης διδακτικής προσέγγισης που ελήφθησαν μετά την πρώτη εφαρμογή της, δικαιώθηκαν στη δεύτερη αυτή μελέτη περίπτωσης.

- Η παρουσίαση στους φοιτητές των φάσεων ανάπτυξης μιας αντικειμενοστρεφούς εφαρμογής και η ενσωμάτωση της ενιαίας εργασίας-εφαρμογής, τους επέτρεψε να αποκτήσουν σχεδιαστικές δεξιότητες (design skills), να συνθέσουν τις γνώσεις

τους και να τις εφαρμόσουν στην υλοποίηση ενός συγκεκριμένου έργου. Οι φοιτητές αξιολόγησαν τη συμβολή αυτού του μαθησιακού πόρου στη βελτίωση των γνώσεων και των δεξιοτήτων σας πολύ υψηλά (Μ.Ο. 4.12 στην κλίμακα από 1 έως 5).

- Ο τρόπος υλοποίησης της εφαρμογής της διδακτικής προσέγγισης, οδήγησε στη δημιουργία μεταξύ των φοιτητών ενεργών κοινοτήτων μάθησης κάνοντας χρήση των διαδικτυακών τεχνολογιών (και το αξιολόγησαν με 4.1).
- Έννοιες τις οποίες οι φοιτητές αξιολογούν ως δύσκολες (σε μια κλίμακα από 1 έως 3), στις εργασίες τους φαίνεται να μπορούν να τις εφαρμόσουν στην πράξη.
- Η πρόταση περί ομαδικών εργασιών δείχνει ότι οι φοιτητές –με τη χρήση του on-line μαθησιακού περιβάλλοντος– εκτίμησαν τη σημασία της δημιουργίας κοινότητας πρακτικών (community of practices).
- Οι φοιτητές διατυπώνουν σαφώς σε σχετική ερώτηση, τη βοήθεια που τους προσέφερε η συστηματική χρήση της διαγραμματικής τεχνικής (class diagrams).

Από συζητήσεις κατά τη διάρκεια των συναντήσεων του e-learning μαθήματος και από αντίστοιχες ανοικτού τύπου ερωτήσεις, προέκυψε η σχετική δυσκολία της χρήσης διαγραμματικής τεχνικής χωρίς τη χρήση κάποιου ειδικού εργαλείου σχεδίασης, αλλά και η δυσκολία της παρουσίασης της φάσης της σχεδίασης μιας αντικειμενοστρεφούς εφαρμογής (class diagrams). Οι φοιτητές, χωρίς την ύπαρξη ενός τέτοιου εργαλείου χρησιμοποιούσαν απλά σχεδιαστικά προγράμματα όπως το Microsoft Visio ή ακόμα και τις σχεδιαστικές δυνατότητες του Microsoft Word. *Συνεπώς κρίθηκε αναγκαίο να ενσωματωθεί στην προτεινόμενη διδακτική προσέγγιση η χρήση ενός σχετικά απλού και φιλικού τέτοιου εργαλείου όπως είναι το ArgoUML (ενότητα 3.8.).*

4.6. Μελέτη περίπτωσης 3 (case study-3)

Στην ενότητα αυτή παρουσιάζεται η τρίτη πιλοτική εφαρμογή της προτεινόμενης διδακτικής προσέγγισης. Σε αυτήν ακολουθήθηκε πλήρως η περιγραφή του τρίτου κεφαλαίου ως προς τη ροή δραστηριοτήτων, το μαθησιακό περιβάλλον και τα μαθησιακά εργαλεία (ενότητες 3.5., 3.6., 3.8.).

4.6.1. Χώρος εφαρμογής

Η διδακτική προσέγγιση εφαρμόστηκε κατά το εαρινό εξάμηνο του 2006 (ξεκίνησε σχετικά αργότερα από τη δεύτερη μελέτη περίπτωσης) με τη μορφή e-learning σεμιναριακού μαθήματος το οποίο διήρκεσε δύομισι μήνες. Ο χώρος εφαρμογής ήταν το Τ.Ε.Ι. Πειραιά, στο πλαίσιο του ΠΜΣ “Information Technology” που διεξάγεται σε συνεργασία με το Paisley University of Scotland. Το e-learning σεμιναριακό μάθημα συνδυάστηκε με το μάθημα του ΠΜΣ “Web Development” (Georgantaki et al., 2007). Η συμμετοχή ήταν προαιρετική και η βαθμολογία των εργασιών που θα παρέδιδαν οι φοιτητές (Φάση II, ενότητα 3.5.), συμμετείχε κατά 10% στη διαμόρφωση της βαθμολογίας του project που θα υλοποιούσαν και θα παρέδιδαν τελικά για το μάθημα (Φάση III, ενότητα 3.5.).

4.6.2. Συμμετέχοντες

Δήλωσαν συμμετοχή στο e-learning σεμιναριακό μάθημα δέκα οκτώ (18) μεταπτυχιακοί φοιτητές, 17 άνδρες και 1 γυναίκα, ηλικίας από 23 έως 52 ετών. Προέρχονταν από τμήματα Τ.Ε.Ι. (14 άτομα) και τμήματα Ανώτατων Εκπαιδευτικών Ιδρυμάτων (Α.Ε.Ι.) (4 άτομα). Οι περισσότεροι από τους συμμετέχοντες είχαν προηγουμένως διδαχθεί ΑΠ με τη χρήση της γλώσσας VB.NET κατά τη διάρκεια του προηγούμενου εξαμήνου στο πλαίσιο του μαθήματος “Software Development” του ΠΜΣ που παρακολουθούσαν. Δύο φοιτητές δεν είχαν παρακολουθήσει αυτό το μάθημα αλλά είχαν έλθει σε επαφή με τον ΑΠ ασχολούμενοι μόνοι τους ή παρακολουθώντας αντίστοιχο μάθημα σε κάποιο Ινστιτούτο Επαγγελματικής Κατάρτισης (Ι.Ε.Κ.) και άλλοι δύο δεν είχαν καμία προηγούμενη επαφή με τον ΑΠ. Επτά (7) φοιτητές είχαν προηγούμενη επαφή με τη γλώσσα Java (από προσωπικό ενδιαφέρον, κατά τη διάρκεια των προπτυχιακών σπουδών τους ή παρακολουθώντας κάποιο σεμινάριο). Το κίνητρο των φοιτητών για τη συμμετοχή τους στο e-learning σεμιναριακό μάθημα, ήταν κυρίως το ενδιαφέρον τους για το αντικείμενο του ΑΠ, η βελτίωση των γνώσεων και των δεξιοτήτων τους στον ΑΠ και έπειτα το ότι τους χρειαζόταν στο πλαίσιο των μεταπτυχιακών τους σπουδών. Συγκεκριμένα οι λόγοι που έκαναν αρχικά τους φοιτητές να παρακολουθήσουν το μάθημα φαίνονται στον

Πίνακα 4.22., όπως προέκυψε από αντίστοιχη ερώτηση του ερωτηματολογίου μετά-αξιολόγησης.

Πίνακας 4.22. Κίνητρα παρακολούθησης (case study-3)

Ερώτηση	Αριθμός φοιτητών που επέλεξαν την απάντηση (max 18)
Για ποιο/ούς λόγους αρχικά είχατε αποφασίσει να παρακολουθήσετε το e-learning course;	
Επειδή με ενδιαφέρει το αντικείμενο.	14
Επειδή ήθελα να μάθω κάτι περισσότερο από ό,τι έχω διδαχθεί ή ήδη γνωρίζω.	9
Επειδή μου χρειαζόταν στο πλαίσιο των Μεταπτυχιακών Σπουδών μου.	8
Επειδή μου χρειαζόταν για επαγγελματικούς λόγους.	3
Με δυσκολεύει το αντικείμενο.	1
Από περιέργεια.	1
Για άλλο λόγο.	1

Ο άλλος λόγος που ένας φοιτητής επικαλέστηκε είναι:

- “Επειδή πίστευα ότι θα αποκτήσω μια καλή γνώση πάνω στη γλώσσα προγραμματισμού JAVA (πράγμα που έγινε) και επειδή η γνώση της JAVA ήταν κάτι που είχα σαν στόχο από παλιότερα απλά ποτέ δεν έβρισκα το χρόνο.”

Να σημειωθεί, ότι οι συμμετέχοντες ήταν σχεδόν καθ’ ολοκληρία εργαζόμενοι με μεγάλους χρονικούς περιορισμούς και βεβαρημένο πρόγραμμα λόγω των σπουδών τους.

4.6.3. Υλοποίηση της εφαρμογής

Κατά τη διάρκεια του e-learning μαθήματος πραγματοποιήθηκαν τρεις (3) συνεδρίες συντονισμού και επίλυσης αποριών και επίσης σε αυτές έγινε η παρουσίαση των φάσεων υλοποίησης της αντικειμενοστρεφούς εφαρμογής με τη χρήση φορητού υπολογιστή και βιντεοπροβολέα και κάνοντας χρήση των προγραμματιστικών περιβαλλόντων ArgoUML, BlueJ και SUN One Studio. Οι συνεδρίες αυτές έγιναν σε εργαστήριο και δόθηκε η δυνατότητα στους φοιτητές να εξασκηθούν και να

εξοικειωθούν με τα προγραμματιστικά περιβάλλοντα. Οι σοβαροί χρονικοί περιορισμοί των συμμετεχόντων δεν επέτρεψαν τη διενέργεια επιπλέον “δια ζώσης” συναντήσεων.

Η διεξαγωγή του e-learning σεμιναριακού μαθήματος ακολούθησε τη ροή δραστηριοτήτων που περιγράφηκε στην ενότητα 3.5. Το on-line μαθησιακό περιβάλλον (ενότητα 3.6.) αυτής της μελέτης περίπτωσης, υλοποιήθηκε με τη δημιουργία δικτυακού τόπου στο περιβάλλον του LMS Moodle (<http://www.moodle.org>).

4.6.4. Αξιολόγηση της εφαρμογής

Οι φοιτητές συμπλήρωσαν τα δύο ερωτηματολόγια (pre-test και post-test) –που παρατίθενται στα Παραρτήματα ΣΤ και Ζ αντίστοιχα– στην αρχή (μετά την πρώτη συνεδρία) και μετά το τέλος του e-learning σεμιναριακού μαθήματος.

Στις εργασίες τις οποίες εκπόνησαν, περιλαμβανόταν μόνον η μεσαίου μεγέθους εφαρμογή (υποενότητα 4.4.5.) την οποία παρέδωσαν κατά τμήματα, καθώς προχωρούσαν τη μελέτη των διδακτικών ενοτήτων του μαθησιακού υλικού και εξελισσόταν αντίστοιχα η παρουσίαση στις δια ζώσης συναντήσεις στην τάξη.

Η τρίτη φάση (Φάση III: Αυτόνομη επίλυση προβλημάτων, από τις φάσεις που περιγράφηκαν στην ενότητα 3.5.) στην τρίτη αυτή εφαρμογή της διδακτικής προσέγγισης, αφορούσε στο project το οποίο οι φοιτητές υλοποίησαν και παρέδωσαν για το μάθημα “Web Development”.

Συνολικά οι φοιτητές που συμμετείχαν στην αξιολόγηση αυτής της εφαρμογής της διδακτικής προσέγγισης ήταν 18, όλοι οι συμμετέχοντες στο μάθημα. Ένας σημαντικός παράγον που επηρεάζει οπωσδήποτε την αξιολόγηση των φοιτητών, αποτελεί το βεβαρημένο πρόγραμμά τους λόγω επαγγελματικών και άλλων υποχρεώσεων και επιπλέον ο φόρτος που συνεπάγονται οι μεταπτυχιακές σπουδές τους.

4.6.4.1. Επεξεργασία ερωτηματολογίων

Ποσοτικά Αποτελέσματα

Από την επεξεργασία των απαντήσεων στα δύο ερωτηματολόγια προέκυψαν τα ποσοτικά στοιχεία που παρουσιάζονται στους παρακάτω πίνακες.

Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στον ΑΠ

Πίνακας 4.23. Αυτοεκτίμηση των φοιτητών για το επίπεδό τους στο γνωστικό αντικείμενο του ΑΠ (case study-3)

Ερώτηση	Μ.Ο. (Από το “pre-test”)	Μ.Ο. (Από το “post- test”)
Ποιο εκτιμάτε ότι είναι το επίπεδό σας στο διαδικασιακό προγραμματισμό;	2.67	3.11
Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον ΑΠ;	2.72	2.83
Πώς εκτιμάτε μετά τη διεξαγωγή του e-learning μαθήματος, την ικανότητά σας να γράψετε προγράμματα σε Java;	*	3.22**

Κωδικοποίηση απαντήσεων:

**Κωδικοποίηση απαντήσεων:

5 = Επαγγελματίας... 1 = Αρχάριος 5 = Πολύ ικανός/ή... 1 = Καθόλου ικανός/ή

* ερώτηση μη διαθέσιμη στο “pre-test” ή το “post-test”

Η αυτοεκτίμηση των φοιτητών έχει αυξηθεί μετά την παρακολούθηση του e-learning μαθήματος και μάλιστα στην τρίτη αυτή μελέτη περίπτωσης η αύξηση στο μέσο όρο των απαντήσεων είναι ιδιαίτερη για το επίπεδό τους στο αντικείμενο του διαδικασιακού προγραμματισμού (από 2.67 σε 3.11), γεγονός που υποδηλώνει τη θετική επίδραση του μαθήματος γενικά για το αντικείμενο του Προγραμματισμού. Αξιολογούν, αρκετά υψηλά (3.22) την ικανότητά τους να γράψουν προγράμματα χρησιμοποιώντας τη γλώσσα προγραμματισμού Java.

Στάση των φοιτητών απέναντι στο αντικείμενο του ΑΠ

Πίνακας 4.24. Προτιμήσεις και στάσεις των φοιτητών σχετικά με το αντικείμενο του ΑΠ (case study-3)

Ερώτηση	Μ.Ο. (Από το “pre- test”)	Μ.Ο. (Από το “post- test”)
Σας αρέσει το αντικείμενο του Προγραμματισμού (ΑΠ στο “post-test”);	4.39	4.23
Δυσκολεύεστε με τον Προγραμματισμό (ΑΠ στο “post-test”);	2.78	2.77
Αντιμετωπίζετε γενικά με φόβο τον Προγραμματισμό (ΑΠ στο “post-test”);	1.89	2.17

Θεωρείτε ότι ο Προγραμματισμός (ΑΠ στο “post-test”) απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;	3.44	3.56
Θεωρείτε ότι μάθατε Προγραμματισμό (ΑΠ στο “post-test”) με αυτά που έχετε διδαχθεί έως τώρα;	3.06	3.33

Κωδικοποίηση απαντήσεων:

5 = Πολύ... 1 = Καθόλου

* ερώτηση μη διαθέσιμη στο “pre-test”

- Η θετική στάση απέναντι στο αντικείμενο, κρίνεται αρκετά υψηλή και διατηρείται και μετά το τέλος του μαθήματος (4.39 πριν, 4.23 μετά).
- Τις δυσκολίες τους τις αξιολόγησαν ίδια πριν και μετά.
- Οι φόβοι των φοιτητών, φαίνεται να έχουν αυξηθεί (χωρίς να είναι ιδιαίτερα υψηλοί) μετά τη διεξαγωγή μαθήματος. Και σε αυτή την περίπτωση, το στοιχείο αυτό θα συζητηθεί στη συνέχεια σε συνδυασμό και με επόμενα ευρήματα.
- Μετά το τέλος του σεμιναρίου θεωρούν “ότι ο ΑΠ απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια” περίπου στον ίδιο βαθμό που το θεωρούσαν και πριν.
- Έκριναν ότι η γνώση που έχουν αποκτήσει σχετικά με τον ΑΠ είναι καλύτερη μετά το μάθημα (από 3.06 αυξήθηκε σε 3.33 κατά μέσο όρο).

Και σε αυτό το case study, το αποτέλεσμα αυτό συγκρινόμενο με το αντίστοιχο του case study-1 (όπου υπήρξε αύξηση από 2.1 σε 3.8), μπορεί να δείχνει υποδεέστερο, αλλά πάλι θα πρέπει να συνεκτιμηθούν και οι διαφορετικοί παράγοντες υλοποίησης των δύο case-studies (1 και 3). Στην πρώτη περίπτωση είχαμε πολύ μεγάλο αριθμό διαλέξεων, ενώ στη δεύτερη η διεξαγωγή έγινε με τη μορφή e-learning σεμιναριακού μαθήματος. Επίσης σε αυτό το case study, οι φοιτητές αξιολόγησαν σχετικά υψηλά τη γνώση τους από πριν (3.06 στην κλίμακα από 1 έως 5).

Στις γενικές ερωτήσεις ανοικτού τύπου:

- “Τι είναι αυτό που σας άρεσε περισσότερο στον Προγραμματισμό (ΑΠ στο post-test) από τη μέχρι τώρα επαφή σας με το αντικείμενο;”

Στο “pre-test” κυριάρχησαν οι παρακάτω απαντήσεις: “Η οργανωτική σκέψη και η μεθοδολογία που χρησιμοποιείται. Επιπλέον η ικανότητα να δημιουργώ εφαρμογές είναι αρκετά συναρπαστική”, “Η δυνατότητα να κάνω τη σκέψη, αλλά κυρίως τη φαντασία μου, πράξη έστω και σε εικονικά περιβάλλοντα”, “Ακρίβεια, αφαίρεση,

αποτελεσματικότητα”, “Η χαρά της δημιουργίας όταν το πρόγραμμα επιτέλους “τρέχει””, “Ότι υπάρχει η δυνατότητα συνεχούς εξέλιξης και έρευνας πάνω στο αντικείμενο, αφού είναι κάτι που μεταβάλλεται συνεχώς και δυναμικά”, “Οι πολλές εφαρμογές που έχει το αντικείμενο του Προγραμματισμού και οι ποικίλες λύσεις που δίνει σε διάφορα προβλήματα”, “Η δυνατότητα να σχεδιάζεις κάτι και να το υλοποιείς με βάση κάποιο στόχο/σκοπό. Η δυνατότητα να βλέπεις να δουλεύει το αντικείμενο της εργασίας σου”, “Η οργανωμένη διαχείριση στην εξεύρεση λύσεων ενός προβλήματος”, “Το γεγονός ότι μπορείς να δημιουργήσεις προγράμματα με το δικό σου προσωπικό στίγμα”, “Δομημένος τρόπος επίλυσης προβλημάτων”.

Και εδώ έχουμε αρκετά σχόλια για τη δημιουργικότητα που χαρακτηρίζει το αντικείμενο και επιπλέον το εξελιζόμενο του, τις πολλές εφαρμογές του και την ακρίβεια-οργάνωση που απαιτεί. Οι περισσότεροι από τους φοιτητές, λόγω μεταπτυχιακού επιπέδου σπουδών, έχουν επισημάνει σημαντικά χαρακτηριστικά του αντικειμένου.

Στο “post-test” δόθηκαν απαντήσεις όπως: “Το ότι οι έννοιες και ο τρόπος που συνδέονται είναι τόσο ξεκάθαρες που σου δίνουν τη δυνατότητα να λύσεις ένα πρόβλημα ή τουλάχιστον ένα μέρος ενός προβλήματος όταν ακόμα βρίσκεσαι στο στάδιο της επεξεργασίας, πριν “καθίσεις” στον υπολογιστή”, “...Τα χαρακτηριστικά του OOP (encapsulation, information hiding, inheritance, polymorphism)”, “Η όλη νοοτροπία δομής και ανάπτυξης μιας πλήρους εφαρμογής”, “Η δυνατότητα που παρέχεται για δόμηση της πληροφορίας με τη χρήση κλάσεων / αντικειμένων”, “Η μεγάλη δυνατότητα οργάνωσης που παρέχει καθώς και η επί μέρους αντιμετώπιση του κάθε μερικού προβλήματος σε σχέση με το όλο”, “Στον αντικειμενοστρεφή προγραμματισμό μου αρέσει το γεγονός ότι το πρόγραμμα χωρίζεται σε κλάσεις, που η καθεμία αποτελεί μια ξεχωριστή οντότητα στο πρόγραμμα. Ο τρόπος σκέψης σε αυτόν είναι παρόμοιος με τον τρόπο που σκέφτομαι γενικά όταν γράφω ένα πρόγραμμα”, “Το γεγονός ότι πρέπει να διατηρείς στο μυαλό σου μία δομή, να σκέφτεσαι συνέχεια με αυτή τη δομή και να μην παρεκκλίνεις από αυτή. Μέσω αυτής της διαδικασίας η μεθοδολογία είναι απαραίτητη, διαφορετικά εύκολα μπορεί κανείς να “χαθεί”. Επειδή λοιπόν είμαι άνθρωπος που του αρέσει ιδιαίτερα η σωστή οργάνωση ο αντικειμενοστρεφής προγραμματισμός πιστεύω ότι τελικά θα μου “ταιριάξει””, “Ότι μπορείς να λύσεις κάθε πρόβλημα έχοντας κατά νου το πραγματικό μοντέλο που θέλει λύση. Με άλλα λόγια η ανάλυση ενός προβλήματος με

χρήση αντικειμενοστρεφών αρχών φαντάζει πιο εύκολη σε σχέση με την ανάλυση που θα χρειαζόταν ένα πρόβλημα στο διαδικασιακό προγραμματισμό”, “Διαφορετική προσέγγιση της εκάστοτε λύσης με δυνατότητα επέκτασης και χρήσης για άλλο πρόβλημα”, “Ότι μπορείς να κάνεις κάτι με πολλούς διαφορετικούς τρόπους”, “Αν και θεωρούσα πιο εύκολο να προγραμματίζω διαφορετικά τις διάφορες εφαρμογές (ASSEMBLY) και παρόλο ότι συνάντησα δυσκολίες στη δομή του αντικειμενοστρεφούς προγραμματισμού, εντούτοις πιστεύω ότι αν κάποιος καταλάβει τη θεωρητική δομή του αντικειμενοστρεφούς προγραμματισμού, είναι πολύ πιο πρακτικός, εξελίξιμος και επαναχρησιμοποιήσιμος τρόπος”, “Η δυνατότητα σύνδεσης του με παραδείγματα από την καθημερινή μας ζωή”.

Όπως και στη δεύτερη μελέτη περίπτωσης έτσι και εδώ, οι απαντήσεις δείχνουν έντονα ότι οι φοιτητές έχουν αντιληφθεί σε πολύ μεγάλο βαθμό τη φιλοσοφία του ΑΠ, τον τρόπο σκέψης που απαιτεί, τα οφέλη χρήσης του, τη συνάφειά του με τον τρόπο λειτουργίας των οντοτήτων στον πραγματικό κόσμο και συνεπώς την επίτευξη του στόχου της προτεινόμενης διδακτικής προσέγγισης που αναφέρθηκε στην ενότητα 3.4. περί της ανάπτυξης Αντικειμενοστρεφούς σκέψης επίλυσης προβλημάτων. Επίσης οι απαντήσεις των φοιτητών, δηλώνουν τη θετική επίδραση της χρήσης του αυθεντικού παραδείγματος και γενικώς αυθεντικών δραστηριοτήτων κατά την εφαρμογή της διδακτικής προσέγγισης (ενότητα 3.5.).

- “Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Προγραμματισμό (ΑΠ στο post-test) από τη μέχρι τώρα επαφή σας με το αντικείμενο;”
- Οι απαντήσεις στο “pre-test” περιελάμβαναν: “Το βασικό πρόβλημα είναι η οργάνωση του κώδικα σε λειτουργικές ενότητες”, “Όταν δεν υπάρχει σαφήνεια στο πού βρίσκεται το συντακτικό λάθος στον κώδικα”, “Νομίζω ότι απαιτείται αρκετή προσπάθεια για εμβάθυνση στο αντικείμενο την οποία προφανώς δεν έχω κάνει”, “Το γεγονός ότι δε σταματάει ποτέ η μελέτη και το ψάξιμο”, “Δε διδάχτηκα ποτέ σωστά μια ή περισσότερες μεθοδολογίες Προγραμματισμού”, “Διαφορετική αποτύπωση σύνταξης για το ίδιο πρόβλημα από διάφορες γλώσσες”, “Οι αλλαγές στη φιλοσοφία του Προγραμματισμού από structured σε OOP”, “Λίγος χρόνος ενασχόλησης λόγω δουλειάς, έλλειψη ικανού αριθμού ωρών διδασκαλίας και απάντησης αποριών”, “Η αρχική επαφή είναι πάντα δύσκολη μέχρι να “κατακτηθεί” κάποια γνώση. Θέλει πολύ τριβή και εξάσκηση που δεν ήταν πάντα σε σωστά

οργανωμένο περιβάλλον”, “Δυσκολία αντιμετώπιση στην Assembly. Οι εντολές ήταν πάρα πολλές και τα προγράμματα τεράστια, οπότε ήταν εύκολο να χαθείς”.

Το πνεύμα των απαντήσεων δηλώνει *δυσκολίες κωδικοποίησης και εύρεσης των συντακτικών λαθών, δυσκολίες με τον τρόπο διδασκαλίας, μη ύπαρξη αρκετού χρόνου για την προσπάθεια που απαιτεί το αντικείμενο, δυσκολίες με γλώσσες προγραμματισμού χαμηλού επιπέδου (low-level programming languages)*.

Στο “*post-test*” ανέφεραν: “Η ημιμαθής ενασχόλησή μου με τον προγραμματισμό γενικότερα, ήταν εμπόδιο στην ταχύτητα με την οποία άρχισα να αντιλαμβάνομαι τα πράγματα στον αντικειμενοστραφή προγραμματισμό. Όχι γιατί δεν ήξερα ήδη αρκετά πράγματα, αλλά γιατί ήξερα κάποια χωρίς να τα έχω κατανοήσει πλήρως”, “Το πέρασμα από το traditional programming στο OOP”, “Η έλλειψη χρόνου, ιδίως στο τέλος του εξαμήνου δημιούργησε κενά στο τελευταίο κεφάλαιο (7^ο)”, “Η πρώτη επαφή μου με τη Visual Basic δε μου έδωσε την εντύπωση ότι πρόκειται για μία ιδιαίτερα δύσκολη γλώσσα προγραμματισμού. Ωστόσο για τη Java δε σχημάτισα την ίδια άποψη. Με δυσκόλεψε το γεγονός ότι τα πάντα στη Java αποτελούν κλάσεις. Υποθέτω ότι μου φαινόταν πιο απλή η δομή της Visual Basic. Επίσης στη Java μου έδινε την αίσθηση ότι “χτυπούσε” συνεχώς, για μια αγκύλη, για ένα κεφαλαίο γράμμα, για το παραμικρό και αυτό ήταν κουραστικό”, “Η εμπειρία που είχα μέχρι τώρα σε όσα αφορούν τον προγραμματισμό είχε να κάνει μόνο με γλώσσες μη αντικειμενοστραφείς. Αυτό με δυσκόλεψε ιδιαίτερα καθώς δεν μπορούσα να ξεφύγω από την “παλιομοδίτικη” λογική της basic ή της pascal, γλώσσες στις οποίες είχα δουλέψει στο παρελθόν αρκετά”, “Δυσκολεύτηκα να καταλάβω κάποια πράγματα που έχουν να κάνουν με το inheritance, interface, πολυμορφισμό”, “Η κατανόηση κάποιων συγκεκριμένων εννοιών τις οποίες όμως τώρα πιστεύω ότι τις έχω κατανοήσει σε αρκετά καλό βαθμό”, “Η γενικότερη και πιο ολοκληρωμένη αντίληψη όλων των σχετιζόμενων εννοιών. Αν κατανοήσεις από την αρχή βασικές έννοιες αντιμετώπιζεις τα πράγματα με περισσότερη ευκολία”, “Η διαχείριση του αντικειμένου και θέματα κληρονομικότητας και πολυμορφισμού”, “Ο Αντικειμενοστρεφής Προγραμματισμός απαιτεί αρκετό χρόνο απασχόλησης για να μπορέσει να διδαχθεί παράλληλα με άλλο μάθημα”, “Η κατανόηση της δομής του και η πρακτική χρησιμοποίηση των εννοιών της κλάσης”.

Όπως και στη δεύτερη μελέτη περίπτωσης έτσι και εδώ, κυριαρχούν απαντήσεις που έχουν να κάνουν με τη διαφορετικότητα της φιλοσοφίας του ΑΠ (που όμως οι απαντήσεις στην παραπάνω ερώτηση του post-test “Τι είναι αυτό που σας άρεσε περισσότερο στον ΑΠ από τη μέχρι τώρα επαφή σας με το αντικείμενο;” δείχνουν μάλλον ότι πρόκειται για κάτι που τους δυσκολεύει μεν, αλλά το αντιλήφθηκαν δε), με δυσκολία στις έννοιες των τελευταίων διδακτικών ενοτήτων, αλλά και απαντήσεις που δηλώνουν δυσκολίες που ξεπεράστηκαν στην πορεία και επιπλέον την έλλειψη χρόνου για να ασχοληθούν όσο θα ήθελαν με το αντικείμενο.

Συμβολή παραγόντων του τρόπου εφαρμογής της διδακτικής προσέγγισης στη μαθησιακή διαδικασία

Πίνακας 4.25. Συμβολή παραγόντων της διδακτικής προσέγγισης στη βελτίωση γνώσεων και δεξιοτήτων (case study-3)

Ερώτηση	M.O. (Από το “pre- test”)	M.O. (Από το “post- test”)
Αξιολογείστε τη σημασία που θεωρείτε ότι (θα έχουν) είχαν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας.		
Υπαρξη μελετών περίπτωσης (case studies) (δηλαδή ολοκληρωμένων παραδειγμάτων) στο μαθησιακό υλικό.	4.50	4.39
Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.	4.22	3.72
Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.	4.56	4.28
Η επίλυση επιλεγμένων ασκήσεων στο πλαίσιο του μαθήματος.	4.39	4.28
Η εκπόνηση εκτενών εργασιών στο πλαίσιο του μαθήματος.	4.50	4.00
Η ενασχόληση με τα περιβάλλοντα που θα είναι διαθέσιμα στο e-learning μάθημα.	4.50	*
Η δομή του e-learning μαθήματος, έτσι ώστε να σε οδηγεί “από το απλό στο σύνθετο” δηλ. από απλούστερες έννοιες να σε εισάγει σταδιακά (“βήμα-βήμα”) σε δυσκολότερες.	*	4.28
Η οργάνωση της ύλης κάθε διδακτικής ενότητας του e-learning μαθήματος (με σύντομη θεωρία, παραδείγματα με κώδικα, λυμένες ασκήσεις, δραστηριότητες σε εκπαιδευτικά περιβάλλοντα).	*	4.44

Κωδικοποίηση απαντήσεων: 5 = Πολύ σημαντική... 1 = Πολύ ασήμαντη

* ερώτηση μη διαθέσιμη στο “pre-test” ή το “post-test”

Από τον Πίνακα 4.25. συνάγονται τα εξής:

- Όλοι οι παράγοντες εκτιμήθηκαν από τους φοιτητές ότι συνέβαλαν κατά πολύ (>4 στην κλίμακα από 1 έως 5) στη βελτίωση των γνώσεων και των δεξιοτήτων τους. Και σε αυτή τη μελέτη περίπτωσης, δικαιώνεται η υιοθέτηση των αρχών (ενότητα 3.4.) που αναφέρθηκαν στην υποενότητα 4.4.4.1., στην εκπαιδευτική φιλοσοφία της προτεινόμενης διδακτικής προσέγγισης.
- Και σε αυτή την περίπτωση όπως στο δεύτερο case study, κάποια μικρή κάμψη όπου παρατηρείται στους Μ.Ο. από το “post-test”, εξηγείται από το γεγονός ότι οι φοιτητές αρχικά έχουν πολύ μεγάλες προσδοκίες συνήθως, ιδίως από ένα μάθημα που διεξάγεται με τρόπο διαφορετικό από τον παραδοσιακό. Η μεγαλύτερη κάμψη (0.5 μονάδα) που παρατηρείται στον παράγοντα “εκπόνηση εκτενών εργασιών στο πλαίσιο του μαθήματος”, κατά ένα μέρος οφείλεται στην έλλειψη χρόνου για την εκπόνηση και παράδοση των εργασιών από τη μεριά των φοιτητών, για τους λόγους που επανειλημμένα αναφέρθηκαν προηγουμένως.
- Και εδώ η επιλογή της σταδιακής εισαγωγής των εννοιών “από το απλό στο σύνθετο” και η χρήση ποικιλίας μαθησιακών πόρων για την παρουσίαση εννοιών αξιολογήθηκαν πολύ υψηλά (4.28 και 4.44 αντίστοιχα).
- Η χαμηλότερη αξιολόγηση του παράγοντα “δυνατότητα συζητήσεων με συμμαθητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος” στο “post-test” (3.72) σε σχέση με άλλους, οφείλεται επίσης σε μεγάλο βαθμό στην έλλειψη χρόνου των φοιτητών για την επαρκή αξιοποίηση αυτού του παράγοντα, όπως προκύπτει και από ευρήματα που θα αναφερθούν στη συνέχεια.
- Αντίθετα η υψηλή αξιολόγηση του παράγοντα “δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος” (4.56 πριν, 4.28 μετά), επιβεβαιώνει τη σημαντικότητα και την ανάγκη επικοινωνίας των φοιτητών με τους διδάσκοντες, όπως ακριβώς διαπιστώθηκε και στην πρώτη μελέτη περίπτωσης (υποενότητα 4.4.4.1.).

Συμβολή μαθησιακού υλικού και διδακτικής προσέγγισης

Πίνακας 4.26. Αξιολόγηση των μαθησιακών πόρων του υλικού μεμονωμένα (case study-3)

Ερώτηση	
Αξιολογήστε το βαθμό με τον οποίο οι παρακάτω μαθησιακοί πόροι συνέβαλαν στη βελτίωση των γνώσεων και των	Μ.Ο. (Από το “post-test”)

δεξιοτήτων σας.	
Λυμένες ασκήσεις, παραδείγματα.	4.39
Αναπάντητες ερωτήσεις, άλυτες ασκήσεις.	3.72
Η μελέτη περίπτωσης (case study).	4.06
Θεωρία.	4.17
Οι δραστηριότητες στο περιβάλλον BlueJ.	4.17
Οι δραστηριότητες στο περιβάλλον ArgoUML.	3.28
Τα αρχεία κώδικα Java που περιλαμβάνονται στο υλικό.	3.89
Το ολοκληρωμένο παράδειγμα που παρέδωσα (εφαρμογή βιβλιοπωλείου).	4.06
Τα υλοποιημένα projects στο περιβάλλον BlueJ που περιλαμβάνονται στο υλικό.	3.94
Οι οδηγίες μελέτης της κάθε διδακτικής ενότητας.	4.06
Το υλικό που προήλθε από τις συζητήσεις στο forum του e-learning μαθήματος.	3.11
Οι συναντήσεις του e-learning μαθήματος και οι διαλέξεις σε αυτές τις συναντήσεις.	3.89

Κωδικοποίηση απαντήσεων: 5 = Πολύ σημαντικός... 1 = Πολύ ασήμαντος

Από τον παραπάνω Πίνακα μπορούν να εξαχθούν τα ακόλουθα συμπεράσματα:

- Εξακολουθεί η πρωτοπορία στην αξιολόγηση των λυμένων ασκήσεων και των παραδειγμάτων του υλικού και σε αυτή τη μελέτη περίπτωσης.
- Οι επόμενοι σημαντικότεροι μαθησιακοί πόροι είναι η θεωρία, οι δραστηριότητες στο περιβάλλον BlueJ, η μεσαίου μεγέθους εφαρμογή που εκπόνησαν και παρέδωσαν (υψηλή αξιολόγηση με Μ.Ο. 4.06 που **δικαιώνει** –όπως και στη δεύτερη μελέτη περίπτωσης– **την απόφαση να ενσωματωθεί στην προτεινόμενη διδακτική προσέγγιση μια τέτοιου είδους εργασία**), η μελέτη περίπτωσης (case study), οι οδηγίες μελέτης που περιέχονταν στο υλικό για την κάθε διδακτική ενότητα, τα υλοποιημένα projects στο περιβάλλον BlueJ, ο πηγαίος κώδικας των παραδειγμάτων και των ασκήσεων που περιέχεται στο υλικό, οι δια ζώσης συναντήσεις του e-learning μαθήματος.
- Ακολουθούν στην αξιολόγηση οι επί μέρους εργασίες (αναπάντητες ερωτήσεις, άλυτες ασκήσεις) που ήταν στην ευχέρεια των φοιτητών να εκπονήσουν για δική τους εξάσκηση, οι δραστηριότητες στο περιβάλλον ArgoUML και το υλικό των συζητήσεων στο forum. Η έλλειψη χρόνου μπορεί να εξηγήσει και τη χαμηλότερη αξιολόγηση του πρώτου και του τρίτου από τους παραπάνω παράγοντες. Για το δεύτερο θα γίνει αναφορά στη συνέχεια.

Πίνακας 4.27. Αξιολόγηση του μαθησιακού υλικού στο σύνολό του και της διδακτικής προσέγγισης (case study-3)

Ερώτηση	M.O. (Από το “post-test”)
Το σεμινάριο με βοήθησε να ξεπεράσω αδυναμίες μου σε σχέση με τον ΑΠ.	3.67
Το σεμινάριο με βοήθησε να ξεπεράσω φοβίες που είχα σε σχέση με τον ΑΠ.	3.22
Πώς θα χαρακτηρίζατε το περιεχόμενο του μαθησιακού υλικού; *	4.06
Το περιεχόμενο του μαθησιακού υλικού επιτυγχάνει τους στόχους που αναφέρονται στην αρχή κάθε διδακτικής ενότητας.	3.94
Οι ερωτήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους στόχους εκμάθησης.	3.89
Οι ασκήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους στόχους εκμάθησης.	3.83
Ικανοποιήθηκαν οι λόγοι για τους οποίους παρακολουθήσατε; **	3.50

Κωδικοποίηση απαντήσεων: 5 = Συμφωνώ απόλυτα... 1 = Διαφωνώ απόλυτα

*Κωδικοποίηση απαντήσεων: 5 = Πολύ ενδιαφέρον, 4 = Ενδιαφέρον, 3 = Αδιάφορο, 2 = Κουραστικό, 1 = Δυσνόητο

**Κωδικοποίηση απαντήσεων: 5 = Πολύ... 1 = Καθόλου

Από τον Πίνακα 4.27. συμπεραίνονται τα εξής:

- Η διδακτική προσέγγιση και το προσφερόμενο μαθησιακό υλικό, παρείχε σημαντική βοήθεια στους φοιτητές για να ξεπεράσουν αδυναμίες και φόβους τους σχετικά με τον ΑΠ. Παρατηρείται ότι και στις δύο προηγούμενες μελέτες περίπτωσης, δηλαδή παρότι στον Πίνακα 4.24. φαίνεται ότι οι φόβοι των φοιτητών σχετικά με το αντικείμενο του ΑΠ αυξήθηκαν μετά το τέλος του σεμιναρίου (από 1.89 σε 2.17), εδώ φαίνεται ότι εκτιμούν αρκετά τη συμβολή του σεμιναρίου στο να ξεπεράσουν φοβίες σε σχέση με τον ΑΠ (3.22).
- Η ποιότητα του μαθησιακού υλικού αξιολογήθηκε πολύ υψηλά (4.06 κατά M.O. στην κλίμακα από 1 έως 5) και σε αυτή τη μελέτη περίπτωσης. Η καταλληλότητα του μαθησιακού υλικού και των ερωτήσεων-ασκήσεων για την επίτευξη των μαθησιακών στόχων, κρίθηκαν επίσης θετικά (κοντά στο 4, στην κλίμακα από 1 έως 5), αποδεικνύοντας και σε αυτή τη μελέτη περίπτωσης την πολύ επαρκή ικανοποίηση της αρχής που υιοθετήθηκε στην προτεινόμενη διδακτική προσέγγιση (ενότητα 3.4.), “**ιδιαίτερη προσοχή στην ποιότητα του μαθησιακού υλικού που σχεδιάζεται**” (Kölling and Rosenberg, 2001b; Buck and Stucki, 2000; Fleury, 2000; Bergin, 2006a, pattern “Quality is Job One”).
- Το επίπεδο ικανοποίησης από το σεμινάριο ήταν υψηλό (3.5) παρά τις δυσκολίες

χρόνου που οι φοιτητές αντιμετώπιζαν.

Μαθησιακά εργαλεία

Οι απαντήσεις των φοιτητών σχετικά με τα σημαντικά χαρακτηριστικά του περιβάλλοντος BlueJ, φαίνονται στον παρακάτω Πίνακα (Georgantaki and Retalis, 2007b).

Πίνακας 4.28. Αξιολόγηση της ευχρηστίας του εκπαιδευτικού προγραμματιστικού περιβάλλοντος BlueJ (case study-3)

Ερώτηση	Αριθμός φοιτητών που επέλεξαν το χαρακτηριστικό (max 18)
Ποια από τα παρακάτω χαρακτηριστικά του BlueJ σας βοήθησαν στη μαθησιακή διαδικασία; Επιλέξτε αυτά που νομίζετε.	
Οι κλάσεις και τα αντικείμενα αποκτούν οπτική υπόσταση.	12
Εύκολη δημιουργία αντικειμένων και οπτικοποίηση τους.	12
Ευκολία χρήσης του περιβάλλοντος.	11
Εύκολος έλεγχος των τιμών των ιδιοχαρακτηριστικών των αντικειμένων.	9
Επιτρέπεται να αλληλεπιδράς απευθείας με τις κλάσεις και τα αντικείμενα.	9
Εύκολος πειραματισμός με άμεσο αποτέλεσμα (feedback).	8
Εύκολη εκτέλεση των μεθόδων και έλεγχός τους (εύκολο πέρασμα παραμέτρων και επιστροφή αποτελέσματος), χωρίς να χρειάζεται να γράψεις κώδικα γι' αυτό το σκοπό.	8
Εύκολο compilation με επισήμανση των συντακτικών λαθών απευθείας στο παράθυρο του editor με μήνυμα και επισήμανση της γραμμής κώδικα με το λάθος.	8
Διαθέτει όψη "implementation" και όψη "interface" για τις κλάσεις.	4
Διαθέτει debugger εύκολο στη χρήση.	5

Όπως και στις προηγούμενες μελέτες περίπτωσης, τα "δυνατά" χαρακτηριστικά του BlueJ, ήταν η οπτικοποίηση των κλάσεων και των αντικειμένων, η απλότητά του και η δυνατότητα αλληλεπίδρασης που προσφέρει (Kölling et al., 2003; Kölling and Rosenberg, 2001b).

- Στην ερώτηση “Σε τι κυρίως πιστεύετε ότι σας βοήθησε το περιβάλλον BlueJ; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ότι άλλο εσείς νομίζετε)”

Οι φοιτητές απάντησαν:

- ◆ Στο ότι αναπαριστά απ’ ευθείας τη σχέση κλάσεων σε ένα project.
- ◆ Δε με βοηθά καθόλου προτιμώ περιβάλλοντα IDE – Sun ONE Studio etc.
- ◆ Κλάσεις, αντικείμενα, ιδιότητες, δημιουργοί, μέθοδοι και κληρονομικότητα.
- ◆ Το BlueJ βοήθησε πολύ στην κατανόηση των εννοιών κλάση, αντικείμενο, κληρονομικότητα και άλλες έννοιες. Μου φάνηκε πολύ καλό περιβάλλον και πολύ ευχάριστο. Μου άρεσε πολύ.
- ◆ Είναι ένα καλό εργαλείο που χωρίς να σου τα δίνει όλα έτοιμα σε βοηθάει προσδιορίζοντας το λάθος γρήγορα, χωρίς να είναι ένα “βαρύ” εργαλείο.
- ◆ Κληρονομικότητα.
- ◆ Ιεραρχία κλάσεων, η σχέση “is a” και η δυναμική κλήση μεθόδου (dynamic binding).
- ◆ Τη γενική φιλοσοφία.
- ◆ Εύκολο στο να καταλάβεις πώς δουλεύει και το ότι είναι απλό στη λειτουργία του.
- ◆ Οπτική υπόσταση του Project.
- ◆ Μέσω του BlueJ, κατάλαβα περισσότερο τις έννοιες του αντικειμένου και το διαχωρισμό της από την κλάση. Επίσης την κληρονομικότητα.
- ◆ Προσομοιώνει τον προγραμματισμό σχηματικά στο μυαλό μας, πολύ σημαντικό αυτό.
- ◆ Εύκολος τρόπος εκτέλεσης των εφαρμογών και παρακολούθηση των αντικειμένων.

Αξιολόγηση δυσκολίας των αντικειμενοστρεφών εννοιών – αρχών

Πίνακας 4.29. Αξιολόγηση της δυσκολίας αντικειμενοστρεφών εννοιών – αρχών (case study-3)

Ερώτηση	
Αξιολογήστε σε τι επίπεδο σας δυσκόλεψαν οι παρακάτω έννοιες/θέματα του ΑΠ.	M.O. (Από το “post-test”)
Κλάση - Αντικείμενο - Ιδιότητες - Μέθοδοι	1.44
Προσδιοριστές ορατότητας (public, private, protected)	1.67

Δημιουργία αντικειμένων - Κατανομή μνήμης	1.83
Κληρονομηση ιδιοτήτων – μεθόδων στις Ιεραρχίες κλάσεων (inheritance)	1.83
Μέθοδος Κατασκευαστή (constructor)	1.89
Μεταβλητές και μέθοδοι κλάσης (static members)	1.89
Υπερίσχυση μεθόδων (methods overriding)	1.94
Υπερφόρτωση μεθόδων (methods overloading)	2.00
Κελυφοποίηση - απόκρυψη πληροφορίας (encapsulation-information hiding) – Διαχωρισμός συμπεριφοράς - υλοποίησης	2.06
Αναφορές σε αντικείμενα (references)	2.17
Δημιουργία ιεραρχιών κλάσεων	2.17
Σχέση “είναι” (“is a”)	2.17
Δυναμική κλήση μεθόδου ή Δυναμική δέσμευση (Dynamic Binding)	2.44
Πολυμορφισμός (Polymorphism)	2.44

Κωδικοποίηση απαντήσεων: 3 = Μεγάλη δυσκολία... 1 = Καθόλου δυσκολία

Οι περισσότερο δύσκολες έννοιες-αρχές για τους φοιτητές, όπως προκύπτει από τον παραπάνω πίνακα είναι αυτές που περιέχονται στην 7^η διδακτική ενότητα της προτεινόμενης διδακτικής προσέγγισης αλλά και οι λειτουργίες δημιουργίας αντικειμένων, υπερφόρτωσης μεθόδων και οι έννοιες των τύπων αναφοράς. Κατά πόσον αντεπεξήλθαν στη δυσκολία αυτών των εννοιών θα συζητηθεί στη συνέχεια, στην ανάλυση των εργασιών που παρέδωσαν.

Ποιοτικά αποτελέσματα

Στις ανοικτού τύπου ερωτήσεις του ερωτηματολογίου μετά-αξιολόγησης δόθηκαν απαντήσεις που παρατίθενται στη συνέχεια.

- **Ερώτηση:** Τι σας άρεσε περισσότερο στο μαθησιακό υλικό του e-learning course;
 - ♦ Τα παραδείγματα με απτές καταστάσεις και τα διαγράμματα που συνόδευαν τον κώδικα.
 - ♦ Η όλη δομή από το μηδέν σε εφαρμογές.
 - ♦ Πλούσιο υλικό με αρκετά παραδείγματα.
 - ♦ Η οργάνωση του υλικού.
 - ♦ Μου άρεσαν τα λυμένα παραδείγματα, αν και μου φάνηκαν λίγα. Θα προτιμούσα να υπήρχαν περισσότερα. Μου άρεσε η θεωρία στην αρχή κάθε ενότητας.

- ◆ Το φόρουμ και η άμεση ανταπόκριση της διδάσκουσας σε κάθε απορία και ερώτηση.
- ◆ Η δομή που περιέχουν τα κεφάλαια.
- ◆ Η δομή του, η σύντομη παράθεση της θεωρίας μαζί με παραδείγματα και η εκπόνηση των εργασιών.
- ◆ Η δομή του και αναλυτική εξήγηση όρων.
- ◆ Ήταν ολοκληρωμένο και πλήρες.
- ◆ Η δομή του.
- ◆ Τα υλοποιημένα παραδείγματα. Προσωπικά με βοηθάει καλύτερα η πρακτική κατανόηση του προγραμματισμού απ' ότι μόνο το θεωρητικό κομμάτι.
- ◆ Η βαρύτητα στη φιλοσοφία του προγραμματισμού παρά στην ίδια τη γλώσσα σε συντακτικό επίπεδο.
- ◆ Οι ασκήσεις και τα project που έπρεπε να παραδοθούν.

Οι απαντήσεις των φοιτητών πέραν αυτών που αναφέρονται στη δομή, οργάνωση, θεωρία, παραδείγματα του υλικού, *δηλώνουν τη θετική επίδραση της χρήσης του αυθεντικού παραδείγματος και γενικώς αυθεντικών δραστηριοτήτων* (ενότητα 3.5.), *τα θετικά σχόλια τους για τη χρήση των διαγραμμάτων κλάσεων και αντικειμένων που συνοδεύουν τον κώδικα σε όλο το υλικό, τη χρήση του forum, την έμφαση στη φιλοσοφία του ΑΠ και όχι στη γλώσσα προγραμματισμού.*

- **Ερώτηση:** Τι δε σας άρεσε;
 - ◆ Χρειάζεται περισσότερος χρόνος.
 - ◆ Ο προγραμματισμός θα έπρεπε να είναι πιο προσαρμοσμένος στις ιδιαίτερες ανάγκες κάθε group και στο υπόβαθρό του σε ύλη και χρόνο.
 - ◆ Υπερβολική εμβάθυνση στη Java χωρίς να είναι το βασικό εργαλείο του μαθήματος. Θα μπορούσαμε να δούμε κάποια βασικά πράγματα στη Java και ύστερα να προχωρήσουμε στη JSP. Αυτό θα μας βοηθούσε πραγματικά στο μάθημα μας.
 - ◆ Σε πολλά σημεία το υλικό δεν ήταν πολύ κατανοητό για κάποιον που δεν είναι και τόσο προχωρημένος σε θέματα αντικειμενοστρεφούς προγραμματισμού. Επίσης οι εκφωνήσεις των εργασιών προς παράδοση ήταν μερικές φορές λίγο ασαφείς. Άφησα αναπάντητα κάποια ερωτήματα εξαιτίας αυτού του λόγου.

- ◆ Νομίζω ότι κάποιες φορές μου φάνηκε λίγο κουραστικό. Δηλαδή όταν έχω ήδη κατανοήσει κάτι, ακολουθούσαν παραδείγματα που αναφέρονταν στο αντικείμενο που ήδη είχε κατανοηθεί. Εγώ όμως δεν μπορούσα να τα προσπεράσω και αναγκαστικά “έχανα” χρόνο διαβάζοντας τα.
- ◆ Κάποιες εργασίες μη παραδοτέες θα μπορούσαν να υπάρχουν ώστε κάποιος που ενδιαφέρεται να ασχοληθεί παραπάνω με το αντικείμενο.
- ◆ Η μικρή διάρκεια του προγράμματος.
- ◆ Είχε αυστηρούς κανόνες.
- ◆ Ο φόρτος εργασιών.
- ◆ Αυτό που δε μου άρεσε είναι ότι υπήρχε πολύ λίγος χρόνος για να αφομοιωθεί η γνώση και για να γίνει περισσότερος πειραματισμός από την πλευρά των σπουδαστών πάνω στο αντικείμενο.
- ◆ Οι πολλές εργασίες για τις οποίες δεν υπήρχε καθόλου ελεύθερος χρόνος.
- ◆ Οι ανούσιες ερωτήσεις από συμφοιτητές στο forum.

Από τα παραπάνω προκύπτει κυρίως, η *πίεση χρόνου που αντιμετώπιζαν οι φοιτητές*. Ενδιαφέρον έχουν και δύο αντιφατικές μεταξύ τους απαντήσεις (περί της ασάφειας κάποιων σημείων του υλικού και αντίθετα περί της επανάληψης υλικού που αναφερόταν σε θέματα που ήδη είχαν κατανοηθεί), που όπως φαίνεται η μια από αυτές έχει δοθεί από εντελώς αρχάριο φοιτητή.

- **Ερώτηση:** Τι θα μπορούσε να σας βοηθήσει στην κατανόηση του γνωστικού αντικειμένου και δεν υπάρχει στο υλικό ή στην οργάνωση του e-learning course;
 - ◆ Θα με βοηθούσε αν είχα περισσότερο χρόνο για την κάλυψη του ίδιου μαθησιακού όγκου.
 - ◆ Λίγο καλλίτερα διαμορφωμένες εκφωνήσεις (για τους μη έχοντες ιδιαίτερο υπόβαθρο), λιγότερες ερωτήσεις, περισσότερες διαλέξεις.
 - ◆ Ο χρόνος που μπορούσαμε να αφιερώσουμε –καθώς οι περισσότεροι είμαστε εργαζόμενοι– ήταν δυστυχώς λίγος. Οπότε θα έπρεπε να δούμε ίσως τη Java από πιο πρακτική σκοπιά, ώστε τα αποτελέσματα να είναι πιο ικανοποιητικά. Θα έπρεπε για παράδειγμα, το e-learning, που αναμφίβολα ήταν ενδιαφέρον, να ενταχθεί ίσως στο μάθημα Software Development αντί στο Web Development, γιατί το Software Development είναι περισσότερο σχετικό. Επίσης το Web Development είναι ένα μάθημα που είναι από μόνο του

χαοτικό και η επιπλέον επιβάρυνσή του με το e-learning έχει προκαλέσει απογοήτευση και δυσάρεστα συναισθήματα άγχους και αγωνίας σε πολλούς από εμάς.

- ◆ Παραδείγματα για την ArgoUML σε μορφή φωτογραφικών στιγμιότυπων από έτοιμα παραδείγματα.
- ◆ Κάποια επιπλέον παραδείγματα
- ◆ Το περιεχόμενο του e-learning θα έπρεπε ίσως να εναρμονιστεί με την εξεταστέα ύλη του Web Development, γιατί έτσι μας επιβάρυνε με επιπλέον χρόνο. Ίσως, επειδή το περιεχόμενό του είναι αρκετά ενδιαφέρον και χρήσιμο, αν αξιολογηθεί ότι επιβάλλεται για την απόκτηση του μεταπτυχιακού, θα μπορούσε να αποτελέσει ένα ξεχωριστό μάθημα
- ◆ Περισσότερα παραδείγματα που να εστιάζουν στη φιλοσοφία του προγραμματισμού μέσα από την καθημερινή μας ζωή.

Προκύπτουν από τα παραπάνω και πάλι *προβλήματα χρόνου*. Επιπλέον ζητείται από έναν φοιτητή η *επίδειξη του περιβάλλοντος ArgoUML με φωτογραφικά στιγμιότυπα*.

- **Ερώτηση:** Ποιοι μαθησιακοί πόροι νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;
 - ◆ Νομίζω ότι και εγώ αλλά και το σύνολο των μαθητών θα κερδίζαμε περισσότερα αν αξιοποιούσα τη χρήση του forum περισσότερο.
 - ◆ Κυρίως τα παραδείγματα και η εξοικείωση με την UML.
 - ◆ Η θεωρία που υπάρχει σε κάθε ενότητα. Οι ερωτήσεις, τα παραδείγματα.
 - ◆ Ίσως κάποια σημεία στην περιγραφή του BlueJ και ArgoUML. Ίσως να δοκιμάζατε τη χρήση κάποιων άλλων περιβαλλόντων εργασίας ίσως πιο εμπορικών.
 - ◆ Οι δραστηριότητες στο ArgoUML.
 - ◆ Να υπάρχει περισσότερο έντυπο υλικό για ανάγνωση.

Φαίνεται οι φοιτητές να χρειάζονται *καλύτερη εξοικείωση με την UML και το ArgoUML*.

- **Ερώτηση:** Ποιες ενότητες/τμήματα του υλικού νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;

- ◆ Πολυμορφισμός.
- ◆ Κυρίως τα παραδείγματα και η εξοικείωση με την UML και η προσαρμογή του υλικού στις ανάγκες κάθε τμήματος.
- ◆ Dynamic binding.
- ◆ Θα μπορούσε να υπάρχει αναφορά στα packages της java και σε κάποιες σημαντικές κλάσεις αυτών.
- ◆ Νομίζω η ενότητα 7.

Επιβεβαιώνεται και εδώ η ανάγκη για καλύτερη εξοικείωση με την UML. Επίσης προκύπτει η πρόταση για βελτίωση της 7^{ης} διδακτικής ενότητας και για παρουσίαση κάποιων σημαντικών κλάσεων από τα πακέτα (packages) της Java. Τέτοιες απαντήσεις στην ίδια ερώτηση, είχαμε και στο δεύτερο case study.

Ερώτηση: Σε τι κυρίως πιστεύετε ότι σας βοήθησε το εργαλείο χρήσης της διαγραμματικής τεχνικής UML, ArgoUML; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ό,τι άλλο εσείς νομίζετε)

- ◆ Δε νομίζω ότι με βοήθησε ιδιαίτερα.
- ◆ Όποιος ασχολείται με OOP πρέπει απαραίτητα να ξέρει πολύ καλά και UML.
- ◆ Σε τίποτα το ιδιαίτερο, θεωρώ ότι πέραν του θεωρητικού κομματιού δε βοηθάει σε κάτι –πιο πολύ μπερδεύει στην αρχή– και έχει κενά στην εξοικείωση του.
- ◆ Βοήθησε στην αποσαφήνιση της κληρονομικότητας.
- ◆ Δυστυχώς μάλλον με μπέρδεψε και με δυσκόλεψε παρά με βοήθησε σε κάτι.
- ◆ Δε μου φάνηκε χρήσιμο. Αντιθέτως έχασα χρόνο στο να καταλάβω τον τρόπο που λειτουργεί το περιβάλλον.
- ◆ Γενικές έννοιες και πληροφορίες.
- ◆ Δε με βοήθησε όσο περίμενα.
- ◆ Τον τρόπο οργάνωσης ενός προβλήματος.
- ◆ Αν και το ArgoUML μου φάνηκε αρκετά εύχρηστο εργαλείο για τη δημιουργία διαγραμμάτων, εντούτοις δε με βοήθησε ιδιαίτερα στην κατανόηση του προγραμματισμού, όσο για παράδειγμα το BlueJ ή το Sun One Studio.
- ◆ Νομίζω ότι περισσότερο με μπέρδεψε παρά με βοήθησε.
- ◆ Γενική εκμάθηση UML.

Από τα παραπάνω προκύπτει η ανάγκη οι φοιτητές να βοηθηθούν περισσότερο για την εξοικείωση με το περιβάλλον του ArgoUML, ή να εξευρεθεί καλύτερη λύση για την υποστήριξη της σχεδίασης στην προτεινόμενη διδακτική προσέγγιση.

Από άλλα σχόλια των φοιτητών, προέκυπτε το ενδιαφέρον τους για το e-learning μάθημα του ΑΠ και διατύπωναν την ιδέα να υπήρχε ως ανεξάρτητο μάθημα στο πρόγραμμα σπουδών τους, ώστε να τους επιτρέπουν οι υποχρεώσεις τους να ασχοληθούν μαζί του αποδοτικότερα. Επίσης ζητούσαν περισσότερες “δια ζώσης” συναντήσεις στην τάξη (και μερικές από αυτές να αφορούν μόνο απορίες τους). Η ιδέα του forum τους φάνηκε αρκετά καλή, αλλά συγχρόνως εκφράζουν για άλλη μια φορά, τους χρονικούς τους περιορισμούς που τους ανάγκαζαν να μην το αξιοποιούν όσο και με τον τρόπο που θα ήθελαν.

Οι φοιτητές εκτίμησαν ιδιαίτερα την επιλογή να εισαχθούν στην αντικειμενοστρεφή φιλοσοφία μέσω ανάλυσης και μοντελοποίησης ενός παραδείγματος από την καθημερινή ζωή (“λαϊκή αγορά”). Χαρακτηριστικά, ένας φοιτητής εξέφρασε στο χώρο των ασύγχρονων συζητήσεων πολύ έντονα τον ενθουσιασμό του, μετά την πρώτη “δια ζώσης” συνάντηση του e-learning μαθήματος στην οποία είχε παρουσιαστεί η ανάλυση του παραδείγματος. Έγραψε ότι μόνο με τη χρήση αυτού του παραδείγματος είχε κατανοήσει πλήρως την αντικειμενοστρεφή φιλοσοφία. Αυτός ο φοιτητής είχε και προηγουμένως διδαχθεί ΑΠ και Java και είχε καταναλώσει πολύ περισσότερο χρόνο μελετώντας, απ’ ότι διήρκεσε αυτή η συνάντηση στην τάξη. Εκτίμησε ιδιαιτέρως το παράδειγμα από την καθημερινή ζωή που επελέγη και επίσης τα περιβάλλοντα-εργαλεία που επελέγησαν. Χαρακτηριστικά έγραψε: “*Το χθεσινό παράδειγμα με βοήθησε να καταλάβω ότι δεν κατάλαβα με πολλές ώρες διδασκαλία Java που έχω αλλού παρακολουθήσει. Η σχηματική απεικόνιση στο BlueJ και το παράδειγμα από την καθημερινή ζωή με έκαναν να καταλάβω ότι ο Προγραμματισμός δεν είναι αριθμοί, αλλά τρόπος σκέψης και αυτό είναι το μυστικό του. Μπορεί να ξέρεις 100 γλώσσες προγραμματισμού, αλλά αν δεν ξέρεις να τοποθετείς τα σωστά αντικείμενα/οντότητες με το σωστό τρόπο, τότε δεν ξέρεις Προγραμματισμό. Αισθάνομαι ότι ανακάλυψα την Αμερική μέσα μου*”.

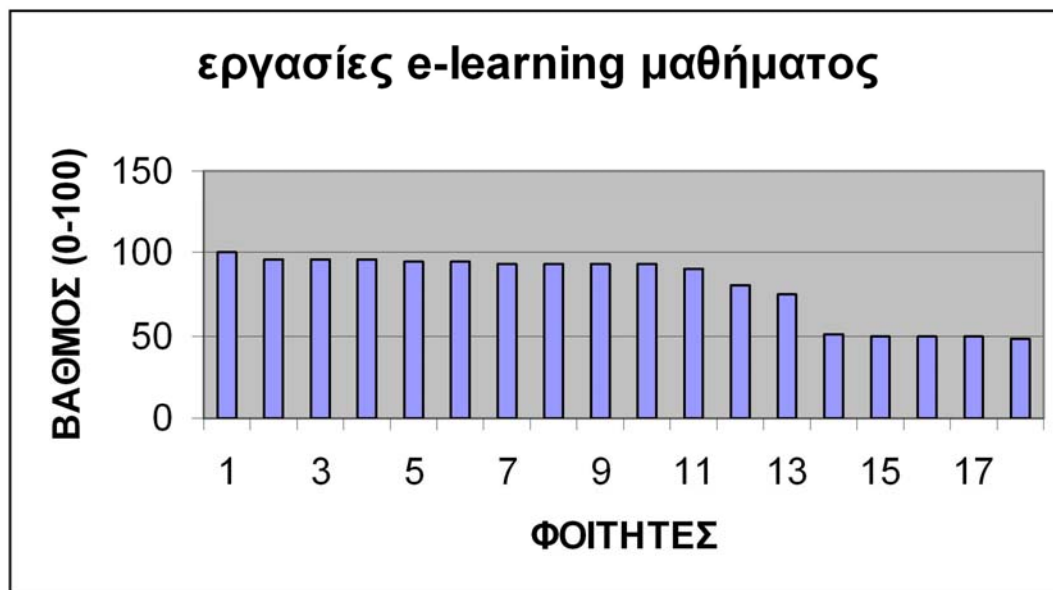
4.6.4.2. Επεξεργασία εργασιών

Μαθησιακά αποτελέσματα

Οι εργασίες που οι φοιτητές εκπόνησαν και παρέδωσαν κατά τη διάρκεια του e-learning σεμιναριακού μαθήματος, περιελάμβαναν κυρίως το μεσαίου μεγέθους project το οποίο υλοποίησαν και παρέδωσαν τμηματικά.

Συνολικά, η ανάλυση των εργασιών τους, έδειξε ότι αντιλήφθηκαν σωστά τις αντικειμενοστρεφείς έννοιες, απέκτησαν αφηρημένη γνώση του αντικειμένου και την εφάρμοσαν με επιτυχία στο πρόβλημα που είχαν να αντιμετωπίσουν. Η μεγάλη πλειοψηφία τους, μπορούσε να μοντελοποιήσει μια αντικειμενοστρεφή εφαρμογή, να σχεδιάσει διαγράμματα κλάσεων στο περιβάλλον ArgoUML και να υλοποιήσει με επιτυχία την εφαρμογή στο περιβάλλον BlueJ ή/και στη συνέχεια στο SUN One Studio.

Η επίδοση των φοιτητών, με κλίμακα βαθμολογίας από 1 έως 100, φαίνεται στο Σχήμα 4.7.



Σχήμα 4.7. Βαθμολογία των εργασιών του e-learning μαθήματος (case study-3)

Όπως προκύπτει, η μεγάλη πλειοψηφία των φοιτητών βαθμολογήθηκε αρκετά υψηλά. Σχετικά με τους πέντε (5) φοιτητές, οι οποίοι κινήθηκαν στη “βάση” της βαθμολογίας, λόγω σοβαρών χρονικών περιορισμών τους δεν μπόρεσαν να παραδώσουν ολοκληρωμένο το τελικό τμήμα της εργασίας τους.

Και σε αυτή τη μελέτη περίπτωσης όπως και στη δεύτερη, παρατηρείται, παρά τη δυσκολία που οι φοιτητές αναφέρουν ότι αντιμετώπισαν σε αντικειμενοστρεφείς έννοιες και αρχές (Πίνακας 4.29.), να αντεπεξέρχονται αρκετά ικανοποιητικά στο να εφαρμόσουν αυτές έννοιες.

4.6.4.3. Ανάπτυξη τελικού project για το μάθημα Web Development

Όπως ήδη αναφέρθηκε, στην τρίτη αυτή μελέτη περίπτωσης η 3^η φάση (Φάση III, ενότητα 3.5.) αφορούσε στην ανάπτυξη ενός τελικού project για Web Development. Σε αυτό περιλαμβανόταν ένα κομμάτι που αφορούσε σε Αντικειμενοστρεφή σχεδίαση, στο οποίο όλοι οι φοιτητές κατάφεραν να ανταποκριθούν με επιτυχία. Το καθ' αυτό project δεν έχει σχέση με τον ΑΠ και δεν ενδιαφέρει περισσότερο. Πάντως όλοι οι φοιτητές πέτυχαν θετικό αποτέλεσμα στη βαθμολογία αυτού του project.

4.6.5. Συμπεράσματα

- Σε αυτή τη μελέτη περίπτωσης αξιολογήθηκε έντονα θετικά η χρήση του αυθεντικού παραδείγματος και γενικώς αυθεντικών δραστηριοτήτων κατά την εφαρμογή της διδακτικής προσέγγισης και επίσης η χρήση της διαγραμματικής τεχνικής, η χρήση του forum, η έμφαση στη φιλοσοφία του ΑΠ και όχι στη γλώσσα προγραμματισμού.
- Οι φοιτητές ανέπτυξαν σχεδιαστικές ικανότητες (design skills), ικανότητες κωδικοποίησης, ελέγχου και αποσφαλμάτωσης (coding, testing and debugging skills) και εφάρμοσαν με επιτυχία τις αντικειμενοστρεφείς έννοιες/αρχές που τους παρουσιάστηκαν και για τις οποίες τους δόθηκε υποστηρικτικό υλικό, στην εφαρμογή που παρέδωσαν.
- Η ανάγκη που προκύπτει είναι να εξοικειωθούν περισσότερο με τη UML και τη χρήση του design tool ArgoUML. Ίσως το παραπάνω να αναδεικνυόταν σε πολύ μικρότερο βαθμό, στην περίπτωση που οι συμμετέχοντες στη μελέτη περίπτωσης δεν είχαν τους χρονικούς περιορισμούς που προαναφέρθηκαν επανειλημμένα και αυτό επέτρεπε τη μεγαλύτερη ενασχόλησή τους, τη διενέργεια περισσότερων “διαζώσης” συναντήσεων και την καλύτερη αξιοποίηση της δυνατότητας των ασύγχρονων συζητήσεων για την επίλυση αποριών τους σε σχέση με τη χρήση του εργαλείου.

Η εξεύρεση μιας καλύτερης λύσης για την υποστήριξη της σχεδίασης στην προτεινόμενη διδακτική προσέγγιση, είναι θέμα υπό διερεύνηση. Η χρήση μιας επέκτασης (extension ή plug-in) στο περιβάλλον BlueJ –το οποίο αξιολογείται σε όλες τις μελέτες περίπτωσης πολύ θετικά για την απλότητα και την ευχρηστία του– που θα επιτρέπει τη σχεδίαση των κλάσεων και την παραγωγή “σκελετικού” κώδικα, θα απάλλασσε τους εκπαιδευόμενους από τις δυσκολίες που αναφέρθηκαν στην τελευταία αυτή μελέτη περίπτωσης και από την ανάγκη εξοικείωσης με ένα επιπλέον περιβάλλον. Περισσότερη συζήτηση αυτής της ιδέας θα γίνει στο επόμενο κεφάλαιο. Συνολικά η αξιολόγηση της τρίτης μελέτης περίπτωσης της προτεινόμενης διδακτικής προσέγγισης (ερωτηματολόγια, εργασίες), δείχνει ότι οι φοιτητές εκτίμησαν τον τρόπο που επελέγη για την εισαγωγή στον ΑΠ και συνεπώς την εκπαιδευτική φιλοσοφία που διέπει την προσέγγιση (ενότητα 3.4.) και επίσης ότι η συνδυασμένη μαθησιακή στρατηγική (blended learning strategy) που ακολουθήθηκε για την εφαρμογή της με συγκεκριμένη ακολουθία μαθησιακών διαδικασιών (learning sequence) (ενότητα 3.5.), υπήρξε αποδοτική ως προς τα μαθησιακά αποτελέσματα, τα αισθήματα και τη στάση των φοιτητών.

4.7. Σύνοψη

Στο κεφάλαιο αυτό έγινε αναλυτική παρουσίαση των εφαρμογών της προτεινόμενης διδακτικής προσέγγισης για τον ΑΠ στη διδακτική πράξη και των μελετών αξιολόγησής τους. Η παρουσίαση, ανέδειξε την εξελικτική ανάπτυξη της προσέγγισης, μέσα από κύκλους αξιολόγησης και βελτιστοποίησης.

Η τελική μορφή, περιέχει μια συγκεκριμένη ροή δραστηριοτήτων, των οποίων ο ακριβής τρόπος υλοποίησης, ιδίως ως προς τα παραδοτέα, μπορεί να διαφοροποιείται ανάλογα με τις συνθήκες της εφαρμογής της προσέγγισης και κυρίως το φόρτο των εκπαιδευομένων.

Με αυτό το κεφάλαιο, ολοκληρώνεται η παρουσίαση της προτεινόμενης διδακτικής προσέγγισης, των εφαρμογών της και της αξιολόγησής της.

Στο επόμενο κεφάλαιο που ακολουθεί παρουσιάζεται μια επισκόπηση των βημάτων που ακολουθήθηκαν κατά την ερευνητική πορεία της διδακτορικής διατριβής, η αξιολόγηση της ερευνητικής πορείας και τα επιτεύγματά της. Αναφέρονται επίσης τα

μελλοντικά ερευνητικά σχέδια που αφορούν στη σχεδιασθείσα διδακτική προσέγγιση για τον ΑΠ.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

РАНЕЕ НЕ ПЕРПА

ΚΕΦΑΛΑΙΟ 5

Συμπεράσματα

5.1. Επισκόπηση της ερευνητικής πορείας

Το ερευνητικό πρόβλημα το οποίο πραγματεύεται η παρούσα διδακτορική διατριβή, είναι η δημιουργία μίας μαθησιακά αποτελεσματικής προσέγγισης για τη διδασκαλία εκπαιδευομένων στη σχεδίαση και ανάπτυξη εφαρμογών Αντικειμενοστρεφούς φιλοσοφίας Προγραμματισμού. Ο στόχος της μεθόδου είναι να βοηθήσει ουσιαστικά (α) στην ανάπτυξη Αντικειμενοστρεφούς σκέψης επίλυσης προβλημάτων (development of OO problem solving skills), (β) στην εμπέδωση βασικών εννοιών-αρχών του παραδείγματος αυτού, (γ) στην ανάπτυξη ικανότητας σχεδίασης, υλοποίησης, ελέγχου, αποσφαλμάτωσης απλών προγραμμάτων σε Java με τη χρήση CASE εργαλείων και προγραμματιστικών περιβαλλόντων και (δ) στην αντιμετώπιση καταγεγραμμένων μαθησιακών προβλημάτων.

Τα βήματα που ακολουθήθηκαν κατά τη διάρκεια της έρευνας, περιελάμβαναν:

- Την εκτενή βιβλιογραφική επισκόπηση τη σχετική με τη διδασκαλία του γνωστικού αντικείμενου του ΑΠ, που αφορά:
 - ◆ στις μαθησιακές δυσκολίες των εκπαιδευομένων που προκύπτουν τόσο από μικρής όσο και από μεγάλης διάρκειας ερευνητικές μελέτες,
 - ◆ στη μελέτη των κύριων διδακτικών προσεγγίσεων και των χαρακτηριστικών τους, που έχουν αναπτυχθεί και προταθεί για τη διδασκαλία του ΑΠ,
 - ◆ στη μελέτη εκπαιδευτικών εργαλείων και προγραμματιστικών περιβαλλόντων που έχουν αναπτυχθεί για την υποστήριξη της διδασκαλίας του ΑΠ,
 - ◆ σε συστάσεις διδασκαλίας που έχουν διατυπωθεί για τη διδασκαλία του ΑΠ σε όλη την παραπάνω βιβλιογραφία.
- Τον καθορισμό της διδακτέας ύλης της απαραίτητης για την εισαγωγή στην Αντικειμενοστρεφή φιλοσοφία Προγραμματισμού και τη μελέτη του οδηγού σπουδών των IEEE/ACM (CC2001, 2001) για τον ΑΠ για το σκοπό αυτό.
- Τη σχεδίαση μαθησιακού υλικού και δραστηριοτήτων σε εκπαιδευτικά εργαλεία και προγραμματιστικά περιβάλλοντα –μέσα στο πλαίσιο ανάπτυξης μιας

ολοκληρωμένης διδακτικής προσέγγισης— και τα δύο προσανατολισμένα στην κατεύθυνση “θεραπείας” καταγεγραμμένων μαθησιακών προβλημάτων.

- Τη δοκιμή ρύθμισης του μαθησιακού υλικού και των εργαλείων σε μια σύντομη εφαρμογή τους (Γεωργαντάκη και Ρετάλης, 2005).
- Τρεις πιλοτικές εφαρμογές της προσέγγισης σε αυθεντικά εκπαιδευτικά περιβάλλοντα στην Τριτοβάθμια εκπαίδευση και τις συστηματικές αξιολογήσεις τους. Με τις αξιολογήσεις διερευνήθηκε η μαθησιακή αποτελεσματικότητα της προσέγγισης και προέκυψαν σημεία που έχρηζαν βελτίωσης. Μέσα από τα συμπεράσματα και τις βελτιώσεις που προέκυπταν σε κάθε εφαρμογή, η διδακτική προσέγγιση αναπτύχθηκε εξελικτικά.
- Την παρουσίαση της τελικής μορφής της προτεινόμενης διδακτικής προσέγγισης η οποία αποτελεί απόρροια των παραπάνω διαδοχικών εφαρμογών, αξιολογήσεων, ενισχύσεων και βελτιώσεων. Η προσέγγιση διέπεται από συγκεκριμένη εκπαιδευτική φιλοσοφία, αρχές, στρατηγικές υλοποίησής της, προτείνει δε μαθησιακές δραστηριότητες με συγκεκριμένη σειρά, κατάλληλο μαθησιακό περιβάλλον για την ένταξή της, αξιοποίηση συγκεκριμένων μαθησιακών εργαλείων και συγκεκριμένους ρόλους για τους εμπλεκόμενους στη μαθησιακή διαδικασία.

5.2. Αποτελέσματα της ερευνητικής πορείας και καινοτομικά στοιχεία

Η τελική προτεινόμενη διδακτική προσέγγιση υιοθετεί τις αρχές της “objects first” φιλοσοφίας με “model first” στρατηγική υλοποίησης, την παιδαγωγική αρχή της εκμάθησης του γνωστικού αντικείμενου κοντά σε ειδικούς (experts), –η οποία αποτελεί χαρακτηριστικό της παιδαγωγικής στρατηγικής της γνωστικής μαθητείας (cognitive apprenticeship)— με πειραματισμό και με έμφαση στη μοντελοποίηση (modeling). Προτείνει να πραγματοποιούνται μαθησιακές δραστηριότητες με μια συγκεκριμένη σειρά (learning sequence) (ενότητα 3.5.) και αξιοποιεί διαδικτυακές τεχνολογίες και εργαλεία.

Συνολικά, η διδακτική προσέγγιση προτείνει:

- τη σταδιακή διδασκαλία Αντικειμενοστρεφών εννοιών (scaffolding) (με εστίαση σε ένα θέμα κάθε φορά) με την αξιοποίηση κατάλληλων εκπαιδευτικών περιβαλλόντων,
- τη χρήση αντιπροσωπευτικών παραδειγμάτων για την παρουσίαση εννοιών,
- την παρουσίαση των δομών της γλώσσας που υλοποιούν τις έννοιες και τη δημιουργία προγραμμάτων με οδηγό τις έννοιες αυτές,
- δραστηριότητες για την ενεργό συμμετοχή των εκπαιδευομένων (active learning) και επίλυση ασκήσεων από τα αρχικά στάδια εκπαίδευσης,
- την καθοδήγηση και σταδιακή υποστήριξη των εκπαιδευομένων μέχρι να συντάξουν κώδικα μόνοι τους, με τη χρήση πολλών έτοιμων παραδειγμάτων κώδικα που τους παρέχονται για ανάγνωση και πειραματισμό, τακτική η οποία σχετίζεται με το μαθησιακό μοντέλο της εφαρμοσμένης “μαθητείας” (applied apprenticeship approach) (Astrachan and Reed, 1995),
- την παρουσίαση και υλοποίηση ολοκληρωμένων προγραμμάτων (μέθοδος main) από τα αρχικά στάδια διδασκαλίας,
- τη χρήση και επαγγελματικού περιβάλλοντος ανάπτυξης εφαρμογών, όπως το SUN One Studio εκτός από ένα εκπαιδευτικό περιβάλλον όπως το BlueJ,
- τη σχεδίαση μαθησιακών πόρων του υλικού με παρεμβάσεις που στοχεύουν στην αντιμετώπιση καταγεγραμμένων μαθησιακών δυσκολιών και συνολικά την προσεχτική σχεδίαση ποιοτικού μαθησιακού υλικού,
- τη χρήση μελετών περίπτωσης (case studies) για την εφαρμογή των αντικειμενοστρεφών εννοιών.

Ως προς το σχεδιασμό του τρόπου εφαρμογής της προσέγγισης και τις επαναλαμβανόμενες εφαρμογές της, υιοθετήθηκαν οι εξής αρχές:

- χρήση της τεχνολογίας και του Διαδικτύου για την επαφή με τους εκπαιδευόμενους και παροχή σε αυτούς εξατομικευμένης βοήθειας,
- υιοθέτηση της αρχής της εκμάθησης του ΑΠ κοντά σε ειδικούς (experts) (παιδαγωγική στρατηγική της γνωστικής μαθητείας, cognitive apprenticeship) (Collins et al., 1989; Brown et al., 1989) με πειραματισμό και με έμφαση στη μοντελοποίηση (modeling),
- υιοθέτηση της συνδυασμένης μαθησιακής στρατηγικής (blended learning strategy) –δηλαδή παραδοσιακής “από-καθέδρας” διδασκαλίας και εμπλοκής των εκπαιδευομένων σε δραστηριότητες με τη χρήση των Διαδικτυακών

τεχνολογιών (μέσω ηλεκτρονικού μαθησιακού περιβάλλοντος)– κατά την εφαρμογή της διδακτικής προσέγγισης και δημιουργία κατάλληλων συνθηκών γι' αυτό,

- καταγραφή των δυσκολιών και των λαθών των εκπαιδευομένων για να ληφθούν υπόψη στη βελτίωση της προσέγγισης,
- επανεξέταση και βελτίωση της προσέγγισης μετά από κάθε εφαρμογή της.

Η παρούσα διατριβή αποτελεί μια συστηματική και μακροχρόνια έρευνα στη διδασκαλία του ΑΠ και μπορεί να συγκριθεί με τις λιγιστές άλλες μακροχρόνιες έρευνες (ενότητα 2.3.).

Πιο συγκεκριμένα, η έρευνα που διεξήχθη κατά τη διάρκεια εκπόνησης της διδακτορικής διατριβής, συγκεντρώνει εμπειρικά δεδομένα επί δύο έτη σε τρεις διαφορετικές εφαρμογές μιας συγκεκριμένης μεθόδου για την εισαγωγή στην Αντικειμενοστρεφή φιλοσοφία Προγραμματισμού, σε αυθεντικά εκπαιδευτικά περιβάλλοντα της Τριτοβάθμιας εκπαίδευσης τόσο σε προπτυχιακό όσο και σε μεταπτυχιακό επίπεδο. *Κάτι ανάλογο σε διάρκεια δε βρέθηκε να έχει επαναληφθεί σε προηγούμενη έρευνα, πλην της περίπτωσης της διδακτορικής διατριβής Ragonis (2004), η οποία πραγματεύεται τη διδασκαλία του ΑΠ με ΟΦ προσέγγιση με τη χρήση της Java και του BlueJ σε μαθητές Δευτεροβάθμιας Εκπαίδευσης (αρχάριους) και τη διερεύνηση των διεργασιών τους και των εντυπώσεων-αντιλήψεων που τους δημιουργήθηκαν. Σε αυτή τη διατριβή διεξήχθη έρευνα επί δύο ακαδημαϊκά έτη (2000-01 και 2001-02) σε μαθητές ηλικίας 15-16 ετών (18 και 29 μαθητές αντίστοιχα κατά χρονιά).*

Στην προαναφερθείσα έρευνα (Ragonis, 2004) –όπως προέκυψε από τη μελέτη της δημοσίευσης Ragonis and Ben-Ari (2005b) που είναι και η πλουσιότερη στην παρουσίαση των ευρημάτων της– τα θέματα διδασκαλίας ως προς τις καθαρά Αντικειμενοστρεφείς έννοιες-θέματα, περιορίστηκαν στις έννοιες κλάση, αντικείμενο, διαδραστική κλήση μεθόδων, δημιουργία κλάσεων και συντεθειμένων κλάσεων. Θέματα όπως υπερφόρτωση μεθόδων, κληρονομικότητα, πολυμορφισμός, δυναμική δέσμευση κ.ά. δεν περιλαμβάνονταν. Δηλαδή το εύρος της διδακτέας ύλης ήταν περιορισμένο και για λόγους στόχων της έρευνας και προσαρμοσμένο ίσως και στο συγκεκριμένο μόνο χώρο εφαρμογής (Δευτεροβάθμια, αρχάριοι).

Αντίθετα, η προτεινόμενη σε αυτή τη διατριβή διδακτική προσέγγιση, καλύπτει σε διδακτικό περιεχόμενο τη θεματική ανάπτυξη που προτείνεται από τις IIEEE και ACM στο CC2001 (2001).

Στην προτεινόμενη προσέγγιση, ενσωματώνεται η χρήση επαγγελματικού περιβάλλοντος ανάπτυξης εφαρμογών για τη θεραπεία προβλημάτων που η ίδια η παραπάνω έρευνα επεσήμανε, δοκιμάστηκε η χρήση και επιπλέον εκπαιδευτικού περιβάλλοντος (προγραμματιστικός μικρόκοσμος Jeroo) και φυσικά ελήφθησαν συνολικά (στη φιλοσοφία της και στην υλοποίησή της) υπόψη οι δυσκολίες που εντόπισε και οι συστάσεις που διατύπωσε η παραπάνω έρευνα.

Επίσης, στην παρούσα έρευνα της διατριβής ο αριθμός των μελετών περίπτωσης που πραγματοποιήθηκαν (3) δεν έχουν πραγματοποιηθεί σε προηγούμενες έρευνες και το δείγμα των συμμετεχόντων είναι ιδιαίτερα αξιολογικό και δεν έχει υπάρξει κάτι αντίστοιχο στην ερευνητική αυτή περιοχή, όπως προέκυψε από τη βιβλιογραφική επισκόπηση. Οι εφαρμογές έγιναν αντίστοιχα με 20, 66 και 18 συμμετέχοντες. Να σημειωθεί εδώ, ότι στο ερευνητικό έργο COOL, που αποτελεί τη σημαντικότερη ίσως ερευνητική προσπάθεια στο χώρο αυτό και το οποίο διεξήχθη με τη συνεργασία τεσσάρων ερευνητικών ιδρυμάτων, ο μέγιστος αριθμός συμμετεχόντων σε πείραμα ήταν 42 άτομα, από τα οποία τα 32 είχαν πλήρη συμμετοχή έως το τέλος του πειράματος. Επιπλέον στην παρούσα έρευνα της διατριβής, οι μελέτες περίπτωσης πραγματοποιήθηκαν σε διαφορετικά περιβάλλοντα, από την άποψη των εκπαιδευτικών ιδρυμάτων (A.E.I., T.E.I.), τις ηλικίες των συμμετεχόντων και το υπόβαθρο των γνώσεών τους. Υπήρξαν επίσης διαφοροποιήσεις ως προς τις συνθήκες εφαρμογής, όπως τα κίνητρα συμμετοχής των συμμετεχόντων, το πλήθος και τον όγκο των παραδοτέων εργασιών, τη συμμετοχή ή όχι στη βαθμολογία υποχρεωτικών μαθημάτων. Συμπερασματικά, η προσέγγιση εφαρμόστηκε και αξιολογήθηκε σε διαφορετικά μεταξύ τους δείγματα και σε διαφορετικές συνθήκες. Κάτι ανάλογο συμβαίνει στο COOL project, το οποίο όμως δεν πραγματεύεται μια διδακτική προσέγγιση για τον ΑΠ, αλλά την πολύπλευρη μελέτη της εκμάθησης του ΑΠ.

Σημαντική πρωτοτυπία αποτελεί η συστηματική μελέτη αξιολόγησης της κάθε εφαρμογής σε χρονοσειρές (επαναλαμβανόμενες στην κάθε εφαρμογή) και η βελτίωση της μεθόδου έπειτα από τις επανειλημμένες αυτές εφαρμογές και αξιολογήσεις, για να

πάρει την τελική προτεινόμενη μορφή της. Ανάμεσα σε έρευνες που έχουν διενεργηθεί δε βρέθηκαν συστηματικές αξιολογήσεις διδακτικών προσεγγίσεων για τη διδασκαλία του ΑΠ. Εντοπίστηκε μια αξιολόγηση στο Moskal et al (2004), που αφορά στη διεξαγωγή CS1 μαθήματος με τη χρήση του εκπαιδευτικού περιβάλλοντος Alice (το οποίο περιγράφεται στην υποενότητα 2.5.1.).

Οι άλλες έρευνες (Ragonis, 2004; Fjuk et al., 2006) δεν αξιοποίησαν προηγούμενα βιβλιογραφικά ευρήματα. Στην προτεινόμενη διδακτική προσέγγιση, από τη διεξοδική βιβλιογραφική επισκόπηση που πραγματοποιήθηκε, συγκεντρώθηκαν αποτελεσματικές “αρχές” (“καλές πρακτικές”, *best practices*) στη διδασκαλία του ΑΠ, που έχουν καταγραφεί ως σύντομες οδηγίες (*guidelines*), ως “σχεδιαστικά πρότυπα” (*design patterns*) ή ως συστάσεις (ενότητα 3.4.) (Georgantaki and Retalis, 2007a; Γεωργαντάκη και Ρετάλης, 2006) και δόθηκε έμφαση σε καταγεγραμμένες μαθησιακές δυσκολίες στην εκμάθηση του ΑΠ, με συγκεκριμένες παρεμβάσεις τόσο στο μαθησιακό υλικό που τη συνοδεύει, όσο και στον τρόπο εφαρμογής της. Το αποτέλεσμα αυτών των παρεμβάσεων ήταν να μην παρατηρηθούν γνωστές μαθησιακές δυσκολίες, αλλά και να διαπιστωθεί η επιμονή κάποιων απ’ αυτές οι οποίες χρήζουν περαιτέρω διερεύνησης για την αντιμετώπιση τους και επίσης να αναδειχθούν άλλες που δεν έχουν αναφερθεί στη βιβλιογραφία (υποενότητες 4.4.4.2. και 4.5.4.2.).

Σε διάφορες έρευνες έχει προταθεί και εφαρμοστεί η “objects first” προσέγγιση στη διδασκαλία του ΑΠ με “model first” υλοποίηση. Κύριες διαφοροποιήσεις της προτεινόμενης στη διδακτορική διατριβή “objects first” προσέγγισης με “model first” στρατηγική υλοποίησης, αποτελούν οι παρεμβάσεις που αυτή ενσωματώνει για την αντιμετώπιση καταγεγραμμένων μαθησιακών προβλημάτων (υποενότητα 3.7.2.), οι συγκεκριμένες επιλογές για τα μαθησιακά εργαλεία (ενότητα 3.8.), η συνδυασμένη μαθησιακή στρατηγική εφαρμογής της προσέγγισης και η ενσωμάτωση σε αυτή χαρακτηριστικών της παιδαγωγικής στρατηγικής της γνωστικής μαθητείας (*cognitive apprenticeship*) (ενότητες 3.5., 3.6., 3.9.). Όλες οι παραπάνω επιλογές αξιολογήθηκαν κατά τις επανειλημμένες εφαρμογές της προτεινόμενης διδακτικής προσέγγισης σε αυθεντικά εκπαιδευτικά περιβάλλοντα.

Η προτεινόμενη προσέγγιση, αποτελεί ουσιαστικά μια τεχνολογικά υποστηριζόμενη ακολουθία μαθησιακών δραστηριοτήτων (ή διαδικασιών) (*learning sequence*)

(ενότητα 3.5.) –δηλαδή ένα μαθησιακό σενάριο (learning script)– με συγκεκριμένα παιδαγωγικά χαρακτηριστικά, που μπορεί να χρησιμοποιηθεί ως ένας οδηγός υλοποίησης της προτεινόμενης “model first” στρατηγικής στη διδασκαλία του ΑΠ σε πραγματικές εκπαιδευτικές συνθήκες. *Τέτοιος οδηγός, δε βρέθηκε στην εκτενή βιβλιογραφική επισκόπηση που πραγματοποιήθηκε.*

5.3. Όροι και συνθήκες εφαρμογής της διδακτικής προσέγγισης όπως προκύπτουν από τις μελέτες περίπτωσης

Στην ενότητα αυτή αναφέρονται οι όροι εφαρμογής της διδακτικής προσέγγισης που προτείνεται από την παρούσα διατριβή και επίσης προδιαγράφονται οι ιδανικές συνθήκες και οι προϋποθέσεις για την επίτευξη του βέλτιστου μαθησιακού αποτελέσματος.

- Η ολοκλήρωση των φάσεων (ενότητα 3.5.) της προσέγγισης, επιτυγχάνεται σε διάστημα τριών (3) μηνών, είτε με αυξημένη φυσική παρουσία του διδάσκοντα (ενότητα 4.4.), είτε δίνοντας έμφαση στην αξιοποίηση των διαδικτυακών τεχνολογιών (ενότητες 4.5., 4.6.).
- Από τις μελέτες περίπτωσης προκύπτει ως βέλτιστο μοντέλο εφαρμογής της το υβριδικό το οποίο συνδυάζει παραδοσιακή “από-καθέδρας” διδασκαλία και εμπλοκή των εκπαιδευομένων σε δραστηριότητες με τη χρήση των Διαδικτυακών τεχνολογιών (μέσω ηλεκτρονικού μαθησιακού περιβάλλοντος).
- Οι ανάγκες της εκπαιδευτικής φιλοσοφίας της προτεινόμενης διδακτικής προσέγγισης απαιτούν τα εξής λογισμικά εργαλεία (ενότητα 3.8.): (i) το UML εργαλείο ArgoUML (<http://argouml.tigris.org/>) για την υποστήριξη της “model first” “πλευράς” της προσέγγισης το οποίο παρέχει τη δυνατότητα παραγωγής “σκελετικού” κώδικα Java, (ii) το εκπαιδευτικό προγραμματιστικό περιβάλλον BlueJ (<http://www.bluej.org/>) που “επιβάλλει” με τα χαρακτηριστικά του τη μοντελοποίηση, υποστηρίζει την οπτικοποίηση εννοιών και σχέσεων και επιτρέπει την αλληλεπίδραση με τις κλάσεις και τα αντικείμενα και (iii) το ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών Java, SUN One Studio. Φυσικά, προϋπόθεση αποτελεί η χρήση του Java2 SDK.
- Επίσης για την ύπαρξη ηλεκτρονικού μαθησιακού περιβάλλοντος (ενότητα 3.6.), απαιτείται να δημιουργούνται δικτυακοί τόποι της εφαρμογής της διδακτικής

προσέγγισης σε ένα Σύστημα Διαχείρισης της Μάθησης (Learning Management System) (LMS), όπως το σύστημα Moodle (<http://www.moodle.org>). Με τη χρήση του, καλύπτονται θέματα διανομής πληροφοριακού και μαθησιακού υλικού, επικοινωνίας, διαχείρισης τάξης κ.ά. Το μαθησιακό υλικό που συνοδεύει και υποστηρίζει τη διδακτική προσέγγιση (ενότητα 3.7.), έχει ενοποιηθεί σε “πακέτο” σύμφωνα με την προδιαγραφή IMS Content Packaging specification (IMS, 2003) και μπορεί να διανέμεται σε ψηφιακό δίσκο (CD) στους εκπαιδευόμενους για τη διευκόλυνση τους, πέραν του ότι θα είναι διαθέσιμο στο δικτυακό τόπο του LMS.

- Η αξιολόγηση της επίδοσης των εκπαιδευομένων, θα γίνεται μέσω μικρών επί μέρους εργασιών και ενός ενιαίου project με συνεκτικότητα που θα αναπτύσσουν και θα παραδίδουν βήμα-βήμα (Φάση II, ενότητα 3.5.) (είτε μόνο μέσω project) και επίσης μέσω τελικών εξετάσεων στις οποίες θα καλούνται να απαντήσουν ερωτήσεις και να επιλύσουν ασκήσεις κατάλληλες για τον έλεγχο της επίτευξης των στόχων της προσέγγισης όπως: (i) κατανόησης των αντικειμενοστρεφών εννοιών και ελέγχου ικανότητας εφαρμογής τους σε προγράμματα Java, (ii) ανάλυσης προβλημάτων με αντικειμενοστρεφή τρόπο σκέψης και υλοποίησης προγραμμάτων σε Java για την επίλυσή τους, (iii) ελέγχου ικανότητας συγγραφής προγραμμάτων σε Java, τα οποία υλοποιούν ένα δοσμένο σχεδιασμό κλάσεων (class diagram design).

Για την αποδοτικότερη υλοποίηση των τριών φάσεων (ενότητα 3.5.) της διδακτικής προσέγγισης, όπως από τα ευρήματα των μελετών περίπτωσης προκύπτει, μπορούν να διατυπωθούν οι επόμενες προδιαγραφές για τις συνθήκες και το περιβάλλον εφαρμογής της που αποτελούν το ιδανικό σενάριο ώστε να επιτυγχάνεται το βέλτιστο μαθησιακό αποτέλεσμα.

- Οι εφαρμογές της προσέγγισης στη διδακτική πράξη δε συμπεριελάμβαναν σημαντικό δείγμα εκπαιδευομένων χωρίς καμία προηγούμενη γνώση στον Προγραμματισμό. Επίσης οι εφαρμογές έδειξαν αποτελεσματικότητα της μεθόδου σε εντελώς αρχάριους εκπαιδευόμενους στο αντικείμενο του ΑΠ. Συνεπώς συνιστάται η χρήση της για την εισαγωγή στον ΑΠ, σε αρχάριους για το αντικείμενο αυτό εκπαιδευόμενους, που έχουν κάποια έστω μικρή προηγούμενη γνώση διαδικασιακού Προγραμματισμού.
- Σημαντικό στοιχείο αποτελεί η διάθεση ικανού χρόνου από τη μεριά των εκπαιδευομένων για την ενεργό συμμετοχή τους σε όλες τις φάσεις υλοποίησης

της προσέγγισης (δια ζώσης συναντήσεις, ασύγχρονη επικοινωνία με τους εκπαιδευτές και τους συνεκπαιδευόμενους, ενεργός ενασχόληση με τις προς παράδοση εργασίες που προβλέπονται από τη διδακτική προσέγγιση) και αυτό να εκτιμάται λαμβάνοντας υπόψη τις συνολικές υποχρεώσεις τους. Συνεπώς, θα πρέπει η προσέγγιση να εφαρμόζεται με δεδομένη τη διαθεσιμότητά τους και στην περίπτωση αυξημένων επαγγελματικών ή άλλων υποχρεώσεών τους να επιμηκύνεται κατάλληλα η διάρκεια εφαρμογής της.

- Θα πρέπει να λαμβάνεται μέριμνα για την κατά το δυνατόν απρόσκοπτη πρόσβαση των εκπαιδευομένων στο Διαδίκτυο (Internet) (διάθεση εργαστηριακών χώρων), στις περιπτώσεις που γνωρίζουμε ότι σε κάποιο βαθμό αυτό δεν μπορεί να διασφαλιστεί από τους ίδιους π.χ. πρωτοετείς φοιτητές. Έτσι θα εξασφαλίζεται η επαφή των εκπαιδευομένων με το on-line μαθησιακό περιβάλλον που δημιουργείται για την εφαρμογή της προσέγγισης.
- Από τα αρχικά στάδια εφαρμογής της προσέγγισης θα πρέπει να επέρχεται κατά το δυνατόν εξοικείωση των εκπαιδευομένων με τη χρήση ασύγχρονης επικοινωνίας (web asynchronous discussion forum) με την κατάλληλη παρακίνηση και ενθάρρυνση από τη μεριά του εκπαιδευτή, ώστε να μην υπάρχει επιφύλαξη στη χρήση της.
- Σημαντικός παράγοντας επιτυχίας της προσέγγισης, αποδεικνύεται η διαθεσιμότητα του εκπαιδευτή για απαντήσεις σε απορίες, διορθώσεις των ασκήσεων και εργασιών, για παρότρυνση, κ.ο.κ. Θα πρέπει η διαθεσιμότητά του να είναι επαρκώς εξασφαλισμένη.
- Θα πρέπει να ενισχύεται η πρακτική εξάσκηση των εκπαιδευομένων (συγγραφή προγραμμάτων, εκπόνηση επί μέρους και εκτενέστερων εργασιών, συζήτηση των παραπάνω) και αυτό θα πρέπει να λαμβάνεται υπόψη σε όλη τη διάρκεια εφαρμογής της προσέγγισης από τον εκπαιδευτή.
- Προσοχή θα πρέπει να δίνεται στην ικανοποιητική επίδειξη της λειτουργίας των μαθησιακών εργαλείων με κάθε τρόπο (δια ζώσης συναντήσεις, επίλυση αποριών χρήσης των εργαλείων με συζητήσεις στο forum), καθώς και στην παρουσίαση και εξήγηση παραδειγμάτων, προγραμμάτων κώδικα κ.λ.π. από τον εκπαιδευτή είτε κατά τις διαλέξεις είτε στο forum ασύγχρονων συζητήσεων.

5.4. Μελλοντικές ερευνητικές κατευθύνσεις

Η προτεινόμενη διδακτική προσέγγιση ενσωματώνει τη χρήση ενός design CASE εργαλείου για την υποστήριξη της “model first” “πλευράς” της, το οποίο διαθέτει τη δυνατότητα παραγωγής “σκελετικού” κώδικα Java για την περαιτέρω ανάπτυξη και τελική συμπλήρωση του στο BlueJ (το οποίο αποτελεί CASE εργαλείο που υποστηρίζει τη μοντελοποίηση) ή/και σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών. Στο εργαλείο αυτό δημιουργείται το μοντέλο των κλάσεων που εντοπίζονται στο προς λύση πρόβλημα, με τα ιδιοχαρακτηριστικά, τις μεθόδους και τις μεταξύ τους σχέσεις και παράγεται “σκελετικός” κώδικας Java από τα μοντέλα αυτά.

Ο δυνατότητες ενός design CASE tool σταματούν σε αυτό το σημείο και η περαιτέρω ανάπτυξη και επίλυση του προβλήματος απαιτεί τη χρήση ενός άλλου περιβάλλοντος για τη συμπλήρωση του κώδικα, αλλά και στη συνέχεια τον έλεγχο, την αποσφαλμάτωση και την εκτέλεση της προγραμματιστικής λύσης του προβλήματος.

Η ενοποίηση των δυνατοτήτων αυτών των δύο περιβαλλόντων θα απάλασσε τους εκπαιδευόμενους από τις δυσκολίες χρήσης ενός design CASE tool, αλλά και θα εξέλειπε η ανάγκη χρήσης ενός επιπλέον περιβάλλοντος στην όλη διαδικασία.

Η ιδέα αυτή ενισχύεται και η αξία της επιβεβαιώνεται από σχετική σύσταση που διατυπώνεται στα συμπεράσματα του COOL project (Fjuk et al., 2006, chapter 11, p. 217) “Leaning a new tool consumes time that could have been spent on the subject matter instead, so introducing more than two tools during a semester is not recommended” (Δε συνιστάται η χρήση περισσότερων από δύο περιβαλλόντων στο ίδιο εξάμηνο, αφού η εκμάθηση χρήσης τους απαιτεί χρόνο που αποβαίνει εις βάρος του ίδιου του διδασκόμενου αντικειμένου).

Ζήτημα υπό διερεύνηση συνεπώς, αποτελεί η ανάπτυξη και χρήση μιας επέκτασης (extension ή plug-in) στο περιβάλλον BlueJ, που θα επιτρέπει τη σχεδίαση του μοντέλου των κλάσεων με τον τρόπο που αυτό πραγματοποιείται σε ένα design CASE tool, δηλαδή με τη χρήση UML διαγραμμάτων κλάσεων, τον καθορισμό των μεταξύ των κλάσεων σχέσεων και την παραγωγή του ανάλογου κώδικα από τα μοντέλα αυτά. Μεγάλο ερευνητικό ενδιαφέρον για τη διδασκαλία του ΑΠ, παρουσιάζει επίσης η κατασκευή εκπαιδευτικών περιβαλλόντων για την οπτικοποίηση της εκτέλεσης αντικειμενοστρεφών προγραμμάτων που θα παρέχουν

μια πλήρη εικόνα του κύκλου ζωής των αντικειμένων και της ροής του προγράμματος (Ala-Mutka, p.9-10; Ben-Ari et al., 2002).

Για την εφαρμογή της διδακτικής προσέγγισης σε συνθήκες αμιγώς ηλεκτρονικής μάθησης (μαθήματος e-learning), με την αξιοποίηση των διαδικτυακών τεχνολογιών (χωρίς “δια ζώσης” συναντήσεις με τους διδάσκοντες), το μαθησιακό υλικό θα μπορούσε να ενισχυθεί με λεπτομερέστερες παρουσιάσεις για την εξήγηση της χρήσης των εκπαιδευτικών περιβαλλόντων και προγραμματιστικών εργαλείων (με τη χρήση αρχείων video). Θα μπορούσε επίσης, να επεκταθεί η διδακτέα ύλη που η διδακτική προσέγγιση περιλαμβάνει στην τρέχουσα μορφή της. Οι μελέτες περίπτωσης έχουν ήδη αναδείξει κάποια θέματα που θα μπορούσαν να ενταχθούν κατάλληλα στο υπάρχον διδακτικό περιεχόμενο, όπως ο χειρισμός λαθών (errors handling). Περαιτέρω είναι δυνατό να διερευνηθεί η ένταξη πιο προχωρημένων ζητημάτων όπως η δημιουργία Γραφικών Διεπιφανειών Χρήστη (Graphical User Interface), ο χειρισμός συμβάντων (event handling) και οι τεχνικές εισόδου/εξόδου. Αντικείμενο διερεύνησης αποτελεί η εξεύρεση, ενσωμάτωση και δοκιμή παρεμβάσεων στη διδακτική προσέγγιση για την αποτελεσματικότερη αντιμετώπιση μαθησιακών δυσκολιών που οι μελέτες περίπτωσης είτε ανέδειξαν είτε έδειξαν να επιμένουν (υποενότητες 4.4.4.2. και 4.5.4.2.). Και βέβαια, πρέπει να σημειωθεί ότι θα έχει ερευνητικό ενδιαφέρον η δοκιμή της προτεινόμενης διδακτικής προσέγγισης για τον ΑΠ με χρήση της γλώσσας VB.NET, εκτός της γλώσσας Java που έχει χρησιμοποιηθεί μέχρι σήμερα.

5.5. Επιλογικά σχόλια

Στη διατριβή αυτή παρουσιάστηκε μια ολοκληρωμένη διδακτική προσέγγιση για την εισαγωγή στην Αντικειμενοστρεφή φιλοσοφία Προγραμματισμού. Η προσέγγιση αυτή στηρίζεται στην επικρατούσα για τη διδασκαλία του ΑΠ φιλοσοφία που είναι η “objects first” (υποενότητα 2.4.1.3.). Υλοποιεί τη φιλοσοφία αυτή με έμφαση στη μοντελοποίηση (“model first” στρατηγική, υποενότητα 2.4.2.3.), δηλαδή εστιάζοντας στις “modeling” πλευρές που από τη φύση του το αντικείμενο ΑΠ, διαθέτει (ένα αντικειμενοστρεφές πρόγραμμα αποτελεί ένα μοντέλο (model), το οποίο προσομοιώνει τη συμπεριφορά ενός πραγματικού ή ενός φανταστικού τμήματος του

κόσμου (Bennedsen and Caspersen, 2004b)). Για την υλοποίηση των παραπάνω, στην προσέγγιση προτείνεται ένα μαθησιακό σενάριο (learning script) που συνίσταται σε μαθησιακές δραστηριότητες με σειρά (learning sequence) (ενότητα 3.5.), οι οποίες υποστηρίζονται τεχνολογικά, αξιοποιώντας διαδικτυακές τεχνολογίες και εργαλεία και διέπονται από την παιδαγωγική αρχή της εκμάθησης του ΑΠ κοντά σε ειδικούς (experts) (γνωστική μαθητεία - cognitive apprenticeship). Για τις ανάγκες υποστήριξης των δραστηριοτήτων, η διδακτική προσέγγιση συνοδεύεται από κατάλληλο μαθησιακό υλικό (παραδείγματα, ασκήσεις, σημειώσεις θεωρίας, κ.ά.) το οποίο σχεδιάστηκε και κατασκευάστηκε αφού ελήφθησαν υπόψη συστάσεις και οδηγίες με προσήλωση στο στόχο της “θεραπείας” μαθησιακών προβλημάτων και μετά από μελέτη του οδηγού σπουδών των IEEE/ACM (CC2001, 2001) για τον ΑΠ, για να καθοριστεί η απαραίτητη διδακτέα ύλη για την εισαγωγή στον ΑΠ.

Οι επανειλημμένες εφαρμογές και συστηματικές αξιολογήσεις της προσέγγισης, απέδειξαν τη μαθησιακή αποτελεσματικότητά της, υπό τις συνθήκες διεξαγωγής των εφαρμογών αυτών. Να σημειωθεί εδώ ότι οι συμμετέχοντες στις εφαρμογές της προσέγγισης ήταν εθελοντές. Οι δυσκολίες που παρουσιάζονται στην εκπαιδευτική έρευνα όσον αφορά τη διασφάλιση της συμμετοχής, επέβαλλαν τον περιορισμό αυτό, δεδομένου ότι δεν ήταν δυνατό να έχουμε ένα εντελώς τυχαία επιλεγμένο δείγμα το οποίο θα συμμετείχε έως το τέλος των μελετών περίπτωσης. Η αποτελεσματικότητα της προσέγγισης συνίσταται, τόσο σε γνωστικό επίπεδο (εμπέδωση εννοιών, ικανότητα υλοποίησης προγραμμάτων με Αντικειμενοστρεφή φιλοσοφία, αντιμετώπιση καταγεγραμμένων δυσκολιών-παρανοήσεων) όσο και σε επίπεδο προτιμήσεων, στάσεων και διάθεσης των εκπαιδευομένων σχετικά με το αντικείμενο του ΑΠ (ενότητα 4.3.). Συστατικά στοιχεία της προσέγγισης (μαθησιακό υλικό και οι πόροι του εξατομικευμένα), στοιχεία που αφορούν στον τρόπο εφαρμογής της στη μαθησιακή διαδικασία (ηλεκτρονικό μαθησιακό περιβάλλον, αξιοποίηση διαδικτυακών τεχνολογιών, εκπόνηση εργασιών, υποστήριξη από τους διδάσκοντες) και επιλογές που έγιναν κατά το σχεδιασμό της και στην εκπαιδευτική της φιλοσοφία (ενότητα 3.4.), αξιολογήθηκαν σημαντικά υψηλά κατά μέσο όρο στις τρεις μελέτες περίπτωσης που διενεργήθηκαν.

Οι συστηματικές αξιολογήσεις, τα σχόλια και η κριτική των συμμετεχόντων, καθώς και η επεξεργασία των παραδοτέων τους, αποτέλεσαν σημαντικό υλικό για τη βελτίωση και την ενίσχυση της προσέγγισης μεταξύ των διαδοχικών εφαρμογών της, αλλά και συνολικά αποτελούν προτάσεις και εναύσματα για τη διερεύνηση

ζητημάτων και την κατάστροψη μελλοντικών ερευνητικών σχεδίων τα οποία αναφέρθηκαν στην προηγούμενη ενότητα.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

РАНЕЕ НЕ ПЕРПА

Βιβλιογραφικές αναφορές

ACM (2002). Obituary. Διαθέσιμο στην ηλεκτρονική διεύθυνση http://www.acm.org/announcements/turing_obit.html (30/4/2007).

Ala-Mutka, K. Problems in Learning and Teaching Programming. A literature study for developing visualizations in the Godewitz-Minerva project. Διαθέσιμο στην ηλεκτρονική διεύθυνση www.cs.tut.fi/~codewitz/literature_study.pdf (29/12/2006).

Allen E., Cartwright R., and Stoler B. (2002). DrJava: A lightweight pedagogic environment for Java. *ACM SIGCSE Bulletin*, 34(1), 137-141.

Alphonse, C., and Ventura. P. (2003). Using graphics to support the teaching of fundamental object-orientation principles in CS1. In *Proceedings companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA '03)*, Anaheim, California, October 26-30, 156-161.

Alphonse, C. and Ventura, P. J. (2002). Object-Orientation in CS1-CS2 by Design. In *Proceedings of Innovation and Technology in Computer Science Education*, Aarhus, Denmark, June 24-26, 70-74.

Anthony, D.L.G. (1996). Patterns for Classroom Education, Pattern Languages of Programming 2, Vlissides, Coplien, Kerth (editors), Addison Wesley, 1996, pp 391. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://ianchaiwriting.50megs.com/classroom-ed.html> (1/10/2006).

Astrachan, O., and Reed, D. (1995). AAA and CS1: The Applied Apprenticeship Approach to CS1. *ACM SIGCSE Bulletin*, 27(1), 1-5.

Barnes, D.J., and Kölling, M. (2005). *Objects First with Java, A Practical Introduction using BlueJ*. Prentice Hall, Second edition.

Barr, M., Holden, S., Phillips, D., and Greening, T. (1999). An exploration of novice programming errors in an object-oriented environment. *ACM SIGCSE Bulletin*, 31(4), 42-46.

Becker, B. W. (2001). Teaching CS1 with Karel the Robot in Java. In *Proceedings of the 32nd SIGCSE Technical Symposium*, ACM Press, 50-54.

Becker, B. W. (2000). Pedagogies for CS1: A survey of Java textbooks. Manuscript. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://www.cs.uwaterloo.ca/~bwbecker/papers/javaTextbooks/> (23/4/2007).

Benander, A., Benander, B., and Sang, J. (2004). Factors related to the difficulty of learning to program in Java-an empirical study of non-novice programmers. *Information and Software Technology*, 46(2), 99-107.

Bennedsen, J., and Caspersen, M. (2004a). Programming in Context - A Model-First Approach to CS1. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, Virginia, March 3-7, 477-481.

Bennedsen, J., and Caspersen, M. (2004b). Teaching Object-Oriented Programming – Towards Teaching a Systematic Programming Process. Paper presented at 18th European Conference on Object-Oriented Programming, 8th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, Oslo, Norway.

Ben-Ari, M. (2004). Situated Learning in Computer Science Education. *Computer Science Education*, 14(2), 85-100.

Ben-Ari, M., Ragonis, N., and Ben-Bassat Levy, R. (2002). A Vision of Visualization in Teaching Object-Oriented Programming. In *Proceeding of 2nd Program Visualization Workshop*, HornstrupCentret, Denmark, 83-89.

Berge, O., Borge, R. E., Fjuk, A., Kaasbøll, J. and Samuelsen, T. (2003a). Learning Object Oriented Programming. Norsk Informatikkonferanse NIK'2003, Tapir Akademisk Forlag, 37-47.

Berge, O., Fjuk, A., and Groven, A. (2003b). Comprehensive Object-Oriented Learning – An Introduction. *Computer Science Education*, 13(4), 331-335.

Bergin, J. (2006a). Some Pedagogical Patterns. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://csis.pace.edu/~bergin/patterns/fewpedpats.html> (1/10/2006).

Bergin, J. (2006b). Fourteen Pedagogical Patterns. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://csis.pace.edu/~bergin/PedPat1.3.html> (1/10/2006).

Bergin, J., Stehlik, M., Roberts, J. and Pattis, R. (2005). *Karel J. Robot: A Gentle Introduction to the Art of Object-Oriented Programming in Java*. USA: Dream Songs Press.

Bergin, J., Stehlik, M., Roberts, J., and Pattis, R. (2003). *Karel J. Robot: A Gentle Introduction to the Art of Object-Oriented Programming in Java*. Published manuscript. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html> (31/12/2006).

Bergin, J. (2000a). Why Procedural is the Wrong First Paradigm if OOP is the Goal. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://www.csis.pace.edu/~bergin/papers/Whynotproceduralfirst.html> (30/4/2007).

Bergin, J. (2000b). Introducing Objects with Karel J. Robot. Position paper presented in Workshop 8 (Tools and Environments for Understanding Object-Oriented Concepts) at ECOOP 2000. June 8, Sophia Antipolis, France. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://csis.pace.edu/~bergin/karel/ecoop2000JBKarel.html> (31/12/2006).

Bergin, J., Stehlik, M., Roberts, J., and Pattis, R. (1997). *Karel++ the Robot: A Gentle Introduction to the Art of Object-Oriented Programming*. 2nd edition, New York, John Wiley & Sons.

Biddle, R., and Tempero, E. (1998). Java pitfalls for beginners. *ACM SIGCSE Bulletin*, 30(2), 48-52.

Bishop, J.M. (1997). A philosophy of teaching Java as a first teaching language. *ACM SIGCSE Bulletin*, 29(1), 140-142.

Booch, G. (1991). *Object Oriented Design with Applications*, The Benjamin/Cummings Publishing Company, Bedwood city, California.

Borge, R. E., and Kaasbøll, J. (2003). What is “OO first”? Submitted as a position paper for the 7th Workshop on Pedagogies and Tools for Learning Object-Oriented

Concepts at 17th European Conference on Object-Oriented Programming, July, Darmstadt, Germany.

Brown, J. S., Collins, A., and Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32-42.

Bruce, K., Danyluk, A., and Murtagh, T. (2001). A library to support a graphics-based object-first approach to CS 1. *ACM SIGCSE Bulletin* 33(1), 6-10.

Brusilovsky, P., Calabrese, E., Hvorecky, Y., Kouchnirenko, A. and Miller, P. (1997). Mini-languages: a way to learn programming principles. *Education and Information Technologies*, 2(1), 65-83.

Buck, D., and Stucki, D.J. (2001). JKarelRobot: A Case Study in Supporting Levels of Cognitive Development in the Computer Science Curriculum. *ACM SIGCSE Bulletin*, 33(1), 16-20.

Buck, D., and Stucki, D. (2000). Design early considered harmful: Graduated exposure to complexity and structure based on levels of cognitive development. *ACM SIGCSE Bulletin*, 32(1), 75-79.

Carter, J., and Fowler, A. (1998). Object Oriented Students?. *ACM SIGCSE Bulletin*, 28(3), 271.

CC2001 Computing Curricula 2001 (final report) (2001). The Joint Task Force on Computing Curricula (IEEE Computer Society and Association for Computing Machinery), December 15. Διαθέσιμο στην ηλεκτρονική διεύθυνση http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf (27/1/2007).

Collins, A., Brown, J. S., and Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. *American Educator*, 6-46.

Collins, A., Brown, J. S., and Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L.B. Resnick (Ed.), *Knowing, learning and instruction: Essays in honour of Robert Glaser*, Hillsdale, NJ: LEA, pp. 453-494.

Cooper, S., Dann, W., and Pausch, R. (2003). Teaching Objects-first in Introductory Computer Science. *ACM SIGCSE Bulletin* 35(1), 191-195.

Cooper, S., Dann, W., and Paush, R. (2000). Alice: a 3-D tool for introductory programming concepts. *Journal of Computing in Small Colleges*, 15(5), 108-117.

Depradine, C., and Gay, G. (2004). Active participation of integrated development environments in the teaching of object-oriented programming. *Computers & Education*, 43(3), 291-298.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed). *Three worlds of CSCL. Can we support CSCL* (p. 61-91). Heerlen, Open Universiteit Nederland.

Eckerdal A. (2006). *Novice Students' Learning of Object-Oriented Programming*. Ph.D. thesis, Department of Information Technology, Uppsala University, Sweden.

Eckerdal, A. and Thune', M. (2005). Novice java programmers' conceptions of 'object' and 'class', and variation theory. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, Monte de Caparica, Portugal, June 27-29, 89-93.

Fernández Muñoz, L., Peña, R. (2002). PIPOO: An Adaptive Language to Learn OO Programming. Position paper presented at 16th European Conference on Object-Oriented Programming, 6th Workshop on Pedagogies and Tools for Learning Object-Oriented Concepts (or workshop "Tools and environments for Learning Object Oriented Concepts"), Malaga, Spain.

Fjuk, A., Karahasanovic, A., Kaasbøll, J. (Eds.) (2006). *Comprehensive Object-Oriented Learning: The Learners Perspective*. Informing Science Press, California.

Fleury, A. E. (2001). Encapsulation and reuse as viewed by java students. *ACM SIGCSE Bulletin*, 33(1), 189-193.

Fleury, A. E. (2000). Programming in Java: Student-constructed rules. *ACM SIGCSE Bulletin*, 32(1), 197-201.

Fricke, A., and Voelter, M. (2000). Seminars: A Pedagogical Pattern Language About Teaching Seminars Effectively. In *Proceedings of EuroPLoP 2000*. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://www.voelter.de/publications/seminars.html> (1/10/2006).

Gamma, E., Helm, R., Johnson R., and Vlissides. J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Georgantaki, S., and Retalis, S. (2007a). A Learning Research Informed Design and Evaluation of a Web-enhanced Object Oriented Programming Seminar. *Journal of Information Systems Education*, 18(2), 243-253.

Georgantaki, S., and Retalis, S. (2007b). Using educational tools for teaching Object Oriented design & programming, *Journal of Information Technology Impact*, 7(2), 111-130.

Georgantaki, S., Psaromiligkos, Y., Retalis, S., Dendrinis, V., and Adamopoulos, D. (2007). Developing a blended learning strategy for teaching Object-Oriented Programming using the "Model First" approach. In *Proceedings of Informatics Education Europe II Conference 2007 (IEEII2007)*, 29-30 November 2007, Thessaloniki, Greece, 87-96.

Goldman, K.J. (2004). A Concepts-First Curriculum for Introductory Computer Science. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, Virginia, March 3-7, 432-436.

Goldman, K.J. (2003). A Demonstration of JPie: An Environment for Live Software Construction in Java. In *Proceedings companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA '03)*, Anaheim, California, October 26-30, 74-75.

Goodyear, P. (1994). Foundations for Courseware Engineering. In R.D. Tennyson, (ed.), *Automatic Instructional Design, Development, and Delivery*. Berlin: Springer-Verlag.

Gray, K. E. and Flatt, M. (2003). Professorj: a gradual introduction to java through language levels. In *Proceedings companion of the 18th Annual ACM SIGPLAN*

Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA '03), Anaheim, California, October 26-30, 170-177.

Groven, A., Hegna, H., and Smørðal, O. (2003). OO learning, a modeling approach. Paper presented at 17th European Conference on Object-Oriented Programming, 7th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, July, Darmstadt, Germany.

Hadjerrouit, S. (1999). A Constructivist Approach to Object-Oriented Design and Programming. In *Proceedings of the 4th ITiCSE*, 171-174.

Hadjerrouit, S. A. (1998). Constructivist Framework for Integrating the Java Paradigm into the Undergraduate Curriculum. In *Proceedings of the 3rd ITiCSE*, 105-107.

Hagan, D. and Markham, S. (2000). Teaching Java with the BlueJ Environment. Paper presented at the Australian Society for Computers in Learning in Tertiary Education (ASCILITE), Coffs Harbour, Australia.

Harrison, C.J., Sallabi, O.M., and Eldridge, S.E. (2005). An initial object-oriented programming language (IOPL) and its implementation. *IEEE, Transactions on Education*, 48(1), 3-10.

Hendrix, T.D., Cross, J.H., and Barowski, L.A. (2004). An extensible framework for providing dynamic data structure visualizations in a lightweight IDE. *ACM SIGCSE Bulletin*, 36(1), 387-391.

Henriksen, P., and Kölling, M. (2004). greenfoot: Combining Object Visualization with Interaction. In *Proceedings companion of the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA '04)*, Vancouver, BC, Canada, 73-82.

Hislop, D. (2002). Mission impossible? Communicating and sharing knowledge via information transfer. *Journal of Information Technology*, 17(3), 165-177.

Holland, S., Griffiths, R., and Woodman, M. (1997). Avoiding object misconceptions. In *Proceedings of the 28th SIGCSE*, 131-134.

Hristova, M., Misra, A., Rutter, M., and Mercuri, R. (2003). Identifying and Correcting Java Programming Errors for Introductory Computer Science Students. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, Reno, Nevada, USA, 153-156.

Hyland, E., and Clynych, G. (2002). Initial experiences gained and initiatives employed in the teaching of Java programming in the Institute of Technology Tallaght, *ACM International Conference Proceeding Series (AICPS)*, Vol. 25, 101-106.

IMS Content Packaging Specification (2003). Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://www.imsglobal.org/content/packaging/index.cfm> (11/10/2006).

Kölling, M., Quig, B., Patterson, A., and Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Journal of Computer Science Education, Special Issue on Learning and Teaching Object Technology*, 13(4), 249-268.

Kölling, M. (2003). The Curse of Hello World. COOL Workshop on Learning and Teaching Object-orientation – Scandinavian Perspectives, Oslo, Norway, October 20th.

Kölling, M., and Rosenberg, J. (2001a). BlueJ - The Hitch Hiker's Guide to Object Orientation. *Journal of object-oriented programming*. June 2001.

Kölling, M., and Rosenberg, J. (2001b). Guidelines for Teaching Object Orientation with Java. *ACM SIGCSE Bulletin* 33(3), 33-36.

Kölling, M., and Rosenberg, J. (2000). Objects first with Java and BlueJ (seminar session). *ACM SIGCSE Bulletin*, 32(1), 429.

Kölling, M. (1999a). The Problem of Teaching Object-Oriented Programming, Part 1: Languages. *Journal of Object-Oriented Programming*, 11(8), 8-15.

Kölling, M. (1999b). The Problem of Teaching Object-Oriented Programming, Part 2: Environments. *Journal of Object-Oriented Programming*, 11(9), 6-12.

Lahtinen, E., Ala-Mutka, K., and Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. In *Proceedings of the 10th Annual SIGCSE conference on*

Innovation and Technology in Computer Science Education, 14 – 18, June 27-29, Monte de Caparica, Portugal, also published in *ACM SIGCSE Bulletin*, 37(3), 14-18.

Lang, B. (2002). Teaching new programmers: a Java tool set as a student teaching aid. In *Proceedings of the inaugural conference on the Principles and Practice of programming*, and in *Proceedings of the 2nd workshop on Intermediate representation engineering for virtual machines*, Dublin, Ireland, 95-100.

Lave, J., and Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*, Cambridge University Press, Cambridge.

Lemay, L., and Cadenhead, R. (1999). *Teach yourself JAVA 2 in 21 days*, SAMS Publishing.

Lewis, J. (2000). Myths about Object-Orientation and Its Pedagogy. In *Proceedings of the 31st SIGCSE technical symposium on Computer Science Education*, Austin, Texas, March, 245-249.

Linden, P. Van der. (1996). *Just JAVA*, The Sunsoft Press JAVA SERIES.

Linn, M., and Clancy, J. (1992). The case for case studies of programming problems. *Communications of the ACM*, 35(3), 121–132.

Luker, P. A. (1994). There's more to oop than syntax!". *ACM SIGCSE Bulletin*, 26(1), 56-60.

Madsen, O.L., (1995). Open Issues in Object-Oriented Programming. *Software Practice and Experience* 25(S4), 1995.

Makrakis, V., Retalis, S., Koutoumanos, A., Papaspyrou, N., and Skordalakis, M. (1998). Evaluating the effectiveness of an ODL Hypermedia System and Courseware at the National Technical University of Athens: A Case Study. *Journal for Universal Computer Science*, 4(3), 259-272.

McIver, L. (2002). Evaluating Languages and Environments for Novice Programmers. *14th Annual Workshop of the Psychology of Programming Interest Group (PPIG 2002)*, Middlesex, UK, 100-110.

Milne, J., and Rowe, G. (2002). Difficulties in Learning and Teaching Programming – Views of Students and Tutors. *Education and Information Technologies*, 7(1), 55-66.

Moskal, B., Lurie, D., and Cooper, S. (2004). Evaluating the effectiveness of a new instructional approach. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, Virginia, March 3-7, 75-79.

Pattis, R. E. (1981). *Karel the Robot: A gentle Introduction to the art of Programming*. John Wiley & Sons.

Powers, K., Cooper, S., Goldman, K., Carlisle, M., McNally, M., and Proulx, V. (2006). Tools for teaching introductory programming: What works? In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*. Houston, Texas, March 1-5, 560-561.

Proulx, V., and Gray, K. E. (2006). Design of Class Hierarchies: An Introduction to OO Program Design. *ACM SIGCSE Bulletin* 38(1), 288-292.

Proulx, V., Raab, J., and Rasala, R. (2002). Objects from the beginning – with GUIs. *ACM SIGCSE Bulletin* 34(3), 65-69.

Psaromiligkos, Y., and Retalis, S. (2003). Re-Evaluating the Effectiveness of a Web-based Learning System: A Comparative Case Study. *Journal of Educational Multimedia and Hypermedia, AACE*, 12 (1), 5-20.

Ragonis, N., and Ben-Ari, M. (2005a). On Understanding the Statics and Dynamics of Object-Oriented Programs. *ACM SIGCSE '05*, 226-230.

Ragonis, N., and Ben-Ari, M. (2005b). A Long-Term Investigation of the Comprehension of OOP Concepts by Novices. *Computer Science Education*, 5(3), 203-221.

Ragonis, N. (2004). *Teaching Object-Oriented Programming to Novices*. Ph.D. thesis, Weizmann Institute of Science, Rehovot, Israel (in Hebrew).

Ragonis, N., and Ben-Ari, M. (2002). Teaching Constructors: A Difficult Multiple Choice. Paper presented at 16th European Conference on Object-Oriented

Programming, 6th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, Malaga, Spain.

Rasala, R., Raab, J., and Proulx, V.K. (2001). Java Power Tools: Model Software for Teaching Object-Oriented Design. *ACM SIGCSE Bulletin* 33(1), 297-301.

Reber, A.S. (1993). *Implicit Learning and Tacit Knowledge: An Essay on the Cognitive Unconscious*. Oxford University Press, New York, NY.

Retalis, S., Makrakis, V., and Skordalakis E. (1998). A methodology for evaluating the effectiveness of the use of new technologies in ODL. In *Proceedings of EDMEDIA-EDTELECOM'98*, Freiburg, Germany.

Roberts, E. (2006). An interactive tutorial system for Java. *ACM SIGCSE Bulletin* 38(1), 334-338.

Roberts, E., and Picard, A. (1998). Designing a Java graphics library for CS1. In *Proceedings of the 3rd Annual SIGCSE conference on Innovation and Technology in Computer Science Education*, Dublin, Ireland, July, 213-218.

Roy, G.G., and Armarego, J. (2004). Teaching programming with objects. ASCILITE (Australasian Society for Computers in Learning in Tertiary Education) Conference, Perth, Western Australia. Διαθέσιμο στην ηλεκτρονική διεύθυνση <http://www.ascilite.org.au/conferences/perth04/procs/roy.html> (4/1/2007).

Sanders, D., and Dorn, B. (2003a). Jeroo: a tool for introducing object-oriented programming, *ACM SIGCSE Bulletin*, 35(1), 201-204.

Sanders, D., and Dorn, B. (2003b). Classroom experience with Jeroo, *Journal of Computing in Small Colleges*, 18(4), 308-316.

Schulte, C. and Niere, J. (2002). Thinking in object structures: teaching modelling in secondary schools. Paper presented at 16th European Conference on Object-Oriented Programming, 6th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, Malaga, Spain.

SmørDAL, O.C. (1999). *Work Oriented Objects - Object Oriented Modelling of Computer Mediated Cooperative Activities: An Activity Theoretical Perspective* -Dr

Scient thesis. (ISBN 82-7368-206-4) (ISSN 0809-3857) Institutt for Informatikk. 1999-05-31 (17) : 115.

Teif, M., and Hazzan, O. (2004). Junior High School Students' Perception of Object Oriented Concepts. Paper presented at 18th European Conference on Object-Oriented Programming, 8th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, Oslo, Norway.

Turner, T. A., and Zachary, L. (2001). Javiva: A Tool for Visualizing and Validating Student-Written Java Programs. In *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, Charlotte, February, 45-49.

Van Haaster, K., and Hagan, D. (2004). Teaching and Learning with BlueJ: an Evaluation of a Pedagogical Tool. *Information Science & Information Technology Education Joint Conference*, Rockhampton, QLD, Australia, 456-470.

Westfall, R. (2001). Hello world considered harmful. *Communications of the ACM*, 44(10), 129-130.

Wiedenbeck, S., and Ramalingam, V. (1999). Novice comprehension of small programs written in the procedural and object-oriented styles, *International Journal of Human-Computer Studies*, 51(1), 71-87.

Wiedenbeck, S., Ramalingam, V., Sarasamma, S. and Corritore C. (1999). A comparison of the comprehension of object-oriented and procedural programs by novice programmers. *Interacting with Computers*, 11(3), 255-282.

Xinogalos, S., Satratzemi, M., and Vassilios Dagdilelis, V. (2006). An introduction to object-oriented programming with a didactic microworld: objectKarel. *Computers & Education*, 47(2), 148-171.

Xinogalos, S., and Satratzemi, M. (2002). An Integrated Programming Environment for Teaching the Object-Oriented Programming Paradigm. LNCS Vol. 2510, In *Proceedings of the 1st EurAsian-ICT, Information and Communication Technology*, Shiraz, Iran, October 29-31, 544-551.

Zimmerman, B. J. and Kitsantas, A. (1999). Acquiring writing revision skill: Shifting from process to outcome self-regulatory goals. *Journal of Educational Psychology*, 91(2), 241-250.

Γεωργαντάκη, Σ. και Ρετάλης, Σ. (2006). Μια διδακτική προσέγγιση για την εκμάθηση των βασικών εννοιών-αρχών του Αντικειμενοστρεφούς Προγραμματισμού. *Θέματα στην Εκπαίδευση, Ειδικό αφιέρωμα: Σύγχρονη έρευνα στη Διδακτική της Πληροφορικής: ερευνητικοί άξονες, μέθοδοι, τεχνικές, εργαλεία*, 7(3), 329-349.

Γεωργαντάκη, Σ. και Ρετάλης, Σ. (2005). Ένα σενάριο διδασκαλίας των βασικών εννοιών-αρχών του αντικειμενοστρεφούς προγραμματισμού, στο Α. Τζιμογιάννης (επιμ.), *Ηλεκτρονικά Πρακτικά εργασιών 3ου Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής», Συνεδρία Σ10*, Κόρινθος, 7-9 Οκτωβρίου.

Ξυνόγαλος, Σ. (2002). *Εκπαιδευτική Τεχνολογία: Ένας Διδακτικός Μικρόκοσμος για την Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό*, Διδακτορική διατριβή, Τμήμα Εφ. Πληροφορικής, Πανεπιστήμιο Μακεδονίας.

Σατρατζέμη, Μ., Ξυνόγαλος, Σ. και Δαγδιλέλης, Β., (2006). Εκπαιδευτικά προγραμματιστικά περιβάλλοντα για τη διδασκαλία του Αντικειμενοστραφούς Προγραμματισμού: Μια επισκόπηση. *Πρακτικά 5^ο Πανελληνίου Συνεδρίου με διεθνή συμμετοχή «Οι ΤΠΕ στην Εκπαίδευση»*, 899-906, Θεσσαλονίκη.

РАНЕЕЗНАМО ТЕРРА

ΠΑΡΑΡΤΗΜΑ Α

Θεματική Ανάπτυξη του Διδακτικού Περιεχομένου

ΕΝΟΤΗΤΑ 1. ΦΙΛΟΣΟΦΙΑ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Στόχοι:

- Να αντιληφθούν οι εκπαιδευόμενοι το διαφορετικό τρόπο αντιμετώπισης των προβλημάτων στο διαδικασιακό (procedural) και τον αντικειμενοστρεφή (object-oriented) προγραμματισμό και τα οφέλη του δεύτερου.
- Να εισαχθούν στις έννοιες κλάση, αντικείμενο, ιδιοχαρακτηριστικά, μέθοδος.
- Να εγκαταστήσουν το απαραίτητο Λογισμικό για τη μεταγλώττιση-εκτέλεση προγραμμάτων σε Java και να συγγράψουν - μεταγλωττίσουν - εκτελέσουν μια τουλάχιστον απλή εφαρμογή Java και ένα Java applet.

Τμήματα Ενότητας

- 1.1. Αντικειμενοστρεφής φιλοσοφία προγραμματισμού
- 1.2. Επαναχρησιμοποιησιμότητα - Επεκτασιμότητα (reusability - expandability)
- 1.3. Διαδικασιακός προγραμματισμός (procedural programming)
- 1.4. Αντικειμενοστρεφής προγραμματισμός (object-oriented programming)
- 1.5. Διαδικασιακός σε αντιδιαστολή με αντικειμενοστρεφή προγραμματισμό
- 1.6. Διαγραμματική παράσταση εννοιών: διάγραμμα κλάσης (class diagram) - διάγραμμα αντικειμένου (object diagram)
- 1.7. Συγγραφή - μεταγλώττιση - εκτέλεση εφαρμογών και μικροεφαρμογών (applets) Java
 - 1.7.1. Εγκατάσταση Java2 SDK standard edition (j2se)
 - 1.7.2. Δημιουργία μιας εφαρμογής Java (Java application)
 - 1.7.3. Δημιουργία ενός Java applet (μίνι εφαρμογή)
 - 1.7.4. Χρήση του SUN One Studio
- 1.8. Ερωτήσεις - Ασκήσεις

ΕΝΟΤΗΤΑ 2. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ: ΑΝΤΙΚΕΙΜΕΝΟ, ΚΛΑΣΗ, ΙΔΙΟΤΗΤΕΣ, ΜΕΘΟΔΟΙ (OBJECT, CLASS, ATTRIBUTES, METHODS)

Στόχοι:

- Να κατανοήσουν και να διακρίνουν οι εκπαιδευόμενοι τις έννοιες κλάση (class) και αντικείμενο (object).
- Να προσδιορίζουν σωστά ιδιοχαρακτηριστικά και μεθόδους σε οντότητες.
- Να κατανοήσουν την κλήση των μεθόδων μιας κλάσης, το πέρασμα παραμέτρων, την εκτέλεση των μεθόδων, την επιστροφή ή όχι τιμής από τη μέθοδο.
- Να μπορούν να υλοποιούν μεθόδους με παραμέτρους ή χωρίς και με τιμή επιστροφής ή χωρίς.
- Να κατανοήσουν τη στατικότητα της κλάσης και τη δυναμικότητα των αντικειμένων που προέρχονται από την κλάση.
- Να εγκαταστήσουν το εκπαιδευτικό περιβάλλον BlueJ, καθώς και το UML tool ArgoUML, να εξοικειωθούν με τη χρήση τους και να πραγματοποιήσουν τις προβλεπόμενες δραστηριότητες και ασκήσεις σε αυτά τα περιβάλλοντα.

Τμήματα Ενότητας

- 2.1. Παραδείγματα κλάσεων με ιδιότητες και μεθόδους - προσδιορισμός ιδιοτήτων και μεθόδων κλάσεων
- 2.2. Στατικότητα κλάσης - Δυναμικότητα στιγμιτύπων της κλάσης
- 2.3. Δραστηριότητες στο εκπαιδευτικό προγραμματιστικό περιβάλλον BlueJ
- 2.4. Δραστηριότητες στο περιβάλλον ArgoUML
- 2.5. Οι μέθοδοι στη Java
- 2.6. Ερωτήσεις - Ασκήσεις

ΕΝΟΤΗΤΑ 3. ΜΕΘΟΔΟΣ ΚΑΤΑΣΚΕΥΑΣΤΗ (CONSTRUCTOR), ΔΗΜΙΟΥΡΓΙΑ ΑΝΤΙΚΕΙΜΕΝΩΝ – ΚΑΤΑΝΟΜΗ ΜΝΗΜΗΣ – ΚΑΤΑΣΤΡΟΦΗ ΑΝΤΙΚΕΙΜΕΝΩΝ, ΑΝΑΦΟΡΕΣ ΣΕ ΑΝΤΙΚΕΙΜΕΝΑ (REFERENCES)

Γενικός στόχος της ενότητας αυτής είναι, οι εκπαιδευόμενοι να κατανοήσουν τον κύκλο ζωής των αντικειμένων (lifecycle of objects, δημιουργία - αρχικοποίηση - ... - καταστροφή) και την ανάθεση αναφορών (references) σε αντικείμενα.

Ειδικοί στόχοι:

- Να κατανοήσουν τη μέθοδο του κατασκευαστή (constructor), να μπορούν να δηλώνουν μεθόδους κατασκευαστών σε μια κλάση και να τις καλούν.
- Να αντιλαμβάνονται την αρχικοποίηση των ιδιοχαρακτηριστικών με διάφορους τρόπους.
- Να μπορούν να δημιουργούν αντικείμενα, να δίνουν αρχικές τιμές στα ιδιοχαρακτηριστικά τους, να συνδέουν τύπους αναφοράς με αυτά (references), να αποσυνδέουν τύπους αναφοράς από τα αντικείμενα.
- Να κατανοούν την καταστροφή αντικειμένων και πότε αυτή συμβαίνει.

Τμήματα Ενότητας

- 3.1. Μέθοδος κατασκευαστή
 - 3.1.1. Σύγκριση κατασκευαστή με άλλες μεθόδους
 - 3.1.2. Δραστηριότητα στο περιβάλλον ArgoUML
 - 3.1.3. Παραδείγματα μιας κλάσης χωρίς τη δήλωση κατασκευαστή και με τη δήλωση ενός ή πολλαπλών κατασκευαστών και χρήση της
 - 3.1.4. Κλήση μεθόδου κατασκευαστή μέσα από άλλη μέθοδο κατασκευαστή (δεσμευμένη λέξη this): παραδείγματα
 - 3.1.5. Δραστηριότητα στο περιβάλλον BlueJ
 - 3.1.6. Τι συμβαίνει όταν δημιουργείται ένα αντικείμενο;
- 3.2. Οπτικοποίηση της δημιουργίας - καταστροφής αντικειμένων, της κατανομής μνήμης και της “σύνδεσης - αποσύνδεσης” αναφορών σε/από αντικείμενα (με το Λογισμικό Powerpoint)
- 3.3. Αναφορές σε αντικείμενα
 - 3.3.1. Τύποι αναφοράς (references types)
 - 3.3.2. Δραστηριότητα στο περιβάλλον BlueJ
 - 3.3.3. Λυμένες ασκήσεις (με σχηματική παράσταση της μνήμης)
- 3.4. Ερωτήσεις - Ασκήσεις

ΕΝΟΤΗΤΑ 4. ΑΦΑΙΡΕΤΙΚΟΤΗΤΑ (ABSTRACTION), ΚΕΛΥΦΟΠΟΙΗΣΗ (ENCAPSULATION) - ΑΠΟΚΡΥΨΗ ΠΛΗΡΟΦΟΡΙΑΣ (INFORMATION HIDING), ΔΙΑΧΩΡΙΣΜΟΣ ΣΥΜΠΕΡΙΦΟΡΑΣ – ΥΛΟΠΟΙΗΣΗΣ (SEPARATION OF BEHAVIOR AND IMPLEMENTATION)

Γενικός στόχος της ενότητας αυτής είναι, οι εκπαιδευόμενοι να κατανοήσουν τις βασικές αρχές της αντικειμενοστρεφούς φιλοσοφίας προγραμματισμού: αφαιρετικότητα (abstraction), κελυφοποίηση (encapsulation), απόκρυψη πληροφορίας (information hiding), διαχωρισμός συμπεριφοράς και υλοποίησης (separation of behavior and implementation).

Ειδικοί στόχοι:

- Να δημιουργούν σωστά διαφορετικές abstractions της ίδιας οντότητας ανάλογα με το πρόβλημα.
- Να κατανοήσουν τη σημασία των προσδιοριστών ορατότητας private και public στη δήλωση ιδιοχαρακτηριστικών και μεθόδων σε μια κλάση.
- Να υλοποιούν κλάσεις οι οποίες ικανοποιούν τις βασικές αρχές της κελυφοποίησης, απόκρυψης πληροφορίας, διαχωρισμού συμπεριφοράς και υλοποίησης χρησιμοποιώντας κατάλληλα τους προσδιοριστές ορατότητας private και public.

Τμήματα Ενότητας

- 4.1. Αφαιρετικότητα (abstraction)
 - 4.1.1. Ορισμός
 - 4.1.2. Παραδείγματα
- 4.2. Κελυφοποίηση - απόκρυψη πληροφορίας (encapsulation - information hiding)
 - 4.2.1. Ορισμός - θεωρία
 - 4.2.2. Μηχανισμός κλάσης
 - 4.2.3. Ιδιωτικό (private) - Δημόσιο (public)
 - 4.2.4. Διαπροσωπεία (interface) - Υλοποίηση (implementation)
- 4.3. Δραστηριότητες στο περιβάλλον ArgoUML
- 4.4. Διαχωρισμός συμπεριφοράς - υλοποίησης (separation of behavior and implementation)

- 4.4.1. Επίδειξη της έννοιας με ένα παράδειγμα κλάσης και προγράμματος χρήσης της
- 4.4.2. Σχηματική απόδοση
- 4.4.3. Λυμένη άσκηση υλοποίησης κλάσης
- 4.4.4. Εμπειρικοί κανόνες
- 4.5. Διαπροσωπεία κλάσης: δραστηριότητα στο περιβάλλον BlueJ
- 4.6. Ερωτήσεις - Ασκήσεις

ΕΝΟΤΗΤΑ 5. ΜΕΘΟΔΟΙ ΚΑΙ ΙΔΙΟΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΛΑΣΗΣ (STATIC OR CLASS MEMBERS)

Στόχοι:

- Να κατανοήσουν οι εκπαιδευόμενοι πότε μια μεταβλητή ή μια μέθοδος θα είναι κλάσης, δηλαδή static.
- Να αντιλαμβάνονται τη μοναδικότητα των μεταβλητών κλάσης για όλα τα αντικείμενα.
- Να μπορούν να καλούν μεθόδους κλάσης με δύο τρόπους και να προσπελαίνουν μεταβλητές κλάσης και στιγμιοτύπου από μεθόδους κλάσης και στιγμιοτύπου με το σωστό τρόπο.

Τμήματα Ενότητας

- 5.1. Μεταβλητές κλάσης (class or static variables)
- 5.2. Μέθοδοι κλάσης (class or static methods)
- 5.3. Δραστηριότητα στο περιβάλλον BlueJ
- 5.4. Εμπειρικοί κανόνες για τις μεταβλητές και τις μεθόδους κλάσης
- 5.5. Λυμένη άσκηση με κλάση που διαθέτει μέλη κλάσης και χρήση τους (κατανομή μνήμης)
- 5.6. Παραδείγματα προγραμμάτων που επιδεικνύουν τις μεταβλητές και τις μεθόδους κλάσης και τη χρήση τους
- 5.7. Ερωτήσεις - Ασκήσεις

ΕΝΟΤΗΤΑ 6. ΥΠΕΡΦΟΡΤΩΣΗ ΜΕΘΟΔΩΝ (METHODS' OVERLOADING)

Στόχοι:

- Να κατανοήσουν οι εκπαιδευόμενοι τι είναι η υπερφόρτωση μεθόδων και πότε έχουμε αποδεκτές υπερφορτωμένες μεθόδους.
- Να μπορούν να υλοποιούν υπερφορτωμένες μεθόδους μέσα σε μια κλάση.
- Να μπορούν να “επιλύουν” το “ποια μέθοδος θα εκτελεστεί” από τις υπερφορτωμένες μεθόδους.

Τμήματα Ενότητας

- 6.1. Τι είναι υπερφόρτωση μεθόδων
- 6.2. Παραδείγματα προγραμμάτων με υπερφορτωμένες μεθόδους
- 6.3. Ερωτήσεις - Ασκήσεις

ΕΝΟΤΗΤΑ 7. ΙΕΡΑΡΧΙΕΣ ΚΛΑΣΕΩΝ (CLASS HIERARCHIES) - ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ (INHERITANCE) – ΠΟΛΥΜΟΡΦΙΣΜΟΣ (POLYMORPHISM) – ΔΥΝΑΜΙΚΗ ΔΕΣΜΕΥΣΗ (DYNAMIC BINDING)

Στόχοι:

- Να κατανοήσουν οι εκπαιδευόμενοι την έννοια της υποκλάσης που προκύπτει προσθέτοντας περισσότερες λεπτομέρειες σε υπάρχουσα κλάση (ειδίκευση-more concrete) και την έννοια της υπερκλάσης που προκύπτει βρίσκοντας κοινά χαρακτηριστικά μεταξύ πολλών κλάσεων (γενίκευση-more abstract).
- Να αντιλαμβάνονται τις ιεραρχίες κλάσεων που δημιουργούνται από υπερκλάσεις και υποκλάσεις και να είναι σε θέση να δημιουργούν ιεραρχίες κλάσεων.
- Να διακρίνουν τις σχέσεις ιεραρχίας από τις σχέσεις κλάσης-αντικειμένου.
- Να αντιλαμβάνονται το μηχανισμό της κληρονομικότητας και τα οφέλη που προσφέρει.
- Να κατανοήσουν την κληρονόμηση ιδιοτήτων και μεθόδων από την υπερκλάση στην υποκλάση σε όλη την έκταση μιας ιεραρχίας κλάσεων και τον προσδιοριστή ορατότητας protected.
- Να μπορούν να γενικεύουν και να ειδικεύουν μεταξύ οντοτήτων και να δημιουργούν ιεραρχίες κλάσεων.
- Να κατανοήσουν την υπερίσχυση μεθόδων (overriding) σε μια ιεραρχία κλάσεων και να μπορούν να αποφασίζουν για το ποια μέθοδος εκτελείται από τις overridden μεθόδους σε κάθε κλήση.

- Να κατανοήσουν τη σχέση “είναι” (is a) μεταξύ κλάσεων ιεραρχίας και την ανάθεση τύπων αναφοράς υποκλάσεων σε τύπους αναφοράς υπερκλάσεων.
- Να μπορούν να αποφασίσουν για την εκτέλεση overridden μεθόδων στις περιπτώσεις που αυτές καλούνται για τύπους αναφοράς διαφορετικού τύπου από ότι το αντικείμενο στο οποίο αυτοί δείχνουν (dynamic binding).
- Να διακρίνουν τις overridden και overloaded μεθόδους.

Τμήματα Ενότητας

- 7.1. Ιεραρχίες κλάσεων - Κληρονομικότητα
 - 7.1.1. Θεωρία και παραδείγματα
 - 7.1.2. Ένα παράδειγμα υπερίσχυσης μεθόδου, που χρησιμοποιούμε πολύ συχνά
 - 7.1.3. Δεσμευμένες λέξεις public, private, protected: Λυμένες ασκήσεις
 - 7.1.4. Δραστηριότητα στο περιβάλλον ArgoUML
- 7.2. Υπερίσχυση μεθόδων (method overriding) – κατασκευαστές και κληρονομικότητα: παραδείγματα, δεσμευμένη λέξη super, δραστηριότητα στο περιβάλλον BlueJ
- 7.3. Πολυμορφισμός (polymorphism) - Δυναμική κλήση μεθόδου (dynamic binding)
 - 7.3.1. Πολυμορφισμός
 - 7.3.1.1. Τύποι πολυμορφισμού
 - 7.3.1.2. Παραδείγματα επίλυσης κλήσης μεθόδου σε υπερφόρτωση και υπερίσχυση μεθόδων (σχηματική παράσταση μνήμης)
 - 7.3.2. Δυναμική κλήση μεθόδου (dynamic binding)
 - 7.3.2.1. Η σχέση “είναι ένα” (“is a”) (θεωρία)
 - 7.3.2.2. Παραδείγματα
 - 7.3.2.3. Μια χρήσιμη αναφορά
 - 7.3.3. Πώς ο σκύλος μου έμαθε πολυμορφισμό (μια περιγραφική διήγηση)
- 7.4. Υπερίσχυση σε αντιδιαστολή με υπερφόρτωση μεθόδων
 - 7.4.1. Θεωρία
 - 7.4.2. Παράδειγμα - λυμένη άσκηση
- 7.5. Αφηρημένες και τελικές κλάσεις και μέθοδοι - οι Διασυνδέσεις ή Διεπαφές (interfaces)
 - 7.5.1. Αφηρημένες κλάσεις
 - 7.5.2. Αφηρημένες κλάσεις και αφηρημένες μέθοδοι

- 7.5.3. Τελικές κλάσεις και μέθοδοι
- 7.5.4. Οι διασυνδέσεις ή Διεπαφές
- 7.5.5. Παράδειγμα - Κληρονόμηση μιας διαπροσωπείας από αφηρημένη κλάση και υλοποίηση της στις υποκλάσεις
- 7.5.6. Δραστηριότητα στο περιβάλλον ArgoUML
- 7.6. Ερωτήσεις - Ασκήσεις

ΕΝΟΤΗΤΑ 8. ΟΛΟΚΛΗΡΩΜΕΝΟ ΠΑΡΑΔΕΙΓΜΑ ή ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ (CASE STUDY)

Στόχοι:

- Να έρθουν σε επαφή οι εκπαιδευόμενοι με μια απλή εφαρμογή και όλη τη διαδικασία ανάλυσης, σχεδιασμού, υλοποίησης και ελέγχου της. Εφαρμόζονται στην πράξη και “επιδεικνύονται” οι αρχές του ΑΠ που έχουν παρουσιασθεί προηγουμένως και επίσης αναδεικνύεται η ανάγκη της παρουσίασης και χρήσης των συλλογών (collection classes) και της χρήσης των iterators για την προσπέλαση των στοιχείων τους.

Παρουσίαση case-study

- Τεκμηρίωση
 - Ένα παράδειγμα σχεδίασης προγράμματος
 - Εισαγωγή
 - Διατύπωση απαιτήσεων
 - Ανάπτυξη της λύσης
 - Μια γενική ανασκόπηση
- Δοκιμή
 - Δοκιμή κλάσεων
 - Δοκιμή εφαρμογής
- Κώδικας κλάσεων
- Υλοποίηση στο BlueJ

ΠΑΡΑΡΤΗΜΑ Β

Ερωτηματολόγιο προ-αξιολόγησης μελέτης περίπτωσης 1

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΠΡΟ-ΑΞΙΟΛΟΓΗΣΗΣ ΓΙΑ ΤΟΝ «ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ»

Αγαπητές φοιτήτριες, Αγαπητοί φοιτητές,

Πρόκειται να παρακολουθήσετε ένα σεμινάριο και να χρησιμοποιήσετε μαθησιακό υλικό που αφορά στον «Αντικειμενοστρεφή Προγραμματισμό».

Με το υλικό αυτό επιδιώκουμε να εντοπίσουμε μαθησιακές σας δυσκολίες στην εκμάθηση αυτού του αντικειμένου και να προτείνουμε τρόπους διδασκαλίας για να ξεπεραστούν οι δυσκολίες αυτές.

Με την προσπάθεια αυτή θέλουμε να συμβάλλουμε στην αποτελεσματικότερη διδασκαλία του «Αντικειμενοστρεφούς Προγραμματισμού».

Η συνεργασία σας είναι εξαιρετικής σημασίας και θα συνδράμει σημαντικά στην προσπάθειά μας.

Το ερωτηματολόγιο που ακολουθεί, **δεν έχει καμία σχέση με την επίδοσή σας σε κάποιο μάθημα, δε θα χρησιμοποιηθεί για κανένα άλλο σκοπό** εκτός από αυτούς που προαναφέρθηκαν και **δε θα μετρήσει καθόλου στη βαθμολογία μαθημάτων**. Αποτελεί μέρος έρευνας για τη Διδακτική του Αντικειμενοστρεφούς Προγραμματισμού.

Παρακαλούμε, **να διαβάσετε προσεκτικά τις παρακάτω ερωτήσεις και να τις απαντήσετε με ειλικρίνεια και όση μεγαλύτερη ακρίβεια μπορείτε**, ώστε να συντελέσετε σε αυτήν την επιστημονική προσπάθεια που γίνεται.

Στις περιπτώσεις που πρέπει να επιλέξετε έναν αριθμό, σημειώστε τον αριθμό που ταιριάζει στην περίπτωσή σας ή που εκφράζει καλύτερα τη γνώμη σας. Παρακαλούμε, επίσης, να αποφύγετε να δηλώνετε ότι έχετε αδιάφορη ή ουδέτερη άποψη στα παρακάτω ερωτήματα εκτός κι αν είστε απόλυτα σίγουροι/ες για την αδιαφορία και ουδετερότητά σας.

Σε ορισμένες περιπτώσεις που σας ζητείται να γράψετε κάτι μόνος/η σας (χωρίς να επιλέξετε από έτοιμες απαντήσεις), γράψτε το όσο πιο συγκεκριμένα μπορείτε.

Σας ευχαριστούμε εκ των προτέρων για την προσπάθειά σας και το χρόνο που θα αφιερώσετε για τη συμπλήρωση του ερωτηματολογίου.

Όνοματεπώνυμο:

.....

1. Φύλο: Γυναίκα Άνδρας

2. Ηλικία:

3. Εξάμηνο Σπουδών:

4. Από ποια κατεύθυνση του Ενιαίου Λυκείου προέρχετε;
Θετική Τεχνολογική

5. Ποια μαθήματα έχετε διδαχθεί στο Ενιαίο Λύκειο σχετικά με Υπολογιστές (σημειώστε αν και όσα θυμάστε);

α) Ανάπτυξη Εφαρμογών σε Προγραμματιστικό περιβάλλον
Βαθμός

β) Εφαρμογές Υπολογιστών

γ) Εφαρμογές Πληροφορικής

δ) Πολυμέσα – Δίκτυα

ε) Τεχνολογία Υπολογιστικών Συστημάτων και Λειτουργικά Συστήματα

στ) Εφαρμογές Λογισμικού

ζ) Άλλο/α

Ποιο/α;

.....
.....
.....

η) Κανένα

6. Ποιο μάθημα έχετε διδαχθεί στο τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων σχετικά με Προγραμματισμό;

α) Αρχές και γλώσσες προγραμματισμού

Ποια γλώσσα/ες προγραμματισμού περιελάμβανε το μάθημα;

.....

β) Αντικειμενοστρεφής σχεδιασμός και Προγραμματισμός

Ποια γλώσσα/ες προγραμματισμού περιελάμβανε το μάθημα;

.....

γ) Άλλο/α

Ποιο/α;

.....
.....
Ποια γλώσσα/ες προγραμματισμού περιελάμβανε το μάθημα;
.....
.....

7. Έχετε διδαχθεί κάπου αλλού εκτός του Ενιαίου Λυκείου και του Πανεπιστημίου Πειραιά, προγραμματισμό;

Όχι Ναι

Αν ναι, πού;

.....
.....

Ποια/ες γλώσσα/ες προγραμματισμού:

.....
.....

8. Έχετε με άλλον τρόπο έλθει σε επαφή με τον Προγραμματισμό; (π.χ. επαγγελματικά, από hobby κ.λ.π.)

Όχι Ναι

Αν ναι, με ποιον;

.....
.....

Με ποια/ες γλώσσα/ες;

.....
.....

9. Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον Προγραμματισμό;

5	4	3	2	1
Επαγγελματίας	Πολύ έμπειρος/η	Μέτρια έμπειρος/η	Λίγο έμπειρος/η	Αρχάριος/α

Στις παρακάτω δηλώσεις επιλέξτε τον αριθμό, που αντιστοιχεί στην απάντηση που περιγράφει καλύτερα αυτό που ισχύει για σας:

10. Πάντα καταφέρνω να λύνω δύσκολα προβλήματα, αν προσπαθήσω αρκετά σκληρά.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

11. Αν κάποιος μου αντιτίθεται, μπορώ να βρω μέσα και τρόπους για να κάνω αυτό που θέλω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

12. Είμαι βέβαιος/η ότι μπορώ να πετύχω τους στόχους μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

13. Είμαι πεπεισμένος/η ότι μπορώ να αντιμετωπίσω αποτελεσματικά απρόσμενα γεγονότα.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

14. Χάρη στην επινοητικότητά μου, μπορώ να χειρίζομαι απρόοπτες καταστάσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

15. Μπορώ να λύνω τα περισσότερα προβλήματα, αν καταβάλλω την απαραίτητη προσπάθεια.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

16. Παραμένω ήρεμος/η όταν συναντώ δυσκολίες, επειδή μπορώ να βασίζομαι στις ικανότητές μου να τις αντιμετωπίσω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

17. Όταν βρεθώ αντιμέτωπος/η με ένα πρόβλημα, μπορώ να βρω αρκετές λύσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

18. Όταν βρίσκομαι σε μια δύσκολη κατάσταση, μπορώ να σκεφτώ μια καλή λύση.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

19. Είμαι ικανός/η να χειριστώ οτιδήποτε παρουσιάζεται στο δρόμο μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

Αξιολογείστε τους τρόπους εκμάθησης που προτιμάτε:

20. Προτιμώ να μελετώ μόνος/η μου τα μαθήματά μου παρά να παρακολουθώ διαλέξεις.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

21. Βαριέμαι γρήγορα όταν ακολουθώ λεπτομερείς διαδικασίες στον τρόπο εκμάθησης-διδασκαλίας.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

22. Πάντα ζητώ να μάθω περισσότερα από όσα παίρνω από τη διδασκαλία ενός μαθήματος, δηλαδή ψάχνω σε άλλες πηγές για μαθησιακό υλικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

23. Προτιμώ να περιέχονται παραδείγματα στο μαθησιακό υλικό των μαθημάτων.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

24. Δίνω μεγαλύτερο βάρος στο πρακτικό μέρος ενός γνωστικού αντικειμένου (ασκήσεις, δραστηριότητες), από ότι στο θεωρητικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

25. Προτιμώ να μαθαίνω τις θεωρητικές λεπτομέρειες του αντικειμένου που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

26. Χρειάζομαι την πνευματική συνδρομή των άλλων σπουδαστών όταν μελετώ ένα αντικείμενο.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

27. Μου αρέσουν περισσότερο εργασίες που βασίζονται στην ομαδική συνεργασία.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

28. Μου αρέσει να πειραματίζομαι με περιβάλλοντα εξομοίωσης που είναι διαθέσιμα για το αντικείμενο που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

Εκτιμείστε τη σημασία που θεωρείτε ότι θα έχουν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας κατά τη διάρκεια του σεμιναρίου.

29. Ύπαρξη των διαλέξεων στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

30. Ύπαρξη του σχετικού με το σεμινάριο υλικού στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

31. Ύπαρξη μελετών περίπτωσης (case studies) (δηλαδή ολοκληρωμένων παραδειγμάτων) στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

32. Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος Moodle.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

33. Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος Moodle.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

34. Η λύση ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

35. Η εκπόνηση εργασιών/ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

36. Η ενασχόληση με τα περιβάλλοντα που θα είναι διαθέσιμα στο σεμινάριο.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

Πείτε τη γνώμη σας όπως προκύπτει από την προηγούμενη εμπειρία σας στον Προγραμματισμό.

37. Σας αρέσει το αντικείμενο του Προγραμματισμού;

5	4	3	2	1
Πολύ	Αρκετά	Λίγο	Καθόλου	Με αφήνει αδιάφορο/η

38. Δυσκολεύεστε με τον Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

39. Αντιμετωπίζετε γενικά με φόβο τον Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

40. Θεωρείτε ότι ο Προγραμματισμός απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

41. Θεωρείτε ότι μάθατε Προγραμματισμό με αυτά που έχετε διδαχθεί έως τώρα;

4	3	2	1
Έμαθα καλά	Έμαθα αρκετά	Απέκτησα μια μικρή ιδέα	Δεν έμαθα καθόλου

42. Τι είναι αυτό που σας άρεσε περισσότερο από τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....

.....

.....

43. Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....

.....

.....

44. Θεωρείτε ότι στα Μαθηματικά είστε:

5	4	3	2	1
Πολύ καλός/η	Καλός/η	Μέτριος/α	Αδύνατος/η	Πολύ αδύνατος/η

45. Έχετε περάσει το μάθημα του Προγραμματισμού (Αρχές και Γλώσσες Προγραμματισμού);

Όχι Ναι

Αν ναι με τι Βαθμό;

Πόσες φορές έχετε συμμετάσχει στις εξετάσεις του μαθήματος;

.....

46. Έχετε περάσει το μάθημα «Αντικειμενοστρεφής σχεδιασμός και Προγραμματισμός»;

Όχι Ναι

Αν ναι με τι Βαθμό;

Πόσες φορές έχετε συμμετάσχει στις εξετάσεις του μαθήματος;

.....

Σας ευχαριστούμε πολύ για τη συμμετοχή σας στην έρευνα και την προσεκτική απάντηση στις ερωτήσεις... Ευχόμαστε καλή συνέχεια στις σπουδές σας...

ΠΑΡΑΡΤΗΜΑ Γ

Ερωτηματολόγιο μετά-αξιολόγησης μελέτης περίπτωσης 1

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΜΕΤΑ-ΑΞΙΟΛΟΓΗΣΗΣ ΓΙΑ ΤΟΝ «ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ»

Αγαπητές φοιτήτριες, Αγαπητοί φοιτητές,

Παρακολουθήσατε ένα σεμινάριο και χρησιμοποιήσατε μαθησιακό υλικό που αφορά στον «Αντικειμενοστρεφή Προγραμματισμό».

Με το υλικό αυτό επιδιώκουμε να εντοπίσουμε μαθησιακές σας δυσκολίες στην εκμάθηση αυτού του αντικειμένου και να προτείνουμε τρόπους διδασκαλίας για να ξεπεραστούν οι δυσκολίες αυτές.

Με την προσπάθεια αυτή θέλουμε να συμβάλλουμε στην αποτελεσματικότερη διδασκαλία του «Αντικειμενοστρεφούς Προγραμματισμού».

Η συνεργασία σας είναι εξαιρετικής σημασίας και θα συνδράμει σημαντικά στην προσπάθειά μας.

Το ερωτηματολόγιο που ακολουθεί, **δεν έχει καμία σχέση με την επίδοσή σας σε κάποιο μάθημα, δε θα χρησιμοποιηθεί για κανένα άλλο σκοπό** εκτός από αυτούς που προαναφέρθηκαν και **δε θα μετρήσει καθόλου στη βαθμολογία μαθημάτων**. Αποτελεί μέρος έρευνας για τη Διδακτική του Αντικειμενοστρεφούς Προγραμματισμού.

Παρακαλούμε, **να διαβάσετε προσεκτικά τις παρακάτω ερωτήσεις και να τις απαντήσετε με ειλικρίνεια και όση μεγαλύτερη ακρίβεια μπορείτε**, ώστε να συντελέσετε σε αυτήν την επιστημονική προσπάθεια που γίνεται.

Οι προτάσεις σας και τα σχόλιά σας είναι ιδιαίτερα σημαντικά για την αξιολόγηση και την περαιτέρω βελτίωση του υλικού. Οι απόψεις σας είναι πολύτιμες και αναγκαίες.

Στις περιπτώσεις που πρέπει να επιλέξετε έναν αριθμό, σημειώστε τον αριθμό που ταιριάζει στην περίπτωσή σας ή που εκφράζει καλύτερα τη γνώμη σας. Παρακαλούμε, επίσης, να αποφύγετε να δηλώνετε ότι έχετε αδιάφορη ή ουδέτερη άποψη στα παρακάτω ερωτήματα εκτός κι αν είστε απόλυτα σίγουροι/ες για την αδιαφορία και ουδετερότητά σας.

Σε ορισμένες περιπτώσεις που σας ζητείται να γράψετε κάτι μόνος/η σας (χωρίς να επιλέξετε από έτοιμες απαντήσεις), γράψτε το όσο πιο συγκεκριμένα μπορείτε.

Σας ευχαριστούμε εκ των προτέρων για την προσπάθειά σας, το χρόνο που θα αφιερώσετε για τη συμπλήρωση του ερωτηματολογίου και την προσεκτική απάντησή στις ερωτήσεις.

Όνοματεπώνυμο:

.....

Αξιολογείστε τη σημασία που θεωρείτε ότι είχαν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας.

1. Ύπαρξη των διαλέξεων στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

2. Ύπαρξη του σχετικού με το μάθημα υλικού στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

3. Ύπαρξη μελετών περίπτωσης (case studies) στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

4. Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του περιβάλλοντος Moodle.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

5. Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του περιβάλλοντος Moodle.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

6. Το υλικό που προήλθε από συζητήσεις μεταξύ συμφοιτητών μου μέσω του περιβάλλοντος Moodle (απλή ανάγνωση μηνυμάτων κι όχι ίδια συμμετοχή στη συζήτηση).

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

7. Το υλικό που προήλθε από συζητήσεις των συμφοιτητών μου με τους διδάσκοντες μέσω του περιβάλλοντος Moodle (απλή ανάγνωση μηνυμάτων κι όχι ίδια συμμετοχή στη συζήτηση).

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

8. Η λύση ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

9. Η εκπόνηση εργασιών/ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

10. Οι διορθώσεις/παρατηρήσεις από τους διδάσκοντες στις εργασίες/ασκήσεις που παρέδιδα.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

11. Η ενασχόληση με τα περιβάλλοντα που ήταν διαθέσιμα στο μάθημα.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

12. Σας αρέσει το αντικείμενο του Προγραμματισμού;

5	4	3	2	1
Πολύ	Αρκετά	Λίγο	Καθόλου	Με αφήνει αδιάφορο/η

13. Δυσκολεύεστε με τον Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

14. Αντιμετωπίζετε γενικά με φόβο τον Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

15. Θεωρείτε ότι ο Προγραμματισμός απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

16. Θεωρείτε ότι μάθατε Προγραμματισμό με αυτά που έχετε διδαχθεί έως τώρα;

4	3	2	1
Έμαθα καλά	Μπήκα στο νόημα	Απέκτησα μια μικρή ιδέα	Δεν έμαθα καθόλου

17. Τι είναι αυτό που σας άρεσε περισσότερο από τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....

.....

.....

18. Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....

.....

.....

Αξιολογήστε τη γνώμη σας για τα παρακάτω:

19. Το σεμινάριο με βοήθησε να ξεπεράσω δυσκολίες που αντιμετώπιζα σε σχέση με τον Προγραμματισμό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

20. Το σεμινάριο με βοήθησε να ξεπεράσω αδυναμίες μου σε σχέση με τον Προγραμματισμό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

21. Το σεμινάριο με βοήθησε να ξεπεράσω φοβίες που είχα σε σχέση με τον Προγραμματισμό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

22. Πώς θα χαρακτηρίζατε το περιεχόμενο του μαθησιακού υλικού;

5	4	3	2	1
Πολύ ενδιαφέρον	Ενδιαφέρον	Αδιάφορο	Κουραστικό	Δυσνόητο

23. Αξιολογήστε το βαθμό με τον οποίο οι παρακάτω μαθησιακοί πόροι συνέβαλαν στη βελτίωση των γνώσεων και των δεξιοτήτων σας.

Μαθησιακός πόρος	ΣΥΜΒΟΛΗ				
	5	4	3	2	1
Λυμένες ασκήσεις, παραδείγματα	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Θεωρία	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Αναπάντητες ερωτήσεις, άλλες ασκήσεις	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Τα αρχεία κώδικα Java που περιλαμβάνονται στο υλικό	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Τα υλοποιημένα projects στο περιβάλλον BLUEJ	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Οι λυμένες ασκήσεις στο περιβάλλον JEROO	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Η μελέτη περίπτωσης (case study)	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη

Σημειώστε την άποψή σας για τα παρακάτω

24. Το περιεχόμενο του μαθησιακού υλικού επιτυγχάνει τους στόχους που αναφέρονται στην αρχή κάθε ενότητας.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

25. Οι ερωτήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους στόχους εκμάθησης.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

26. Οι ασκήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους στόχους εκμάθησης.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

27. Η ποσότητα της δουλειάς που απαιτήθηκε για τη μελέτη του υλικού ήταν μεγάλη.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

28. Η προσπάθεια που καταβλήθηκε για την απάντηση/λύση των ερωτήσεων/ασκήσεων του υλικού ήταν μεγάλη.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

29. Ανατρέξτε και σε άλλες πηγές πέραν αυτών που παρέχονται στο υλικό για την εκμάθηση του γνωστικού αντικείμενου;

Όχι Ναι

Αν ναι, ποιες;

.....

30. Πόσο ασχοληθήκατε με τη συγγραφή προγραμμάτων σε JAVA κατά τη διάρκεια του σεμιναρίου;

5	4	3	2	1
5-10 ώρες την εβδομάδα	2-5 ώρες την εβδομάδα	2 ώρες την εβδομάδα	Λιγότερο από 2 ώρες την εβδομάδα	Καθόλου

31. Παρακάτω παρατίθενται έννοιες/θέματα της αντικειμενοστρεφούς φιλοσοφίας προγραμματισμού. Σημειώστε ποιες από αυτές σας δυσκόλεψαν και ιεραρχήστε τη δυσκολία μεταξύ τους σημειώνοντας αριθμούς δίπλα τους. Ξεκινήστε την αρίθμηση από 1 (αυτό που σας δυσκόλεψε λιγότερο) και για κάθε έννοια δυσκολότερη κατά την άποψή σας, αυξάνετε την αρίθμηση κατά 1.

- Κλάση - Αντικείμενο - Ιδιότητες - Μέθοδοι
- Κελυφοποίηση–απόκρυψη πληροφορίας (encapsulation–information hiding) - Διαχωρισμός συμπεριφοράς-υλοποίησης
- Προσδιοριστές ορατότητας (public, private)
- Δημιουργία αντικειμένων - Κατανομή μνήμης
- Μέθοδος Κατασκευαστή (constructor)
- Αναφορές σε αντικείμενα (references)
- Μεταβλητές – μέθοδοι κλάσης
- Υπερφόρτωση μεθόδων (methods overloading)
- Δημιουργία Ιεραρχιών κλάσεων
- Κληρονόμηση ιδιοτήτων – μεθόδων στις Ιεραρχίες κλάσεων (inheritance)
- Σχέση “is a”
- Υπερίσχυση μεθόδων (methods overriding)
- Δυναμική κλήση μεθόδου (Dynamic Binding)
- Πολυμορφισμός (Polymorphism)

32. Πώς εκτιμάτε μετά τη διεξαγωγή του σεμιναρίου, την ικανότητά σας να γράψετε προγράμματα σε Java;

5	4	3	2	1
Πολύ ικανός/η	Αρκετά ικανός/η	Μέτρια ικανός/η	Λίγο ικανός/η	Καθόλου ικανός/η

33. Για ποιο/ούς λόγους αρχικά είχατε αποφασίσει να παρακολουθήσετε το σεμινάριο;

- α) Με δυσκολεύει το αντικείμενο
- β) Δεν έχω περάσει το/α σχετικό/α μάθημα/τα
- γ) Επειδή με ενδιαφέρει το αντικείμενο
- δ) Ήθελα κάτι περισσότερο από ότι έχω διδαχθεί
- ε) Δεν είχα κατανοήσει το αντικείμενο παρακολουθώντας το/α μάθημα/τα στο Τμήμα
- στ) Για άλλο λόγο

Ποιον;

.....

.....

34. Ικανοποιήθηκαν οι λόγοι για τους οποίους παρακολουθήσατε;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

Ερωτήσεις ανοιχτού τύπου

35. Τι σας άρεσε περισσότερο στο μαθησιακό υλικό;

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....
36. Τι δε σας άρεσε;

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

37. Τι θα μπορούσε να σας βοηθήσει στην κατανόηση του γνωστικού αντικειμένου και δεν υπάρχει στο υλικό ή στην οργάνωση του σεμιναρίου;

.....
.....
.....
.....
.....
.....
.....
.....
.....

38. Ποιοι μαθησιακοί πόροι νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;

.....
.....
.....
.....
.....
.....
.....
.....

39. Ποιες ενότητες/τμήματα του υλικού νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;

.....

.....

.....

.....

.....

.....

.....

.....

.....

40. Σε τι κυρίως πιστεύετε ότι σας βοήθησε το περιβάλλον BLUEJ; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ότι άλλο εσείς νομίζετε)

.....

.....

.....

.....

Μπορείτε να αποδώσετε τη βοήθεια αυτή σε κάποια χαρακτηριστικά του περιβάλλοντος, όπως τα παρακάτω; Επιλέξτε αυτά που νομίζετε.

- Ευκολία χρήσης του περιβάλλοντος
- Οι κλάσεις και τα αντικείμενα αποκτούν οπτική υπόσταση
- Επιτρέπεται να αλληλεπιδράς απευθείας με τις κλάσεις και τα αντικείμενα
- Εύκολη δημιουργία αντικειμένων και οπτικοποίηση τους
- Εύκολη εκτέλεση των μεθόδων και έλεγχός τους (εύκολο πέρασμα παραμέτρων και επιστροφή αποτελέσματος), χωρίς να χρειάζεται να γράψεις κώδικα γι' αυτό το σκοπό
- Εύκολος έλεγχος των τιμών των ιδιοχαρακτηριστικών των αντικειμένων
- Εύκολο compilation με επισήμανση των συντακτικών λαθών απευθείας στο παράθυρο του editor με μήνυμα και επισήμανση της γραμμής κώδικα με το λάθος
- Διαθέτει όψη implementation και όψη interface για τις κλάσεις
- Διαθέτει debugger εύκολο στη χρήση
- Εύκολος πειραματισμός με άμεσο αποτέλεσμα (feedback)
- Άλλο/α, τι;

РАНЕКЪМЪО РЕПАА

ΠΑΡΑΡΤΗΜΑ Δ

Ερωτηματολόγιο προ-αξιολόγησης μελέτης περίπτωσης 2

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΠΡΟ-ΑΞΙΟΛΟΓΗΣΗΣ ΓΙΑ ΤΟ e-learning ΜΑΘΗΜΑ «ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ»

Αγαπητές φοιτήτριες, Αγαπητοί φοιτητές,

Πρόκειται να συμμετέχετε σε ένα e-learning μάθημα και να χρησιμοποιήσετε μαθησιακό υλικό που αφορά στον «Αντικειμενοστρεφή Προγραμματισμό».

Με το υλικό αυτό επιδιώκουμε να εντοπίσουμε μαθησιακές σας δυσκολίες στην εκμάθηση αυτού του αντικειμένου και να προτείνουμε τρόπους διδασκαλίας για να ξεπεραστούν οι δυσκολίες αυτές.

Με την προσπάθεια αυτή θέλουμε να συμβάλλουμε στην αποτελεσματικότερη διδασκαλία του «Αντικειμενοστρεφούς Προγραμματισμού».

Η συνεργασία σας είναι εξαιρετικής σημασίας και θα συνδράμει σημαντικά στην προσπάθειά μας.

Το ερωτηματολόγιο που ακολουθεί, αποτελεί μέρος έρευνας για τη Διδακτική του Αντικειμενοστρεφούς Προγραμματισμού.

Παρακαλούμε, να διαβάσετε προσεκτικά τις παρακάτω ερωτήσεις και να τις απαντήσετε με ειλικρίνεια και όση μεγαλύτερη ακρίβεια μπορείτε, ώστε να συντελέσετε σε αυτήν την επιστημονική προσπάθεια που γίνεται.

Στις περιπτώσεις που πρέπει να επιλέξετε έναν αριθμό, σημειώστε τον αριθμό που ταιριάζει στην περίπτωσή σας ή που εκφράζει καλύτερα τη γνώμη σας. Παρακαλούμε, επίσης, να αποφύγετε να δηλώνετε ότι έχετε αδιάφορη ή ουδέτερη άποψη στα παρακάτω ερωτήματα εκτός κι αν είστε απόλυτα σίγουροι/ες για την αδιαφορία και ουδετερότητά σας.

Σε ορισμένες περιπτώσεις που σας ζητείται να γράψετε κάτι μόνο/η σας (χωρίς να επιλέξετε από έτοιμες απαντήσεις), γράψτε το όσο πιο συγκεκριμένα μπορείτε.

Σας ευχαριστούμε εκ των προτέρων για την προσπάθειά σας και το χρόνο που θα αφιερώσετε για τη συμπλήρωση του ερωτηματολογίου.

Όνοματεπώνυμο:

.....

1. Φύλο: Γυναίκα Άνδρας

2. Ηλικία:

3. Εξάμηνο Σπουδών:

4. Από ποια κατεύθυνση του Ενιαίου Λυκείου προέρχετε;

Θετική Τεχνολογική

5. Ποια μαθήματα έχετε διδαχθεί στο Ενιαίο Λύκειο σχετικά με Υπολογιστές

(σημειώστε αν και όσα θυμάστε);

α) Ανάπτυξη Εφαρμογών σε Προγραμματιστικό περιβάλλον

Βαθμός

β) Εφαρμογές Υπολογιστών

γ) Εφαρμογές Πληροφορικής

δ) Πολυμέσα – Δίκτυα

ε) Τεχνολογία Υπολογιστικών Συστημάτων και Λειτουργικά Συστήματα

στ) Εφαρμογές Λογισμικού

ζ) Άλλο/α

Ποιο/α;

.....
.....
.....

η) Κανένα

6. Ποιο μάθημα έχετε διδαχθεί στο τμήμα Διδακτικής της Τεχνολογίας και

Ψηφιακών Συστημάτων σχετικό με Προγραμματισμό;

α) Αρχές και γλώσσες προγραμματισμού

Ποια γλώσσα/ες προγραμματισμού περιελάμβανε το μάθημα;

.....

β) Αντικειμενοστρεφής σχεδιασμός και Προγραμματισμός

Ποια γλώσσα/ες προγραμματισμού περιελάμβανε το μάθημα;

.....

γ) Άλλο/α

Ποιο/α;

.....

.....
Ποια γλώσσα/ες προγραμματισμού περιελάμβανε το μάθημα;
.....

7. Έχετε διδαχθεί κάπου αλλού εκτός του Ενιαίου Λυκείου και του Πανεπιστημίου Πειραιά, προγραμματισμό;

Όχι Ναι

Αν ναι, πού;

.....
.....

Ποια/ες γλώσσα/ες προγραμματισμού:

.....
.....

8. Έχετε με άλλον τρόπο έλθει σε επαφή με τον Προγραμματισμό; (π.χ. επαγγελματικά, από hobby κ.λ.π.)

Όχι Ναι

Αν ναι, με ποιον;

.....
.....

Με ποια/ες γλώσσα/ες;

.....
.....

9. Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον Προγραμματισμό;

5	4	3	2	1
Επαγγελματίας	Πολύ έμπειρος/η	Μέτρια έμπειρος/η	Λίγο έμπειρος/η	Αρχάριος/α

Στις παρακάτω δηλώσεις επιλέξτε τον αριθμό, που αντιστοιχεί στην απάντηση που περιγράφει καλύτερα αυτό που ισχύει για σας:

10. Πάντα καταφέρνω να λύνω δύσκολα προβλήματα, αν προσπαθήσω αρκετά σκληρά.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

11. Αν κάποιος μου αντιτίθεται, μπορώ να βρω μέσα και τρόπους για να κάνω αυτό που θέλω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

12. Είμαι βέβαιος/η ότι μπορώ να πετύχω τους στόχους μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

13. Είμαι πεπεισμένος/η ότι μπορώ να αντιμετωπίσω αποτελεσματικά απρόσμενα γεγονότα.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

14. Χάρη στην επινοητικότητα μου, μπορώ να χειρίζομαι απρόοπτες καταστάσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

15. Μπορώ να λύνω τα περισσότερα προβλήματα, αν καταβάλλω την απαραίτητη προσπάθεια.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

16. Παραμένω ήρεμος/η όταν συναντώ δυσκολίες, επειδή μπορώ να βασίζομαι στις ικανότητές μου να τις αντιμετωπίζω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

17. Όταν βρεθώ αντιμέτωπος/η με ένα πρόβλημα, μπορώ να βρω αρκετές λύσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

18. Όταν βρίσκομαι σε μια δύσκολη κατάσταση, μπορώ να σκεφτώ μια καλή λύση.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

19. Είμαι ικανός/η να χειριστώ οτιδήποτε παρουσιάζεται στο δρόμο μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

Αξιολογήστε τους τρόπους εκμάθησης που προτιμάτε:

20. Προτιμώ να μελετώ μόνος/η μου τα μαθήματά μου παρά να παρακολουθώ διαλέξεις.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

21. Βαριέμαι γρήγορα όταν ακολουθώ λεπτομερείς διαδικασίες στον τρόπο εκμάθησης-διδασκαλίας.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

22. Πάντα ζητώ να μάθω περισσότερα από όσα παίρνω από τη διδασκαλία ενός μαθήματος, δηλαδή ψάχνω σε άλλες πηγές για μαθησιακό υλικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

23. Προτιμώ να περιέχονται παραδείγματα στο μαθησιακό υλικό των μαθημάτων.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

24. Δίνω μεγαλύτερο βάρος στο πρακτικό μέρος ενός γνωστικού αντικείμενου (ασκήσεις, δραστηριότητες), από ότι στο θεωρητικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

25. Προτιμώ να μαθαίνω τις θεωρητικές λεπτομέρειες του αντικείμενου που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

26. Χρειάζομαι την πνευματική συνδρομή των άλλων σπουδαστών όταν μελετώ ένα αντικείμενο.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

27. Μου αρέσουν περισσότερο εργασίες που βασίζονται στην ομαδική συνεργασία.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

28. Μου αρέσει να πειραματίζομαι με περιβάλλοντα εξομοίωσης που είναι διαθέσιμα για το αντικείμενο που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

Εκτιμείστε τη σημασία που θεωρείτε ότι θα έχουν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας κατά τη διάρκεια του e-learning μαθήματος.

29. Ύπαρξη των διαλέξεων στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

30. Ύπαρξη του σχετικού υλικού στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

31. Ύπαρξη μελετών περίπτωσης (case studies) (δηλαδή ολοκληρωμένων παραδειγμάτων) στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

32. Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

33. Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

34. Η λύση ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

35. Η εκπόνηση εργασιών/ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

36. Η ενασχόληση με τα περιβάλλοντα που θα είναι διαθέσιμα στο e-learning μάθημα.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

Πείτε τη γνώμη σας όπως προκύπτει από την προηγούμενη εμπειρία σας στον Προγραμματισμό.

37. Σας αρέσει το αντικείμενο του Προγραμματισμού;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

38. Δυσκολεύεστε με τον Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

39. Αντιμετωπίζετε γενικά με φόβο τον Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

40. Θεωρείτε ότι ο Προγραμματισμός απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

41. Θεωρείτε ότι μάθατε Προγραμματισμό με αυτά που έχετε διδαχθεί έως τώρα;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

42. Τι είναι αυτό που σας άρεσε περισσότερο από τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....
.....

43. Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....
.....

44. Θεωρείτε ότι στα Μαθηματικά είστε:

5	4	3	2	1
Πολύ καλός/η	Καλός/η	Μέτριος/α	Αδύνατος/η	Πολύ αδύνατος/η

45. Έχετε περάσει το μάθημα του Προγραμματισμού (Αρχές και Γλώσσες Προγραμματισμού);

Όχι Ναι

Αν ναι με τι Βαθμό;

Πόσες φορές έχετε συμμετάσχει στις εξετάσεις του μαθήματος;

.....

46. Είχατε περάσει το μάθημα «Αντικειμενοστρεφής σχεδιασμός και Προγραμματισμός» (Απαντήστε αν φοιτούσατε στο Τμήμα το 2003-04 και πριν);

Όχι Ναι

Αν ναι με τι Βαθμό;

Πόσες φορές είχατε συμμετάσχει στις εξετάσεις του μαθήματος;

.....

47. Έχετε περάσει το μάθημα «Αντικειμενοστρεφής Προγραμματισμός»;

Όχι Ναι

Αν ναι με τι Βαθμό;

Πόσες φορές έχετε συμμετάσχει στις εξετάσεις του μαθήματος;

.....

Σας ευχαριστούμε πολύ για τη συμμετοχή σας στην έρευνα και την προσεκτική απάντηση στις ερωτήσεις... Ευχόμαστε καλή συνέχεια στις σπουδές σας...

ΠΑΡΑΡΤΗΜΑ Ε

Ερωτηματολόγιο μετά-αξιολόγησης μελέτης περίπτωσης 2

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΜΕΤΑ-ΑΞΙΟΛΟΓΗΣΗΣ ΓΙΑ ΤΟ e-learning ΜΑΘΗΜΑ «ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ»
--

Αγαπητές φοιτήτριες, Αγαπητοί φοιτητές,

Παρακολουθήσατε ένα e-learning μάθημα και χρησιμοποιήσατε μαθησιακό υλικό που αφορά στον «Αντικειμενοστρεφή Προγραμματισμό».

Με το υλικό αυτό επιδιώκουμε να εντοπίσουμε μαθησιακές σας δυσκολίες στην εκμάθηση αυτού του αντικειμένου και να προτείνουμε τρόπους διδασκαλίας για να ξεπεραστούν οι δυσκολίες αυτές.

Με την προσπάθεια αυτή θέλουμε να συμβάλλουμε στην αποτελεσματικότερη διδασκαλία του «Αντικειμενοστρεφούς Προγραμματισμού».

Η συνεργασία σας είναι εξαιρετικής σημασίας και θα συνδράμει σημαντικά στην προσπάθειά μας.

Το ερωτηματολόγιο που ακολουθεί, αποτελεί μέρος έρευνας για τη Διδακτική του Αντικειμενοστρεφούς Προγραμματισμού.

Παρακαλούμε, **να διαβάσετε προσεκτικά τις παρακάτω ερωτήσεις και να τις απαντήσετε με ειλικρίνεια και όση μεγαλύτερη ακρίβεια μπορείτε**, ώστε να συντελέσετε σε αυτήν την επιστημονική προσπάθεια που γίνεται.

Οι προτάσεις σας και τα σχόλιά σας είναι ιδιαίτερα σημαντικά για την αξιολόγηση και την περαιτέρω βελτίωση του υλικού. Οι απόψεις σας είναι πολύτιμες και αναγκαίες.

Στις περιπτώσεις που πρέπει να επιλέξετε έναν αριθμό, σημειώστε τον αριθμό που ταιριάζει στην περίπτωσή σας ή που εκφράζει καλύτερα τη γνώμη σας. Παρακαλούμε, επίσης, να αποφύγετε να δηλώνετε ότι έχετε αδιάφορη ή ουδέτερη άποψη στα παρακάτω ερωτήματα εκτός κι αν είστε απόλυτα σίγουροι/ες για την αδιαφορία και ουδετερότητά σας.

Σε ορισμένες περιπτώσεις που σας ζητείται να γράψετε κάτι μόνος/η σας (χωρίς να επιλέξετε από έτοιμες απαντήσεις), γράψτε το όσο πιο συγκεκριμένα μπορείτε.

Σας ευχαριστούμε εκ των προτέρων για την προσπάθειά σας, το χρόνο που θα αφιερώσετε για τη συμπλήρωση του ερωτηματολογίου και την προσεκτική απάντηση στις ερωτήσεις.

Όνοματεπώνυμο:

.....

Στις παρακάτω δηλώσεις επιλέξτε τον αριθμό, που αντιστοιχεί στην απάντηση που περιγράφει καλύτερα αυτό που ισχύει για σας:

1. Πάντα καταφέρνω να λύνω δύσκολα προβλήματα, αν προσπαθήσω αρκετά σκληρά.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

2. Αν κάποιος μου αντιτίθεται, μπορώ να βρω μέσα και τρόπους για να κάνω αυτό που θέλω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

3. Είμαι βέβαιος/η ότι μπορώ να πετύχω τους στόχους μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

4. Είμαι πεπεισμένος/η ότι μπορώ να αντιμετωπίσω αποτελεσματικά απρόσμενα γεγονότα.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

5. Χάρη στην επινοητικότητά μου, μπορώ να χειρίζομαι απρόοπτες καταστάσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

6. Μπορώ να λύνω τα περισσότερα προβλήματα, αν καταβάλλω την απαραίτητη προσπάθεια.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

7. Παραμένω ήρεμος/η όταν συναντώ δυσκολίες, επειδή μπορώ να βασίζομαι στις ικανότητές μου να τις αντιμετωπίσω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

8. Όταν βρεθώ αντιμετώπος/η με ένα πρόβλημα, μπορώ να βρω αρκετές λύσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

9. Όταν βρίσκομαι σε μια δύσκολη κατάσταση, μπορώ να σκεφτώ μια καλή λύση.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

10. Είμαι ικανός/η να χειριστώ οτιδήποτε παρουσιάζεται στο δρόμο μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

Αξιολογήστε τους τρόπους εκμάθησης που προτιμάτε:

11. Προτιμώ να μελετώ μόνος/η μου τα μαθήματά μου παρά να παρακολουθώ διαλέξεις.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

12. Βαριέμαι γρήγορα όταν ακολουθώ λεπτομερείς διαδικασίες στον τρόπο εκμάθησης-διδασκαλίας.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

13. Πάντα ζητώ να μάθω περισσότερα από όσα παίρνω από τη διδασκαλία ενός μαθήματος, δηλαδή ψάχνω σε άλλες πηγές για μαθησιακό υλικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

14. Προτιμώ να περιέχονται παραδείγματα στο μαθησιακό υλικό των μαθημάτων.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

15. Δίνω μεγαλύτερο βάρος στο πρακτικό μέρος ενός γνωστικού αντικείμενου (ασκήσεις, δραστηριότητες), από ότι στο θεωρητικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

16. Προτιμώ να μαθαίνω τις θεωρητικές λεπτομέρειες του αντικείμενου που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

17. Χρειάζομαι την πνευματική συνδρομή των άλλων σπουδαστών όταν μελετώ ένα αντικείμενο.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

18. Μου αρέσουν περισσότερο εργασίες που βασίζονται στην ομαδική συνεργασία.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

19. Μου αρέσει να πειραματίζομαι με περιβάλλοντα εξομοίωσης που είναι διαθέσιμα για το αντικείμενο που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

Αξιολογήστε τη σημασία που θεωρείτε ότι είχαν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας κατά τη διάρκεια του e-learning μαθήματος.

20. Ύπαρξη του σχετικού με το e-learning μάθημα υλικού στο WWW (Σύνδεσμος “Εγγραφα” του e-class).

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

21. Ύπαρξη ολοκληρωμένων παραδειγμάτων (case studies), στο υλικό του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

22. Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

23. Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

24. Η δομή του e-learning μαθήματος, έτσι ώστε να σε οδηγεί “από το απλό στο σύνθετο” δηλ. από απλούστερες έννοιες να σε εισάγει σταδιακά (“βήμα-βήμα”) σε δυσκολότερες.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

25. Η οργάνωση της ύλης κάθε διδακτικής ενότητας του e-learning μαθήματος (με σύντομη θεωρία, παραδείγματα με κώδικα, λυμένες ασκήσεις, δραστηριότητες σε εκπαιδευτικά περιβάλλοντα).

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

26. Η επίλυση επιλεγμένων ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

27. Η εκπόνηση εκτενών εργασιών στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

28. Η ανάρτηση στο ηλεκτρονικό μαθησιακό περιβάλλον, προτεινόμενων λύσεων των ασκήσεων που περιέχονταν στις εργασίες που παρέδιδες.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

Απαντήστε για τον Αντικειμενοστρεφή Προγραμματισμό.

29. Σας αρέσει το αντικείμενο του Αντικειμενοστρεφούς Προγραμματισμού;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

30. Δυσκολεύεστε με τον Αντικειμενοστρεφή Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

31. Αντιμετωπίζετε γενικά με φόβο τον Αντικειμενοστρεφή Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

32. Θεωρείτε ότι ο Αντικειμενοστρεφής Προγραμματισμός απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

33. Θεωρείτε ότι μάθατε Αντικειμενοστρεφή Προγραμματισμό με αυτά που έχετε διδαχθεί έως τώρα;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

34. Τι είναι αυτό που σας άρεσε περισσότερο από τον Αντικειμενοστρεφή Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....

35. Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Αντικειμενοστρεφή Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....

36. α) Ποιο εκτιμάτε ότι είναι το επίπεδό σας στο Διαδικασιακό (procedural) Προγραμματισμό (γλώσσα C);

5	4	3	2	1
Επαγγελματίας	Πολύ έμπειρος/η	Μέτρια έμπειρος/η	Λίγο έμπειρος/η	Αρχάριος/α

β) Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον Αντικειμενοστρεφή (object-oriented) Προγραμματισμό (γλώσσα Java);

5	4	3	2	1
Επαγγελματίας	Πολύ έμπειρος/η	Μέτρια έμπειρος/η	Λίγο έμπειρος/η	Αρχάριος/α

Αξιολογήστε τη γνώμη σας για τα παρακάτω:

37. Το e-learning μάθημα με βοήθησε να ξεπεράσω αδυναμίες μου σε σχέση με τον Αντικειμενοστρεφή Προγραμματισμό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

38. Το e-learning μάθημα με βοήθησε να ξεπεράσω φοβίες που είχα σε σχέση με τον Αντικειμενοστρεφή Προγραμματισμό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

39. Πώς θα χαρακτηρίζατε το περιεχόμενο του μαθησιακού υλικού του e-learning μαθήματος;

5	4	3	2	1
Πολύ ενδιαφέρον	Ενδιαφέρον	Αδιάφορο	Κουραστικό	Δυσνόητο

40. Αξιολογήστε το βαθμό με τον οποίο οι παρακάτω μαθησιακοί πόροι του e-learning μαθήματος, συνέβαλαν στη βελτίωση των γνώσεων και των δεξιοτήτων σας.

Μαθησιακός πόρος	ΣΥΜΒΟΛΗ				
	5	4	3	2	1
Λυμένες ασκήσεις, παραδείγματα	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Θεωρία	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη

	5	4	3	2	1
Οι δραστηριότητες στο περιβάλλον BLUEJ (που περιέχονταν στο υλικό και πραγματοποιούσα κατά τη μελέτη μου)	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Τα αρχεία κώδικα Java που περιλαμβάνονται στο υλικό	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Τα υλοποιημένα projects στο περιβάλλον BLUEJ που περιλαμβάνονται στο υλικό	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Το ολοκληρωμένο παράδειγμα (case study) που περιλαμβάνεται στο υλικό	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Οι οδηγίες μελέτης της κάθε διδακτικής ενότητας	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Το ολοκληρωμένο παράδειγμα που παρέδωσα (εφαρμογή βιβλιοπωλείου)	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Οι εργασίες που παρέδωσα	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Το υλικό που	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη

προήλθε από τις συζητήσεις στην “Περιοχή συζητήσεων” (forum) του e-learning μαθήματος	σημαντική				ασήμαντη
	5	4	3	2	1
Οι συναντήσεις του e-learning μαθήματος και οι διαλέξεις σε αυτές τις συναντήσεις	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη

Σημειώστε την άποψή σας για τα παρακάτω:

41. Το περιεχόμενο του μαθησιακού υλικού επιτυγχάνει τους στόχους που αναφέρονται στην αρχή κάθε ενότητας.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

42. Οι ερωτήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους μαθησιακούς στόχους.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

43. Οι ασκήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους μαθησιακούς στόχους.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

44. Χρησιμοποιήσατε και άλλες πηγές πέραν αυτών του υλικού του e-learning μαθήματος, για την εκμάθηση του γνωστικού αντικειμένου;

Όχι Ναι

Αν ναι, ποιες;

45. Παρακάτω παρατίθενται έννοιες/θέματα της αντικειμενοστρεφούς φιλοσοφίας προγραμματισμού. Αξιολογήστε σε τι επίπεδο σας δυσκόλεψαν.

Έννοιες/θέματα	Επίπεδο δυσκολίας		
	3	2	1
Κλάση - Αντικείμενο - Ιδιότητες - Μέθοδοι	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Δημιουργία αντικειμένων - Κατανομή μνήμης	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Μέθοδος Κατασκευαστή (constructor)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Αναφορές σε αντικείμενα (references)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Κελυφοποίηση - απόκρυψη πληροφορίας (encapsulation-information hiding) - Διαχωρισμός συμπεριφοράς - υλοποίησης	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Προσδιοριστές ορατότητας (public, private, protected)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Μεταβλητές και μέθοδοι κλάσης (static members)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Υπερφόρτωση μεθόδων (methods overloading)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Δημιουργία ιεραρχιών κλάσεων	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Κληρονομία ιδιοτήτων - μεθόδων στις Ιεραρχίες κλάσεων (inheritance)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Σχέση «είναι» (“is a”)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Υπερίσχυση μεθόδων (methods overriding)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Δυναμική κλήση μεθόδου ή Δυναμική δέσμευση (Dynamic Binding)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Πολυμορφισμός (Polymorphism)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία

46. Πώς εκτιμάτε μετά τη διεξαγωγή του e-learning μαθήματος, την ικανότητά σας να γράψετε προγράμματα σε Java;

5	4	3	2	1
Πολύ ικανός/η	Αρκετά ικανός/η	Μέτρια ικανός/η	Λίγο ικανός/η	Καθόλου ικανός/η

47. Πόσο σίγουροι/ες αισθάνεστε ότι θα επιτύχετε στις εξετάσεις του μαθήματος;

5	4	3	2	1
Πολύ σίγουρος/η	Αρκετά σίγουρος/η	Μέτρια σίγουρος/η	Λίγο σίγουρος/η	Καθόλου σίγουρος/η

Ερωτήσεις ανοιχτού τύπου

(γράψτε τη γνώμη σας, όσο πιο συγκεκριμένα μπορείτε)

48. Τι σας άρεσε περισσότερο στο μαθησιακό υλικό;

.....
.....

49. Τι δε σας άρεσε;

.....
.....

50. Τι θα μπορούσε να σας βοηθήσει στην κατανόηση του γνωστικού αντικειμένου και δεν υπάρχει στο υλικό ή στην οργάνωση του e-learning μαθήματος;

.....
.....

51. Ποιοι μαθησιακοί πόροι νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;

.....
.....

52. Ποιες ενότητες/τμήματα του υλικού νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;

.....
.....

53. Σε τι κυρίως πιστεύετε ότι σας βοήθησε το περιβάλλον BLUEJ; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ό,τι άλλο εσείς νομίζετε)

.....
.....

Μπορείτε να αποδώσετε τη βοήθεια αυτή σε κάποια χαρακτηριστικά του περιβάλλοντος BlueJ, όπως τα παρακάτω; Επιλέξτε αυτά που νομίζετε.

- Ευκολία χρήσης του περιβάλλοντος
- Οι κλάσεις και τα αντικείμενα αποκτούν οπτική υπόσταση
- Επιτρέπεται να αλληλεπιδράς απευθείας με τις κλάσεις και τα αντικείμενα
- Εύκολη δημιουργία αντικειμένων και οπτικοποίησης τους
- Εύκολη εκτέλεση των μεθόδων και έλεγχός τους (εύκολο πέρασμα παραμέτρων και επιστροφή αποτελέσματος), χωρίς να χρειάζεται να γράφεις κώδικα γι' αυτό το σκοπό
- Εύκολος έλεγχος των τιμών των ιδιοχαρακτηριστικών των αντικειμένων
- Εύκολο compilation με επισήμανση των συντακτικών λαθών απευθείας στο παράθυρο του editor με μήνυμα και επισήμανση της γραμμής κώδικα με το λάθος
- Διαθέτει όψη implementation και όψη interface για τις κλάσεις
- Διαθέτει debugger εύκολο στη χρήση
- Εύκολος πειραματισμός με άμεσο αποτέλεσμα (feedback)
- Άλλο/α, τι;

.....

54. Σε τι κυρίως πιστεύετε ότι σας βοήθησε η συστηματική χρήση της διαγραμματικής τεχνικής (διαγράμματα κλάσεων και αντικειμένων) στο μαθησιακό υλικό; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ό,τι άλλο εσείς νομίζετε)

.....

Σας ευχαριστούμε για τη συνεργασία σας!
Ευχόμαστε καλή συνέχεια στις σπουδές σας...

ΠΑΡΑΡΤΗΜΑ ΣΤ

Ερωτηματολόγιο προ-αξιολόγησης μελέτης περίπτωσης 3

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΠΡΟ-ΑΞΙΟΛΟΓΗΣΗΣ ΓΙΑ ΤΟΝ «ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ»

Αγαπητοί σπουδαστές, Αγαπητές σπουδάστριες,

Πρόκειται να σας δοθεί μαθησιακό υλικό που αφορά στον «Αντικειμενοστρεφή Προγραμματισμό».

Με το υλικό αυτό επιδιώκουμε να εντοπίσουμε μαθησιακές σας δυσκολίες στην εκμάθηση αυτού του αντικειμένου και να προτείνουμε τρόπους διδασκαλίας για να ξεπεραστούν οι δυσκολίες αυτές.

Με την προσπάθεια αυτή θέλουμε να συμβάλλουμε στην αποτελεσματικότερη διδασκαλία του «Αντικειμενοστρεφούς Προγραμματισμού».

Η συνεργασία σας είναι εξαιρετικής σημασίας και θα συνδράμει σημαντικά στην προσπάθειά μας.

Το ερωτηματολόγιο που ακολουθεί, **δεν έχει καμία σχέση με την επίδοσή σας στο μάθημα, δε θα χρησιμοποιηθεί για κανένα άλλο σκοπό** εκτός από αυτούς που προαναφέρθηκαν και **δε θα μετρήσει καθόλου στη βαθμολογία μαθημάτων**. Αποτελεί μέρος έρευνας για τη Διδακτική του Αντικειμενοστρεφούς Προγραμματισμού.

Παρακαλούμε, να διαβάσετε προσεκτικά τις παρακάτω ερωτήσεις και να τις απαντήσετε με ειλικρίνεια και όση μεγαλύτερη ακρίβεια μπορείτε, ώστε να συντελέσετε σε αυτήν την επιστημονική προσπάθεια που γίνεται.

Στις περιπτώσεις που πρέπει να επιλέξετε έναν αριθμό, σημειώστε τον αριθμό που ταιριάζει στην περίπτωσή σας ή που εκφράζει καλύτερα τη γνώμη σας. Παρακαλούμε, επίσης, να αποφύγετε να δηλώνετε ότι έχετε αδιάφορη ή ουδέτερη άποψη στα παρακάτω ερωτήματα εκτός κι αν είστε απόλυτα σίγουροι/ες για την αδιαφορία και ουδετερότητά σας.

Σε ορισμένες περιπτώσεις που σας ζητείται να γράψετε κάτι μόνος/η σας (χωρίς να επιλέξετε από έτοιμες απαντήσεις), γράψτε το όσο πιο συγκεκριμένα μπορείτε.

Σας ευχαριστούμε εκ των προτέρων για την προσπάθειά σας, το χρόνο που θα αφιερώσετε για τη συμπλήρωση του ερωτηματολογίου και την προσεκτική απάντηση στις ερωτήσεις.

Σας ευχόμαστε καλή επιτυχία στις προσπάθειές σας!

Όνοματεπώνυμο:

.....

1. Φύλο: Γυναίκα Άνδρας

2. Ηλικία:

3. Σπουδές:

Τμήμα Α.Ε.Ι. ή Τ.Ε.Ι. από το οποίο αποφοιτήσατε (Τμήμα, Σχολή, Εκπαιδευτικό Ίδρυμα):

.....
.....

Άλλες σπουδές που έχετε πραγματοποιήσει:

.....
.....

4. Έχετε διδαχθεί ξανά «Αντικειμενοστρεφή Προγραμματισμό»;

Όχι Ναι

Αν ναι πού;

.....
.....

Ποια/ες γλώσσα/ες προγραμματισμού διδαχθήκατε;

.....
.....

5. Ποια/ες άλλες γλώσσες προγραμματισμού έχετε διδαχθεί κατά τη διάρκεια των σπουδών σας;

.....
.....

6. Έχετε διδαχθεί κάπου αλλού εκτός των σπουδών σας, προγραμματισμό;

Όχι Ναι

Αν ναι, πού;

.....
.....

Γλώσσα/ες προγραμματισμού:

.....
.....

7. Έχετε έλθει σε επαφή με τον Προγραμματισμό με άλλον τρόπο; (π.χ. επαγγελματικά, από hobby κ.λ.π.)

Όχι Ναι

Αν ναι, με ποιον;

.....
.....

Με ποια/ες γλώσσα/ες;

.....

.....

8.

α) Ποιο εκτιμάτε ότι είναι το επίπεδό σας στο Διαδικασιακό (procedural) Προγραμματισμό (C, Pascal κ.λ.π.);

5	4	3	2	1
Επαγγελματίας	Πολύ έμπειρος/η	Μέτρια έμπειρος/η	Λίγο έμπειρος/η	Αρχάριος/α

β) Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον Αντικειμενοστρεφή (Object-oriented) Προγραμματισμό (C++, Java κ.λ.π.);

5	4	3	2	1
Επαγγελματίας	Πολύ έμπειρος/η	Μέτρια έμπειρος/η	Λίγο έμπειρος/η	Αρχάριος/α

Στις παρακάτω δηλώσεις επιλέξτε τον αριθμό, που αντιστοιχεί στην απάντηση που περιγράφει καλύτερα αυτό που ισχύει για σας:

9. Πάντα καταφέρνω να λύνω δύσκολα προβλήματα, αν προσπαθήσω αρκετά σκληρά.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

10. Αν κάποιος μου αντιτίθεται, μπορώ να βρω μέσα και τρόπους για να κάνω αυτό που θέλω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

11. Είμαι βέβαιος/η ότι μπορώ να πετύχω τους στόχους μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

12. Είμαι πεπεισμένος/η ότι μπορώ να αντιμετωπίσω αποτελεσματικά απρόσμενα γεγονότα.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

13. Χάρη στην επινοητικότητα μου, μπορώ να χειρίζομαι απρόοπτες καταστάσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

14. Μπορώ να λύνω τα περισσότερα προβλήματα, αν καταβάλλω την απαραίτητη προσπάθεια.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

15. Παραμένω ήρεμος/η όταν συναντώ δυσκολίες, επειδή μπορώ να βασίζομαι στις ικανότητές μου να τις αντιμετωπίζω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

16. Όταν βρεθώ αντιμέτωπος/η με ένα πρόβλημα, μπορώ να βρω αρκετές λύσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

17. Όταν βρίσκομαι σε μια δύσκολη κατάσταση, μπορώ να σκεφτώ μια καλή λύση.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

18. Είμαι ικανός/η να χειριστώ οτιδήποτε παρουσιάζεται στο δρόμο μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

Αξιολογείστε τους τρόπους εκμάθησης που προτιμάτε:

19. Προτιμώ να μελετώ μόνος/η μου τα μαθήματά μου παρά να παρακολουθώ διαλέξεις.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

20. Βαριέμαι γρήγορα όταν ακολουθώ λεπτομερείς διαδικασίες στον τρόπο εκμάθησης-διδασκαλίας.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

21. Πάντα ζητώ να μάθω περισσότερα από όσα παίρνω από τη διδασκαλία ενός μαθήματος, δηλαδή ψάχνω σε άλλες πηγές για μαθησιακό υλικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

22. Προτιμώ να περιέχονται παραδείγματα στο μαθησιακό υλικό των μαθημάτων.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

23. Δίνω μεγαλύτερο βάρος στο πρακτικό μέρος ενός γνωστικού αντικείμενου (ασκήσεις, δραστηριότητες), από ότι στο θεωρητικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

24. Προτιμώ να μαθαίνω τις θεωρητικές λεπτομέρειες του αντικείμενου που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

25. Χρειάζομαι την πνευματική συνδρομή των άλλων σπουδαστών όταν μελετώ ένα αντικείμενο.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

26. Μου αρέσουν περισσότερο εργασίες που βασίζονται στην ομαδική συνεργασία.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

27. Μου αρέσει να πειραματίζομαι με περιβάλλοντα εξομοίωσης που είναι διαθέσιμα για το αντικείμενο που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

Εκτιμείστε τη σημασία που θεωρείτε ότι θα έχουν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας κατά τη διάρκεια του e-learning μαθήματος.

28. Η ύπαρξη ολοκληρωμένων παραδειγμάτων στο WWW.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

29. Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

30. Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

31. Η επίλυση ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

32. Η εκπόνηση εργασιών στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

33. Η ενασχόληση με τα περιβάλλοντα εξομοίωσης που θα είναι διαθέσιμα στο e-learning μάθημα.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

Πείτε τη γνώμη σας όπως προκύπτει από την προηγούμενη εμπειρία σας στον Προγραμματισμό.

34. Σας αρέσει το αντικείμενο του Προγραμματισμού;

5	4	3	2	1
Πολύ	Αρκετά	Λίγο	Καθόλου	Με αφήνει αδιάφορο/η

35. Δυσκολεύεστε με τον Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

36. Αντιμετωπίζετε γενικά με φόβο τον Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

37. Θεωρείτε ότι ο Προγραμματισμός απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

38. Θεωρείτε ότι μάθατε Προγραμματισμό με αυτά που έχετε διδαχθεί έως τώρα;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

39. Τι είναι αυτό που σας άρεσε περισσότερο από τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....
.....
.....

40. Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....
.....
.....

Σας ευχαριστούμε για τη συνεργασία σας!

РАНЕЕЗНАМО ТЕРПАА

ΠΑΡΑΡΤΗΜΑ Ζ

Ερωτηματολόγιο μετά-αξιολόγησης μελέτης περίπτωσης 3

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΜΕΤΑ-ΑΞΙΟΛΟΓΗΣΗΣ ΓΙΑ ΤΟΝ «ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ»

Αγαπητοί σπουδαστές, Αγαπητές σπουδάστριες,

Παρακολουθήσατε ένα e-learning course και χρησιμοποιήσατε μαθησιακό υλικό που αφορά στον «Αντικειμενοστρεφή Προγραμματισμό».

Με το υλικό αυτό επιδιώκουμε να εντοπίσουμε μαθησιακές δυσκολίες στην εκμάθηση αυτού του αντικειμένου και να προτείνουμε τρόπους διδασκαλίας, για να ξεπεραστούν οι δυσκολίες αυτές.

Με αυτή την προσπάθεια θέλουμε να συμβάλλουμε στην αποτελεσματικότερη διδασκαλία του «Αντικειμενοστρεφούς Προγραμματισμού».

Η συνεργασία σας είναι εξαιρετικής σημασίας και θα συνδράμει σημαντικά στην προσπάθειά μας.

Το ερωτηματολόγιο που ακολουθεί, **δεν έχει καμία σχέση με την επίδοσή σας στο μάθημα, δε θα χρησιμοποιηθεί για κανένα άλλο σκοπό** εκτός από αυτούς που προαναφέρθηκαν και **δε θα μετρήσει καθόλου στη βαθμολογία μαθημάτων**. Αποτελεί μέρος έρευνας για τη Διδακτική του Αντικειμενοστρεφούς Προγραμματισμού.

Παρακαλούμε, να διαβάσετε **προσεκτικά** τις παρακάτω ερωτήσεις και να τις απαντήσετε με **ειλικρίνεια** και **όση μεγαλύτερη ακρίβεια μπορείτε**, ώστε να συντελέσετε σε αυτήν την επιστημονική προσπάθεια που γίνεται.

Οι προτάσεις και τα σχόλιά σας είναι ιδιαίτερα σημαντικά για την αξιολόγηση του e-learning course και την περαιτέρω βελτίωση του. Οι απόψεις σας είναι πολύτιμες και αναγκαίες.

Στις περιπτώσεις που πρέπει να επιλέξετε έναν αριθμό, σημειώστε τον αριθμό που ταιριάζει στην περίπτωσή σας ή που εκφράζει καλύτερα τη γνώμη σας. Παρακαλούμε, επίσης, να αποφύγετε να δηλώνετε ότι έχετε αδιάφορη ή ουδέτερη άποψη στα παρακάτω ερωτήματα, εκτός κι αν είστε απόλυτα σίγουροι/ες για την αδιαφορία και ουδετερότητά σας.

Σε ορισμένες περιπτώσεις που σας ζητείται να γράψετε κάτι μόνος/η σας (χωρίς να επιλέξετε από έτοιμες απαντήσεις), γράψτε το όσο πιο συγκεκριμένα μπορείτε.

Σας ευχαριστούμε εκ των προτέρων για την προσπάθειά σας, το χρόνο που θα αφιερώσετε για τη συμπλήρωση του ερωτηματολογίου και την προσεκτική απάντησή στις ερωτήσεις.

Σας ευχόμαστε καλή επιτυχία στις προσπάθειές σας!

Όνοματεπώνυμο:

.....
(υποχρεωτικό πεδίο, απαραίτητο για να συνδυαστούν τα 2 ερωτηματολόγια προ-αξιολόγησης και μετά-αξιολόγησης)

Στις παρακάτω δηλώσεις επιλέξτε τον αριθμό, που αντιστοιχεί στην απάντηση που περιγράφει καλύτερα αυτό που ισχύει για σας:

1. Πάντα καταφέρνω να λύνω δύσκολα προβλήματα, αν προσπαθήσω αρκετά σκληρά.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

2. Αν κάποιος μου αντιτίθεται, μπορώ να βρω μέσα και τρόπους για να κάνω αυτό που θέλω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

3. Είμαι βέβαιος/η ότι μπορώ να πετύχω τους στόχους μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

4. Είμαι πεπεισμένος/η ότι μπορώ να αντιμετωπίσω αποτελεσματικά απρόσμενα γεγονότα.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

5. Χάρη στην επινοητικότητά μου, μπορώ να χειρίζομαι απρόοπτες καταστάσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

6. Μπορώ να λύνω τα περισσότερα προβλήματα, αν καταβάλλω την απαραίτητη προσπάθεια.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

7. Παραμένω ήρεμος/η όταν συναντώ δυσκολίες, επειδή μπορώ να βασίζομαι στις ικανότητές μου να τις αντιμετωπίσω.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

8. Όταν βρεθώ αντιμέτωπος/η με ένα πρόβλημα, μπορώ να βρω αρκετές λύσεις.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

9. Όταν βρίσκομαι σε μια δύσκολη κατάσταση, μπορώ να σκεφτώ μια καλή λύση.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

10. Είμαι ικανός/η να χειριστώ οτιδήποτε παρουσιάζεται στο δρόμο μου.

4	3	2	1
Ισχύει απόλυτα	Μάλλον ισχύει	Μάλλον δεν ισχύει	Δεν ισχύει καθόλου

Αξιολογήστε τους τρόπους εκμάθησης που προτιμάτε:

11. Προτιμώ να μελετώ μόνος/η μου τα μαθήματά μου παρά να παρακολουθώ διαλέξεις.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

12. Βαριέμαι γρήγορα, όταν ακολουθώ λεπτομερείς διαδικασίες στον τρόπο εκμάθησης-διδασκαλίας.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

13. Πάντα ζητώ να μάθω περισσότερα από όσα παίρνω από τη διδασκαλία ενός μαθήματος, δηλαδή ψάχνω σε άλλες πηγές για μαθησιακό υλικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

14. Προτιμώ να περιέχονται παραδείγματα στο υλικό των μαθημάτων.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

15. Δίνω μεγαλύτερο βάρος στο πρακτικό μέρος ενός γνωστικού αντικείμενου (ασκήσεις, δραστηριότητες), από ότι στο θεωρητικό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

16. Προτιμώ να μαθαίνω τις θεωρητικές λεπτομέρειες του αντικείμενου που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

17. Χρειάζομαι την πνευματική συνδρομή των άλλων σπουδαστών, όταν μελετώ ένα αντικείμενο.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

18. Μου αρέσουν περισσότερο εργασίες που βασίζονται στην ομαδική συνεργασία.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

19. Μου αρέσει να πειραματίζομαι με περιβάλλοντα εξομοίωσης που είναι διαθέσιμα για το αντικείμενο που μελετώ.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

Αξιολογήστε τη σημασία που θεωρείτε ότι είχαν οι παρακάτω παράγοντες στη βελτίωση των γνώσεων και των δεξιοτήτων σας κατά τη διάρκεια του e-learning course.

20. Η ύπαρξη ολοκληρωμένων παραδειγμάτων (case studies), στο υλικό του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

21. Η δυνατότητα συζητήσεων με συμφοιτητές μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

22. Η δυνατότητα συζητήσεων με τους διδάσκοντες μέσω του ηλεκτρονικού μαθησιακού περιβάλλοντος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

23. Η δομή του e-learning μαθήματος, έτσι ώστε να σε οδηγεί “από το απλό στο σύνθετο” δηλ. από απλούστερες έννοιες να σε εισάγει σταδιακά (“βήμα-βήμα”) σε δυσκολότερες.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

24. Η οργάνωση της ύλης κάθε διδακτικής ενότητας του e-learning μαθήματος (με σύντομη θεωρία, παραδείγματα με κώδικα, λυμένες ασκήσεις, δραστηριότητες σε εκπαιδευτικά περιβάλλοντα).

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

25. Η επίλυση επιλεγμένων ασκήσεων στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

26. Η εκπόνηση εκτενών εργασιών στο πλαίσιο του μαθήματος.

5	4	3	2	1
Πολύ σημαντικός	Σημαντικός	Αδιάφορος	Ασήμαντος	Πολύ ασήμαντος

Απαντήστε για τον Αντικειμενοστρεφή Προγραμματισμό.

27. Σας αρέσει το αντικείμενο του Αντικειμενοστρεφούς Προγραμματισμού;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

28. Δυσκολεύεστε με τον Αντικειμενοστρεφή Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

29. Αντιμετωπίζετε γενικά με φόβο τον Αντικειμενοστρεφή Προγραμματισμό;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

30. Θεωρείτε ότι ο Αντικειμενοστρεφής Προγραμματισμός απαιτεί πολλές ικανότητες και μεγάλη προσπάθεια;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

31. Θεωρείτε ότι μάθατε Αντικειμενοστρεφή Προγραμματισμό με αυτά που έχετε διδαχθεί έως τώρα;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

32. Τι είναι αυτό που σας άρεσε περισσότερο από τον Αντικειμενοστρεφή Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....

.....

.....

.....

.....

.....

33. Τι είναι αυτό που σας δημιούργησε δυσκολίες με τον Αντικειμενοστρεφή Προγραμματισμό από τη μέχρι τώρα επαφή σας με το αντικείμενο;

.....

.....

.....

.....

.....

.....

Αξιολογήστε τη γνώμη σας για τα παρακάτω:

34. Το e-learning μάθημα με βοήθησε να ξεπεράσω αδυναμίες μου σε σχέση με τον Αντικειμενοστρεφή Προγραμματισμό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

35. Το e-learning μάθημα με βοήθησε να ξεπεράσω φοβίες που είχα σε σχέση με τον Αντικειμενοστρεφή Προγραμματισμό.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

36. Πώς θα χαρακτηρίζατε το περιεχόμενο του μαθησιακού υλικού;

5	4	3	2	1
Πολύ ενδιαφέρον	Ενδιαφέρον	Αδιάφορο	Κουραστικό	Δυσνόητο

37. Αξιολογήστε το βαθμό με τον οποίο οι παρακάτω μαθησιακοί πόροι συνέβαλαν στη βελτίωση των γνώσεων και των δεξιοτήτων σας.

Μαθησιακός πόρος	ΣΥΜΒΟΛΗ				
	5	4	3	2	1
Λυμένες ασκήσεις, παραδείγματα	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Θεωρία	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Αναπάντητες ερωτήσεις, άλυτες ασκήσεις	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Οι δραστηριότητες στο περιβάλλον BLUEJ	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
Οι δραστηριότητες	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη

στο περιβάλλον ArgoUML					
	5	4	3	2	1
Τα αρχεία κώδικα Java που περιλαμβάνονται στο υλικό	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
	5	4	3	2	1
Τα υλοποιημένα projects στο περιβάλλον BLUEJ που περιλαμβάνονται στο υλικό	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
	5	4	3	2	1
Το ολοκληρωμένο παράδειγμα (case study) που περιλαμβάνεται στο υλικό	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
	5	4	3	2	1
Οι οδηγίες μελέτης της κάθε διδακτικής ενότητας	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
	5	4	3	2	1
Το ολοκληρωμένο παράδειγμα που παρέδωσα (εφαρμογή βιβλιοπωλείου)	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
	5	4	3	2	1
Το υλικό που προήλθε από τις συζητήσεις στο forum του e-learning μαθήματος	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη
	5	4	3	2	1
Οι συναντήσεις του e-learning μαθήματος και οι διαλέξεις σε αυτές τις συναντήσεις	Πολύ σημαντική	Σημαντική	Ουδέτερη	Ασήμαντη	Πολύ ασήμαντη

Σημειώστε την άποψή σας για τα παρακάτω

- 38.** Το περιεχόμενο του μαθησιακού υλικού επιτυγχάνει τους στόχους που αναφέρονται στην αρχή κάθε ενότητας.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

- 39.** Οι ερωτήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους μαθησιακούς στόχους.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

- 40.** Οι ασκήσεις του μαθήματος ήταν κατάλληλες σε σχέση με τους μαθησιακούς στόχους.

5	4	3	2	1
Συμφωνώ απόλυτα	Συμφωνώ	Ουδέτερη άποψη	Διαφωνώ	Διαφωνώ απόλυτα

- 41.** Χρησιμοποιήσατε και άλλες πηγές πέραν αυτών του υλικού του e-learning μαθήματος, για την εκμάθηση του γνωστικού αντικειμένου;

Όχι Ναι

Αν ναι, ποιες;

.....
.....
.....

- 42.** Παρακάτω παρατίθενται έννοιες/θέματα της αντικειμενοστρεφούς φιλοσοφίας προγραμματισμού. Αξιολογήστε σε τι επίπεδο σας δυσκόλεψαν.

Έννοιες/θέματα	Επίπεδο δυσκολίας		
Κλάση - Αντικείμενο - Ιδιότητες - Μέθοδοι	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Δημιουργία αντικειμένων - Κατανομή	3	2	1

	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Μέθοδος Κατασκευαστή (constructor)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Αναφορές σε αντικείμενα (references)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Κελυφοποίηση - απόκρυψη πληροφορίας (encapsulation-information hiding) - Διαχωρισμός συμπεριφοράς - υλοποίησης	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Προσδιοριστές ορατότητας (public, private, protected)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Μεταβλητές και μέθοδοι κλάσης (static members)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Υπερφόρτωση μεθόδων (methods overloading)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Δημιουργία ιεραρχιών κλάσεων	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Κληρονομία ιδιοτήτων - μεθόδων στις Ιεραρχίες κλάσεων (inheritance)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Σχέση «είναι» (“is a”)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Υπερίσχυση μεθόδων (methods overriding)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Δυναμική κλήση μεθόδου ή Δυναμική δέσμευση (Dynamic Binding)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία
Πολυμορφισμός (Polymorphism)	3	2	1
	Μεγάλη δυσκολία	Μικρή δυσκολία	Καθόλου δυσκολία

43. Πώς εκτιμάτε μετά τη διεξαγωγή του e-learning course, την ικανότητά σας να γράψετε προγράμματα σε Java;

5	4	3	2	1
Πολύ ικανός/η	Αρκετά ικανός/η	Μέτρια ικανός/η	Λίγο ικανός/η	Καθόλου ικανός/η

44.

α) Ποιο εκτιμάτε ότι είναι το επίπεδό σας στο Διαδικασιακό (procedural) Προγραμματισμό (C, Pascal κ.λ.π.);

5	4	3	2	1
Επαγγελματίας	Πολύ έμπειρος/η	Μέτρια έμπειρος/η	Λίγο έμπειρος/η	Αρχάριος/α

β) Ποιο εκτιμάτε ότι είναι το επίπεδό σας στον Αντικειμενοστρεφή (Object-oriented) Προγραμματισμό μετά την παρακολούθηση του e-learning course;

5	4	3	2	1
Επαγγελματίας	Πολύ έμπειρος/η	Μέτρια έμπειρος/η	Λίγο έμπειρος/η	Αρχάριος/α

45. Για ποιο/ούς λόγους αρχικά είχατε αποφασίσει να παρακολουθήσετε το e-learning course; (Μπορείτε να σημειώσετε περισσότερα από ένα)

- α) Επειδή με ενδιαφέρει το αντικείμενο
- β) Επειδή με δυσκολεύει το αντικείμενο
- γ) Επειδή ήθελα να μάθω κάτι περισσότερο από ό,τι έχω διδαχθεί ή ήδη γνωρίζω
- δ) Επειδή μου χρειαζόταν στο πλαίσιο των Μεταπτυχιακών Σπουδών μου
- ε) Επειδή μου χρειαζόταν για επαγγελματικούς λόγους
- στ) Από περιέργεια
- ζ) Για άλλο λόγο

Ποιον;

.....

46. Ικανοποιήθηκαν οι λόγοι για τους οποίους παρακολουθήσατε;

5	4	3	2	1
Πολύ	Αρκετά	Μέτρια	Λίγο	Καθόλου

Ερωτήσεις ανοιχτού τύπου

(γράψτε τη γνώμη σας, όσο πιο συγκεκριμένα μπορείτε)

47. Τι σας άρεσε περισσότερο στο μαθησιακό υλικό του e-learning course;

.....

48. Τι δε σας άρεσε;

.....
.....
.....
.....
.....
.....

49. Τι θα μπορούσε να σας βοηθήσει στην κατανόηση του γνωστικού αντικειμένου και δεν υπάρχει στο υλικό ή στην οργάνωση του e-learning course;

.....
.....
.....
.....
.....
.....

50. Ποιοι μαθησιακοί πόροι νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;

.....
.....
.....
.....

51. Ποιες ενότητες/τμήματα του υλικού νομίζετε ότι χρειάζονται περαιτέρω βελτίωση;

.....
.....
.....
.....

52. Σε τι κυρίως πιστεύετε ότι σας βοήθησε το περιβάλλον BLUEJ; (έννοιες που σας βοήθησε να κατανοήσετε – αποσαφηνίσετε ή ό,τι άλλο εσείς νομίζετε)

.....
.....
.....

РАНЕЕЗНАМО ТЕРРА