



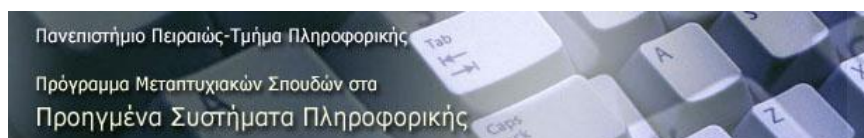
## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Προηγμένα Συστήματα Πληροφορικής»



### Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Εκπαιδευτικό λογισμικό σε UML</b>
Όνοματεπώνυμο Φοιτητή	<b>Κουνάδη Ιωάννα του Σπυρίδων</b>
Αριθμός Μητρώου	<b>ΜΠΣΠ/08017</b>
Κατεύθυνση	<b>Συστήματα Υποστήριξης Αποφάσεων</b>
Επιβλέπων	<b>Μαρία Βίβου, Καθηγήτρια</b>



Ημερομηνία Παράδοσης **Νοέμβριος 2010**

---

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μαρία Βίρβου

Δημήτριος Αποστόλου

Χαράλαμπος Κωνσταντόπουλος

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>Περίληψη</b> .....	3
<b>Εισαγωγή</b> .....	4
<b>ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ ΣΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ</b> .....	5
1.1. <b>Ορισμός</b> .....	5
1.2. <b>Ιστορικά στοιχεία</b> .....	5
1.3. <b>Πλεονεκτήματα</b> .....	6
1.4. <b>Η επίδραση της εκπαίδευσης</b> .....	7
1.4.1. <b>Θεωρίες μάθησης</b> .....	7
1.4.2. <b>Εκπαιδευτική τεχνολογία</b> .....	7
1.4.3. <b>Νέες μορφές εκπαίδευσης</b> .....	8
1.4.4. <b>Λειτουργίες της εκπαίδευσης – Οικονομική διάσταση</b> .....	8
1.4.5. <b>Εκπαιδευτική έρευνα – Εξατομικευμένη μάθηση</b> .....	9
1.4.6. <b>Ο ηλεκτρονικός υπολογιστής στην εκπαίδευση</b> .....	10
1.5. <b>Εκπαιδευτικό λογισμικό – μελέτες περίπτωσης</b> .....	10
1.5.1. Geometre’s Sketchpad .....	10
1.5.2. Modellus .....	13
1.5.3. Interactive physics .....	14
<b>ΚΕΦΑΛΑΙΟ 2 : ΕΙΣΑΓΩΓΗ ΣΤΗΝ UML</b> .....	15
2.1. <b>Ορισμός</b> .....	15
2.1.1. <b>Εκδόσεις της UML</b> .....	15
2.2. <b>Πεδίο χρήσης</b> .....	16
2.3. <b>Βασικές έννοιες</b> .....	16
2.3.1. <b>Βασικά Διαγράμματα UML</b> .....	19
2.4. <b>Παραδείγματα χρήσης</b> .....	22
2.4.1. <b>Αγορά μετοχών από το διαδίκτυο</b> .....	22
2.4.2. <b>Διαγράμματα</b> .....	24
<b>ΚΕΦΑΛΑΙΟ 3 : ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ</b> .....	26
3.1. <b>Γλώσσες προγραμματισμού</b> .....	26
3.1.1. Java .....	26
3.1.2. Html .....	28
3.1.3. Visual Basic .....	29
3.2. <b>Ολοκληρωμένα περιβάλλοντα προγραμματισμού (IDE)</b> .....	31
3.2.1. NetBeans .....	31
3.2.2. Visual studio .....	32
3.2.3. Eclipse .....	33
3.3. <b>Εργαλεία λογισμικού</b> .....	34
3.3.1. Blender .....	34

3.3.2.	Multimedia builder .....	35
3.3.3.	GIMP.....	36
3.3.4.	Scilab.....	37
3.3.5.	Drupal .....	38
<b>ΚΕΦΑΛΑΙΟ 4: ΤΟ ΛΟΓΙΣΜΙΚΟ JEDUML (JAVA EDUCATIONAL UML).....</b>		<b>39</b>
4.1.	<b>Σκοπός</b> .....	39
4.2.	<b>Τεχνικά χαρακτηριστικά</b> .....	39
4.3.	<b>Δομή</b> .....	40
4.4.	<b>Τεκμηρίωση</b> .....	40
4.5.	<b>Βιβλιοθήκες</b> .....	41
4.6.	<b>Πηγαίος κώδικας</b> .....	43
4.6.1.	<b>Η γραφική διεπαφή</b> .....	44
4.6.2.	<b>Ο σχεδιαστής διαγραμμάτων</b> .....	46
4.7.	<b>Λειτουργία</b> .....	46
<b>Συμπεράσματα - Περίληψη</b> .....		<b>59</b>
<b>Επίλογος</b> .....		<b>59</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ</b> .....		<b>60</b>

## Περίληψη

Η ανάπτυξη και εξάπλωση των τεχνολογιών πληροφορικής και επικοινωνιών (ΤΠΕ) δεν θα μπορούσε να αφήσει ανεπηρέαστο και τον χώρο της εκπαίδευσης κάθε είδους και βαθμίδας. Έτσι δημιουργήθηκε το εκπαιδευτικό λογισμικό (educational software) και η εκπαίδευση με χρήση υπολογιστή (computer driven education). Θέμα της πτυχιακής εργασίας είναι η ανάπτυξη ενός λογισμικού εκπαιδευτικού σκοπού, που θα αφορά της εκμάθησης της Unified Modelling Language (UML).

Η εργασία ακολουθεί της εξής δόμηση:

- § Το 1<sup>ο</sup> κεφάλαιο παρέχει μια εισαγωγή στο επιστημονικό πεδίο του εκπαιδευτικού λογισμικού. Συγκεκριμένα, γίνεται ορισμός του πεδίου, μελετάται πώς προέκυψε η ανάγκη για εκπαιδευτικό λογισμικό, την εξάπλωση, τις τάσεις που επικρατούν, παιδαγωγικά – τεχνικά ζητήματα. Επίσης, παρουσιάζονται παραδείγματα σχετικών λογισμικών.
- § Το 2<sup>ο</sup> κεφάλαιο πραγματοποιεί μια εισαγωγή στην UML. Τί είναι, πώς προέκυψε, που χρησιμεύει. Παρουσιάζονται βασικές έννοιες και παραδείγματα χρήσης της.
- § Το 3<sup>ο</sup> κεφάλαιο ασχολείται με τεχνολογίες και εργαλεία για την παραγωγή εκπαιδευτικού λογισμικού. Εδώ θα γίνει μια σύντομη παρουσίαση σε τεχνολογίες και εργαλεία που χρησιμοποιούνται για την δημιουργία τέτοιου λογισμικού.
- § Το 4<sup>ο</sup> κεφάλαιο αναφέρεται στην υλοποίηση του λογισμικού JEDUML. Τι έχει υλοποιηθεί, πώς υλοποιήθηκε, πώς λειτουργεί.

## Summary

The rapid development and expansion of information technology (IT) couldn't let unaffected the scientific field of education. Educational software and computer driven education was developed in order to serve the needs that were emerged to the educational sector. The subject of this assignment is the development of an educational software which will concern the learning of Unified Modelling Language (UML).

The assignment has the following structure:

- § The first chapter makes an introduction to the scientific field of educational software. More specifically, we define the educational software scientific field, we study the factors which made the need of educational software to emerge, the expansion, the trends and the technical issues which emerged.
- § The second chapter makes an introduction to UML. What is UML, which needs serves, which are it's basic components. We present basic information and cases studies.
- § The third chapter deals with technologies and tools which can be used in educational software development.
- § The forth chapter refers to our implementation. It presents what was made, how it was implemented and how it works.

## Εισαγωγή

Τα τελευταία χρόνια παρατηρήθηκε ταυτόχρονα με την ανάπτυξη και εξάπλωση των τεχνολογιών πληροφορικής και η ανάπτυξη προϊόντων λογισμικού εκπαιδευτικού χαρακτήρα. Ωστόσο, η ανάπτυξη εκπαιδευτικού υλικού δεν οφείλεται αποκλειστικά στην ανάπτυξη που υπήρξε στον κλάδο της πληροφορικής αλλά προέκυψε ως επιτακτική ανάγκη αναδυόμενη κατευθείαν από τον κλάδο της εκπαίδευσης. Οι νέες συνθήκες που διαμορφώθηκαν στον τομέα της εκπαίδευσης και η ανάγκη για «δια βίου μάθηση» συντέλεσαν στην ανάπτυξη προγραμμάτων εκπαιδευτικού χαρακτήρα τα οποία θα ανταποκρίνονταν σε κάθε βαθμίδα και είδος εκπαίδευσης.

Ανάλογα με τη μορφή της χρήσης και τον εκπαιδευτικό στόχο, το εκπαιδευτικό λογισμικό μπορεί να είναι λογισμικό εξάσκησης-εγκύμνασης, εκπαίδευσης-φροντιστηρίου, λύσης προβλημάτων, προσομοιώσεων, εκπαιδευτικών παιχνιδιών, μοντελοποίησης. Το εκπαιδευτικό λογισμικό συμβάλει στη φιλικότερη, ελκυστικότερη, πλουσιότερη και πολύπλευρη παρουσίαση της ύλης, τη βιωματική προσέγγιση της γνώσης, την ενεργοποίηση του μαθητή μέσα από δημιουργικές δραστηριότητες, πειραματισμό και διερεύνηση, τη συμπύκνωση πολλών μακροσκελών κειμένων σε οπτικοακουστικά μηνύματα με μεγάλη περιεκτικότητα πληροφορίας, τη μείωση του χρόνου που αφιερώνει ο μαθητής και του κόπου που καταβάλλει για την αφομοίωση της ύλης-περιεχομένου και την προώθηση της συνεργατικής αλλά και της εξατομικευμένης μάθησης (οι μαθητές στο πλαίσιο κοινών δραστηριοτήτων μαθαίνουν να συνεργάζονται αλλά και ο κάθε μαθητής ξεχωριστά μπορεί να ακολουθήσει τους δικούς του ρυθμούς μάθησης).

Στα πλαίσια της παρούσας εργασίας γίνεται αναφορά στο επιστημονικό πεδίο του εκπαιδευτικού λογισμικού, την ιστορική του εξέλιξη, τα πλεονεκτήματα και την επίδρασή του στον τομέα της μάθησης. Επίσης, γίνεται εκτενής αναφορά στην UML (Unified Modeling Language), η οποία είναι και το αντικείμενο του εκπαιδευτικού λογισμικού που αναπτύχθηκε. Επιπλέον, γίνεται αναφορά στις τεχνολογίες που μπορούν να χρησιμοποιηθούν για την ανάπτυξη εκπαιδευτικού λογισμικού και λεπτομερής επεξήγηση και παρουσίαση του λογισμικού που αναπτύχθηκε (JEDUML) για τον σκοπό της εργασίας.

## ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ ΣΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ

Σε αυτό το κεφάλαιο θα πραγματοποιήσουμε μια εισαγωγή στο επιστημονικό πεδίο του εκπαιδευτικού λογισμικού. Συγκεκριμένα, θα ορίσουμε το πεδίο, θα μελετήσουμε πώς προέκυψε η ανάγκη για εκπαιδευτικό λογισμικό, την εξάπλωσή του, τάσεις που επικρατούν και παιδαγωγικά – τεχνικά ζητήματα. Επίσης, παρουσιάζουμε παραδείγματα σχετικών λογισμικών.

### 1.1. Ορισμός

Το λογισμικό αποτελεί το ένα από τα δύο βασικά χαρακτηριστικά για τη λειτουργία του ηλεκτρονικού υπολογιστή ( το άλλο είναι το υλικό ) και διακρίνεται σε δύο μεγάλες κατηγορίες: στο λογισμικό συστήματος και στο λογισμικό εφαρμογών. Εκπαιδευτικό λογισμικό [1] είναι λογισμικό που ανήκει στο λογισμικό εφαρμογών και με την αυστηρή έννοια του όρου θεωρείται το λογισμικό που εμπεριέχει διδακτικούς στόχους, ολοκληρωμένα σενάρια, αλληγορίες με παιδαγωγική σημασία και επιφέρει συγκεκριμένα διδακτικά και μαθησιακά αποτελέσματα. Φυσικά, ο όρος «εκπαιδευτικό λογισμικό» πρέπει να συμπεριλαμβάνει και πακέτα εφαρμογών επιμορφωτικού, εγκυκλοπαιδικού και ψυχαγωγικού τύπου. Λογισμικά, τα οποία μπορούν να χρησιμοποιηθούν για εκπαιδευτικούς σκοπούς, μπορούν να χαρακτηρισθούν διάφοροι τύποι πακέτων, όπως τα παρακάτω:

- Γλώσσες προγραμματισμού
- Πακέτα εφαρμογών
- Προσομοιώσεις
- Παιχνίδια
- Διαδίκτυο
- Εκπαιδευτικά συστήματα εικονικής πραγματικότητας
- Πακέτα εξάσκησης και πρακτικής
- Εκπαιδευτικές εφαρμογές πολυμέσων

Ανάλογα με τη μορφή της χρήσης και τον εκπαιδευτικό στόχο, υπάρχουν έξι βασικοί τύποι εκπαιδευτικού λογισμικού:

- Εξάσκησης-εκγύμνασης
- Εκπαίδευσης-φροντιστηρίου
- Λύσης προβλημάτων
- Προσομοιώσεων
- Εκπαιδευτικών παιχνιδιών
- Μοντελοποίησης

### 1.2. Ιστορικά στοιχεία

Η χρήση της ψηφιακής τεχνολογίας στην εκπαίδευση ξεκίνησε στις δυτικές Ευρωπαϊκές χώρες από τη δεκαετία του '60, ενώ λίγο αργότερα ακολούθησαν και οι χώρες της Ανατολικής Ευρώπης. Στη χώρα μας, όμως, μόλις τη δεκαετία του '90 εμφανίστηκαν οι πρώτοι ηλεκτρονικοί υπολογιστές στα σχολεία, ενώ την ίδια, περίπου, περίοδο οι οικιακοί ηλεκτρονικοί υπολογιστές έκαναν την εμφάνισή τους, σποραδικά στην αρχή, αλλά με αυξανόμενους ρυθμούς. Έτσι, στα τέλη της δεκαετίας του '90 οι κάτοχοι ηλεκτρονικού υπολογιστή μπορούσαν να βρουν αρκετούς τίτλους εκπαιδευτικού λογισμικού στα ράφια των καταστημάτων.

Μπορούμε να χωρίσουμε την ιστορία του εκπαιδευτικού λογισμικού σε τρεις περιόδους. Τα CD-ROM της δεκαετίας του '90 χαρακτηρίζουν την πρώτη εποχή του εκπαιδευτικού λογισμικού. Ακολούθησε η δεύτερη εποχή, με λογισμικό που συνεργάζεται με κατάλληλες συσκευές και είναι ανοιχτό ως προς το περιεχόμενο. Στην τρίτη εποχή, που ζούμε τώρα, το λογισμικό προσφέρεται



περισσότερο στη μορφή της διαδικτυακής υπηρεσίας και έχει περισσότερα επικοινωνιακά χαρακτηριστικά.

Στην πρώτη περίοδο του εκπαιδευτικού λογισμικού (δεκαετία '90) τα πιο πολλά λογισμικά ήταν κλειστά ως προς το περιεχόμενο. Αυτό σημαίνει ότι κάθε τίτλος λογισμικού έφερε συγκεκριμένο περιεχόμενο, όπως ακριβώς ένας τίτλος βιβλίου περιέχει ένα συγκεκριμένο κείμενο. Έτσι, για παράδειγμα, μπορούσε κανείς να βρει σε μορφή CD-ROM κάποιες εγκυκλοπαίδειες ή CD με γρίφους και μαθηματικά παιχνίδια. Ορισμένα παιχνίδια, εξ' άλλου, σχεδιάζονταν έτσι ώστε να έχουν εκπαιδευτική αξία. Οι περισσότεροι από τους τίτλους που μοίρασε το Υπουργείο Παιδείας στα σχολεία της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης την περασμένη χρονιά ανήκουν σε αυτή την κατηγορία. Συνήθως οι τίτλοι αυτοί παρουσιάζουν κάποιον ήρωα που καθοδηγείται από το παιδί μέσα από κάποιες περιπέτειες - εκπαιδευτικά παιχνίδια. Σε άλλες περιπτώσεις, το λογισμικό περιέχει ταινίες και εποπτικό υλικό ή εγκυκλοπαιδικές πληροφορίες στις οποίες ο μαθητής μπορεί να ανατρέξει, ή μπορούν να προβληθούν και να χρησιμοποιηθούν ομαδικά από όλη την τάξη.

Τα μειονεκτήματα των εκπαιδευτικών CD-ROM σε συνδυασμό με την πρόοδο της τεχνολογίας των περιφερειακών και των φορητών συσκευών οδήγησε στη δεύτερη εποχή του εκπαιδευτικού λογισμικού, που χρονικά τοποθετείται από το τέλος της δεκαετίας του '90 έως και σήμερα. Σε αυτή την περίοδο άρχισαν να χρησιμοποιούνται συστηματικά στην εκπαίδευση συσκευές όπως ο ψηφιακός προβολέας ή ο διαδραστικός πίνακας, αντικαθιστώντας τους προβολείς διαφανειών - slides και τους παραδοσιακούς μαυροπίνακες. Αυτό συνέβη πρώτα στις προηγμένες χώρες, και σταδιακά φτάνει και στη χώρα μας.

Οι τίτλοι που εμφανίστηκαν στις αρχές της δεκαετίας του '00 είναι, συνήθως, ανοιχτοί ως προς το περιεχόμενό τους και χρησιμοποιούνται ευρύτατα ακόμα και σήμερα. Το λογισμικό απεγκλωβίστηκε από τα δεσμά της πεπερασμένης διδακτέας ύλης, και οι νέοι τίτλοι σχεδιάστηκαν έτσι ώστε είτε να αφορούν σε ένα ολόκληρο γνωστικό αντικείμενο, είτε να λειτουργούν ως εργαλεία γενικής εκπαιδευτικής χρήσης. Αντί το λογισμικό να λειτουργεί σαν βιβλίο με ήχο και εικόνα, χρησιμοποιείται πλέον για την κατασκευή εκπαιδευτικού υλικού, είτε για την διευκόλυνση της χρήσης συγκεκριμένων συσκευών που εντάσσουν οι εκπαιδευτικοί στη δουλειά τους (π.χ. διαδραστικός πίνακας, προβολέας).

Στην τρίτη περίοδο, που μπορούμε να θεωρήσουμε ότι ξεκίνησε περίπου πριν από πέντε χρόνια, δεν έχουμε μόνο τίτλους λογισμικού, αλλά υπηρεσίες και εφαρμογές. Το λογισμικό είναι κάτι που το τοποθετείς στον υπολογιστή σου και εκτελεί κάποιες λειτουργίες. Με τις διαδικτυακές υπηρεσίες, όμως, οι λειτουργίες αυτές είναι διαθέσιμες απ' ευθείας, μέσω διαδικτύου, χωρίς να χρειαστεί καμία ιδιαίτερη εγκατάσταση σε κάποιον υπολογιστή. Αυτές οι υπηρεσίες είναι διαθέσιμες ταυτόχρονα σε όλους, κοστίζουν πολύ λιγότερο και, αν σχεδιαστούν σωστά, είναι αρκετά εύχρηστες.

### 1.3. Πλεονεκτήματα

Ο βασικός σκοπός ύπαρξης του εκπαιδευτικού λογισμικού δεν είναι η αντικατάσταση του έντυπου βιβλίου ή εγχειριδίου με ένα ηλεκτρονικό που θα διαβάζεται από την οθόνη του υπολογιστή, αλλά η συμπλήρωση και ο εμπλουτισμός του διδακτικού υλικού. Γενικότερα συμβάλει:

- Στη φιλικότερη, ελκυστικότερη, πλουσιότερη και πολύπλευρη παρουσίαση της ύλης
- Στη βιωματική προσέγγιση της γνώσης
- Στην ενεργοποίηση του μαθητή μέσα από δημιουργικές δραστηριότητες, πειραματισμό και διερεύνηση
- Στη συμπύκνωση πολλών μακροσκελών κειμένων σε οπτικοακουστικά μηνύματα με μεγάλη περιεκτικότητα πληροφορίας
- Στη μείωση του χρόνου που αφιερώνει ο μαθητής και του κόπου που καταβάλλει για την αφομοίωση της ύλης-περιεχομένου
- Στην προώθηση της συνεργατικής αλλά και της εξατομικευμένης μάθησης (οι μαθητές στο πλαίσιο κοινών δραστηριοτήτων μαθαίνουν να

συνεργάζονται αλλά και ο κάθε μαθητής ξεχωριστά μπορεί να ακολουθήσει τους δικούς του ρυθμούς μάθησης).

#### 1.4. Η επίδραση της εκπαίδευσης

Την εκκίνηση για τη χρησιμοποίηση της τεχνολογίας γενικότερα στην εκπαίδευση, έδωσε η εργασία και οι μελέτες ψυχο-παιδαγωγών, με στόχο την εύρεση των αποτελεσματικότερων μεθόδων μάθησης. Τα προϊόντα αυτών των μελετών ήταν οι λεγόμενες «θεωρίες μάθησης», για τις οποίες έγινε προσπάθεια να υλοποιηθούν πολλές φορές με την χρησιμοποίηση της τεχνολογίας της εποχής.

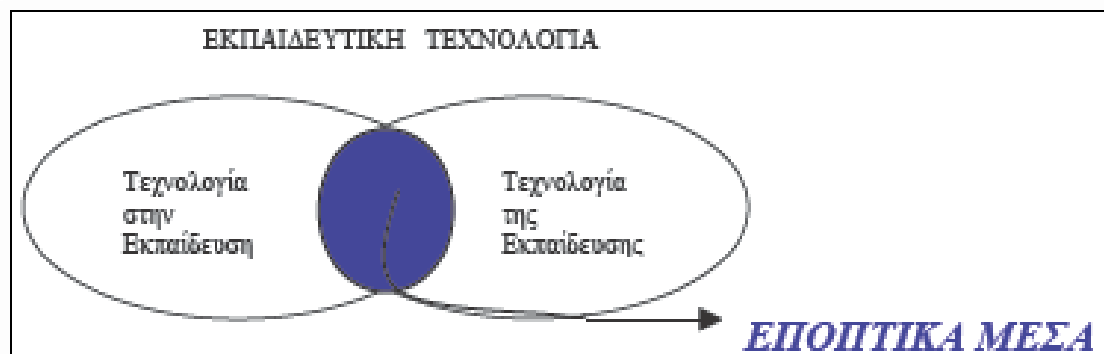
##### 1.4.1. Θεωρίες μάθησης

Στην ιστορική εξέλιξη τους μέχρι σήμερα, υπάρχουν σε πολύ γενικό πλαίσιο τρεις σχολές θεωριών μάθησης. Ο συμπεριφορισμός που προσδιορίζει σαν τον κυριότερο παράγοντα μάθησης την επίδραση του εξωτερικού περιβάλλοντος, η οποία διαμορφώνει την συμπεριφορά του μαθητή, ο γνωστικισμός που, αντίθετα από τον συμπεριφορισμό, προσδιορίζει σαν κύριο παράγοντα μάθησης αυτό που στον ίδιο το μαθητή εσωτερικά συμβαίνει, αναλυτικότερα ότι η μάθηση είναι η αλλαγή-εξέλιξη μιας ήδη υπάρχουσας γνώσης και ο οικοδομητισμός που περιγράφει την μάθηση σαν μια αλλαγή του νοήματος των πραγμάτων η οποία επιτυγχάνεται μέσα από εμπειρίες, στις οποίες η γνώση οικοδομείται σε μια σύνθετη συνέργια της προηγούμενης γνώσης του μαθητή, του κοινωνικού περιεχομένου της και του προβλήματος που παρουσιάζεται για επίλυση. Εμφανίζονται δε και ρεύματα διδακτικής πρακτικής τα οποία μπορούν να διαχωριστούν σε δύο μεγάλες κατηγορίες την δασκαλο-κεντρική και τη μαθητοκεντρική-θεώρηση τα οποία εξυπηρετούν την μία ή την άλλη ή και την συνεργασία των σχολών και εκφράζονται με επιμέρους διδακτικές προσεγγίσεις όπως η διάλεξη, ο διάλογος με τους μαθητές, η επίδειξη, η εξατομικευμένη μάθηση δηλαδή η μάθηση με γνωστικό υλικό έτσι διαμορφωμένο που να ταιριάζει στις γνωστικές ανάγκες του κάθε μαθητή, η ανακαλυπτική μάθηση ή προσπάθεια δηλαδή ο μαθητής, ενεργώντας σε κατάλληλα περιβάλλοντα, να ανακαλύψει μόνος του χωρίς να του δοθεί έτοιμο ένα κομμάτι γνώσης, η ομαδοσυνεργατική μάθηση δηλαδή η συνεργασία των μαθητών σε ομάδα απέναντι σε ένα αντικείμενο μάθησης κλπ.

##### 1.4.2. Εκπαιδευτική τεχνολογία

Ο ιδιαίτερος τομέας της εκπαίδευσης-δημιουργείται κάπου στη δεκαετία του 50- ο οποίος ασχολείται με την δυνατότητα εισαγωγής κάθε μορφής τεχνολογίας στην καθημερινή πρακτική της, λέγεται «Εκπαιδευτική Τεχνολογία» και πρέπει να θεωρείται σαν ένα είδος «ομπρέλλα», η οποία περιλαμβάνει την εφαρμογή και αξιοποίηση γνώσεων τόσο από το χώρο των θετικών επιστημών (φυσική, μαθηματικά, χημεία, τεχνολογία) όσο και από το χώρο των θεωρητικών επιστημών (ψυχολογία, κοινωνιολογία, φιλοσοφία, γλωσσολογία, σημειολογία...), για την δημιουργία ενός συστήματος τεχνολογικών διδακτικών παρεμβάσεων στη διδασκαλία.

Φυσικά η είσοδος οποιουδήποτε είδους τεχνολογίας στην διδασκαλία, περνά μέσα από τις θεωρίες μάθησης. Σύμφωνα με τον Σιμάτο (Σιμάτος Α. , 1994), ο ακριβής προσδιορισμός της ουσίας της εκπαιδευτικής τεχνολογίας μπορεί να γίνει κατανοητός βάσει του παρακάτω διαγράμματος.



**Εικόνα 1. Τα μέρη της Εκπαιδευτικής Τεχνολογίας**

Ο όρος «τεχνολογία στην Εκπαίδευση» χαρακτηρίζει την μεταφορά των οπτικοακουστικών μέσων στην διδασκαλία και χαρακτηρίζει την πρώτη περίοδο της ανάπτυξης της εκπαιδευτικής τεχνολογίας, με δάνεια από τις μέχρι τότε επικρατούσες θεωρίες μάθησης, περισσότερο δε του συμπεριφορισμού. Ο εγκέφαλος του μαθητή σε αυτή την περίοδο αντιμετωπίζεται σαν άδειο δοχείο και αναζητείται το καταλληλότερο εργαλείο ώστε να επιταχυνθεί κατά το μέγιστο δυνατό η μεταφορά της γνώσης από το περιβάλλον σε αυτό. Ερευνήθηκε η δύναμη της εικόνας στην μεταφορά της γνώσης σε σχέση με τον γραπτό ή προφορικό λόγο, τα κίνητρα μάθησης που είχε ο μαθητής με το καινούργιο τρόπο παρουσίασης των πληροφοριών σε αυτόν και η επίδραση της ενίσχυσης στην προσδοκώμενη συμπεριφορά του μαθητή δια μέσου όλων αυτών των μέσων. Ο δεύτερος όρος «τεχνολογία της εκπαίδευσης» χαρακτηρίζει τη νεώτερη αντίληψη της μάθησης που έχει σχέση με τις γνωστικές θεωρίες και τον οικοδομισμό, εστιάζοντας εκτός από τις επιδράσεις του περιβάλλοντος με εικόνα, ήχο ή και συνδυασμό και των δύο και στις εσωτερικές διεργασίες που γίνονται στον κάθε μαθητή για την λειτουργία της μάθησης. Η επικάλυψη με το όνομα «εποπτικά μέσα» που φαίνεται στο διάγραμμα 3 θέλει να τονίσει απλώς την διαφορετική αντιμετώπιση στην εκπαιδευτική διαδικασία των ίδιων κατά τα άλλα τεχνολογικών μέσων, η οποία και φανερώνει την με την πάροδο των ετών μετάβαση της ίδιας της εκπαιδευτικής τεχνολογίας από τις αντιλήψεις του συμπεριφορισμού σε αυτές των γνωστικών θεωριών μάθησης και τα αποτελέσματα των ερευνών της Γνωστικής Ψυχολογίας.

Επομένως το Εκπαιδευτικό Λογισμικό και η ηλεκτρονική μάθηση ως τεχνολογικά προϊόντα ανήκουν στην αρμοδιότητα της Εκπαιδευτικής Τεχνολογίας, η οποία έχει προχωρήσει πλέον σε δυναμικά δάνεια γνώσεων και από την περιοχή της Πληροφορικής και των Επικοινωνιών.

#### 1.4.3. **Νέες μορφές εκπαίδευσης**

Πέραν τούτων, υπάρχουν στην εποχή μας κοινωνικές τάσεις οι οποίες πιέζουν για τη δημιουργία διαφορετικών τύπων μάθησης – εκπαίδευσης – κατάρτισης από τους τυπικούς, δηλαδή το οργανωμένο σχολείο το οποίο στοχεύει σε συγκεκριμένες ηλικιακές ομάδες. Έτσι κάτω από αυτές τις κοινωνικές επιταγές, αναφερόμαστε στην δια βίου εκπαίδευση η οποία προσπαθεί να καλύψει τα γνωστικά κενά ανθρώπων που έχουν από καιρό αφήσει το ρόλο του μαθητή, στην εξ' αποστάσεως εκπαίδευση που προσπαθεί να εκπαιδεύσει ανθρώπους σύμφωνα με τον χρόνο που διαθέτουν και στον χώρο που επιλέγουν, στην εκπαίδευση ενηλίκων η οποία προετοιμάζει δασκάλους και διδακτικές πρακτικές για την δια βίου εκπαίδευση, στην διαρκή εκπαίδευση – κατάρτιση του προσωπικού μεγάλων επιχειρήσεων ή οργανισμών ώστε να μπορούν οι εργαζόμενοι να συμπεριλάβουν στην καθημερινή τους εργασία τα ομολογουμένως συνεχώς αυξανόμενα επιτεύγματα της τεχνολογίας κλπ.

#### 1.4.4. **Λειτουργίες της εκπαίδευσης – Οικονομική διάσταση**

Οι δύο βασικές λειτουργίες της Εκπαίδευσης είναι: (Ράπτης Α., Ράπτη Α., 1999).

1. Η *ατομική λειτουργία* δηλαδή η ανάπτυξη της προσωπικότητας του ατόμου
2. Η *κοινωνική λειτουργία* δηλαδή η ανάπτυξη της κουλτούρας μιας κοινωνίας, μέσω της οποίας λειτουργεί ή κοινωνική αναπαραγωγή και ο μετασχηματισμός.

Η ίδια η εκπαίδευση επηρεάζεται από ιστορικούς, κοινωνικούς και πολιτιστικούς παράγοντες. Για να είναι σε θέση όμως μια οργανωμένη κοινωνία να προσφέρει στα μέλη της τις παραπάνω λειτουργίες της εκπαίδευσης, θα πρέπει να διαθέτει τους οικονομικούς πόρους, οι οποίοι θα στηρίξουν την προσφορά αυτή. Αυτό γίνεται περισσότερο αντιληπτό, στην εκπαίδευση σε θετικές ή τεχνολογικές επιστήμες, οι οποίες απαιτούν, εκτός από την ύπαρξη των κατάλληλων εκπαιδευτικών και κατάλληλη υλικοτεχνική υποδομή. Για παράδειγμα εάν θεωρήσουμε τις νέες τεχνολογίες ως κοινωνικό εργαλείο αναπαραγωγής και αναλογιστούμε τη ραγδαία ανάπτυξη που παρουσιάζουν στην εποχή μας, μπορούμε ευκολότερα να μιλήσουμε για την Εκπαίδευση γύρω από αυτές και με οικονομικά μεγέθη. Να μιλήσουμε για το «κόστος της Εκπαίδευσης», για το Προϊόν της Εκπαίδευσης, για τους πελάτες της Εκπαίδευσης και τα διαμορφούμενα target groups σε αυτούς. Αξίζει να σημειωθεί ότι από τους μεγαλύτερους πελάτες στις σύγχρονες μορφές εκπαίδευσης διεθνώς, είναι οι στρατιωτικοί οργανισμοί, με στόχο την εκπαίδευση του προσωπικού τους, οι οποίοι βέβαια ενδιαφέρονται για το κόστος που καλούνται να αντιμετωπίσουν και αναζητούν τρόπους ελαχιστοποίησής του. Γενικότερα αυτή η οικονομική διάσταση της Εκπαίδευσης, με στόχο προφανώς το λιγότερο δυνατό κόστος για το καλύτερο δυνατό αποτέλεσμα, συμβάλλει εν μέρει στη χρήση της ηλεκτρονικής μάθησης όσο και στη προσπάθεια δημιουργία διεθνών προτύπων σε αυτή.

#### 1.4.5. Εκπαιδευτική έρευνα – Εξατομικευμένη μάθηση

Ο τομέας της εκπαίδευσης ο οποίος ασχολείται με την έρευνα γύρω από τη υλοποίηση των θεωριών μάθησης λέγεται «Εκπαιδευτική Έρευνα». Μπορούμε να πούμε ότι η τεχνολογία των Επικοινωνιών και Πληροφορικής έχει σήμερα μια διττή σχέση με την Εκπαιδευτική Έρευνα. Από τη μία χρησιμοποιείται σαν εργαλείο για την διαχείριση των αποτελεσμάτων των ερευνών της και από την άλλη η ηλεκτρονική μάθηση αξιολογείται μέσα από αυτή.

Ένα παράδειγμα εκπαιδευτικής έρευνας, είναι οι μελέτες οι οποίες συνέκριναν την μάθηση στην τάξη του τυπικού σχολείου, όπου ο δάσκαλος αναφέρεται με τον ίδιο τρόπο για το γνωστικό αντικείμενο σε όλους τους μαθητές, με την μάθηση σε εκείνη την περίπτωση όπου ο δάσκαλος αναφέρεται στο γνωστικό αντικείμενο σύμφωνα με τις ανάγκες του κάθε μαθητή (εξ' ατομικευμένη μάθηση). Αυτό έγινε μέσα στα πλαίσια ελέγχου της αξιοπιστίας των γνωστικών θεωριών, οι οποίες θεωρούν την μάθηση μια προσωπική περιπέτεια για κάθε μαθητή και τους τρόπους που αυτή επιτυγχάνεται άμεσα εξαρτώμενους από τον ίδιο τον μαθητή. Μερικά από τα αποτελέσματα αυτών των ερευνών είναι ότι:

- ταχύτητα προόδου του μαθητή, ποικίλει με παράγοντα από τρία έως επτά – ακόμα και σε τάξεις με προσεκτικά επιλεγμένους μαθητές. (Gettinger, M. , 1984).
- μαθητής στην τυπική αίθουσα διδασκαλίας κατά μέσο όρο κάνει περίπου 0.1 ερωτήσεις την ώρα. (Graesser, A.C.,& Person, N.K. , 1994).
- την εξ' ατομικευμένη μάθηση, παρέχεται αυξημένη δυνατότητα επικοινωνίας μεταξύ δασκάλου και μαθητή με αποτέλεσμα ο μαθητής να έχει την ευχέρεια για περίπου 120 ερωτήσεις ή απαντήσεις την ώρα. (Graesser, A.C.,& Person, N.K., 1994).
- επιτυχία των μαθητών της εξ' ατομικευμένης μάθησης, μπορεί να υπερβεί αυτήν των μαθητών της τυπικής τάξης – μια βελτίωση η οποία μπορεί να θεωρηθεί ισοδύναμη με την αύξηση της απόδοσης του 98% των μαθητών έναντι του 50%. (Bloom, B.S., 1984).

Η εξ' ατομικευμένη μάθηση έχει μερικές φορές μεγάλη αποτελεσματικότητα. Δυστυχώς όμως η εφαρμογή της στην αίθουσα διδασκαλίας είναι πολύ δύσκολη. Στην ουσία μπορεί να επιτευχθεί στην περίπτωση στην οποία είναι διαθέσιμος ένας δάσκαλος για ένα μαθητή. Το τελευταίο αποκλείει φυσικά την μέθοδο από το να γίνει επίσημη εκπαιδευτική στρατηγική για το δημόσιο ή το

ιδιωτικό σύστημα εκπαίδευσης, αλλά ακόμα και για εκπαιδευτικά περιβάλλοντα επιχειρήσεων, διότι δεν είναι εφικτή αλλά και στην περίπτωση που θα ήταν, δεν είναι οικονομικά σύμφωρη.

Πιθανώς αυτή την προφανή αδυναμία χρησιμοποίησής της, να μπορεί να καλύψει η ηλεκτρονική μάθηση, με την ικανότητά της να παρέχει διδακτικό υλικό σύμφωρο με τις απαιτήσεις του ενδιαφερόμενου σε πραγματικό χρόνο αλλά και σε λογικό κόστος. Διάφορες εμπειρικές μελέτες στο θέμα αύξησαν το ενδιαφέρον των κρατικών αρμοδίων σε διεθνές επίπεδο για την χρησιμοποίηση της τεχνολογίας των Επικοινωνιών και Πληροφορικής. Αυτές οι μελέτες απέδειξαν ότι σε αντίθεση με την τυπική μορφή εκπαίδευσης στην αίθουσα διδασκαλίας, η τεχνολογία των Επικοινωνιών και της Πληροφορικής μπορεί να δώσει κατάλληλης μορφής εκπαιδευτικό περιεχόμενο, με κατάλληλη αλληλουχία, έτσι ώστε το περιεχόμενο αυτό να ταιριάζει πολύ καλύτερα στο τρόπο μάθησης, στα ενδιαφέροντα και τους στόχους του κάθε μαθητή ξεχωριστά. (Fletcher, J.D., 2003). Τα παραπάνω επίσης δηλώνουν, ότι η ηλεκτρονική μάθηση μπορεί να υποστηρίξει και τις νέες μορφές εξέλιξης αυτές της *εξ' αποστάσεως μάθησης, δια βίου μάθησης* ...κλπ χρησιμοποιώντας την τεχνολογία του Διαδικτύου.

#### 1.4.6. **Ο ηλεκτρονικός υπολογιστής στην εκπαίδευση**

Γενικότερα οι θιασώτες των νέων τεχνολογιών στο χώρο της εκπαίδευσης προσεγγίζουν την χρησιμοποίηση του υπολογιστή στην καθημερινότητα τους μέσα από πέντε διαφορετικές λειτουργίες: (Ράπτης Α., Ράπτη Α., 1999).

1. Τη διδασκαλία της πληροφορικής ως ξεχωριστού γνωστικού αντικειμένου με στόχο την προώθηση του τεχνικού αναλφαριθμητισμού και την απόκτηση των κοινωνικά απαραίτητων δεξιοτήτων
2. Τη χρήση του υπολογιστή ως πηγή πληροφόρησης, καθώς και μέσου διασυνδέσεων και επικοινωνίας
3. Την αξιοποίηση του υπολογιστή ως εποπτικού και επικοινωνιακού μέσου για την υποβοήθηση της διδασκαλίας.
4. Τη χρήση του υπολογιστή ως γνωστικού εργαλείου καθώς και πεδίου μελέτης για τη διερεύνηση γνωστικών δομών και μοντέλων
5. Τη χρήση του υπολογιστή για παίξιμο παιχνιδιών.

Αν όλες αυτές οι λειτουργίες θα μπορούσαν να ενεργήσουν σύμφωνα με τις ανάγκες τα ενδιαφέροντα και τον τρόπο μάθησης του κάθε μαθητή ξεχωριστά, η χρήση του υπολογιστή στην διδασκαλία θα μπορούσε από μερικούς να ειπωθεί όχι μόνο σαν εναλλακτική αλλά και σαν αναντικατάστατη.

Συνεπώς, εκείνο το σημείο, το οποίο από παιδαγωγικής άποψης φαίνεται να κυριαρχεί για την χρησιμοποίηση της ηλεκτρονικής μάθησης, είναι η υλοποίηση αυτής της δυνατότητας της εκπλήρωσης των αναγκών του κάθε μαθητή ξεχωριστά, χωρίς βέβαια να εμποδίζεται η διαμέσου αυτής προσέγγιση και συμμετοχή σε ομάδες συμμαθητών του για την από κοινού αντιμετώπιση «εμπειριών μάθησης». Και σε αυτή τη δυνατότητα θα πρέπει να εστιάσουν οι ασχολούμενοι μαζί της για να επιφέρουν μια σημαντική αλλαγή στην εκπαίδευση.

#### 1.5. **Εκπαιδευτικό λογισμικό – μελέτες περίπτωσης**

Σε αυτήν την ενότητα θα παρουσιάσουμε κάποια λογισμικά τα οποία είναι χαρακτηριστικά παραδείγματα και τα οποία αποτέλεσαν και οδηγό για εμάς κατά την εκπόνηση της εργασίας.

##### 1.5.1. Geometre's Sketchpad

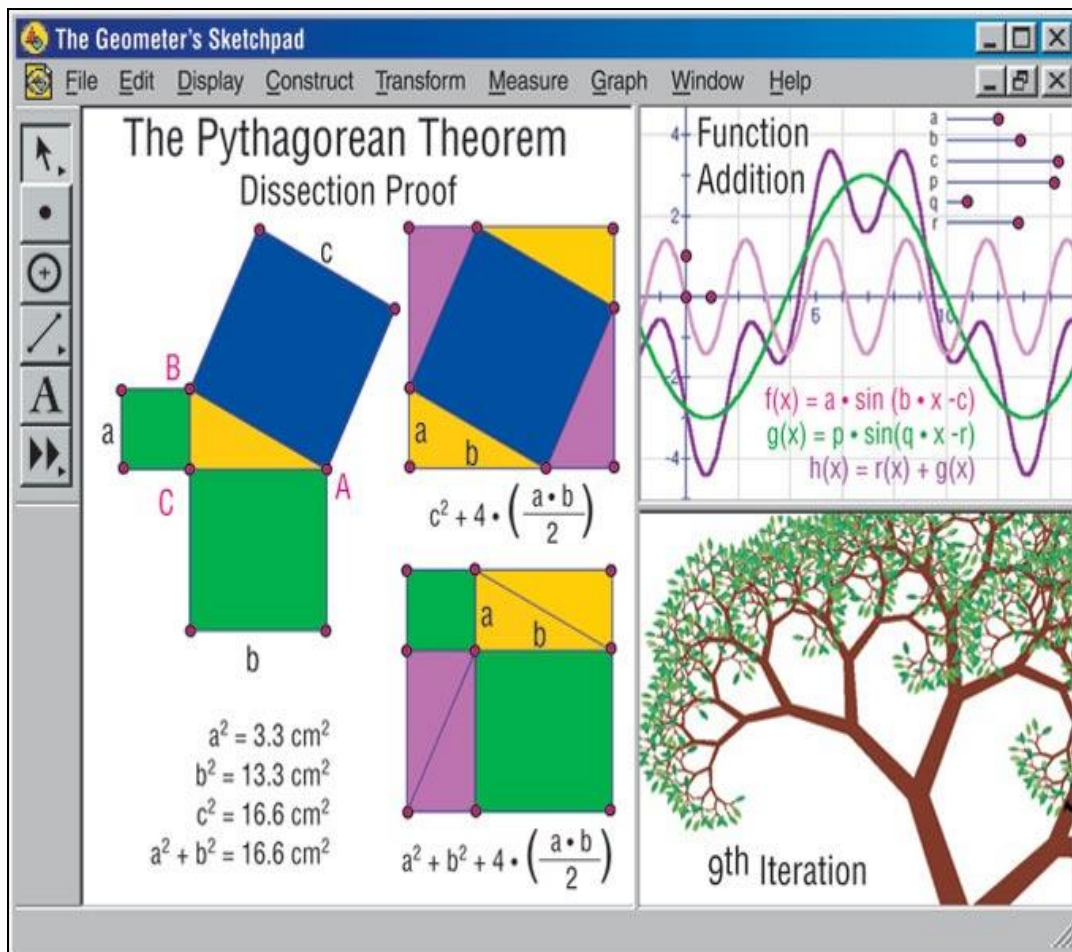
Το «The Geometer's Sketchpad» [7,8] είναι ένα ισχυρό εργαλείο για τη διδασκαλία της Γεωμετρίας, της Άλγεβρας και της Τριγωνομετρίας. Ο σχεδιασμός και η κατασκευή του στηρίχθηκαν σε πολύχρονες έρευνες στην περιοχή της διδακτικής των μαθηματικών. Είναι ένα διεθνώς δοκιμασμένο εργαλείο μάθησης για το οποίο υπάρχει πλούσια βιβλιογραφία και τεκμηρίωση.

Το «The Geometer's Sketchpad» είναι ιδανικό για την οργάνωση δραστηριοτήτων *διερευνητικής μάθησης* στο σχολικό εργαστήριο και στο σπίτι. Αξιοποιεί τις δυνατότητες των νέων τεχνολογιών λαμβάνοντας υπόψη τις νέες τάσεις για διερευνητική προσέγγιση στη σχεδίαση του λογισμικού (με πολλαπλές αναπαραστάσεις, άμεσο χειρισμό κ.τ.λ.). Με τις δυνατότητες που διαθέτει βοηθά στην κατανόηση με ολοκληρωμένο τρόπο εννοιών και διαδικασιών μέσα από την *επίλυση προβλημάτων* και τον *πειραματισμό*.

Το «The Geometer's Sketchpad» είναι ένα «ανοικτό» περιβάλλον διερευνητικής μάθησης. Οι δυνατότητές του είναι τόσο ευρείες που αν και αρχικά σχεδιασμένο για τις ανάγκες της γυμνασιακής εκπαίδευσης σήμερα συνιστάται από την Πέμπτη τάξη του Δημοτικού μέχρι τις τελευταίες τάξεις του Λυκείου. Οι δυνατότητες αυτές το μετέτρεψαν σε ένα εκπαιδευτικό εργαλείο με απεριόριστο αριθμό εφαρμογών. Αν και σχεδιάστηκε αρχικά για Γεωμετρία, σήμερα οι μαθητές μπορούν να το χρησιμοποιήσουν για να εξερευνήσουν την Άλγεβρα, την Τριγωνομετρία, την Τέχνη, την Επιστήμη και πολλά άλλα.

Η παιδαγωγική προσέγγιση του λογισμικού στηρίζεται στην άποψη ότι η μάθηση προϋποθέτει την ενεργητική συμμετοχή των μαθητών στο κοινωνικό πλαίσιο της τάξης, το οποίο αποτελείται πρωταρχικά από τον εκπαιδευτικό (μέσω των διδακτικών του παρεμβάσεων), το μαθητή και από την αλληλεπίδραση των μαθητών με τα διδακτικά εργαλεία (π.χ. βιβλία, υπολογιστικά περιβάλλοντα κ.α.). Αυτό προϋποθέτει ότι το περιβάλλον διδασκαλίας είναι πλούσιο και παρέχει δυνατότητες για απόκτηση ικανότητας για διερεύνηση, ανάλυση και σύνθεση, αντιμετώπιση και κατανόηση εννοιών μέσα από την οργάνωση νοητικών δομών και την ενεργή αλληλεπίδραση μαθητή, προσφερομένων εργαλείων και διδάσκοντα. Ακολουθώντας αυτή την προσέγγιση, το λογισμικό:

- Προσφέρει ένα θεματικό πλαίσιο το οποίο διευρύνει την φαντασία των ενεργητικά ενασχολούμενων με αυτό δημιουργώντας κίνητρο για μάθηση.
- Η δυνατότητα σχεδίασης και κατασκευής δι-διάστατων αντικειμένων προσφέρει την δυνατότητα ενεργητικής ενασχόλησης των μαθητών και εμπλουτίζει τις γνωστικές και μεταγνωστικές τους εικόνες.
- Αποτελείται από «εικόνες» οι οποίες παίζουν τον ρόλο «φυσικών μεταφορών» ή/και «οπτικών αναπαραστάσεων» μαθηματικών εννοιών με δυνατότητα διαβάθμισης της γνωστικής τους επεξεργασίας (π.χ. διαισθητική, πρακτική και φορμαλιστική επεξεργασία εννοιών).
- Η χρήση της γεωμετρίας γίνεται μέσω οπτικών και λεκτικών κωδίκων οι οποίοι με την κατάλληλη διδακτική παρέμβαση (π.χ. προτεινόμενη μέσα από ένα διδακτικό σενάριο, ή/και με την παρέμβαση του καθηγητή) βοηθούν στην εποικοδόμηση μαθηματικών εννοιών.



Εικόνα 2. Χρησιμοποιώντας το λογισμικό

Η διδακτική προσέγγιση σε ένα τέτοιο περιβάλλον προσφέρει και ενθαρρύνει την:

- Απόκτηση εμπειριών από την ενεργή ενασχόληση των μαθητών με το φυσικό και γεωμετρικό χώρο που παρέχεται (π.χ. κατασκευή, σχεδιασμός, παρατήρηση)
- Επικοινωνία αυτών των εμπειριών και των προσπαθειών τους για την προσέγγιση των γεωμετρικών εννοιών που επεξεργάζονται μέσα από κατάλληλες ασκήσεις και τεχνολογικά εργαλεία.
- Συνειδητή προσέγγιση των εμπειριών που αποκτούν οι μαθητές με την ενασχόλησή τους στο περιβάλλον της γεωμετρίας και τη χρήση των τεχνολογικών εργαλείων ως προς το γνωστικό αντικείμενο που εκάστοτε διδάσκεται
- Χρήση πολλαπλών αναπαραστάσεων της μαθηματικής έννοιας και προσέγγισής της με τεχνολογικά και παραδοσιακά εργαλεία (π.χ. κανόνας, διαβήτης, σχεδιαστικά υλικά).

Η σημαντικότερη δυνατότητα του «The Geometer's Sketchpad» είναι το "direct manipulation", η δυνατότητα δηλαδή της άμεσης διαχείρισης των μαθηματικών αντικειμένων και σχημάτων και την επεξεργασία των γεωμετρικών και εννοιών ολιστικά και από διαφορετικές οπτικές γωνίες. Ο καθηγητής / μαθητής, αφού δημιουργήσει ένα σχήμα μπορεί να το μεγεθύνει, να το μετακινήσει, να εξετάσει αν συμπίπτει με άλλο παρόμοιο, πράγμα που βέβαια δεν μπορεί να γίνει με τους παραδοσιακούς τρόπους διδασκαλίας.

Η δυνατότητα της κίνησης και της ταυτόχρονης παρακολούθησης της αλλαγής των διαφόρων στοιχείων και μεγεθών του σχήματος, δίνει τη δυνατότητα της χρήσης της «εικασίας» και του πειραματισμού στη διδακτική πράξη, κάτι που έχει μεγάλη ανάγκη η διδακτική των Μαθηματικών.

Το «The Geometer's Sketchpad» μπορεί να χρησιμοποιηθεί και σαν εργαλείο επίλυσης προβλημάτων, όπως π.χ. στην εύρεση γεωμετρικών τόπων, αφού παρέχει τη δυνατότητα να διαγράφεται στην οθόνη η γραμμή που σχηματίζεται από τις διαδοχικές θέσεις ενός επιλεγμένου σημείου κατά την κίνηση των παραμετρικών στοιχείων του σχήματος.

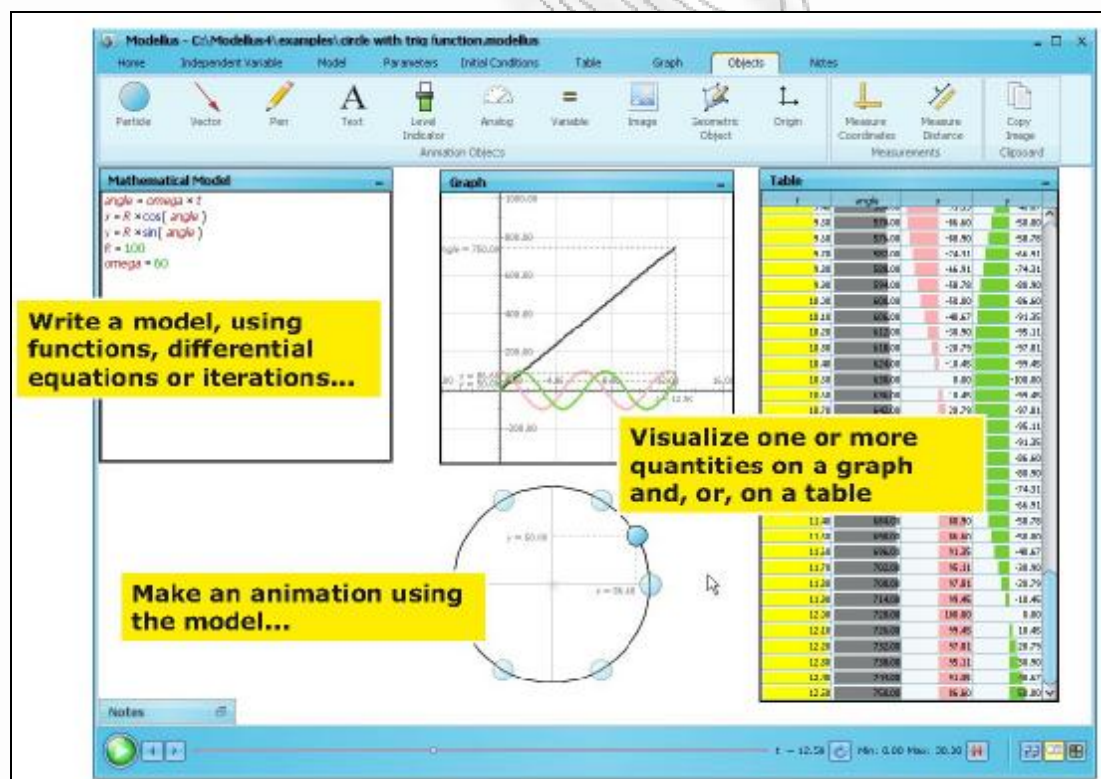
### 1.5.2. Modellus

Το λογισμικό Modellus [7,9], το οποίο σχεδιάστηκε από μία ομάδα επιστημόνων με την καθοδήγηση του καθηγητή Vítor Duarte Teodoro από το Πανεπιστήμιο Lisbon της Πορτογαλίας, είναι ένα ισχυρό εργαλείο, ιδιαίτερα χρήσιμο για τη διδασκαλία των θετικών επιστημών. Κυρίως μπορεί να χρησιμοποιηθεί για την υποστήριξη των Μαθηματικών, της Φυσικής, της Χημείας, των Οικονομικών και δευτερευόντως της Βιολογίας.

Ανήκει στην κατηγορία του ανοικτού τύπου περιβάλλοντος-εργαλείο για modeling, πειραματισμό και simulation, απαραίτητο για την ανάπτυξη μαθηματικών μοντέλων και την επεξεργασία τους μέσα από γραφικές παραστάσεις, πίνακες και animations.

Στον πυρήνα του προγράμματος υπάρχει μια περιοχή εργασίας (παράθυρο) στην οποία ο μαθητής μπορεί να γράψει το μαθηματικό μοντέλο με μορφή εξισώσεων ή ορισμών μεγεθών. Στη συνέχεια, το σύστημα αναλαμβάνει να πραγματοποιήσει την αναπαράσταση της εξέλιξης του φαινομένου που υπακούει στο μαθηματικό μοντέλο.

Το Modellus αξιοποιεί πολλές προηγούμενες προσπάθειες που έγιναν στην κατεύθυνση της δημιουργίας ενός λογισμικού κατάλληλου για μοντελοποιήσεις σε ποικίλες γνωστικές περιοχές.



Εικόνα 3. Χρησιμοποιώντας το λογισμικό

Είναι ένα λογισμικό πολύ εύκολο στην εκμάθηση και τη χρήση του και άμεσα προσαρμόσιμο στις ανάγκες της Δευτεροβάθμιας εκπαίδευσης. Χρησιμοποιείται από καθηγητές διαφόρων ειδικοτήτων όπως μαθηματικούς, φυσικούς, χημικούς ακόμα και από βιολόγους, οι οποίοι μπορούν να επινοούν σενάρια για διαφορετικά μαθήματα αξιοποιώντας τα διαθέσιμα εργαλεία του λογισμικού. Συνιστάται για μαθήματα της Γ΄ Γυμνασίου καθώς και για όλες τις τάξεις του Λυκείου.

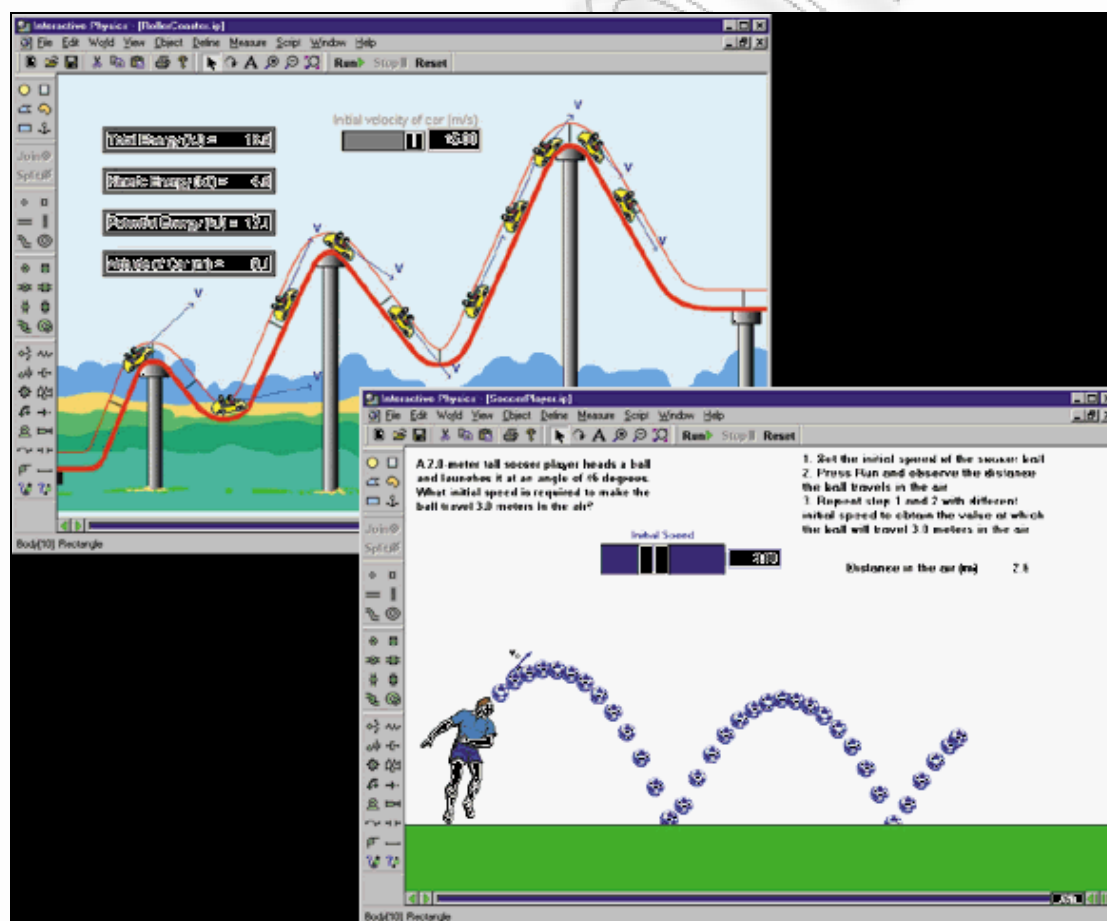


Μέσα από ήδη σχεδιασμένα πειράματα από τον κατάλογο των έτοιμων μοντέλων που συνοδεύουν το λογισμικό ή ακόμα και με τον σχεδιασμό νέων μοντέλων δίνεται στους μαθητές η δυνατότητα να περιγράφουν, να ερμηνεύουν ακόμα και να προβλέπουν διάφορα φαινόμενα, να ασκούνται στη διαδικασία της μοντελοποίησης διαφόρων φαινομένων ή καταστάσεων και να καλλιεργούν νοητικές ικανότητες για την αντιμετώπιση προβλημάτων.

Από παιδαγωγική σκοπιά το Modellus προσφέρει πολλές δυνατότητες διερευνητικής μάθησης. Μέσα από πολλαπλές αναπαραστάσεις (προσομοιώσεις, πίνακες τιμών, αρχικές συνθήκες, αποθηκεύσεις ειδικών καταστάσεων και ανάκλησή τους), τον άμεσο χειρισμό των αντικειμένων, τη δυνατότητα σκηνοθεσίας του περιβάλλοντος και τις μοντελοποιήσεις οι μαθητές μπορούν να κατανοούν καινούργιες έννοιες και διαδικασίες μέσα από την επίλυση προβλημάτων και τον πειραματισμό.

### 1.5.3. Interactive physics

Το εκπαιδευτικό λογισμικό διερευνητικού χαρακτήρα Interactive Physics [7,11] είναι ένα «ανοιχτό» περιβάλλον μάθησης. Είναι ένα πλήρες εργαστήριο κίνησης στον υπολογιστή που συνδυάζει μια απλή διεπαφή (user interface) με μια δυνατή μηχανή που προσομοιώνει τις βασικές αρχές της Κλασικής Μηχανικής του Νεύτωνα.



Εικόνα 4. Χρησιμοποιώντας το λογισμικό

Με το Interactive Physics ο εκπαιδευτικός δημιουργεί προσομοιώσεις σχεδιάζοντας αντικείμενα στην οθόνη και ζωντανεύοντάς τα με κίνηση. Υπάρχουν διαθέσιμα ελατήρια, σχοινιά, αποσβεστήρες, μετρητές και μια ποικιλία άλλων σχηματικών αντικειμένων. Κάνοντας κλικ στην Εκτέλεση, η προσομοίωση τίθεται σε κίνηση. Η ισχυρή μηχανή προσομοίωσης του Interactive Physics ορίζει πώς θα κινηθούν τα αντικείμενα και παρουσιάζει μια πολύ ρεαλιστική κίνηση.

Μεγέθη όπως η ταχύτητα, η επιτάχυνση, η ροπή, η γωνιακή ροπή, η κινητική ενέργεια και η τριβή μπορούν να υπολογιστούν ενώ εκτελείται η προσομοίωση. Οι μετρήσεις αυτές μπορούν να εμφανιστούν με αριθμούς, με γραφικές παραστάσεις ή κινούμενα διανύσματα.

Με το Interactive Physics παρέχεται η δυνατότητα δοκιμής εναλλακτικών υποθετικών σεναρίων. Οι μαθητές μπορούν να κάνουν προβλέψεις, να εκτελούν προσομοιώσεις και να βλέπουν άμεσα τα αποτελέσματα. Επίσης, το Interactive Physics διαθέτει ένα ενσωματωμένο σύστημα προγραμματισμού (scripting) που επεκτείνει τις δυνατότητές του.

Μερικοί από τους γενικούς διδακτικούς στόχους του λογισμικού είναι:

- Να ασκηθούν οι μαθητές στην περιγραφή, στην ερμηνεία και στην πρόβλεψη φαινομένων
- Να ασκηθούν στη διαδικασία μοντελοποίησης φαινομένων ή καταστάσεων
- Να καλλιεργήσουν νοητικές δεξιότητες για την αντιμετώπιση προβλημάτων

## ΚΕΦΑΛΑΙΟ 2 : ΕΙΣΑΓΩΓΗ ΣΤΗΝ UML

Στο κεφάλαιο αυτό θα πραγματοποιήσουμε μια εισαγωγή στην UML. Θα παρουσιάσουμε βασικές έννοιες, τα πεδία χρήσης και θα δούμε παραδείγματα χρήσης, ώστε ο αναγνώστης να είναι όσο χρειάζεται εξοικειωμένος πριν την παρουσίαση του λογισμικού εκμάθησης.

### 2.1. Ορισμός

Η UML [10,11,12,13] είναι μία γλώσσα μοντελοποίησης που μπορεί να χρησιμοποιηθεί τόσο από τον άνθρωπο όσο και από τις μηχανές. Χρησιμοποιείται για τον προσδιορισμό, την οπτικοποίηση με γραφικά σύμβολα καθώς και για την κατασκευή ενός συστήματος λογισμικού. Όπως θα δούμε παρακάτω, περιέχει ένα σύνολο διαγραμμάτων, τα οποία χρησιμοποιούνται για την περιγραφή των προδιαγραφών του λογισμικού που θέλουμε να υλοποιήσουμε.

Επίσης, επιτρέπει τη μοντελοποίηση των συστημάτων με βάση τις αρχές των αντικειμενοστραφών μοντέλων και χρησιμοποιεί κυρίως διαγράμματα για να εκφράσει αυτή την αντικειμενοστραφή ανάλυση και σχεδίαση έργων λογισμικού. Άλλο ένα πολύ σημαντικό σημείο που πρέπει να τονιστεί είναι ότι η γλώσσα αυτή είναι ανεξάρτητη της μεθοδολογίας μοντελοποίησης.

Η UML εφαρμόζεται για τη λύση object-oriented (OO) προβλημάτων. Για την επίλυση τέτοιου είδους προβλημάτων, θα πρέπει πρώτα να κατασκευαστεί κάποιο μοντέλο. Τα μοντέλα αποτελούνται από αντικείμενα που αλληλεπιδρούν στέλνοντας μηνύματα μεταξύ τους.

Τα αντικείμενα μπορούν να θεωρηθούν ως ζωντανοί οργανισμοί που έχουν χαρακτηριστικά γνώρισμα, μπορούν να επιτελέσουν διάφορες λειτουργίες και μπορούν να έχουν διαφορετικές συμπεριφορές. Οι τιμές των χαρακτηριστικών γνωρισμάτων των αντικειμένων καθορίζουν και την κατάστασή τους. Όσον αφορά τις κλάσεις, αυτές μπορούν να συμπεριλάβουν τα χαρακτηριστικά (ως δεδομένα) και τις συμπεριφορές (ως μεθόδους και λειτουργίες) σε μία ενιαία κι ευδιάκριτη οντότητα. Τα αντικείμενα αποτελούν στιγμιότυπα των κλάσεων.

#### 2.1.1. Εκδόσεις της UML

Σ' αυτό το σημείο θα αναφερθούμε σε κάποια ιστορικά στοιχεία της UML. Αρχικά, το 1995 ενδοποιήθηκαν κάποιες μέθοδοι αντικειμενοστραφούς μοντελοποίησης που είχαν αναπτυχθεί τη δεκαετία του 1990 κι έτσι δημιουργήθηκε η 1η έκδοση της UML. Ύστερα, το 1997, η έκδοση 1.0 της UML υιοθετείται ως πρότυπο του Object Management Group (OMG). Την ίδια χρονιά, έγιναν κάποιες ενσωματώσεις στην έκδοση αυτή και καταλήξαμε στη βασική έκδοση 1.1. Η σημασιολογία της έκδοσης αυτής δεν είχε αναπτυχθεί αρκετά, πρόβλημα το οποίο διορθώθηκε στην έκδοση 2.0. Ακολούθως, το 1998, έγιναν κάποιες τροποποιήσεις στην έκδοση 1.1 και φτάσαμε στην έκδοση 1.2. Αυτή η έκδοση μπορεί να θεωρηθεί κι ως beta επειδή οι αλλαγές αφορούσαν διορθώσεις τύπων και γραμματικών λαθών.

Το 2000 πήραμε την έκδοση 1.3, όπου οι πιο σημαντικές αλλαγές είχαν να κάνουν με τα διαγράμματα Use Cases και Activity. Πιο συγκεκριμένα, στα διαγράμματα Use Cases της έκδοσης

1.2 υπήρχαν 2 σχέσεις “uses” και “extends”. Η έκδοση 1.3 προσφέρει την “include” αντί της “uses” και προσθέτει στην “extends” τη γενίκευση (generalization). Επομένως, έχουμε τις “include” (περιλαμβάνει), “extends” (επεκτείνει) και τη generalization (γενικεύει). Επίσης την ίδια χρονιά, διορθώθηκαν κάποια σφάλματα κι έγιναν κάποιες τροποποιήσεις ώστε να καταλήξουμε στην έκδοση 1.4. Η έκδοση 1.5 της UML βγήκε στην αγορά το 2003.

Η έκδοση 2.0 της UML δημιουργήθηκε το 2005 και είναι η βασική έκδοση που χρησιμοποιείται με κάποιες παραλλαγές μέχρι και σήμερα. Μία σημαντική προσθήκη στην έκδοση 2.0 που δείχνει και τη διαφορά με την έκδοση 1.x αφορά τα στοιχεία μιας εσωτερικής δομής ενός διαγράμματος κλάσεων, όπου πλέον μπορούμε να δούμε τα χαρακτηριστικά μιας κλάσης. Πιο συγκεκριμένα, μπορούμε να δούμε όλες τις σχέσεις των αντικειμένων μιας κλάσης, πράγμα που δεν μπορούσε να συμβεί στην έκδοση 1.x., μιας και στην τελευταία φαινόταν μόνο οι σχέσεις aggregation.

Άλλη μια διαφορά της έκδοσης 1.x με την έκδοση 2.0 είναι ότι στην 1η υπήρχαν δράσεις και δραστηριότητες ενώ στη 2η υπάρχουν εσωτερικές κι εξωτερικές δραστηριότητες αντίστοιχα. Οι εσωτερικές δραστηριότητες διαρκούν λίγο, σχετίζονται συνήθως με μεταβάσεις και δεν μπορούν να διακοπούν, ενώ οι εξωτερικές δραστηριότητες διαρκούν περισσότερο, κάθε κατάσταση μπορεί να έχει μία δραστηριότητα που σχετίζεται αποκλειστικά με αυτή την κατάσταση και μπορούν να διακοπούν από διάφορα γεγονότα. Τέσσερα είναι τα βασικά σημεία των εκδόσεων 2.x: Superstructure, που καθορίζει τη σημασιολογία των διαγραμμάτων και των στοιχείων των μοντέλων. Παρέχει καλύτερη υποστήριξη στην βασισμένη στα συστατικά, ανάπτυξη, καλύτερες λύσεις για τη μοντελοποίηση των συμπεριφορών και βελτιώνει τις προδιαγραφές της αρχιτεκτονικής, Infrastructure, που καθορίζει το μοντέλο στο οποίο βασίζεται το Superstructure. Παρέχει έναν επαναχρησιμοποιήσιμο πυρήνα μετά-γλώσσας, που χρησιμοποιείται για τον ορισμό της ίδιας της UML και μηχανισμούς για τη ρύθμιση της γλώσσας, Object Constraint Language (OCL), που καθορίζει τους κανόνες για τα στοιχεία των μοντέλων και UML Diagram Interchange, που καθορίζει τον τρόπο με τον οποίο ανταλλάσσονται τα διαγράμματα της UML.

## 2.2. Πεδίο χρήσης

Όπως προαναφέραμε, UML εφαρμόζεται για τη λύση object-oriented (OO) προβλημάτων. Για την επίλυση τέτοιου είδους προβλημάτων, θα πρέπει πρώτα να κατασκευαστεί κάποιο μοντέλο. Τα μοντέλα αποτελούνται από αντικείμενα που αλληλεπιδρούν στέλνοντας μηνύματα μεταξύ τους.

Επίσης, επιτρέπει τη μοντελοποίηση των συστημάτων με βάση τις αρχές των αντικειμενοστραφών μοντέλων και χρησιμοποιεί κυρίως διαγράμματα για να εκφράσει αυτή την αντικειμενοστραφή ανάλυση και σχεδίαση έργων λογισμικού.

Υπάρχουν διάφοροι τομείς όπου χρησιμοποιείται η γλώσσα αυτή, μερικοί από τους οποίους αναφέρονται παρακάτω:

- Οι consultancy εταιρείες, οι οποίες δίνουν λύσεις στον τομέα της βιομηχανίας, παρέχοντας συμβουλευτικές υπηρεσίες χρησιμοποιώντας τη UML ως εργαλείο μοντελοποίησης. Μεγάλες εταιρείες είναι η Cargemini, KPMG, Accenture, PA Consulting, Boston Consulting.
- Οι system engineers για να κάνουν capture τα απαιτούμενα ενός project.
- Οι software και hardware engineers, οι οποίοι χρησιμοποιούν τη UML σαν οδηγό για development μοντέλων και μεθοδολογιών.
- Οι μάνατζερ, στους οποίους η UML παρέχει ένα τρόπο να μοντελοποιήσουν τις στρατηγικές τους.

## 2.3. Βασικές έννοιες

Στη UML 2 υπάρχουν δύο βασικές κατηγορίες διαγραμμάτων: τα διαγράμματα δομής και τα διαγράμματα συμπεριφοράς. Τα διαγράμματα δομής μας δείχνουν τη στατική δομή του συστήματος που μοντελοποιείται. Αντίστοιχα, τα διαγράμματα συμπεριφοράς μας δείχνουν τη δυναμική συμπεριφορά μεταξύ των αντικειμένων του συστήματος. Στοιχεία που έχουν να κάνουν

με τη δυναμική συμπεριφορά είναι οι μέθοδοι, οι συνεργασίες και οι δραστηριότητες των αντικειμένων. Με άλλα λόγια, το δυναμικό μοντέλο μας δείχνει τον τρόπο με τον οποίο αποκρίνεται το σύστημα στις ενέργειες των χρηστών ή σε άλλα εξωτερικά ερεθίσματα και πώς διαμορφώνεται η εσωτερική του κατάσταση κατά τη λειτουργία του. Επίσης, η δυναμική συμπεριφορά απεικονίζει την αλληλεπίδραση μεταξύ των αντικειμένων.

Πρώτα θα αναφερθεί επιγραμματικά η ανάλυση του συστήματος και αργότερα οι τύποι των διαγραμμάτων καθώς και λίγα λόγια για τη λειτουργία τους.

Αρχικά, έχουμε τη περιγραφή της λειτουργίας του συστήματος από την πλευρά του χρήστη, δηλαδή μοντελοποιούνται οι απαιτήσεις. Αυτό μπορεί να γίνει με χρήση των διαγραμμάτων περιπτώσεων χρήσης (use case).

Ύστερα χρειάζεται να μοντελοποιήσουμε τη στατική δομή του συστήματος, πράγμα που γίνεται με χρήση των διαγραμμάτων κλάσεων (class), αντικειμένων (object), συστατικών (component) και ανάπτυξης (deployment).

Ακολουθεί η μοντελοποίηση των δυναμικών χαρακτηριστικών του συστήματος. Αυτό πραγματοποιείται με χρήση των διαγραμμάτων ακολουθίας (sequence) κι επικοινωνίας (communication).

Αργότερα, έχουμε τη μοντελοποίηση της συμπεριφοράς συγκεκριμένων αντικειμένων μέσω των διαγραμμάτων κατάστασης (state machine).

Τέλος, βρίσκεται η μοντελοποίηση των βημάτων εκτέλεσης μιας διαδικασίας μέσω των διαγραμμάτων δραστηριότητας (activity).

Τα διαγράμματα δομής (structure) είναι ένας τύπος διαγράμματος που απεικονίζει τη δομή του συστήματος, επομένως χρησιμοποιούνται για την εισαγωγή στην αρχιτεκτονική των συστημάτων λογισμικού. Με άλλα λόγια παρουσιάζουν τα στοιχεία ενός συστήματος, τα οποία είναι ανεξάρτητα του χρόνου. Σ' αυτά ανήκουν τα διαγράμματα class, component, composite structure, deployment, object και package.

Τα διαγράμματα κλάσεων (class) περιγράφουν τη δομή ενός συστήματος, δηλαδή τις κλάσεις, τα χαρακτηριστικά των κλάσεων καθώς και τις σχέσεις μεταξύ τους. Όπως θα δούμε παρακάτω, οι σχέσεις μεταξύ των κλάσεων είναι απαραίτητες για την εκτέλεση των διαγραμμάτων περιπτώσεων χρήσεων.

Τα διαγράμματα συστατικών (component) έχουν να κάνουν με την κοκκίωση ενός συστήματος, δηλαδή με το πώς χωρίζεται ένα σύστημα σε συστατικά και μας δείχνουν τις εξαρτήσεις μεταξύ των συστατικών αυτών, όπου συστατικό είναι μία φυσική μονάδα υλοποίησης κώδικα. Οι εξαρτήσεις δείχνουν τον τρόπο με τον οποίο οι αλλαγές που πραγματοποιούνται σ' ένα συστατικό μπορούν να επηρεάσουν τη λειτουργία των άλλων συστατικών. Αξίζει να τονιστεί ότι αναπαριστά σε υψηλό επίπεδο τη δομή του κώδικα.

Τα διαγράμματα σύνθετης δομής (composite structure) περιγράφουν την εσωτερική δομή μιας κλάσης και τις συνεργασίες που μπορεί να υπάρχουν μεταξύ διαφόρων στιγμιότυπων της κλάσης αυτής. Όπως και τα διαγράμματα κλάσεων, έτσι και τα διαγράμματα σύνθετης δομής αναπαριστούν ρόλους μέσα στο σύστημα και συνδέσεις μεταξύ τους.

Τα διαγράμματα ανάπτυξης (deployment) αναπαριστούν την αρχιτεκτονική καθώς και τα περιβάλλοντα υλοποίησης του υλικού (hardware). Αναπαριστώνται οι υπολογιστές και οι συσκευές (κόμβοι) που χρησιμοποιούνται από το σύστημα καθώς και οι συνδέσεις τους, όπου κόμβοι είναι τα φυσικά αντικείμενα που έχουν μνήμη και δυνατότητα επεξεργασίας. Παρουσιάζουν την αντιστοίχιση του λογισμικού σε συσκευές, δηλαδή χρησιμοποιούνται για να δείξουν ποια συστατικά αντιστοιχούν σε ποιους κόμβους.

Τα διαγράμματα αντικειμένων (object) αναπαριστούν την δομή ενός μοντελοποιημένου συστήματος σε μία συγκεκριμένη χρονική στιγμή, δηλαδή μοντελοποιούν τα αντικείμενα. Πιο συγκεκριμένα, μπορούμε να πούμε ότι αναπαριστώνται οι σχέσεις μεταξύ των κλάσεων. Τα διαγράμματα αντικειμένων είναι παρόμοια με τα διαγράμματα κλάσεων και χρησιμοποιούν την ίδια σημειογραφία (notation). Όπως θα δείξουμε και παρακάτω, τα διαγράμματα αντικειμένων δεν είναι πολύ σημαντικά, αλλά χρησιμοποιούνται όταν τα διαγράμματα κλάσεων είναι ιδιαίτερα πολύπλοκα.

Τα διαγράμματα πακέτων (package) μας δείχνουν τον τρόπο με τον οποίο υποδιαιρείται ένα σύστημα σε λογικά σύνολα δείχνοντας τις εξαρτήσεις μεταξύ αυτών των συνόλων.

Τα διαγράμματα συμπεριφοράς (behavior) είναι ένας τύπος διαγράμματος που απεικονίζει τα χαρακτηριστικά γνωρίσματα που έχουν να κάνουν με τη συμπεριφορά ενός συστήματος. Μιας και τα διαγράμματα συμπεριφοράς απεικονίζουν τη συμπεριφορά του συστήματος, χρησιμοποιούνται κατά κύριο λόγο για την περιγραφή της λειτουργίας συστημάτων λογισμικού. Σ' αυτά ανήκουν τα διαγράμματα activity, state machine και use case, καθώς και τα 4 διαγράμματα αλληλεπίδρασης.

Τα διαγράμματα δραστηριοτήτων (activity) αναπαριστούν μια σειριακή ροή των δραστηριοτήτων (workflow ή ροή εργασίας) των στοιχείων που εκτελούνται σ' ένα σύστημα μέσα σε μια περίπτωση χρήσης. Δηλαδή δείχνει τη ροή από μια διεργασία σε μια άλλη. Ακόμα, καταγράφει μια συγκεκριμένη επιχειρησιακή διαδικασία και περιγράφει τι συμβαίνει κατά την εξέλιξη μιας περίπτωσης χρήσης μεταξύ των κλάσεων. Κύριο στοιχείο των διαγραμμάτων αυτών είναι οι ενέργειες (actions).

Τα διαγράμματα καταστάσεων (state machine) παρέχουν τη σημειογραφία για την περιγραφή πολλών συστημάτων – από προγράμματα υπολογιστών μέχρι και επιχειρησιακές διαδικασίες. Αποτελούν συμπληρωματικό διάγραμμα των διαγραμμάτων κλάσεων, δηλαδή χρησιμοποιούνται σε μία κλάση προκειμένου να μοντελοποιήσουν τη δυναμική συμπεριφορά των αντικειμένων της.

Ειδικότερα, απεικονίζει το σύστημα ως μια μηχανή πεπερασμένων καταστάσεων, όπου θα πρέπει ένα αντικείμενο να βρίσκεται σε μια συγκεκριμένη κατάσταση και αφού δεχτεί κάποιο εξωτερικό ερέθισμα, να μεταβαίνει από αυτή την κατάσταση σε κάποια άλλη. Όλα τα διαγράμματα κατάστασης ξεκινούν σε μια αρχική κατάσταση εκκίνησης του αντικειμένου που μοντελοποιείται. Μετά την αρχική κατάσταση, το αντικείμενο αλλάζει καταστάσεις. Οι συνθήκες που βασίζονται στις δραστηριότητες μιας κατάστασης του αντικειμένου μπορούν να επηρεάσουν την επόμενη κατάσταση του αντικειμένου. Αυτή η εναλλαγή καταστάσεων συμβαίνει σε διαφορετικές χρονικές στιγμές και αφορά τις καταστάσεις στις οποίες μπορεί να βρεθεί ένα αντικείμενο κατά την εκτέλεση μιας περίπτωσης χρήσης. Αξίζει να σημειωθεί ότι αφορούν μόνο κλάσεις, στιγμιότυπα των οποίων περνούν από διαφορετικές καταστάσεις.

Τα διαγράμματα περιπτώσεων χρήσης (use case) παρουσιάζουν τις λειτουργίες των χρηστών ενός συστήματος και τις εξαρτήσεις μεταξύ των χρηστών. Στο διάγραμμα αυτό, μπορεί εύκολα κάποιος να κατανοήσει τους χρήστες του συστήματος και τις περιπτώσεις χρήσης αφού οι τελευταίες συνδέονται με τους χρήστες με γραμμές που αναπαριστούν συσχετίσεις επικοινωνίας.

Όπου ως εξαρτήσεις μπορούν να θεωρηθούν οι λειτουργικές σχέσεις μεταξύ των χρηστών καθώς και η αλληλεπίδραση μεταξύ εξωτερικών χρηστών και του συστήματος. Επομένως, ένας χρήστης επικοινωνεί με το σύστημα μέσω της αποστολής και λήψης μηνυμάτων. Ακόμα, μια περίπτωση χρήσης μπορεί να έχει αρκετούς χρήστες κι ένας χρήστης με τη σειρά του μπορεί να συμμετέχει σε πολλές περιπτώσεις χρήσης.

Ως περιπτώσεις χρήσης θεωρούμε οποιοδήποτε πιθανό αποτέλεσμα ενός σεναρίου, δηλαδή όλες τις πιθανές συμπεριφορές των χρηστών. Τα σενάρια με τη σειρά τους μπορούν να έχουν επιτυχημένη ή αποτυχημένη κατάληξη. Για κάθε επιτυχημένο σενάριο, αποθηκεύονται τα βήματά του και ύστερα παρουσιάζονται όλα τα υπόλοιπα σενάρια που έχουν επιτυχή κατάληξη (με διαφορετική ακολουθία βημάτων).

Περιέχει 3 σχέσεις :

- “extends” (επεκτείνει), όπου σε μια περίπτωση χρήσης μπορούμε να προσθέσουμε επιπλέον ενέργειες και συνεπώς να την επεκτείνουμε,
- “includes” (περιλαμβάνει), όπου μια περίπτωση χρήσης περιλαμβάνει τα χαρακτηριστικά μιας άλλης και
- “generalization” (γενικεύει), όπου μια περίπτωση χρήσης γενικεύει κάποια άλλη, όμοια με αυτήν, περίπτωση χρήσης. Χρησιμοποιείται για την παρουσίαση των επιτυχημένων σεναρίων που έχουν όμως διαφορετική ακολουθία βημάτων από την αρχική.

Τα διαγράμματα αλληλεπίδρασης (interaction) είναι ένα υποσύνολο των διαγραμμάτων συμπεριφοράς που εστιάζουν στη ροή ελέγχου και δεδομένων (με άλλα λόγια στις αλληλεπιδράσεις) μεταξύ των αντικειμένων του μοντελοποιημένου συστήματος. Αλληλεπίδραση

είναι το σύνολο των μηνυμάτων που ανταλλάσσονται μεταξύ των αντικειμένων. Σ' αυτά ανήκουν τα διαγράμματα communication, interaction overview, sequence και timing diagrams. Πιο συγκεκριμένα, οι αλληλεπιδράσεις μεταξύ των αντικειμένων αναπαρίστανται από τα διαγράμματα επικοινωνίας και ακολουθίας (communication and sequence diagrams).

Τα διαγράμματα επικοινωνίας (communication) αναπαριστούν τη φυσική σύνδεση των αντικειμένων σε ένα σύστημα κι εστιάζουν στις σχέσεις μεταξύ των αντικειμένων που ανταλλάσσουν μηνύματα. Με άλλα λόγια, δείχνουν τις αλληλεπιδράσεις μεταξύ των αντικειμένων που συμβαίνουν με χρήση μηνυμάτων ακολουθίας. Είναι παρόμοιο διάγραμμα με το διάγραμμα ακολουθίας με τη διαφορά ότι αναπαριστά και τις σχέσεις μεταξύ των αντικειμένων και απεικονίζονται τα μηνύματα πάνω σε ένα διάγραμμα αντικειμένων κι όχι κατά μήκος του άξονα του χρόνου. Συνδυάζουν τα διαγράμματα κλάσεων, ακολουθίας και περιπτώσεων χρήσης και παρέχουν πληροφορίες τόσο για τη στατική δομή όσο και για τη δυναμική συμπεριφορά του συστήματος.

Τα περιληπτικά διαγράμματα αλληλεπίδρασης (interaction overview) αποτελούν ένα τύπο διαγραμμάτων δραστηριοτήτων, στα οποία οι κόμβοι αναπαριστούν διαγράμματα αλληλεπίδρασης. Αναπαριστούν τη ροή των αλληλεπιδράσεων μεταξύ των αντικειμένων σε υψηλό επίπεδο ανάλυσης.

Τα διαγράμματα ακολουθίας (sequence) μοντελοποιούν τις αλληλεπιδράσεις μεταξύ των αντικειμένων στο χρόνο. Δηλαδή μας δείχνουν τον τρόπο με τον οποίο τα αντικείμενα επικοινωνούν μεταξύ τους μέσω μιας ακολουθίας μηνυμάτων. Μοιάζει με το διάγραμμα δραστηριότητας μόνο που αντί για διαδρομές, απεικονίζεται η σειρά των ενεργειών στον άξονα του χρόνου. Επίσης, δείχνει ποιος χρήστης ξεκινάει μια δραστηριότητα κι αν συνεργάζεται με κάποιον άλλο χρήστη. Σε αντίθεση με τα διαγράμματα περιπτώσεων χρήσης, παρουσιάζεται μόνο το επιτυχημένο σενάριο κι όχι εναλλακτικά σενάρια που καταλήγουν κι αυτά σε επιτυχή κατάληξη. Ακόμα, αναπαρίστανται η χρονική σειρά εκτέλεσης σε 2 διαστάσεις : στην οριζόντια διάσταση των αντικειμένων και στην κάθετη διάσταση του χρόνου.

Τα διαγράμματα χρονισμού (timing) αποτελούν ένα συγκεκριμένο τύπο διαγραμμάτων αλληλεπίδρασης, στον οποίο εστιάζονται οι περιορισμοί για συγχρονισμό.

### 2.3.1. Βασικά Διαγράμματα UML

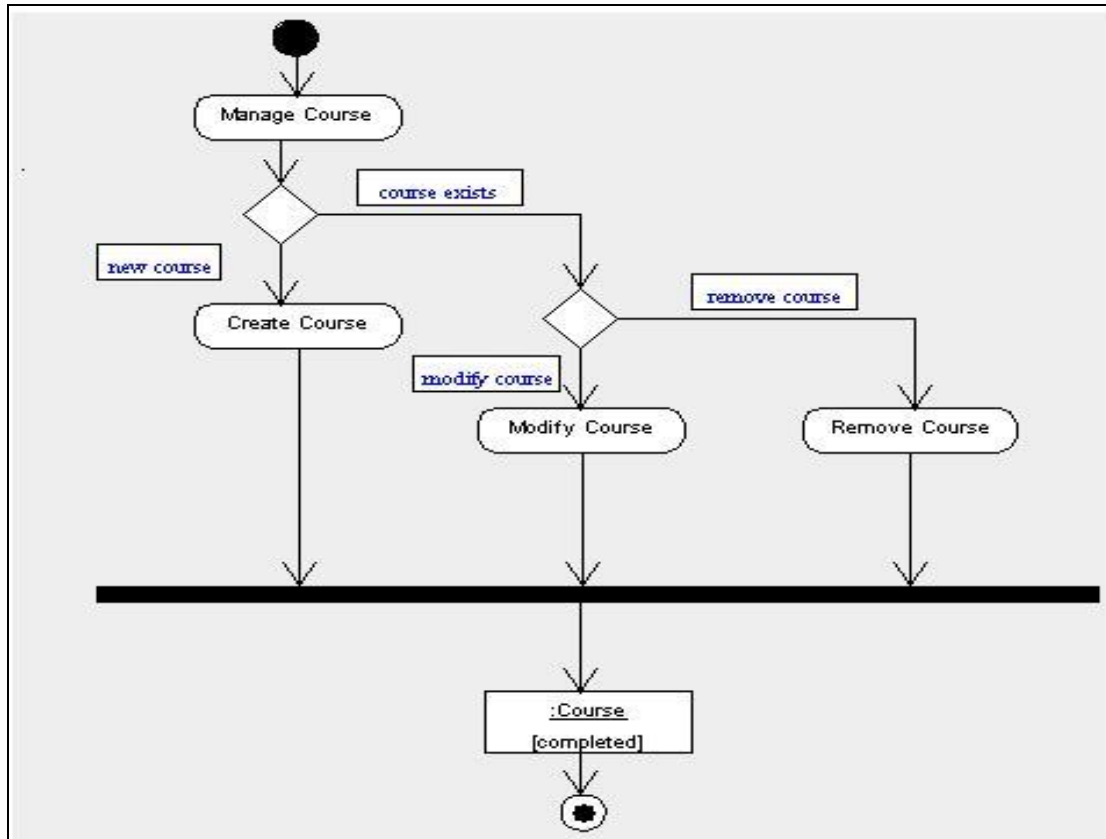
Ως βασικά διαγράμματα της UML θεωρούμε τα ακόλουθα 7 : activity, class, component, deployment, sequence, state machine και use case.

Διαγράμματα	Προτεραιότητα εκμάθησης
Activity	High
Class	High
Communication	Low
Component	Medium
Composite Structure	Low
Deployment	Medium
Interaction Overview	Low
Object	Low
Package	Low
Sequence	High
State Machine	Medium
Timing	Low
Use Case	Medium

Πίνακας 1. Προτεραιότητα εκμάθησης διαγραμμάτων UML

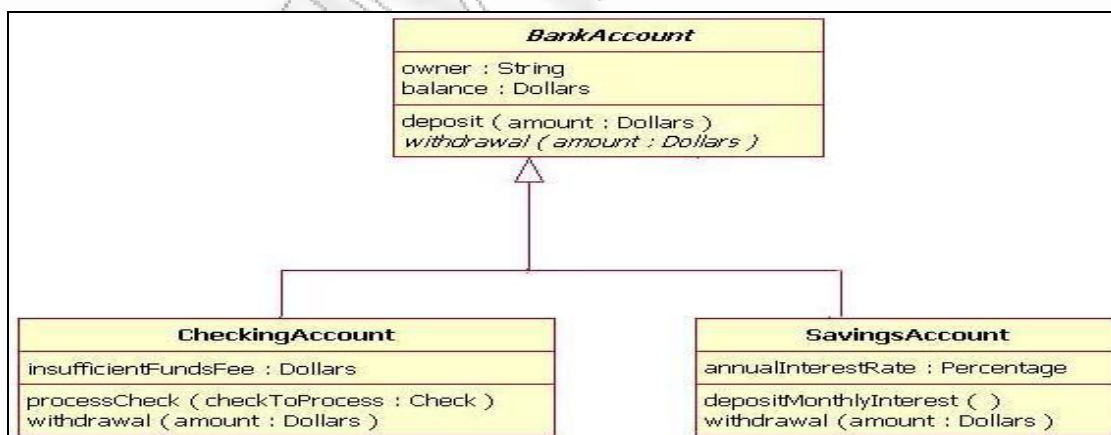
Τα διαγράμματα δραστηριοτήτων (**activity**) αναπαριστούν το workflow των στοιχείων που εκτελούνται σ' ένα σύστημα. Χρησιμοποιούνται για τη μοντελοποίηση των εσωτερικών λειτουργιών

του συστήματος, δηλαδή για την περιγραφή της λειτουργίας ενός οργανισμού ή για τη μοντελοποίηση των διαδικασιών του οργανισμού.



Εικόνα 5. Διάγραμμα Activity (<http://www.developer.com/design/article.php/2247041>)

Τα διαγράμματα κλάσεων (class) περιγράφουν τη δομή ενός συστήματος, δηλαδή τις κλάσεις, τα χαρακτηριστικά των κλάσεων καθώς και τις σχέσεις μεταξύ τους.

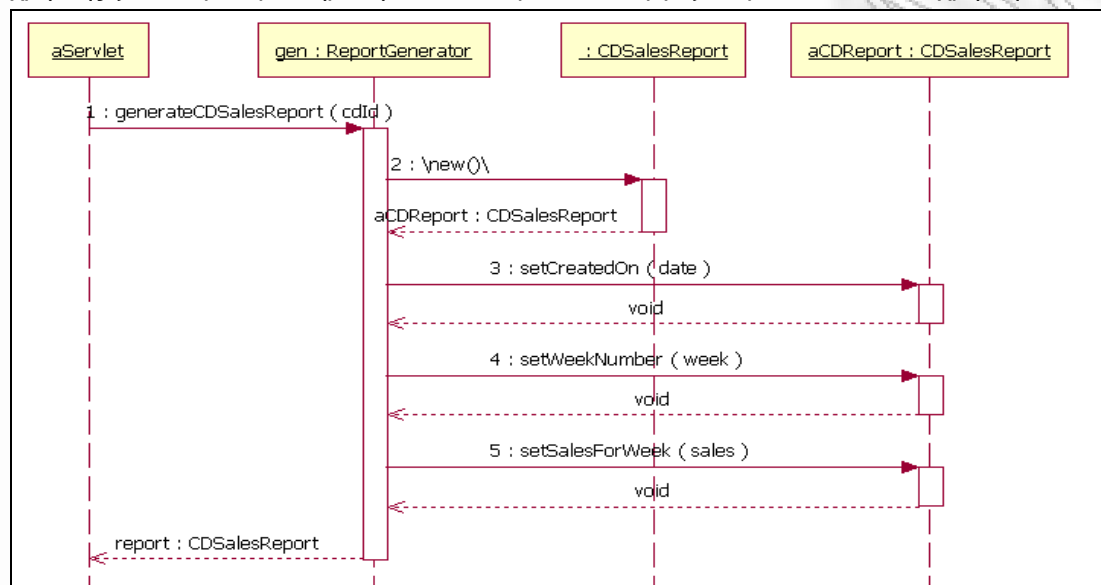


Εικόνα 6. Διάγραμμα Class (<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/>)

Τα διαγράμματα συστατικών (component) απεικονίζουν τα συστατικά που συνθέτουν ένα σύστημα, μια εφαρμογή ή μία επιχείρηση. Δηλαδή παρουσιάζονται τα συστατικά, καθώς και οι εξαρτήσεις και οι αλληλεπιδράσεις μεταξύ τους. Επιπλέον, τα διαγράμματα αυτά αναπαριστούν τα interfaces που χρησιμοποιούνται από τα συστατικά για την επικοινωνία μεταξύ τους.

Τα διαγράμματα ανάπτυξης (**deployment**) αναπαριστούν την αρχιτεκτονική του συστήματος. Εκεί ανήκουν οι κόμβοι, οι συνδέσεις των κόμβων και τα περιβάλλοντα υλοποίησης του υλικού και του λογισμικού.

Τα διαγράμματα ακολουθίας (**sequence**) μοντελοποιούν τις αλληλεπιδράσεις μεταξύ των αντικειμένων κατά μήκος του άξονα του χρόνου και μας δείχνουν τον τρόπο με τον οποίο τα αντικείμενα επικοινωνούν μεταξύ τους μέσω μιας ακολουθίας μηνυμάτων. Επίσης, δείχνει ποιος χρήστης ξεκινάει μια δραστηριότητα κι αν κι εφόσον συνεργάζεται με κάποιον άλλο χρήστη.

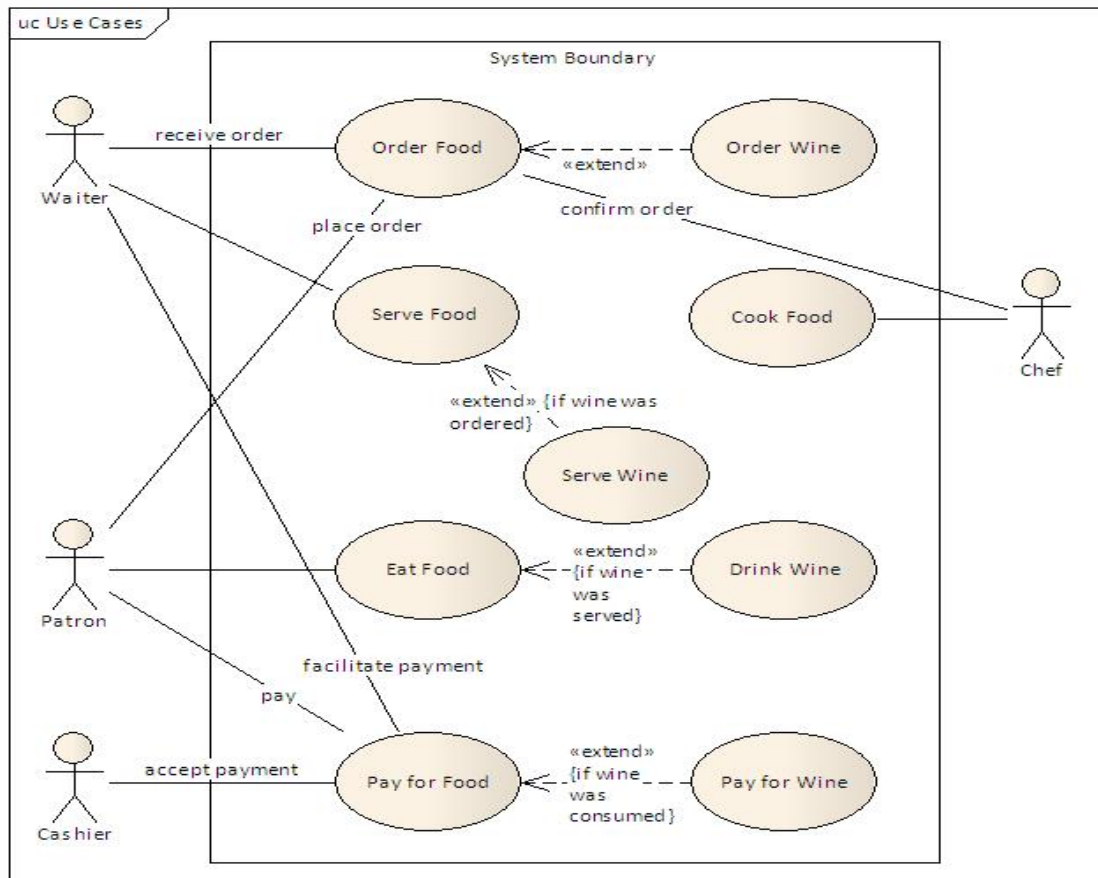


Εικόνα 7. Διάγραμμα Sequence (<http://www.ibm.com/developerworks/rational/library/769.html>)

Τα διαγράμματα καταστάσεων (**state machine**) απεικονίζουν το σύστημα ως μια μηχανή πεπερασμένων καταστάσεων, όπου θα πρέπει ένα αντικείμενο ή μία αλληλεπίδραση να βρίσκεται σε μια συγκεκριμένη κατάσταση. Εφόσον το αντικείμενο δεχτεί κάποιο εξωτερικό ερέθισμα, μεταβαίνει από τη μία κατάσταση σε κάποια άλλη.

Τα διαγράμματα περιπτώσεων χρήσης (**use case**) επικεντρώνονται στα χαρακτηριστικά του συστήματος και στις λειτουργικές απαιτήσεις που πρέπει να παρέχει, δηλαδή παρουσιάζουν τις λειτουργίες των χρηστών ενός συστήματος και τις εξαρτήσεις μεταξύ των χρηστών. Σε αντίθεση με τα διαγράμματα δραστηριοτήτων, αυτά χρησιμοποιούνται για τη μοντελοποίηση των εξωτερικών λειτουργιών του συστήματος, δηλαδή για την περιγραφή του τρόπου με τον οποίο αλληλεπιδρά το σύστημα με τους χρήστες του ή με άλλα πληροφοριακά συστήματα. Είναι σημαντικό διάγραμμα γιατί περιγράφει το στόχο και ορίζει το domain του συστήματος.





Εικόνα 8. Διάγραμμα Use Case (Wikipedia)

## 2.4. Παραδείγματα χρήσης

Σε αυτήν την ενότητα θα δούμε ορισμένα παραδείγματα, για να εξοικειωθούμε με την χρήση της UML.

### 2.4.1. Αγορά μετοχών από το διαδίκτυο

Πρωταγωνιστής: Αγοραστής

Έκταση: Personal Advisors / Finance Package (PAF)

Επίπεδο: Χρήστη

Εταίροι και συμφέροντα: -

Αγοραστής: Θέλει να αγοράσει μετοχές οι οποίες θα προστεθούν αυτομάτως στο χαρτοφυλάκιο PAF.

Χρηματιστηριακή: Χρειάζεται όλες τις πληροφορίες της αγοράς

Προϋπόθεση: Ο χρήστης έχει ανοίξει το PAF.

Δεσμεύσεις: Θα πρέπει να έχουν καταγραφεί αρκετά στοιχεία στα logs του συστήματος ώστε το PAF μπορεί να εντοπίσει ότι κάτι πήγε στραβά και να ζητήσει σχετικές λεπτομέρειες από τον χρήστη.

Εκπλήρωση: Ο δικτυακός τόπος έχει επιβεβαιώσει την αγορά, τα logs και το χαρτοφυλάκιο του πελάτη ενημερώνονται κατάλληλα.

Βασικό Σενάριο

Ο αγοραστής επιλέγει να αγοράσει μετοχές μέσω διαδικτύου.

Το PAF παίρνει από τον χρήστη το όνομα του δικτυακού τόπου που θα χρησιμοποιηθεί για την αγορά.

Το PAF συνδέεται με τον δικτυακό τόπο, διατηρώντας τον έλεγχο.

Ο αγοραστής αναζητά και αγοράζει μετοχές από τον δικτυακό τόπο.

Το PAF λαμβάνει τις απαντήσεις από τον δικτυακό τόπο και ενημερώνει το χαρτοφυλάκιο του αγοραστή.

Το PAF δείχνει στον αγοραστή τη νέα κατάσταση του χαρτοφυλακίου του.

Εναλλακτικά σενάρια

α. Ο αγοραστής επιθυμεί να αγοράσει μετοχές από έναν δικτυακό τόπο τον οποίο δεν υποστηρίζει το PAF:

Το σύστημα παίρνει νέα επιλογή από τον αγοραστή, με το ενδεχόμενο να ακυρωθεί η περίπτωση χρήσης.

β. Αποτυχία επικοινωνίας με τον δικτυακό τόπο:

Το σύστημα ενημερώνει τον αγοραστή για την αποτυχία και επιστρέφει στο προηγούμενο βήμα.

Ο αγοραστής είτε σταματάει την περίπτωση χρήσης είτε προσπαθεί εκ νέου.

γ. Ο υπολογιστής κρεμάει ή σβήνει στη διάρκεια της δοσοληψίας:

???

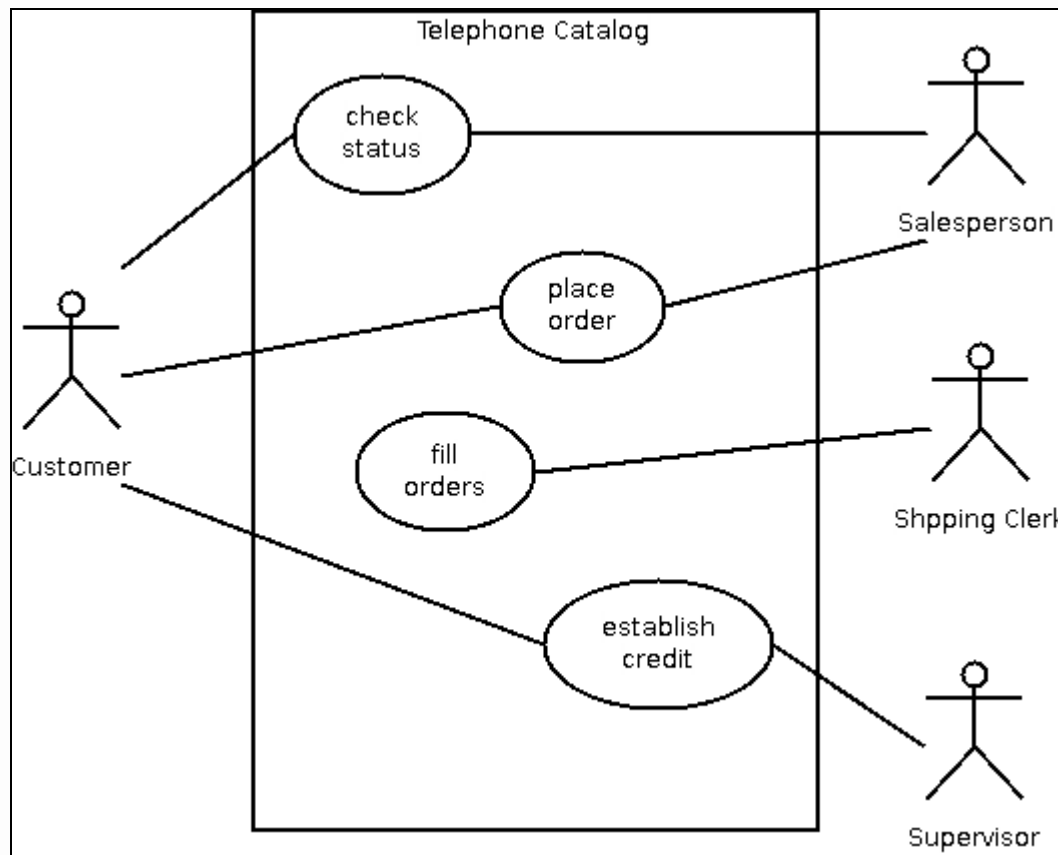
δ. Ο δικτυακός τόπος δεν επιβεβαιώνει την αγορά αλλά τη βάζει σε αναμονή:

Το PAF καταγράφει την καθυστέρηση και θέτει σε λειτουργία χρονόμετρο προκειμένου να ρωτήσει τον αγοραστή για το πέρας της εντολής αγοράς.

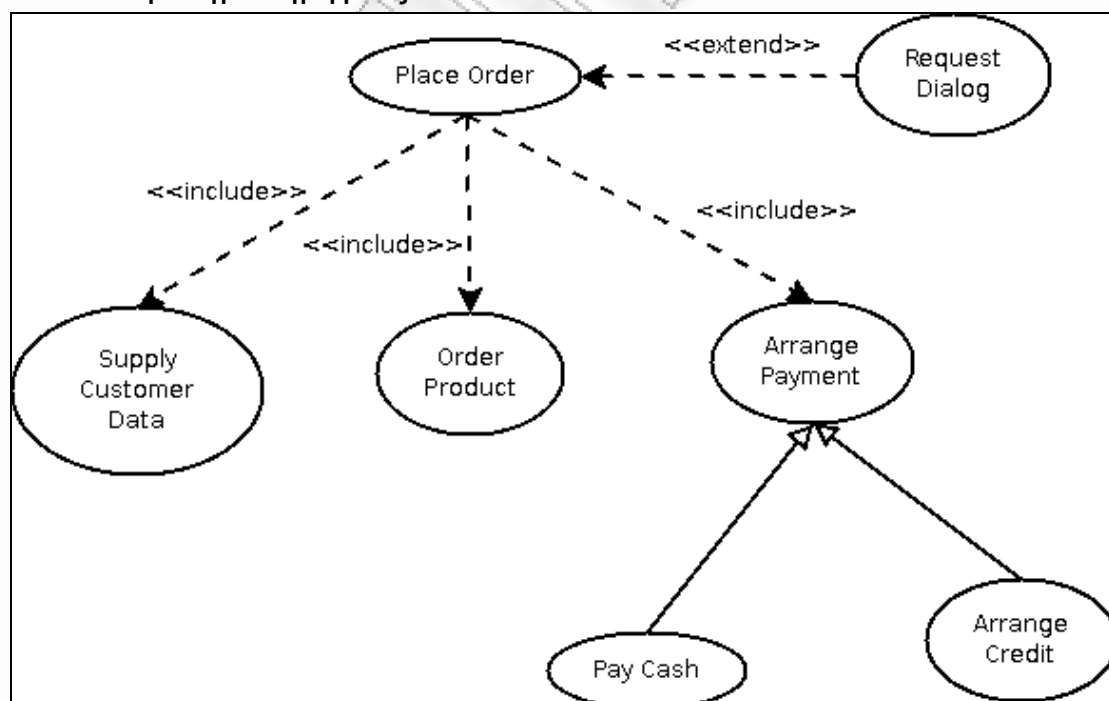
ε. Ο δικτυακός τόπος δεν επιστρέφει τις απαιτούμενες πληροφορίες για την αγορά:

Το PAF γράφει στα logs την έλλειψη πληροφορίας και απαιτεί από τον αγοραστή να προχωρήσει σε ενημέρωση αμφισβητούμενης αγοράς. (<http://foo.bar>)

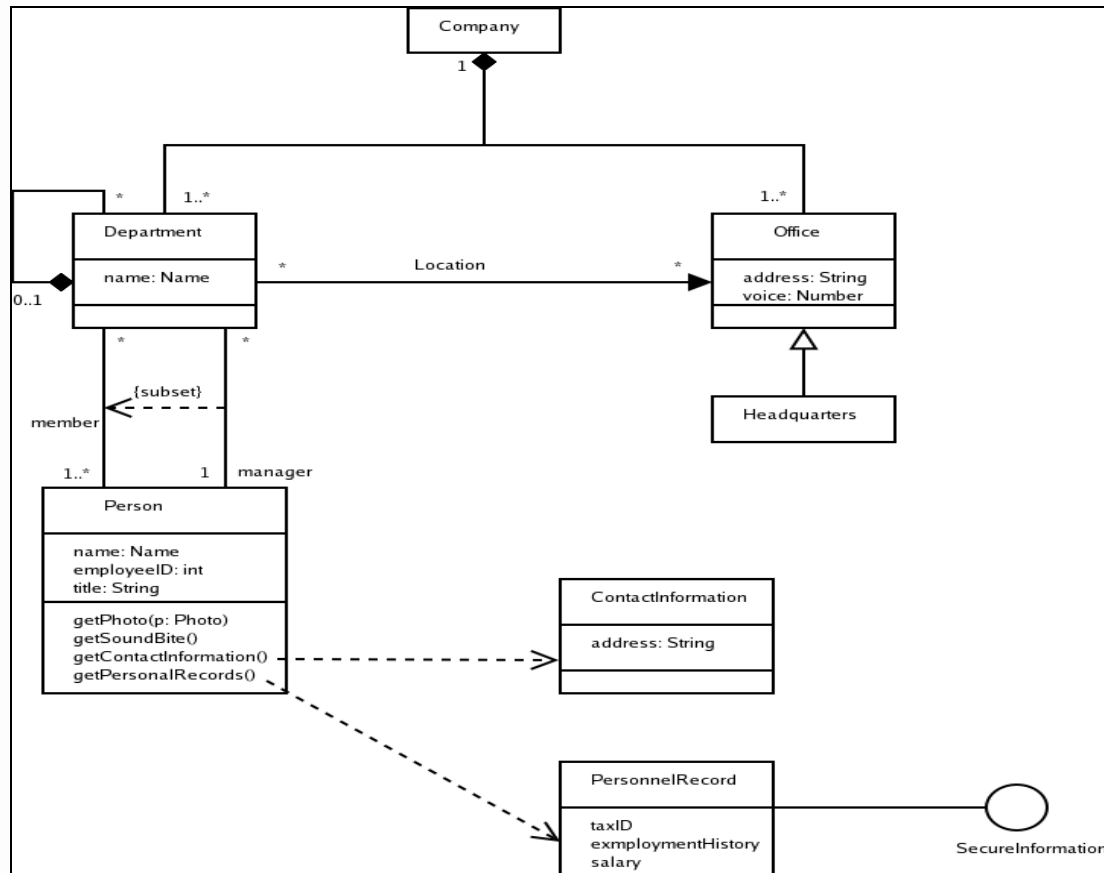
2.4.2. Διαγράμματα



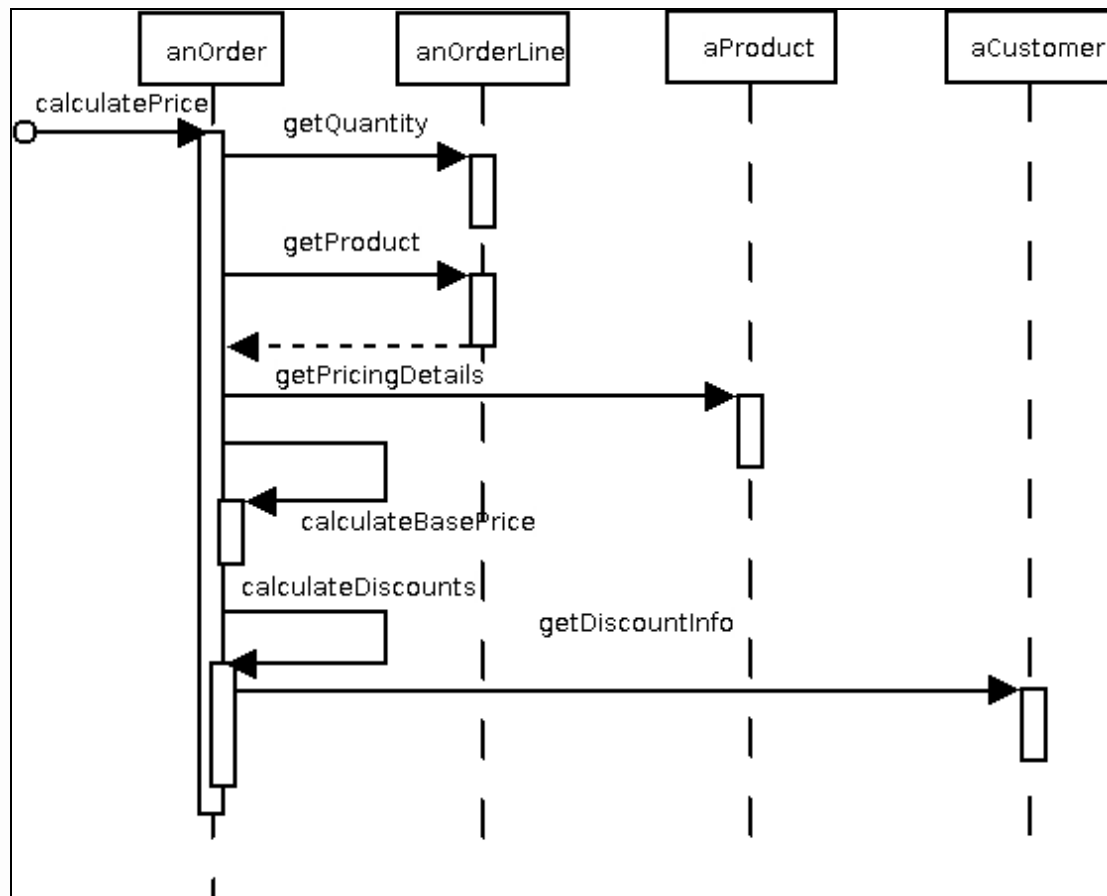
Εικόνα 9: Παράδειγμα διαγράμματος UML



Εικόνα 10: Παράδειγμα διαγράμματος UML



Εικόνα 11: Παράδειγμα διαγράμματος UML



Εικόνα 12: Παράδειγμα διαγράμματος UML

## ΚΕΦΑΛΑΙΟ 3 : ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ

Στο παρόν κεφάλαιο θα κάνουμε μια επισκόπηση των τεχνολογιών και των προγραμματιστικών εργαλείων που χρησιμοποιούνται για την παραγωγή εκπαιδευτικού λογισμικού. Ιδιαίτερη έμφαση θα δοθεί σε ό,τι χρησιμοποιήθηκε για την δική μας υλοποίηση.

### 3.1. Γλώσσες προγραμματισμού

Όπως έχουμε δει ως τώρα, ένα λογισμικό εκπαιδευτικού σκοπού δεν παύει να είναι ένα λογισμικό εφαρμογών. Συνεπώς για την κατασκευή του μας απασχολούν τα ίδια ζητήματα που απασχολούν τον προγραμμαστή μιας οποιαδήποτε εφαρμογής. Ποιό είναι το target group του λογισμικού; Πώς πρέπει να χρησιμοποιείται; Σε τι λειτουργικό σύστημα θα εκτελείται; Ποια είναι τα γνωρίσματα που πρέπει να έχει; Έτσι λοιπόν κάνουμε επιλογές για τα εργαλεία που θα χρησιμοποιήσουμε. Ένα από αυτά τα εργαλεία είναι η γλώσσα προγραμματισμού. Στην διάρκεια της ιστορίας της επιστήμης των υπολογιστών, έχουν αναπτυχθεί πάρα πολλές γλώσσες, η κάθε μια με τα δικά της χαρακτηριστικά, σκοπούς και τελικό αντίκτυπο. Παρακάτω θα παρουσιάσουμε μια σειρά γλωσσών που χρησιμοποιούνται κατά κόρον στην παραγωγή εκπαιδευτικού λογισμικού.

#### 3.1.1. Java

Η Java[2,3,4] είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Στις αρχές του 1991, στη Sun αναζητούσαν το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικρο-συσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της

εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς αποφάνθησαν ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Η επίσημη εμφάνιση της Java αλλά και του HotJava (πλοηγός με υποστήριξη Java) στη βιομηχανία της πληροφορικής έγινε το Μάρτιο του 1995 όταν η Sun την ανακοίνωσε στο συνέδριο Sun World 1995. Από εκεί και πέρα η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα (GPL) στο μεταγλωττιστή (javac) και στο πακέτο ανάπτυξης (jdk).

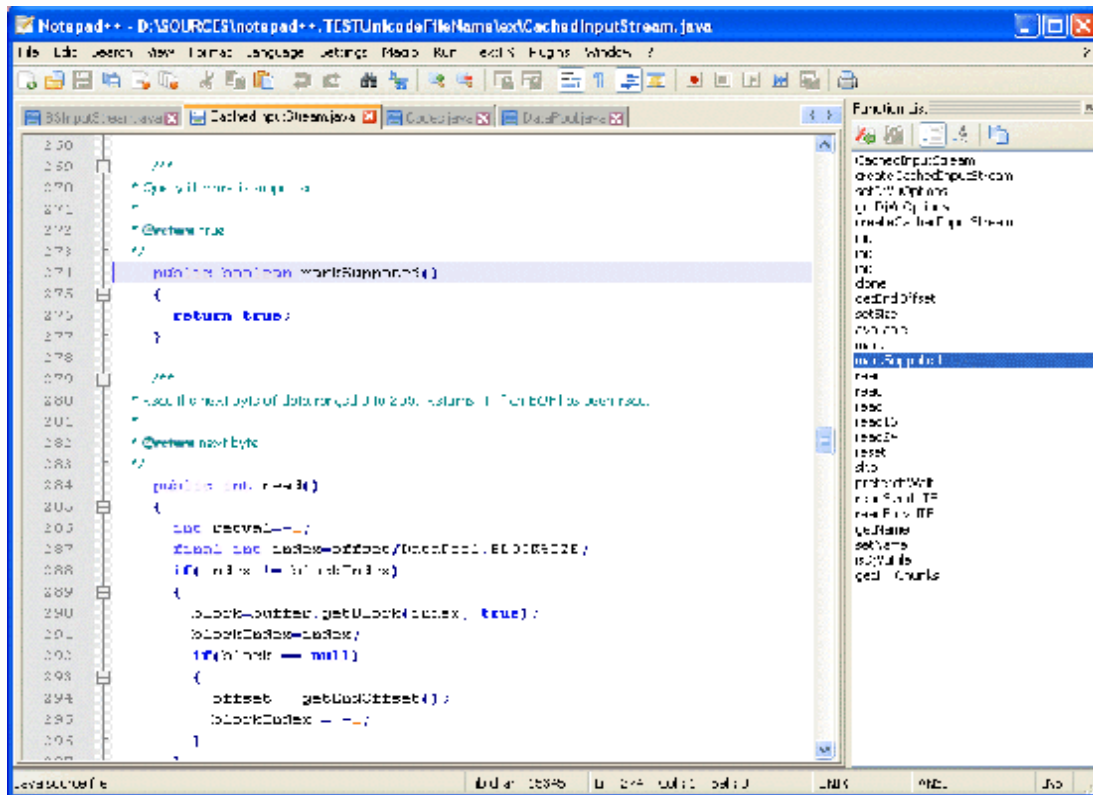
Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός (assembly) κώδικας που εκτελείται σε Windows είναι διαφορετικός από αυτόν που εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά).



**Εικόνα 13. Το λογότυπο της Java.**

Αφού γραφεί κάποιο πρόγραμμα σε Java τότε μεταγλωττίζεται μέσω του εργαλείου javac, το οποίο παράγει έναν αριθμό από αρχεία .class (=bytecode). Το bytecode είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν προσπαθήσουμε λοιπόν να εκτελέσουμε την εφαρμογή μας το Java Virtual Machine που πρέπει να είναι εγκατεστημένο στο μηχανήμα μας, θα αναλάβει να διαβάσει τα αρχεία .class και να τα μεταφράσει σε γλώσσα και εντολές μηχανής (assembly) που υποστηρίζει το λειτουργικό μας σύστημα και ο επεξεργαστής μας, έτσι ώστε να εκτελεστεί (να σημειώσουμε εδώ ότι αυτό συμβαίνει με την παραδοσιακή Εικονική Μηχανή (Virtual Machine). Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν πολύχρηστα τμήματα bytecode απ' ευθείας σε εγγενή κώδικα (native code) με αποτέλεσμα να βελτιώνεται η ταχύτητα). Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Πρέπει να πούμε ότι η JVM είναι λογισμικό εξαρτημένο από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοση του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδιομηχανές κλπ.

Ο,τιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Από την άλλη μεριά ούτε ο χρήστης μπορεί να κατεβάσει «κακό» κώδικα από το δίκτυο και να τον εκτελέσει. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα καταναμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως.



Εικόνα 14. Μια γεύση από κώδικα Java

### 3.1.2. Html

Τα αρχικά HTML προέρχονται από τις λέξεις HyperText Markup Language. Η HTML δεν είναι μια γλώσσα προγραμματισμού (αν και ονομάζουμε όσους την χρησιμοποιούν HTML programmers). Είναι μια περιγραφική γλώσσα, δηλαδή ένας ειδικός τρόπος γραφής κειμένου και κλήσης άλλων αρχείων ή εφαρμογών βασισμένος σε οδηγίες (tags). Ο Web client αναγνωρίζει αυτόν τον ειδικό τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν.

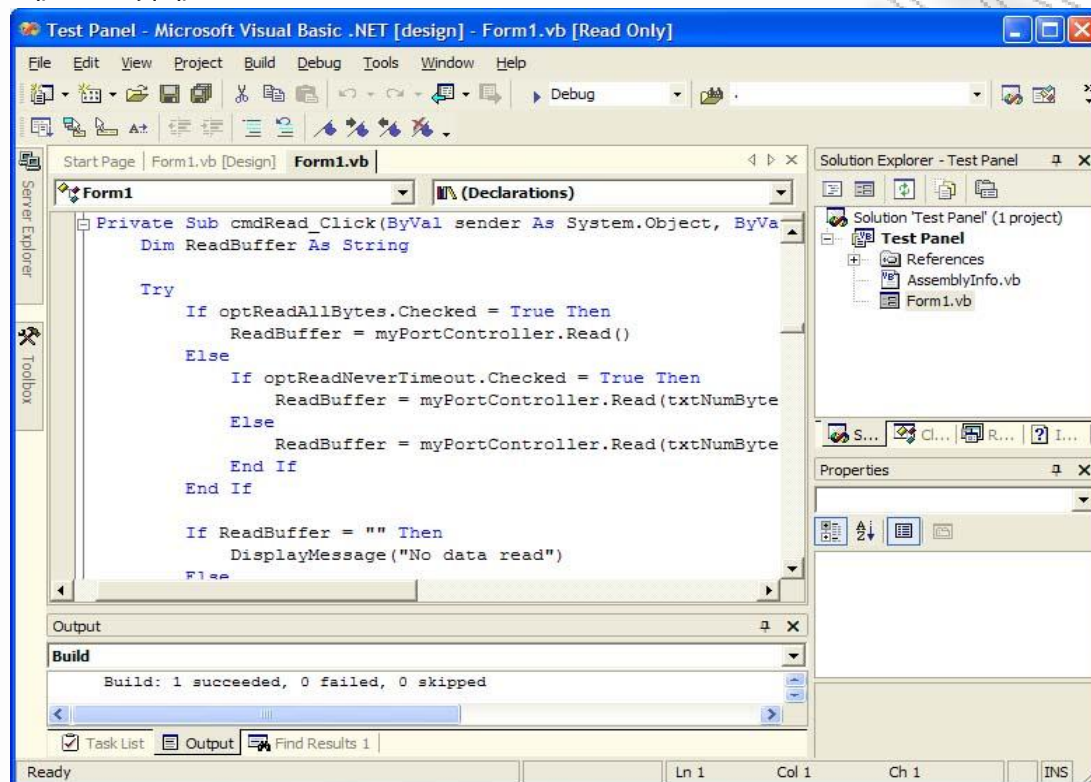
Όπως είπαμε, η HTML [2,5,6] είναι μία γλώσσα σήμανσης (*markup language*), δηλαδή ένας ειδικός τρόπος γραφής κειμένου. Ο καθένας μπορεί να δημιουργήσει ένα αρχείο HTML χρησιμοποιώντας απλώς έναν επεξεργαστή κειμένου. Αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων στα διάφορα υπολογιστικά συστήματα. Ο browser αναγνωρίζει αυτόν τον τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν. Αξίζει να σημειωθεί ότι η html είναι η πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας. Η html χρησιμοποιεί τις ειδικές επικέτες (τα tags) να δώσει τις απαραίτητες οδηγίες στον browser. Τα tags είναι εντολές που συνήθως ορίζουν την αρχή ή το τέλος μιας λειτουργίας. Τα tags βρίσκονται πάντα μεταξύ των συμβόλων < και >. Π.χ. <BODY> Οι οδηγίες είναι case insensitive, δεν επηρεάζονται από το αν έχουν γραφτεί με πεζά (μικρά) ή κεφαλαία. Ένα αρχείο HTML πρέπει να έχει κατάληξη htm ή html.

Για να μπορούν οι browser να ερμηνεύουν σχεδόν απόλυτα σωστά την html έχουν θεσπιστεί κάποιοι κανόνες. Αυτοί οι κανόνες είναι γνωστοί ως προδιαγραφές. Επομένως σχεδόν κάθε είδος υπολογιστή μπορεί να δείξει το ίδιο καλά μια ιστοσελίδα. Οι πρώτες προδιαγραφές ήταν η html 2.0. Πρόβλημα προέκυψε όταν η Microsoft και η Netscape πρόσθεσαν στην html τέτοιες δυνατότητες που στην αρχή τουλάχιστον ήταν συμβατές μόνο με συγκεκριμένους browser. Ακόμη και σήμερα υπάρχουν διαφορές στην απεικόνιση κάποιας σελίδας από διαφορετικούς browsers. Ιδιαίτερο είναι το πρόβλημα όταν η ιστοσελίδα, εκτός από "καθαρή" HTML περιλαμβάνει και εφαρμογές Javascript.





λειτουργικότητα. Καθώς υπάρχουν προεπιλεγμένα χαρακτηριστικά και ενέργειες για τα επιμέρους στοιχεία, μπορεί να δημιουργηθεί ένα απλό πρόγραμμα χωρίς ο προγραμματιστής να γράψει πολλές γραμμές κώδικα. Στις προηγούμενες εκδόσεις υπήρχαν προβλήματα επιδόσεων, αλλά με τους ταχύτερους υπολογιστές και τη μεταγλώττιση εγγενούς κώδικα αυτό παύει να είναι ένα τόσο σημαντικό ζήτημα.



Εικόνα 16. Μια γεύση από κώδικα VB.net.

Οι φόρμες δημιουργούνται χρησιμοποιώντας τεχνικές drag-and-drop. Χρησιμοποιείται ένα εργαλείο για την τοποθέτηση στοιχείων ελέγχου (π.χ. πλαίσια κειμένου, κουμπιά, κλπ.) στη φόρμα (παράθυρο). Τα στοιχεία ελέγχου έχουν χαρακτηριστικά και χειριστές συμβάντων συνδεδεμένους με αυτά. Οι προεπιλεγμένες τιμές παρέχονται όταν δημιουργείται το στοιχείο ελέγχου, αλλά μπορούν να τροποποιηθούν από τον προγραμματιστή. Πολλές τιμές χαρακτηριστικών είναι δυνατό να τροποποιηθούν κατά το χρόνο εκτέλεσης από ενέργειες του χρήστη ή αλλαγές του περιβάλλοντος, παρέχοντας έτσι μια δυναμική εφαρμογή. Για παράδειγμα, μπορεί να εισαχθεί κώδικας στον χειριστή συμβάντων αλλαγής διαστάσεων της φόρμας, ώστε ένα στοιχείο ελέγχου να παραμένει πάντα στο κέντρο της φόρμας ή να μεγαλώσει ώστε να την γεμίσει, κλπ. Με την προσθήκη κώδικα μέσα σε ένα χειριστή συμβάντων για το πάτημα των πλήκτρων σε ένα πλαίσιο κειμένου, το πρόγραμμα μπορεί αυτόματα να μετατρέψει το εισαγόμενο κείμενο σε κεφαλαία ή πεζά ή ακόμα και να εμποδίσει ορισμένους από τους χαρακτήρες να εμφανιστούν.

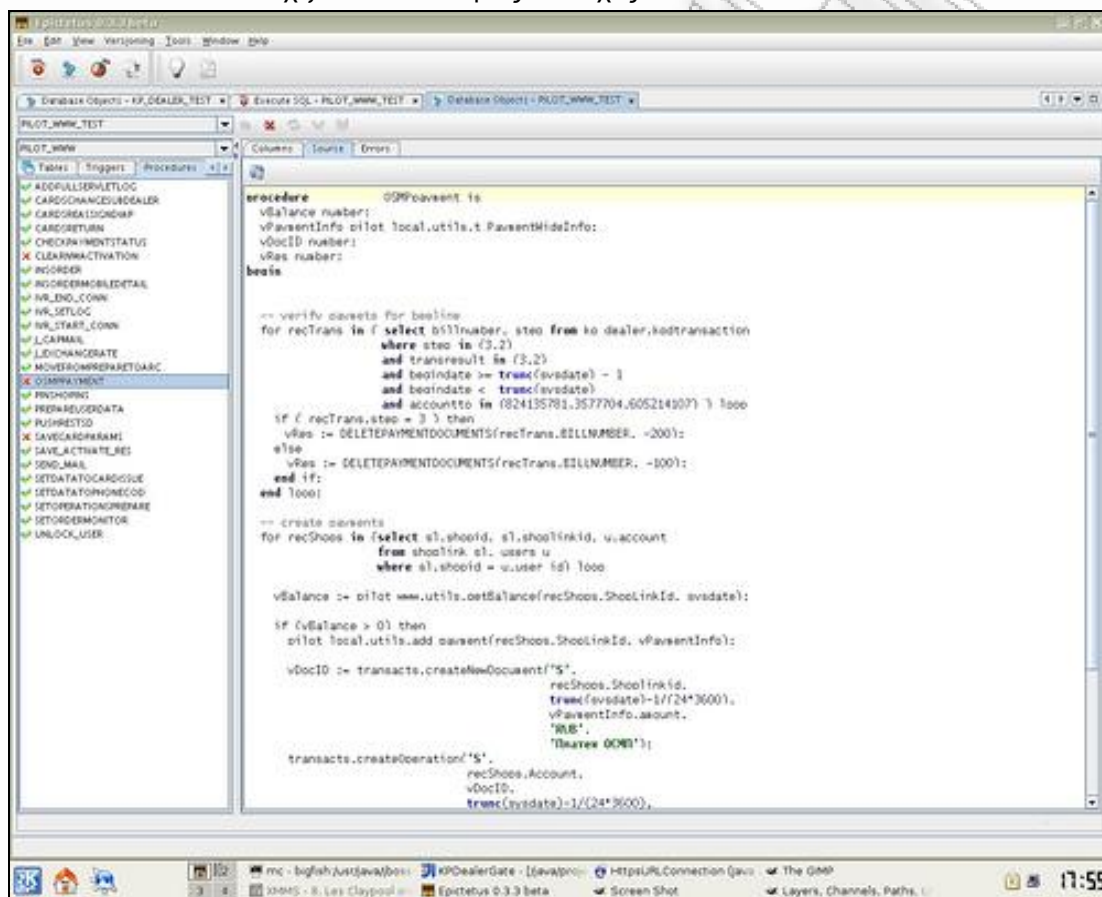
Με τη Visual Basic είναι δυνατή η δημιουργία εκτελέσιμων (EXE) αρχείων, στοιχείων ελέγχου ActiveX ή αρχείων DLL, αλλά χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών για τα Windows και τη διασύνδεση συστημάτων βάσεων δεδομένων. Πλαίσια διαλόγου με λιγότερες λειτουργίες μπορούν να χρησιμοποιηθούν για pop-up δυνατότητες. Τα στοιχεία ελέγχου παρέχουν τις βασικές λειτουργίες της εφαρμογής, ενώ οι προγραμματιστές μπορούν να εισαγάουν επιπλέον λογική μέσα στο κατάλληλο χειριστή γεγονότων. Για παράδειγμα, ένα πτυσσόμενο πλαίσιο θα εμφανίζει αυτόματα μια λίστα που θα επιτρέπει στο χρήστη να επιλέξει οποιοδήποτε στοιχείο. Ένας χειριστής γεγονότων καλείται όταν ένα αντικείμενο είναι επιλεγμένο, και στη συνέχεια μπορεί να εκτελεστεί πρόσθετος κώδικας που δημιουργείται από τον προγραμματιστή για να εκτελεστεί κάποια ενέργεια που βασίζεται στο στοιχείο που έχει επιλεγεί.

### 3.2. Ολοκληρωμένα περιβάλλοντα προγραμματισμού (IDE)

Ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment, IDE) είναι μία σουίτα λογισμικού που βοηθάει στην ανάπτυξη προγραμμάτων υπολογιστή. Συνήθως ένα IDE περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή, συνδέτη, σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές.

#### 3.2.1. NetBeans

Το NetBeans [12] είναι ένα επιτυχημένο ερευνητικό έργο ανοικτού κώδικα (open source) με μεγάλο αριθμό χρηστών, μια αναπτυσσόμενη κοινότητα, με περισσότερους από 100 συνεργάτες παγκοσμίως. Η Sun Microsystems ίδρυσε το ερευνητικό έργο ανοικτού κώδικα NetBeans τον Ιούνιο του 2000 και συνεχίζει να είναι ο κύριος ανάδοχος.



Εικόνα 17. Μια γέυση από τη χρήση του λογισμικού

Σήμερα δύο ερευνητικά έργα υπάρχουν: Το NetBeans IDE και το NetBeans Platform. Το NetBeans IDE είναι ένα περιβαλλοντικό εργαλείο IDE - ένα εργαλείο στους προγραμματιστές για να γράψουν, να κάνουν compile, debug και να αναπτύξουν προγράμματα. Είναι γραμμένο σε Java - αλλά μπορεί να υποστηρίξει όλες τις γλώσσες προγραμματισμού. Υπάρχει επίσης ένας μεγάλος αριθμός υπομονάδων (modules) που βοηθάνε στην επέκταση της λειτουργικότητας του NetBeans IDE. Το NetBeans IDE είναι ένα ελεύθερο προϊόν δίχως περιορισμούς στον τρόπο χρησιμοποίησής του.

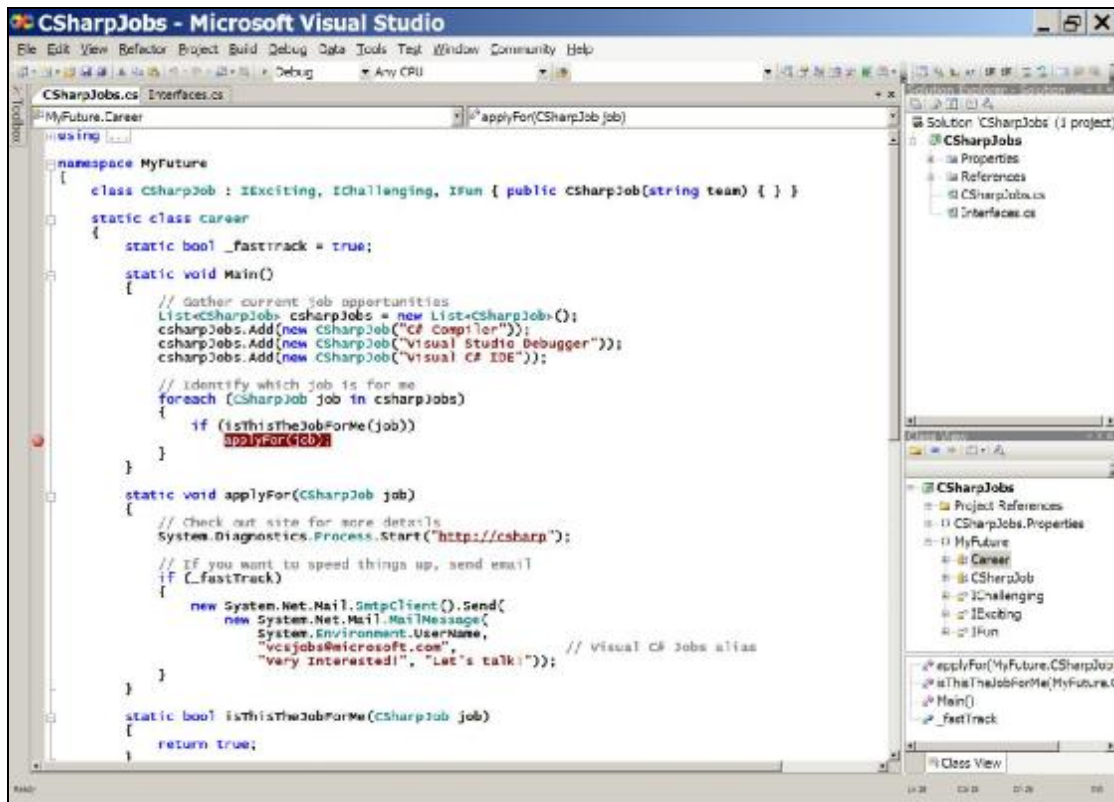
Το netbeans.org είναι το σπίτι της NetBeans κοινότητα ανοιχτής πηγής (open source community) η οποία είναι αφοσιωμένη στο χτίσιμο ενός παγκοσμίας τάξεως IDE. Το netbeans.org επιτρέπει στους χρήστες περισσότερων από 160 χωρών παγκοσμίως να είναι σε επαφή με πηγές γνώσεων και άτομα που περιβάλλουν το NetBeans.

Διαθέσιμο επίσης είναι το NetBeans Platform; ένα framework αποτελούμενο από υπομονάδες (modular) που χρησιμοποιείται σαν βάση λογισμικού για τη δημιουργία μεγάλων desktop εφαρμογών. Οι ISV συνεργάτες διαθέτουν προσθήκες, επιπρόσθετα προγράμματα (plugins) που εύκολα συνενώνονται στο Platform και μπορούν επίσης να χρησιμοποιηθούν για την ανάπτυξη άλλων εργαλείων και λύσεων. Ο πηγαίος κώδικας (source code) είναι διαθέσιμος για επαναχρησιμοποίηση κάτω από το Common Development and Distribution License (CDDL).

### 3.2.2. Visual studio

Το Windows Visual Studio [13] προσφέρει ένα ευρύ φάσμα εργαλείων για όλες τις φάσεις ανάπτυξης λογισμικού, συμπεριλαμβανομένης της δημιουργίας, δοκιμής, ανάπτυξης, εντοπισμού και διαχείρισης και επιτρέπει στους προγραμματιστές να επικοινωνούν μέσω ηλεκτρονικών υπολογιστών, διακομιστών, του Web και φορητών συσκευών. Το Visual Studio Team System 2008 προσφέρει εργαλεία ανάπτυξης εφαρμογών όπως:

- Σύστημα έργου - Χρησιμοποιείται για τη διαχείριση των δεδομένων που απαιτούνται για την ανάπτυξη εφαρμογών των έργων σας
- Επεξεργασία κώδικα - Εργαλεία για δημιουργία και τροποποίηση κειμένου και κώδικα
- Επανασχεδιασμός προγράμματος και διόρθωση σφαλμάτων - Εργαλεία για τη βελτίωση κώδικα και τον εντοπισμό και την επίλυση λογικών σφαλμάτων
- Δημιουργία αναφορών - Εργαλεία που βοηθούν στην τήρηση αρχείου και τη διαχείριση εργασιών
- Πλατφόρμα - Εργαλεία για τη δημιουργία εφαρμογών για τις τεχνολογίες Office, Windows CE, .NET και Windows
- Για προχωρημένους - Εργαλεία για το σχεδιασμό, την υλοποίηση, την αξιοποίηση, την ανάλυση την επαλήθευση και την αξιολόγηση της προόδου ανάπτυξης των εφαρμογών
- Γλώσσα - Επιλογές που περιλαμβάνονται στο JScript 8.0, το Visual Basic 2008, το Visual C# 2008 και το Visual C++ 2008



Εικόνα 18. Μια γεύση από τη χρήση του λογισμικού

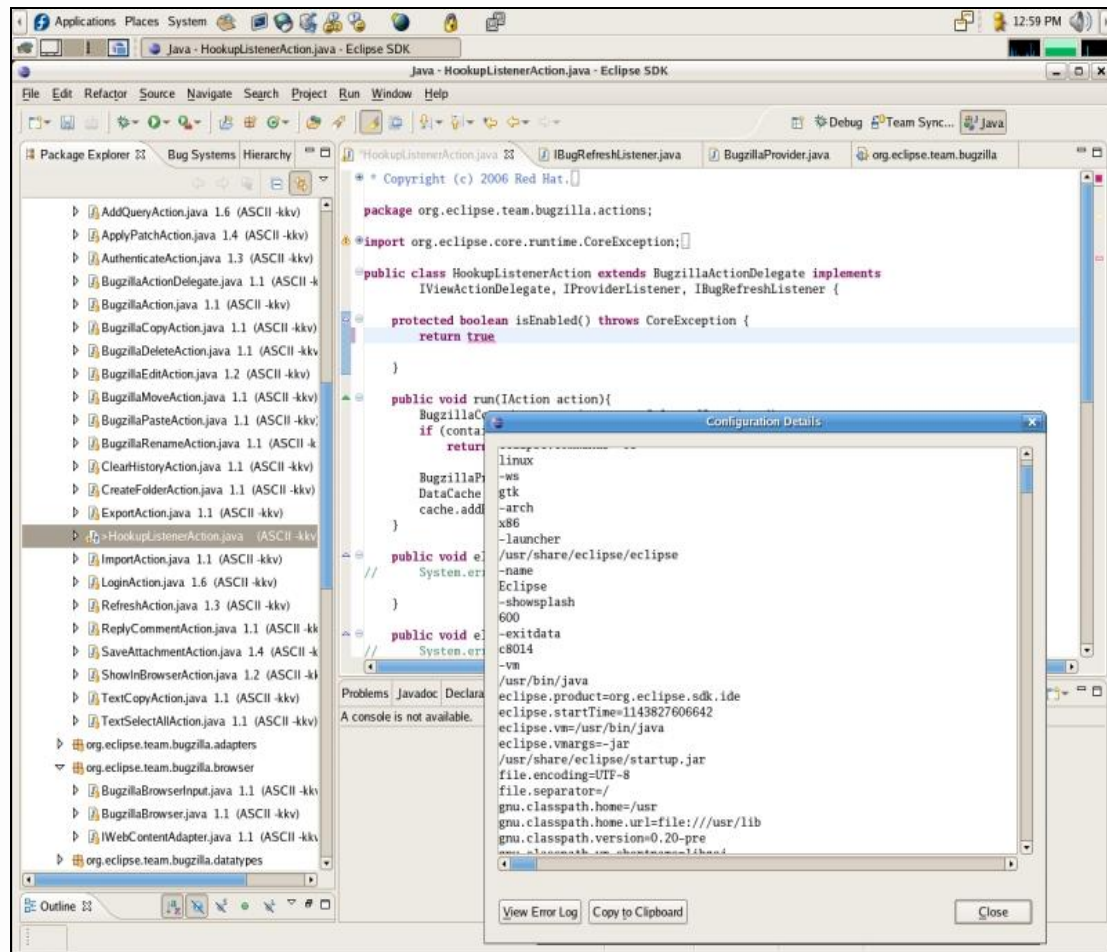
Στις δυνατότητες που προσφέρει αυτό το IDE, συμπεριλαμβάνονται και τα εξής:

- Συνεργασία μεταξύ όλων των μελών της ομάδας ανάπτυξης και επιτάχυνση στην ανάπτυξη εφαρμογών χρησιμοποιώντας τα εργαλεία συνεργασίας του Visual Studio Team System
- Ανάπτυξη επιχειρηματικού λογισμικού σε διάφορες γλώσσες προγραμματισμού.
- Οργανωμένο σύστημα έργων και αρχείων με το Solution Explorer.
- Πρόσβαση και χρήση εργαλείων για την ανάπτυξη και τον εντοπισμό σφαλμάτων μιας εφαρμογής συμπεριλαμβανομένων των εργαλείων .NET Framework, του Error Lookup, του Spy++ και του Visual Studio Command Prompt
- Επισήμανση και περιήγηση σε γραμμές κώδικα χρησιμοποιώντας σελιδοδείκτες.
- Έγχρωμα διακριτικά στον κώδικα για ευκολότερο εντοπισμό χρησιμοποιώντας τη δυνατότητα χρωματισμού κώδικα
- Εισαγωγή έτοιμου κώδικα χρησιμοποιώντας τη δυνατότητα με τα τμήματα κώδικα.

### 3.2.3. Eclipse

Το Eclipse [2] είναι ένα λογισμικό που ανήκει στην κατηγορία των Ολοκληρωμένων Περιβαλλόντων Ανάπτυξης ( I.D.E. ). Ο χρήστης του λογισμικού αυτού είναι σε θέση να δημιουργήσει projects σε γλώσσες προγραμματισμού όπως C, C++ αλλά, κυρίως, σε γλώσσα JAVA [11]. Το Eclipse είναι μία ευρύτατα διαδεδομένη πλατφόρμα ανάπτυξης εφαρμογών η οποία διατίθεται στην default έκδοσή της για Java developers. Μεγάλο όπλο του Eclipse είναι

η δυνατότητα προσθήκης (άπειρων) plugins ώστε να υποστηρίζει δεκάδες επιπλέον γλώσσες και δυνατότητες. Επιπλέον είναι διαθέσιμο σε όλες τις πλατφόρμες.



Εικόνα 19. Χρησιμοποιώντας το λογισμικό

### 3.3. Εργαλεία λογισμικού

Στην παράγραφο αυτή θα παρουσιάσουμε μια σειρά από εργαλεία λογισμικού τα οποία χρησιμοποιούνται για την ανάπτυξη εφαρμογών διαχείριση μάθησης και εν γένει για εκπαιδευτικό λογισμικό.

#### 3.3.1. Blender

Το Blender [4] είναι ένα λογισμικό δημιουργίας τρισδιάστατων εικόνων με πάρα πολλές δυνατότητες σε εφέ και άλλα εργαλεία. Είναι διαθέσιμο για Windows όσο και για Linux/Unix. Το Blender είναι λογισμικό ανοικτού κώδικα και κυκλοφορεί υπό την άδεια GPL.

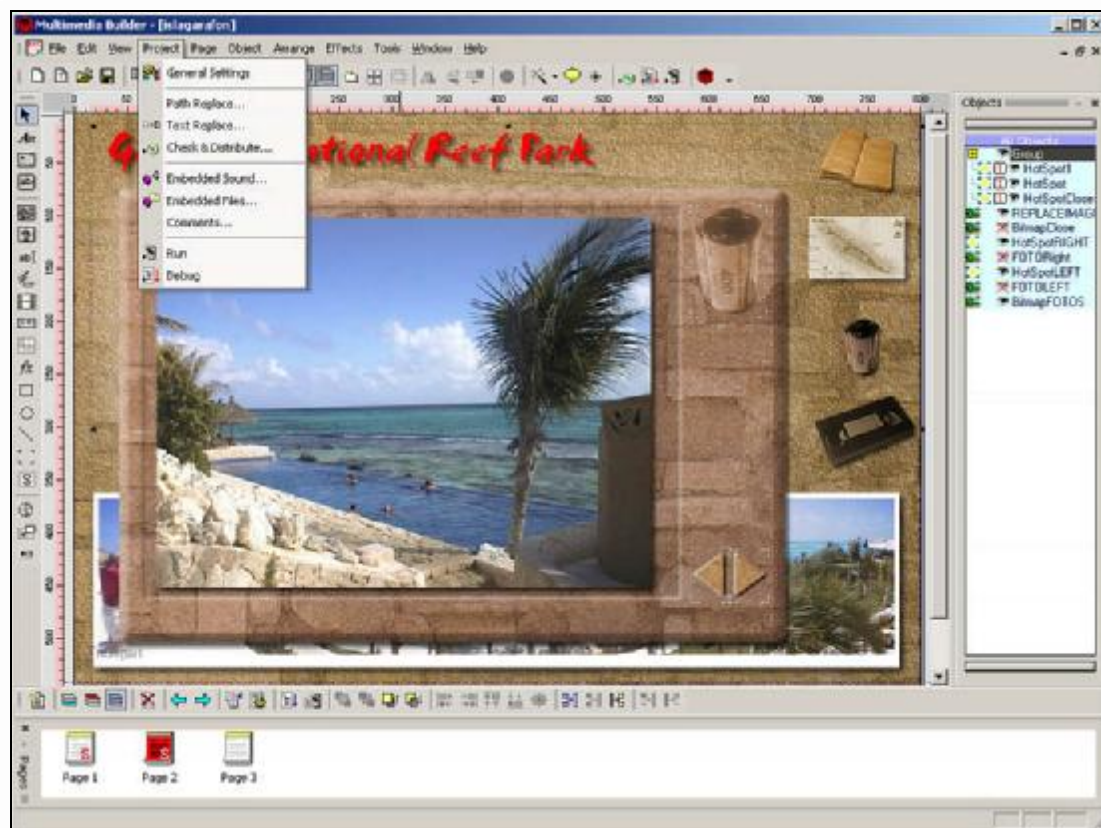
Εκτός από στατικές εικόνες, μπορούμε να δημιουργήσουμε πολλά περισσότερα όπως μοντέλα, ταινίες, animation, ακόμα και παιχνίδια.



Εικόνα 20. Χρησιμοποιώντας το λογισμικό

### 3.3.2. Multimedia builder

Το Multimedia Builder [1] είναι ένα πρόγραμμα, ένα εργαλείο ανάπτυξης Εκπαιδευτικού Λογισμικού με τη δυνατότητα χρήσης υπερμέσων (Hypermedia), κατάλληλο για εκπαιδευτικούς, αλλά και μαθητές. (Τα υπερμέσα εμπεριέχουν τα υπερκείμενα και τα πολυμέσα. Το υπερκείμενο ως γνωστό είναι μια μη γραμμική διάταξη κειμένων, τα οποία συνδέονται μεταξύ τους με λέξεις ή σήματα - «κόμβους» και «συνδέσμους». Οι σύνδεσμοι ενώνουν ένα κόμβο με έναν άλλο ή πολλαπλούς κόμβους σχηματίζοντας έτσι μια υπερμεσική διάταξη της πληροφορίας, η οποία είναι μη γραμμική και ο τρόπος προσπέλασης ή διαδρομής από κόμβο σε κόμβο είναι μια διαδικασία που επιλέγεται από το χρήστη). Τα πολυμέσα περιλαμβάνουν περισσότερα του ενός μέσα, όπως κείμενα, ήχους, εικόνες, γραφικά, κινούμενα σχέδια, video κ.λ.π.

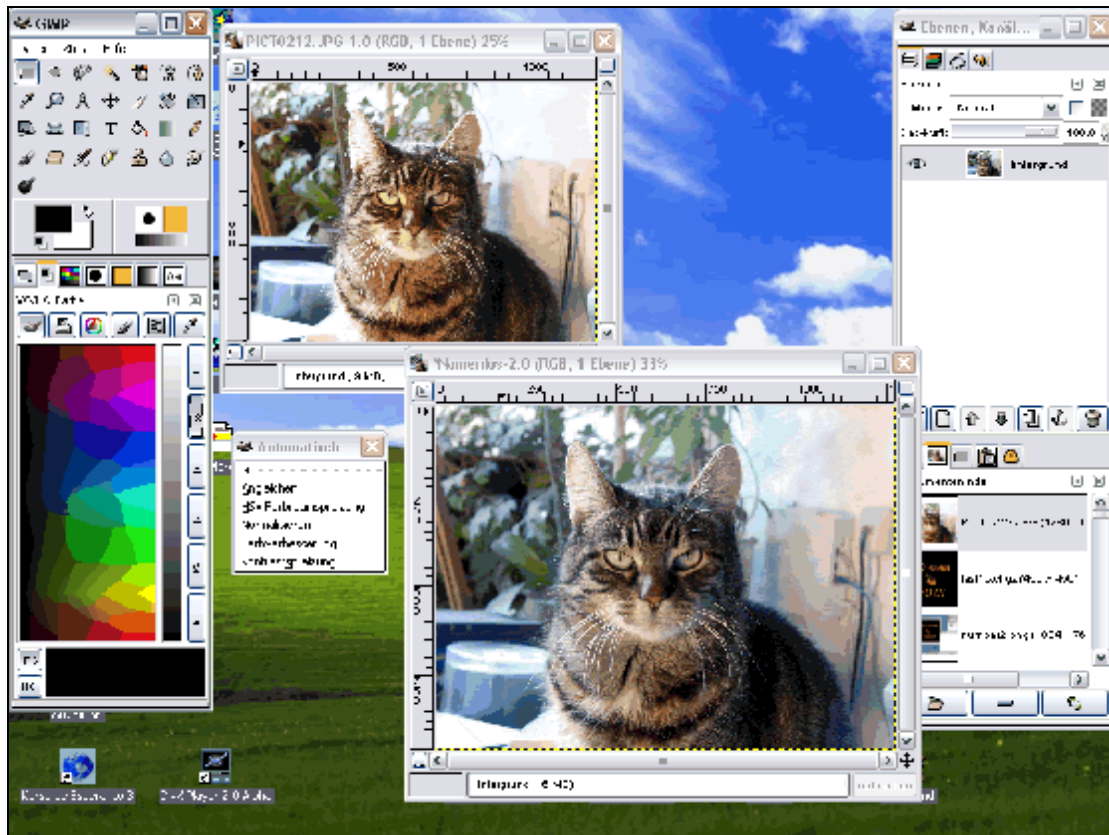


**Εικόνα 21. Χρησιμοποιώντας το λογισμικό**

Το πακέτο αυτό παρέχει τη δυνατότητα χρήσης πλήθους εργαλείων (π.χ., επεξεργαστή κειμένου με τα γραφικά του και με δυνατότητα δημιουργίας υπερκειμένων και υπερμέσων, σχεδιαστικό πρόγραμμα, εξελιγμένη μορφή της Logo, λογιστικό φύλλο και βάση δεδομένων, εικόνα και βίντεο της προτίμησης των χρηστών, ήχο (που μπορεί να είναι και οι φωνές των μαθητών ή η μουσική της προτίμησης τους κτλ, ) και αποτελεί παράδειγμα λογισμικού για την εφαρμογή της εποικοδομιστικής μάθησης με όλα τα χαρακτηριστικά που τη συνοδεύουν. Υπάρχει επίσης δυνατότητα σύνδεσης του προγράμματος με το διαδίκτυο με όλα τα γνωστά οφέλη που μπορεί να προκύψουν, εφόσον οι δραστηριότητες είναι παιδαγωγικά σχεδιασμένες.

### 3.3.3. GIMP

Το GIMP [3], που σημαίνει GNU Image Manipulation Program, είναι ένα πρόγραμμα επεξεργασίας γραφικών και εικόνας. Μερικές χρήσεις του είναι η δημιουργία γραφικών, η αλλαγή διαστάσεων μιας φωτογραφίας καθώς και η διόρθωση του χρωματικού τόνου και της φωτεινότητας, και πολλές άλλες. Το GIMP μπορεί επίσης να χρησιμοποιηθεί για δημιουργία κινούμενων εικόνων σε μορφή Gif.



**Εικόνα 22. Χρησιμοποιώντας το λογισμικό**

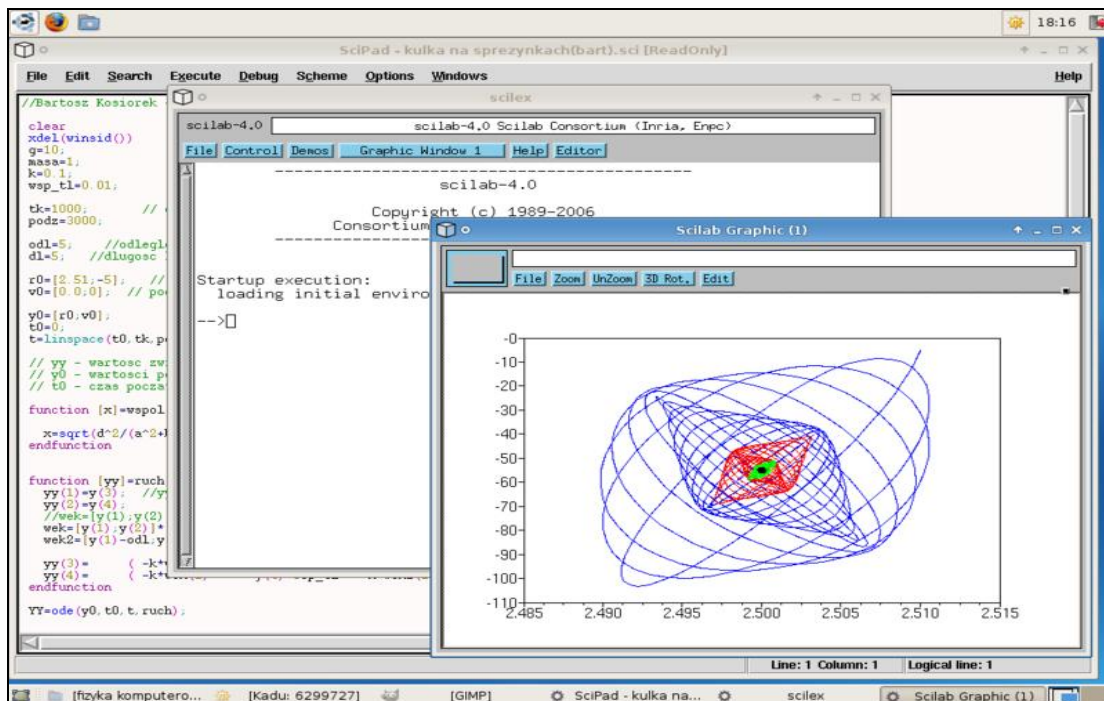
Το GIMP ξεκίνησε το 1995 από τον Spencer Kimball και τον Peter Mattis. Αποτελεί πλέον μέρος του GNU Project και διατίθεται με την GNU General Public License. Η εργαλειοθήκη GTK+, που αναπτύχθηκε αρχικά για το πρόγραμμα, έχει βρει χρήση σε πολλές ακόμα εφαρμογές, και αποτελεί σήμερα τη βάση γνωστών περιβαλλόντων εργασίας, όπως τα GNOME και Xfce.

### 3.3.4. Scilab

Το Scilab [5] είναι ένα απλό, αλλά ισχυρό, εργαλείο προγραμματισμού που επιτρέπει την εύκολη χρήση μαθηματικών συναρτήσεων, στατιστικών μεθόδων, και τη δημιουργία τυχαίων αριθμών για προσομοίωση.

Είναι και αυτό λογισμικό ανοικτού κώδικα που διατίθεται δωρεάν υπό την GPL άδεια χρήσης.





Εικόνα 23. Χρησιμοποιώντας το λογισμικό

### 3.3.5. Drupal

Το Drupal [12,13] είναι ένα αρθρωτό σύστημα διαχείρισης περιεχομένου (Content Management System, CMS) ανοικτού/ελεύθερου λογισμικού, γραμμένο στη γλώσσα προγραμματισμού PHP. Το Drupal, όπως πολλά σύγχρονα CMS, επιτρέπει στο διαχειριστή συστήματος να οργανώνει το περιεχόμενο, να προσαρμόζει την παρουσίαση, να αυτοματοποιεί διαχειριστικές εργασίες και να διαχειρίζεται τους επισκέπτες του ιστοτόπου και αυτούς που συνεισφέρουν. Παρόλο που υπάρχει μια πολύπλοκη προγραμματιστική διεπαφή, οι περισσότερες εργασίες μπορούν να γίνουν με λίγο ή και καθόλου προγραμματισμό. Το Drupal ορισμένες φορές περιγράφεται ως "[υποδομή για εφαρμογές ιστού]", καθώς οι δυνατότητές του προχωρούν παραπέρα από τη διαχείριση περιεχομένου, επιτρέποντας ένα μεγάλο εύρος υπηρεσιών και συναλλαγών.

Το Drupal μπορεί να εκτελεστεί σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των λειτουργικών συστημάτων Windows, Mac OS X, Linux, FreeBSD, ή οποιασδήποτε πλατφόρμας που υποστηρίζει είτε το διακομιστή ιστοσελίδων Apache HTTP Server (έκδοση 1.3+), είτε το Internet Information Services (έκδοση IIS5+), καθώς επίσης και τη γλώσσα προγραμματισμού PHP (έκδοση 4.3.3+). Το Drupal απαιτεί μια βάση δεδομένων όπως η MySQL και η PostgreSQL για την αποθήκευση του περιεχομένου και των ρυθμίσεών του.



Εικόνα 24. Χρησιμοποιώντας το λογισμικό

## ΚΕΦΑΛΑΙΟ 4: ΤΟ ΛΟΓΙΣΜΙΚΟ JEDUML (JAVA EDUCATIONAL UML)

Το κεφάλαιο αυτό αναφέρεται στην υλοποίηση του εκπαιδευτικού λογισμικού JEDUML. Τι έχει φτιαχτεί, πώς έχει υλοποιηθεί και πώς λειτουργεί. Επίσης, γίνεται μια ποιοτική και ποσοτική αξιολόγηση του λογισμικού.

### 4.1. Σκοπός

Στα πλαίσια αυτής της εργασίας υλοποιήθηκε το λογισμικό JEDUML, του οποίου το όνομα προέρχεται από την φράση Java EDucational UML. Σκοπός ήταν να αποτελέσει ένα λογισμικό που θα μπορεί να υποστηρίξει την εξοικείωση με τις βασικές έννοιες της Unified Modelling Language, τόσο σε θεωρητικό, όσο και σε πρακτικό επίπεδο.

Το λογισμικό αυτό απευθύνεται κυρίως σε μαθητές και σπουδαστές που χρειάζονται έναν πιο αυτοματοποιημένο τρόπο για να γνωριστούν με την UML και να αρχίσουν να την χρησιμοποιούν. Σκοπός της παρούσας υλοποίησης δεν είναι η παραγωγή ενός full-featured λογισμικού γενικής χρήσης. Δεν επιθυμούσαμε δηλαδή να καταλήξουμε στην δημιουργία ενός ακόμα CASE Tool γιατί αυτό θα ήταν και κοπιώδες και εκτός των ορίων της παρούσας εργασίας.

Στις παρακάτω ενότητες παρουσιάζονται πιο αναλυτικά τα χαρακτηριστικά του λογισμικού.

### 4.2. Τεχνικά χαρακτηριστικά

Το JEDUML υλοποιήθηκε με την γλώσσα προγραμματισμού Java, για την οποία έγινε αναφορά σε προηγούμενο κεφάλαιο. Η ανάπτυξη έγινε στο ολοκληρωμένο περιβάλλον προγραμματισμού Net Beans, σε λειτουργικό σύστημα Microsoft Windows XP.

Υπάρχει η απαίτηση για να λειτουργήσουν σωστά ορισμένα χαρακτηριστικά, το λογισμικό να εκτελείται στο ίδιο λειτουργικό σύστημα και με την παρουσία του φυλλομετρητή Mozilla Firefox. Ωστόσο, αυτό δεν εμποδίζει την συνολική λειτουργία ενώ με ελάχιστο optimization μπορεί να δείξει

το πλήρες δυναμικό του σε οποιοδήποτε σύστημα. Αυτό άλλωστε είναι αναμενόμενο και επιθυμητό, λόγω της πλατφόρμας ανάπτυξης που επιλέχθηκε.

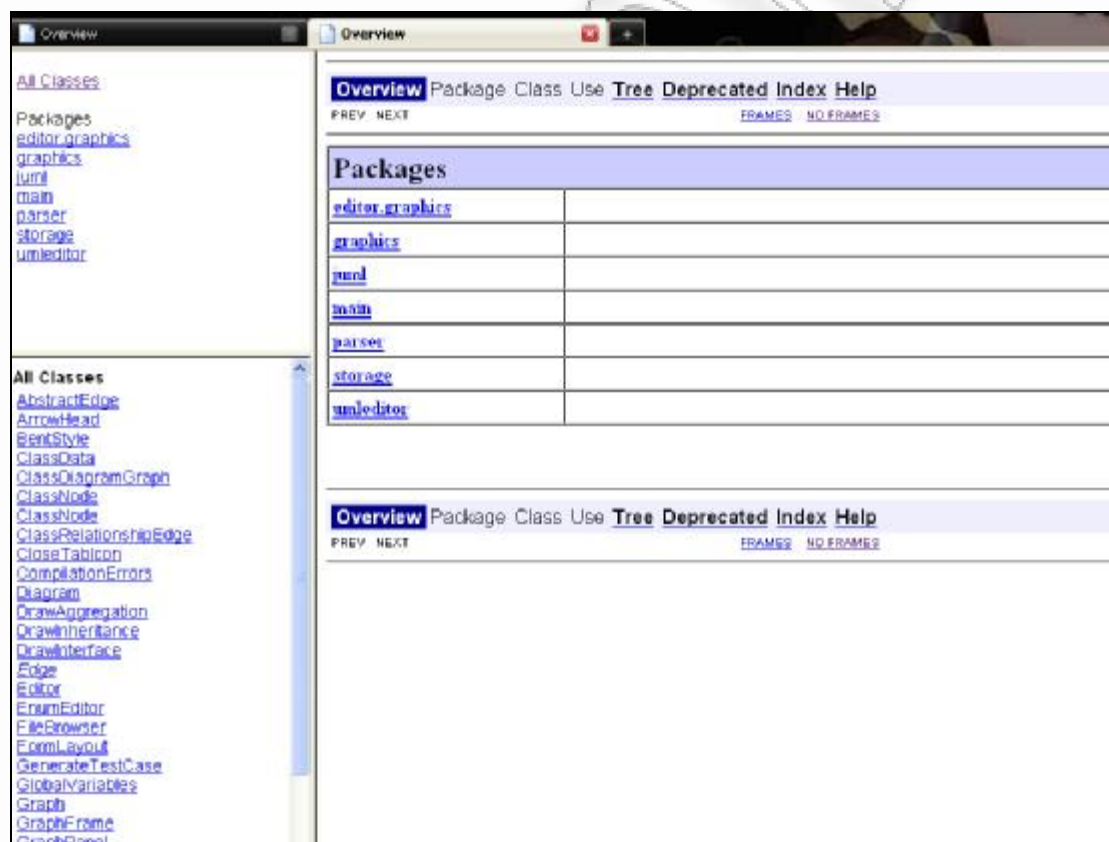
Το λογισμικό να είναι ανοικτής αρχιτεκτονικής, δηλαδή να έχει ανοικτό τον πηγαίο του κώδικα, για όποιον θέλει να τροποποιήσει, να ενισχύσει ή να αφαιρέσει χαρακτηριστικά από το λογισμικό.

### 4.3. Δομή

Το JEDUML διατίθεται σε ένα πακέτο το οποίο αποτελείται από τον πηγαίο κώδικα, τις βιβλιοθήκες που χρησιμοποιήθηκαν, το εκτελέσιμο του λογισμικού, σε μορφή jar και την τεκμηρίωση.

### 4.4. Τεκμηρίωση

Στον φάκελο doc βρίσκεται η τεκμηρίωση του πηγαίου κώδικα, οργανωμένη σαν Javadoc, για ευκολότερη πλοήγηση και ανάγνωση. Στην παρακάτω εικόνα φαίνεται η αρχική σελίδα της τεκμηρίωσης, το αρχείο Index.html. Από το κάθε frame μπορεί να υπάρξει πρόσβαση σε κάθε κλάση, οποιουδήποτε πακέτου.



Εικόνα 25. Αρχική σελίδα της τεκμηρίωσης

Overview Package <b>Class</b> Use Tree Deprecated Index Help	
<a href="#">PREV CLASS</a> <a href="#">NEXT CLASS</a>	<a href="#">FRAMES</a> <a href="#">NO FRAMES</a>
SUMMARY: <a href="#">NESTED</a>   <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>	DETAIL: <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>
<b>umeditor</b> <b>Class FormLayout</b> java.lang.Object └─umeditor.FormLayout <b>All Implemented Interfaces:</b> java.awt.LayoutManager  class <b>FormLayout</b> extends java.lang.Object implements java.awt.LayoutManager  A layout manager that lays out components along a central axis	
Field Summary	
private static int	<a href="#">GAP</a>
private int	<a href="#">height</a>
private int	<a href="#">left</a>
private int	<a href="#">right</a>
Constructor Summary	
<a href="#">FormLayout</a> ()	
Method Summary	
void	<a href="#">addLayoutComponent</a> (java.lang.String name, java.awt.Component comp)

Εικόνα 26. Τεκμηρίωση τυχαίας κλάσης

#### 4.5. Βιβλιοθήκες

Το JEDUML χρησιμοποιεί εκτός από τις βασικές βιβλιοθήκες της JAVA, οι οποίες φαίνονται στον φάκελο /lib, και τις βιβλιοθήκες xpp3 [13] και xstream[14].

Η [13] είναι μια βιβλιοθήκη που χρησιμοποιείται για το serialization XML σε αντικείμενα και αντίστροφα. Στο παρακάτω snippet, φαίνεται ένα μικρό παράδειγμα χρήσης.

```
package com.thoughtworks.xstream;
public class Cd {
    private String id;

    private Cd bonusCd;

    Cd(String id, Cd bonusCd) {
        this.id = id;
        this.bonusCd = bonusCd;
    }

    Cd(String id) {
        this.id = id;
    }

    public String getId() {
        return id;
    }
}
```

```
public Cd getBonusCd() {  
    return bonusCd;  
}  
}
```

Η [14] ή αλλιώς Xml Pull Parser είναι ένας streaming pull XML parser που διακρίνεται για τον γρήγορο και αποδοτικό χειρισμό των αντικειμένων εισόδου. Το Pull parsing είναι ένα επίπεδο αφάιρησης του χειρισμού XML αντικειμένων και συγκεκριμένα κρύβει από τον προγραμματιστή το tokenizing. Στο παρακάτω snippet φαίνεται ένα παράδειγμα χρήσης.

```
public Person parsePerson(XmlPullParser parser)  
    throws ValidationException, XmlPullParserException  
{  
    Person person = new Person();  
    while(true) {  
        int eventType = parser.nextTag();  
        if(eventType == XmlPullParser.START_TAG) {  
            String tag = parser.getStartTagName();  
            if("name".equals(tag)) {  
                if(person.name != null) {  
                    throw new ValidationException(  
                        "only one person name is allowed");  
                }  
                person.name = parser.nextText();  
            } else if("home_address".equals(tag)) {  
                if(person.homeAddress != null) {  
                    throw new ValidationException(  
                        "only one home address is allowed");  
                }  
                person.homeAddress = parseAddress(parser);  
            } else if("work_address".equals(tag)) {  
                if(person.workAddress != null) {  
                    throw new ValidationException(  
                        "only one work address is allowed");  
                }  
                person.workAddress = parseAddress(parser);  
            } else {  
                throw new ValidationException(  
                    "unknown field "+tag+" in person record");  
            }  
        } else if(eventType == XmlPullParser.END_TAG) {  
            break;  
        }  
    }  
    if(person.name == null) {  
        throw new ValidationException(  

```

```

        "person name is required");
    }
    return person;
}

```

#### 4.6. Πηγαίος κώδικας

Ο πηγαίος κώδικας είναι συγκεντρωμένος σε τρία πακέτα (packages). Το πρώτο είναι το src/juml, το άλλο είναι το src/editor και το τρίτο είναι το src/Tests.

Το src/Tests περιλαμβάνει HTML και flash αρχεία που υλοποιούν τα τέστ αξιολόγησης, για τα οποία γίνεται λόγος στην συνέχεια. Κάθε τέστ αποτελείται από ένα HTML αρχείο και ένα interactive flash. Στην δημιουργία τους συνέδραμε και το λογισμικό Quiz builder [15]. Στο παρακάτω snippet, παρουσιάζεται η δομή αυτών των αρχείων.

```

...
</head>

<body bgcolor="#F8F8F8">

<CENTER>

    <!-- the comment and script is used to workaround the KB912945 patch -->
    <!--[if gte IE 6]> <comment id="to_be_rewritten"> <![endif]-->

    <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#v
ersion=8,0,0,0" WIDTH="650" HEIGHT="480" id="quiz"> <PARAM NAME=movie
VALUE="Test1/Test1.swf">
    <PARAM NAME=quality VALUE=high>
    <PARAM NAME=base VALUE="Test1/">
    <PARAM NAME=bgcolor VALUE=#F8F8F8>
    <EMBED src="Test1/Test1.swf" quality=high base="Test1/" bgcolor=#F8F8F8
WIDTH="650" HEIGHT="480" NAME="quiz" ALIGN="" TYPE="application/x-shockwave-
flash" PLUGINS PAGE="http://www.macromedia.com/go/getflashplayer"> </EMBED>
    </OBJECT>

    </comment><script language="Javascript" src="Test1/ieupdate.js"></script>

</CENTER>

</body>
</html>

```

Το juml περιλαμβάνει όλες τις λειτουργίες του κυρίως προγράμματος. Περιλαμβάνει σε αντίστοιχα πακέτα τα λυμένα προβλήματα, τις ασκήσεις εξάσκησης και τα κεφάλαια της θεωρίας σε μορφή αρχείων HTML. Επίσης, περιλαμβάνει το γραφικό περιβάλλον για το οποίο θα γίνει λόγος ξεχωριστά. Τέλος, το πακέτο editor υλοποιεί το σχεδιαστή διαγραμμάτων, ο οποίος παρουσιάζεται παρακάτω.

#### 4.6.1. Η γραφική διεπαφή

Η γραφική διεπαφή με τον χρήστη είναι παραθυρική. Για αυτόν τον σκοπό χρησιμοποιήθηκαν JFrames. Το κυρίως παράθυρο της εφαρμογής υλοποιείται στα αρχεία JUMLView.java και JUMLApp.java. Παρακάτω φαίνονται δύο snippets από τα αντίστοιχα αρχεία.

```
import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
```

```
import java.awt.event.ActionEvent;
import java.awt.print.*;
import java.awt.event.ActionListener;
import javax.swing.Timer;
import javax.swing.Icon;
import javax.swing.JDialog;
import javax.swing.JButton;
import javax.swing.JFrame;
import sun.java2d.pipe.DrawImage;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.net*;
....
public class JUMLView extends FrameView {
....
}
```

Το πρόγραμμα κάνει χρήση του σχετικού API για παραθυρικές εφαρμογές. Κάθε επιλογή, συνδέεται με ένα συγκεκριμένο action, όπως αυτό που παρατίθεται που κάνει launch ένα problem section.

```
@Action
public void DisplayProblem2() {

    JTextPane tp = new JTextPane();
    JScrollPane js = new JScrollPane();
    js.getViewPort().add(tp);
    JFrame jf2 = new JFrame();
    jf2.getContentPane().add(js);
    jf2.pack();
    jf2.setSize(480,300);
    jf2.setVisible(true);

    try {
        URL url = getClass().getResource("Problems/Problem2.html");
```

```
    tp.setPage(url);
  }
  catch (Exception e) {
    e.printStackTrace();
  }
}
```

```
import org.jdesktop.application.Application;
import org.jdesktop.application.SingleFrameApplication;

/**
 * The main class of the application.
 */
public class JUMLApp extends SingleFrameApplication {

    /**
     * At startup create and show the main frame of the application.
     */
    @Override protected void startup() {
        show(new JUMLView(this));
    }

    /**
     * This method is to initialize the specified window by injecting resources.
     * Windows shown in our application come fully initialized from the GUI
     * builder, so this additional configuration is not needed.
     */
    @Override protected void configureWindow(java.awt.Window root) {
    }

    /**
     * A convenient static getter for the application instance.
     * @return the instance of JUMLApp
     */
    public static JUMLApp getApplication() {
        return Application.getInstance(JUMLApp.class);
    }

    /**
     * Main method launching the application.
     */
    public static void main(String[] args) {
        launch(JUMLApp.class, args);
    }
}
```



#### 4.6.2. Ο σχεδιαστής διαγραμμάτων

Ο σχεδιαστής διαγραμμάτων αποτελείται από πέντε πακέτα, κάθε ένα από τα οποία υλοποιεί ένα module του σχεδιαστή. Για περισσότερες λεπτομέρειες ως προς την ιεραρχία και την χρήση των κλάσεων, διατίθεται και η τεκμηρίωση του κώδικα σε μορφή javadocs, στον φάκελο doc/index.html.

Το πακέτο main υλοποιεί την γραφική διεπαφή του σχεδιαστή. Το αρχείο GUI.java για παράδειγμα σηκώνει το σχετικό JFrame μέσα στο οποίο απεικονίζονται όλα τα components.

Το πακέτο parser, υλοποιεί την δυνατότητα που έχει ο σχεδιαστής, όπως θα παρουσιαστεί παρακάτω να μετατρέπει ένα project σε πηγαίο κώδικα java.

Το πακέτο storage, υλοποιεί μεθόδους για την αποθήκευση των σχημάτων που δημιουργούνται, καθώς και του κώδικα που παράγεται αυτόματα.

Το πακέτο graphics, υλοποιεί τα σχήματα σε ανώτερο επίπεδο. Πρόκειται για ένα επίπεδο αφαίρεσης μεγαλύτερο από αυτό που υλοποιείται στο πακέτο umleditor.

Το πακέτο umleditor τελος, υλοποιεί το πεδίο σχεδίασης. Πρόκειται για ένα JFrame, το οποίο εξοπλίζουμε με την δυνατότητα σχεδίασης. Για αυτό το σκοπό υλοποιούνται κλάσεις σχημάτων, όπως φαίνεται σε αρχεία όπως τα ArrowHead.java, BentStyle.java και άλλα και μεθόδους διαδραστικής απεικόνισης τους, όπως στα formLayout.java, GraphFrame.java

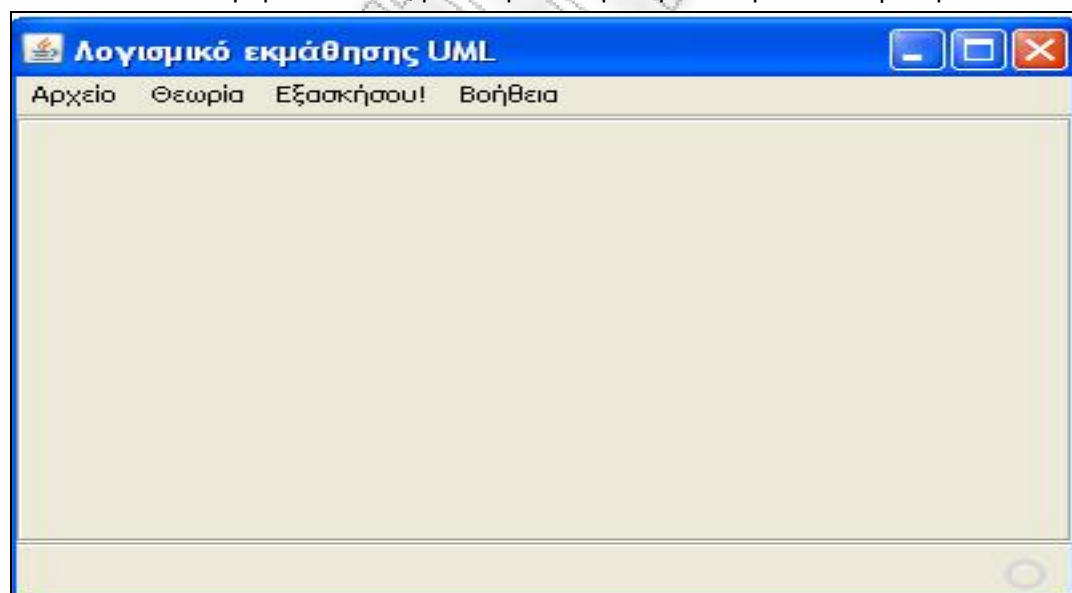
#### 4.7. Λειτουργία

Ο γενικός τρόπος που χρησιμοποιείται για την εκκίνηση του λογισμικού JEDUML είναι μέσω της γραμμής εντολών, σε οποιοδήποτε σύστημα. Η εντολή που χρησιμοποιούμε φαίνεται αμέσως παρακάτω:

```
Java -jar <μονοπάτι/για/το/JEDUML.jar>
```

Σε περιβάλλον Microsoft Windows, μπορούμε εναλλακτικά να εκτελέσουμε το Batch script, **DOUBLECLICKME.bat** και το JEDUML να εκτελεστεί αυτόματα.

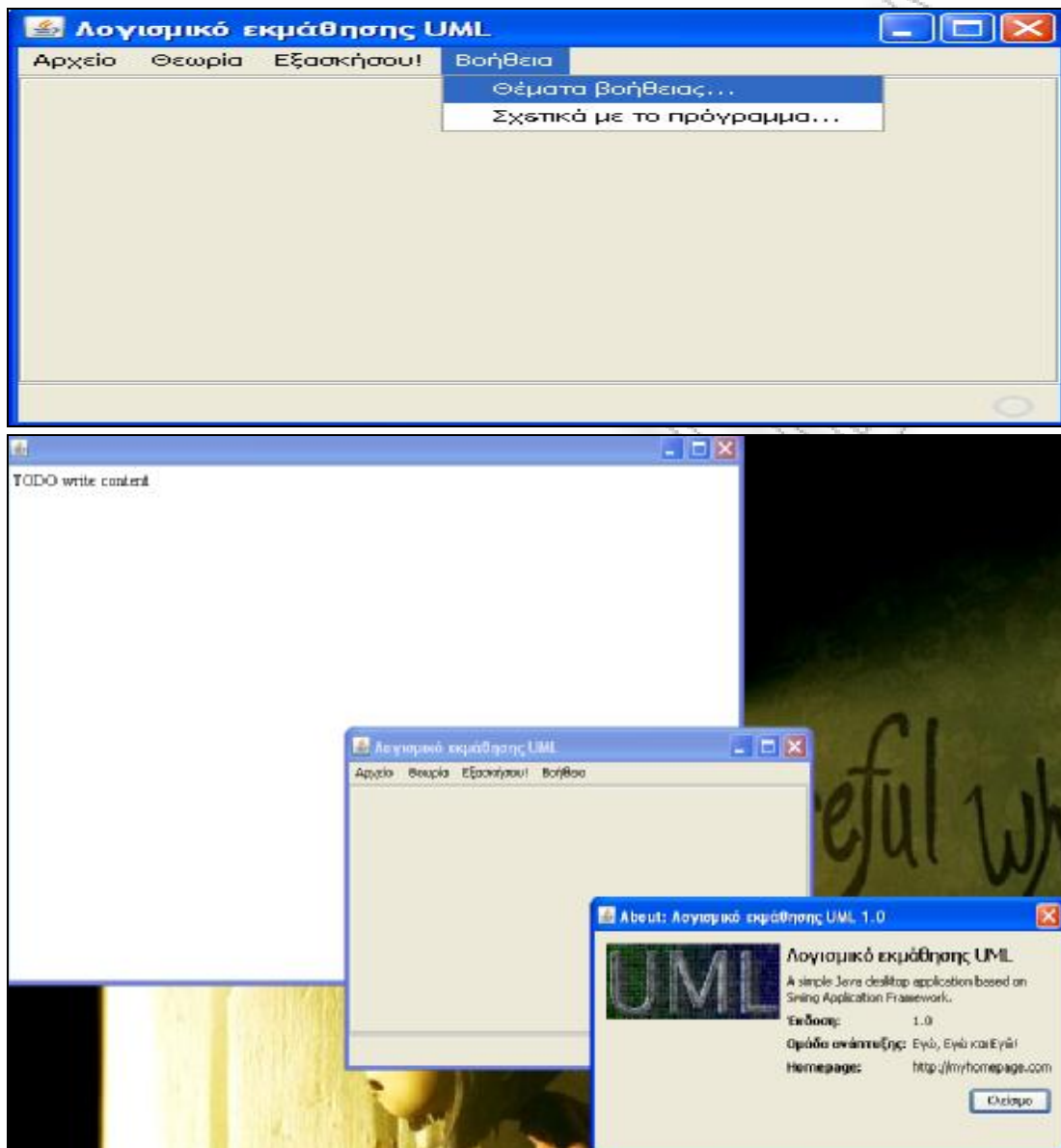
Όταν το λογισμικό εκκινείτε, βλέπουμε το παράθυρο που φαίνεται στην παρακάτω εικόνα.



Εικόνα 27. Αρχική σελίδα προγράμματος

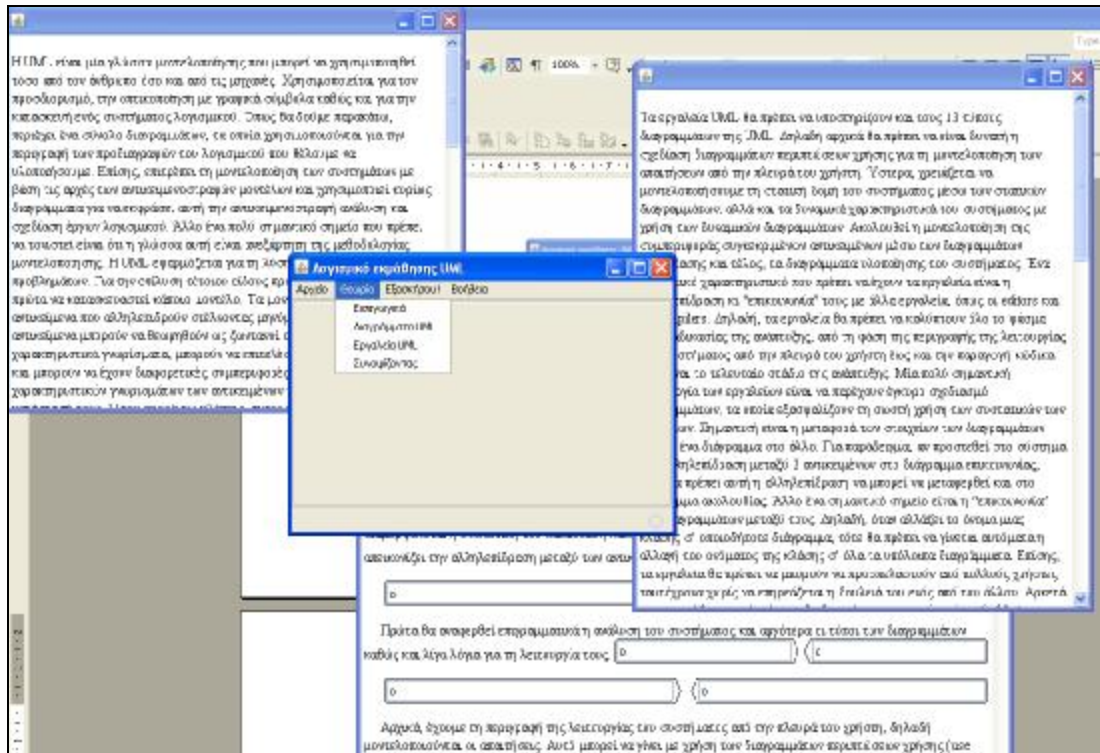
Ακόμα ο χρήστης δεν έχει κάνει κάποια επιλογή, οπότε το panel είναι κενό. Στην κορυφή του παραθύρου βλέπουμε τέσσερα menu items, με τις ονομασίες «Αρχείο», «Θεωρία», «Εξασκήσου!» και «Βοήθεια».

Στις παρακάτω εικόνες παρουσιάζονται τα περιεχόμενα του menu «Βοήθεια». Το πρώτο εμφανίζει ένα HTML section, οποίο μπορούμε να τοποθετήσουμε ένα help manual και το δεύτερο εμφανίζει διάφορες πληροφορίες για το πρόγραμμα καθώς και το λογότυπο του λογισμικού.



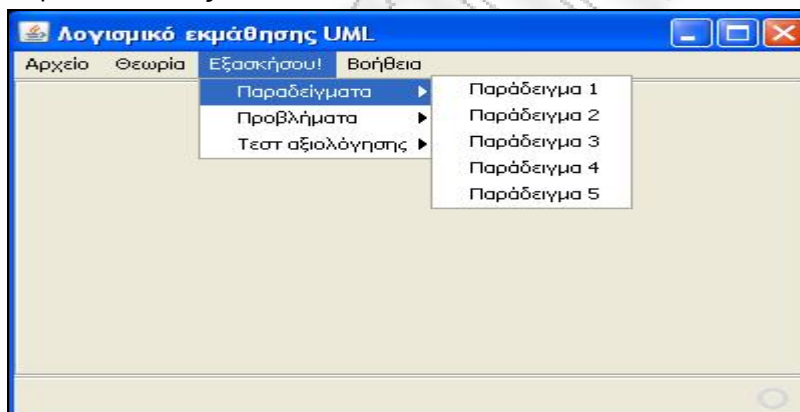
Εικόνα 29. Περιεχόμενα του menu «Βοήθεια»

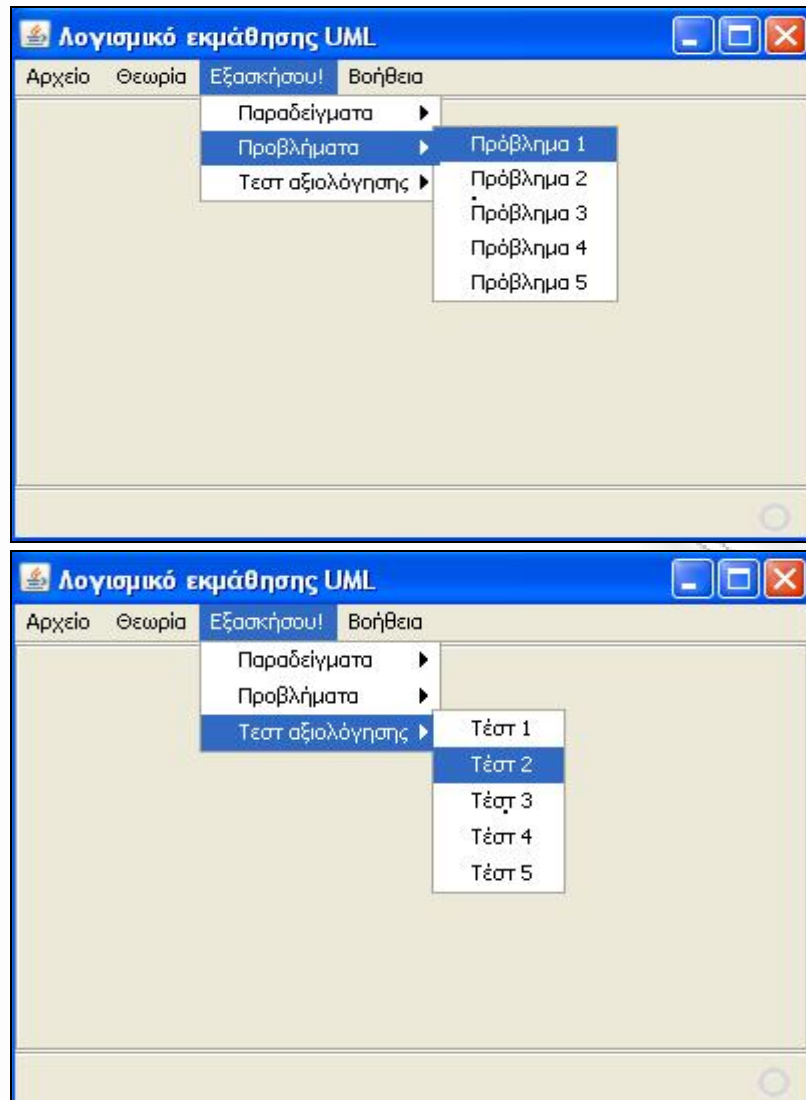
Το μενού «θεωρία» περιέχει τέσσερις επιλογές που κάθε μια εμφανίζει ένα section με αντίστοιχη θεωρία για την UML. Για παράδειγμα, η επιλογή «Εργαλεία» είναι ένα θεωρητικό κεφάλαιο που μιλάει για τα εργαλεία που χρησιμοποιούνται για την ανάπτυξη UML μοντέλων. Δείτε την εικόνα που ακολουθεί



Εικόνα 30. Περιεχόμενα του μενού «Θεωρία»

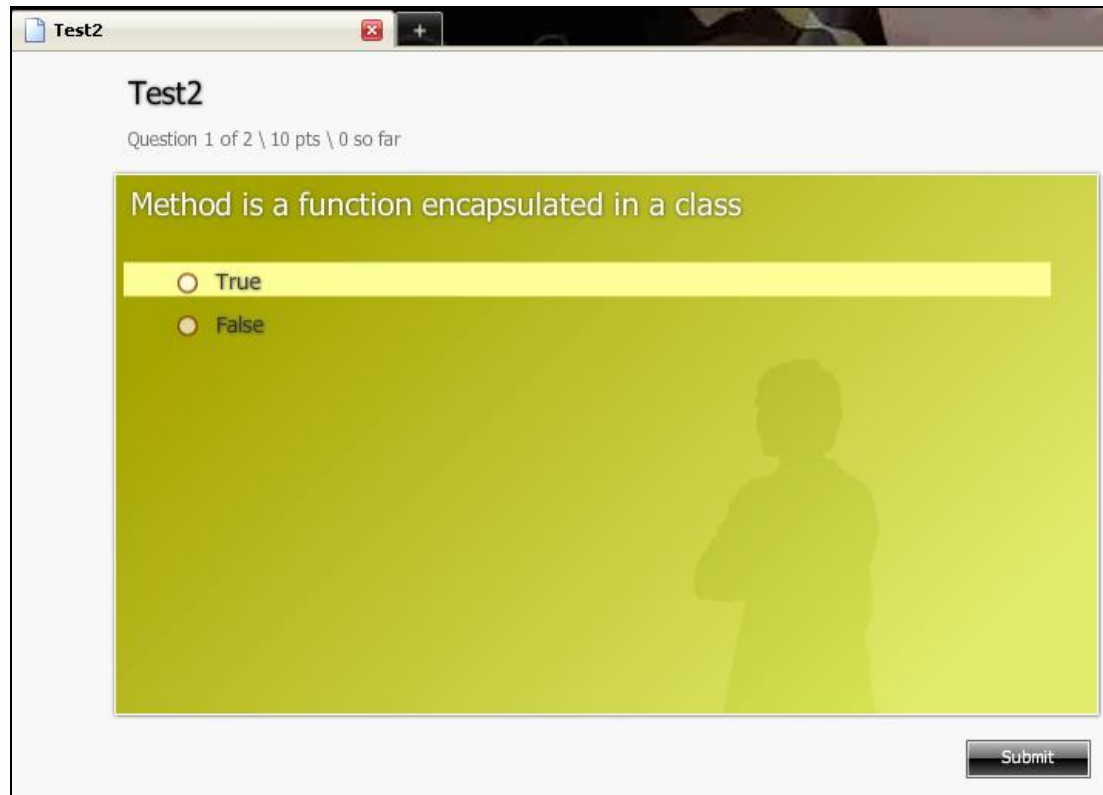
Στις παρακάτω εικόνες παρουσιάζονται τα περιεχόμενα του μενού «Εξασκήσου». Στην ενότητα αυτή υπάρχει υλικό για την εξάσκηση του χρήστη στις αντίστοιχες έννοιες που πραγματεύεται η θεωρία. Υπάρχουν τρεις τρόποι για αυτό. Είναι τα λυμένα παραδείγματα, οι άλυτες ασκήσεις, τις οποίες καλείται να λύσει ο χρήστης και τα τεστ αυτο-αξιολόγησης. Δείτε τις παρακάτω εικόνες.



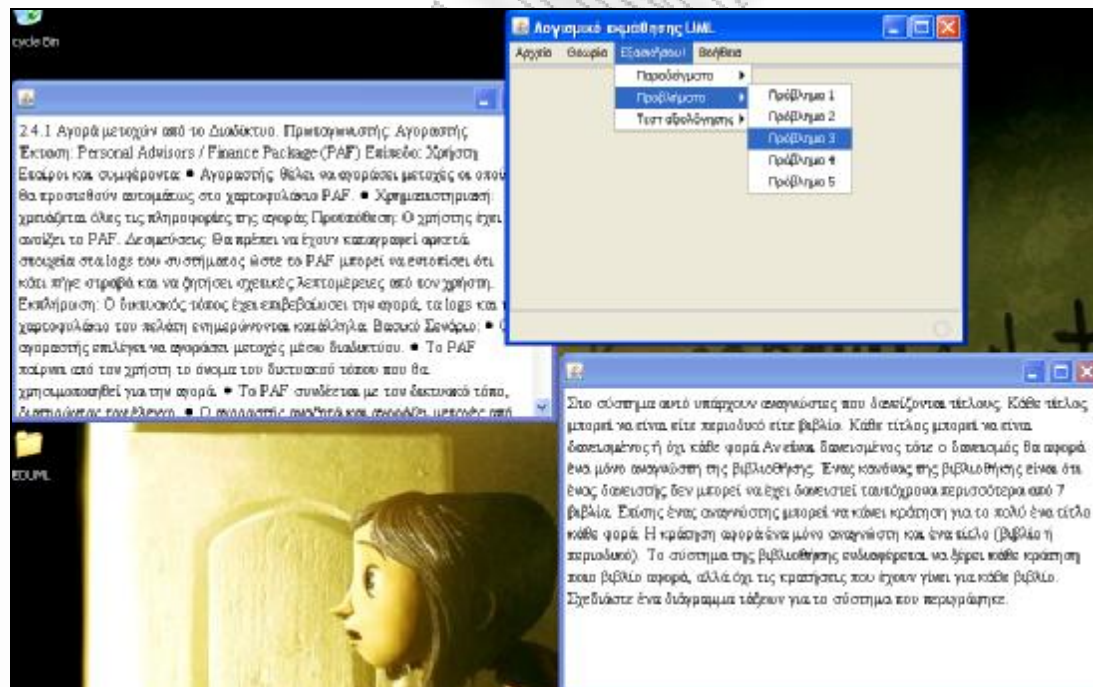


Εικόνα 31. Περιεχόμενα του menu «Εξασκήσου!»

Τα παραδείγματα είναι προβλήματα όπου παρατίθενται οι εκφωνήσεις και οι λύσεις τους, στα προβλήματα παρατίθενται μόνο οι προδιαγραφές και ο χρήστης καλείται να τα λύσει. Τα τεστ αυτοαξιολόγησης είναι τεστ πολλαπλής επιλογής όπου ο χρήστης απαντάει σε ερωτήσεις που έχουν να κάνουν με την θεωρία που διδάχτηκε. Δείτε τις παρακάτω εικόνες.



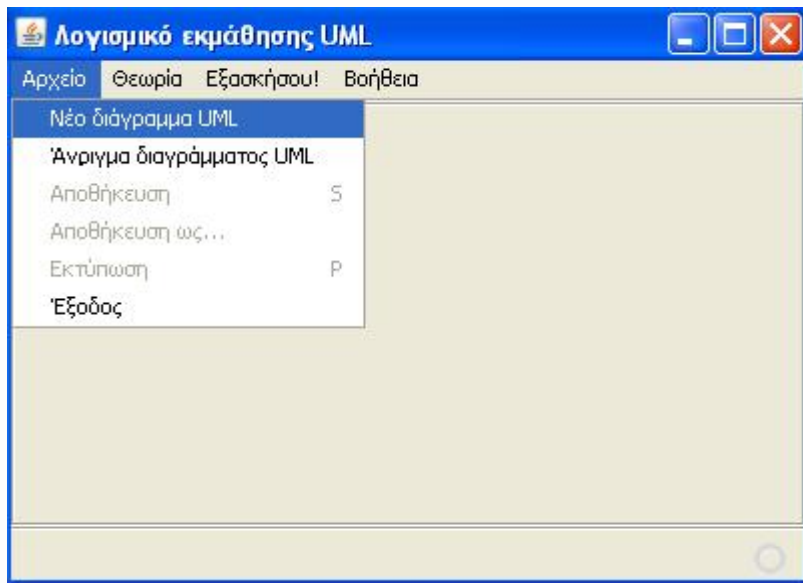
Εικόνα 32. Τεστ Αξιολόγησης



Εικόνα 33. Προβλήματα προς λύση

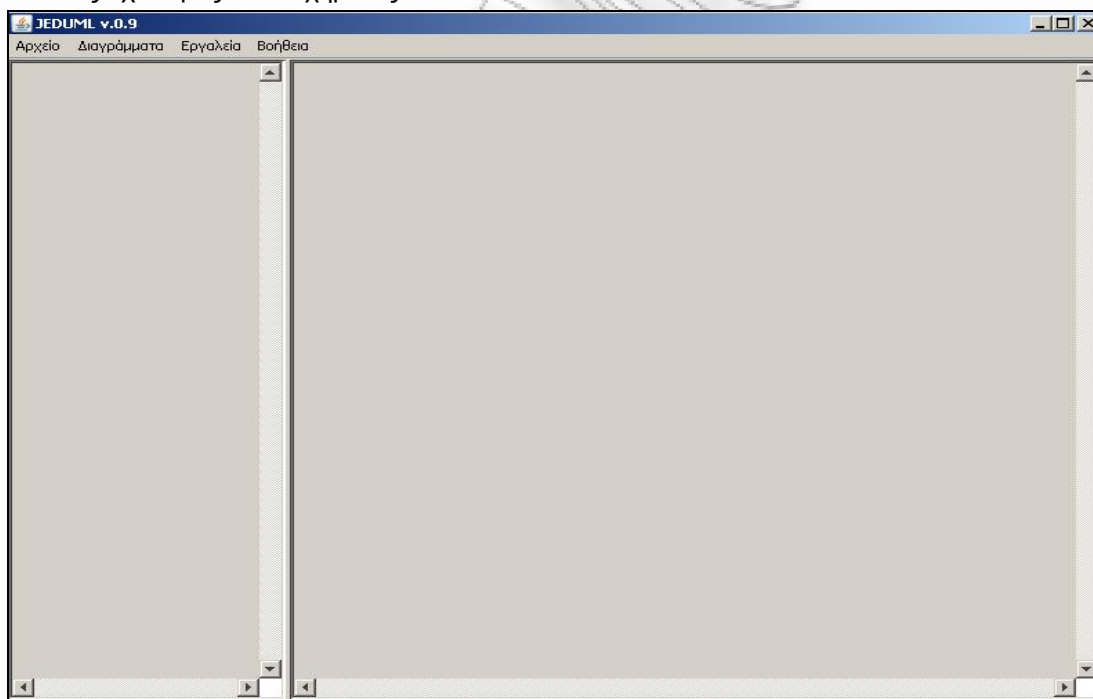
Τα προβλήματα προς λύση αλλά και κάθε άλλου είδους πρακτική εξάσκηση πάνω στην UML, μπορεί να γίνει με τον ειδικό UML editor που περιλαμβάνει το λογισμικό. Όπως προαναφέραμε, ο σκοπός ήταν μόνο να αναπτύξουμε ένα απλό τρόπο που να εξοικειώνει τον χρήστη με τα βασικά στοιχεία της UML και όχι η ανάπτυξη ενός full-featured UML CASE Tool. Από

το μενού «Αρχείο» επιλέγοντας είτε «Νέο διάγραμμα UML» είτε «Άνοιγμα διαγράμματος UML» ανοίγει ο editor, όπως φαίνεται στις παρακάτω εικόνες.



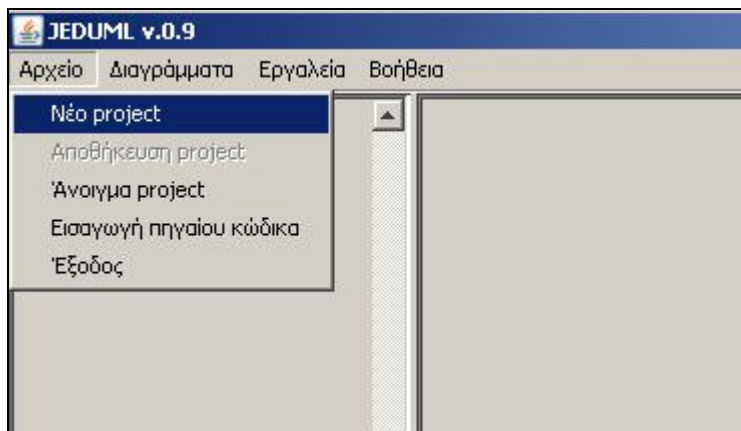
Εικόνα 34. Δημιουργία νέου διαγράμματος UML

Στην παρακάτω εικόνα φαίνεται η επιφάνεια του editor, χωρίς να έχει γίνει καμία δουλειά, κανένας σχεδιασμός UML σχήματος.



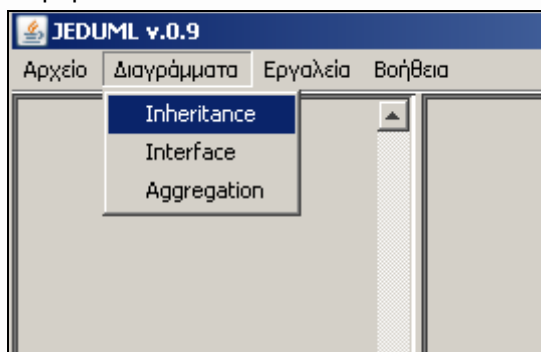
Εικόνα 35. Η επιφάνεια του editor

Στις παρακάτω εικόνες φαίνονται πιο αναλυτικά οι διαθέσιμες επιλογές. Στην επιλογή «αρχείο» οι διαθέσιμες επιλογές είναι η δημιουργία νέου σχήματος, το άνοιγμα ενός υπάρχοντος ( σε μορφή XML, όπως θα δούμε παρακάτω ) και η εισαγωγή πηγαίου κώδικα σε Java.



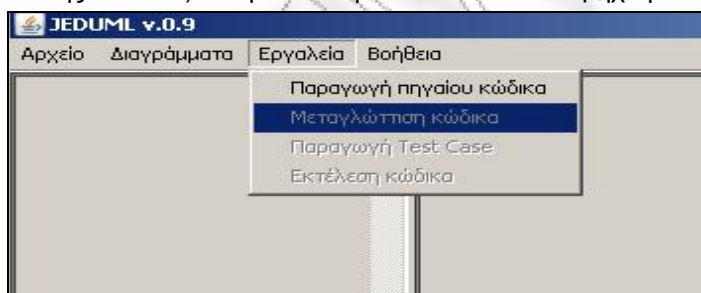
**Εικόνα 36. Δημιουργία νέου project**

Στην παρακάτω εικόνα παρουσιάζονται οι επιλογές διαγραμμάτων. Υπάρχει επιλογή για διάγραμμα κληρονομικότητας, διεπαφής και συνδυασμού. Οι επιλογές αυτές βρίσκονται σε πειραματικό στάδιο.



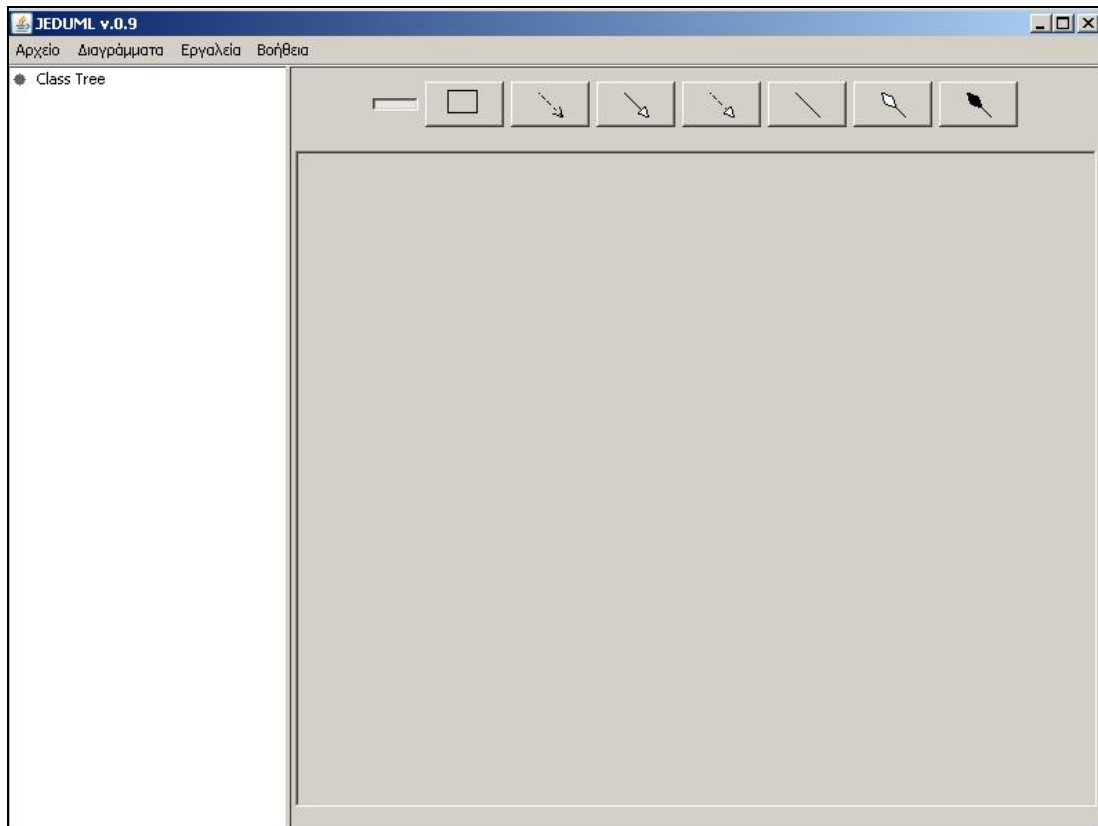
**Εικόνα 37. Επιλογές διαγραμμάτων**

Στην παρακάτω εικόνα φαίνονται τα διάφορα παρεχόμενα εργαλεία στα οποία συγκαταλέγονται η παραγωγή πηγαίου κώδικα σε Java, η μεταγλώττιση, η παραγωγή Test Case ( επίσης σε Java ) και η εκτέλεση του κώδικα που παρήχθη.



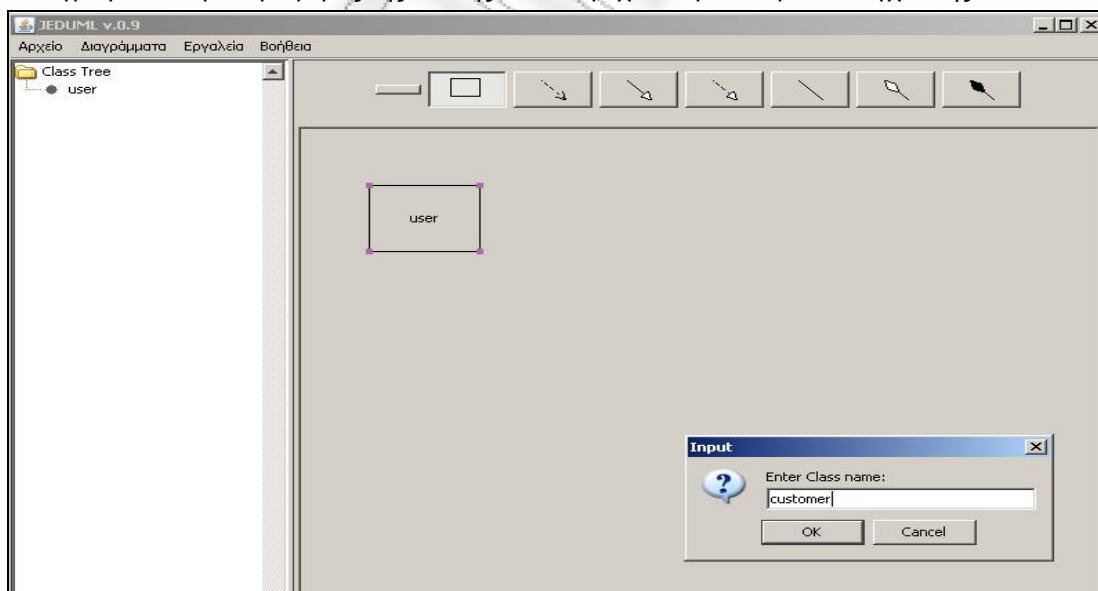
**Εικόνα 38. Παρεχόμενα εργαλεία**

Στην παρακάτω εικόνα φαίνεται η επιφάνεια σχεδίασης ενός καινούριου UML διαγράμματος.



**Εικόνα 39. Επιφάνεια σχεδίασης UML διαγράμματος**

Εδώ φαίνεται η δημιουργία μιας καινούριας κλάσης. Αυτόματα το λογισμικό μας ζητάει να ορίσουμε το όνομα της. Αμέσως μετά, η κλάση θα απεικονιστεί σαν ένα τετράγωνο και θα καταγραφεί στο αριστερό μέρος της οθόνης, που παρέχει ακόμα ένα μέσο ελέγχου της.



**Εικόνα 40. Δημιουργία νέας κλάσης**

Επίσης, με τη χρήση κάποιου από τα βέλη που διατίθενται μπορούμε να ενώσουμε τις κλάσεις που έχουμε δημιουργήσει. Η επεξήγηση των βελών παρατίθεται παρακάτω.





Εισαγωγή(Import). Σημαίνει ότι ένα πακέτο εισάγει όλα τα στοιχεία (κλάσεις κλπ) ενός άλλου πακέτου.



Γενίκευση(Generalization). Το αντικείμενο (στην αρχή του βέλους) κληρονομεί μέρος των χαρακτηριστικών του από ένα «γενικό» αντικείμενο (στο τέλος του βέλους).



Πραγμάτωση(Realization). Η κλάση υλοποιεί τις μεθόδους και τα μέλη που έχει ορίσει ένα interface.



Συσχέτιση(Association). Μια γενική συσχέτιση μεταξύ των μελών δύο αντικειμένων.

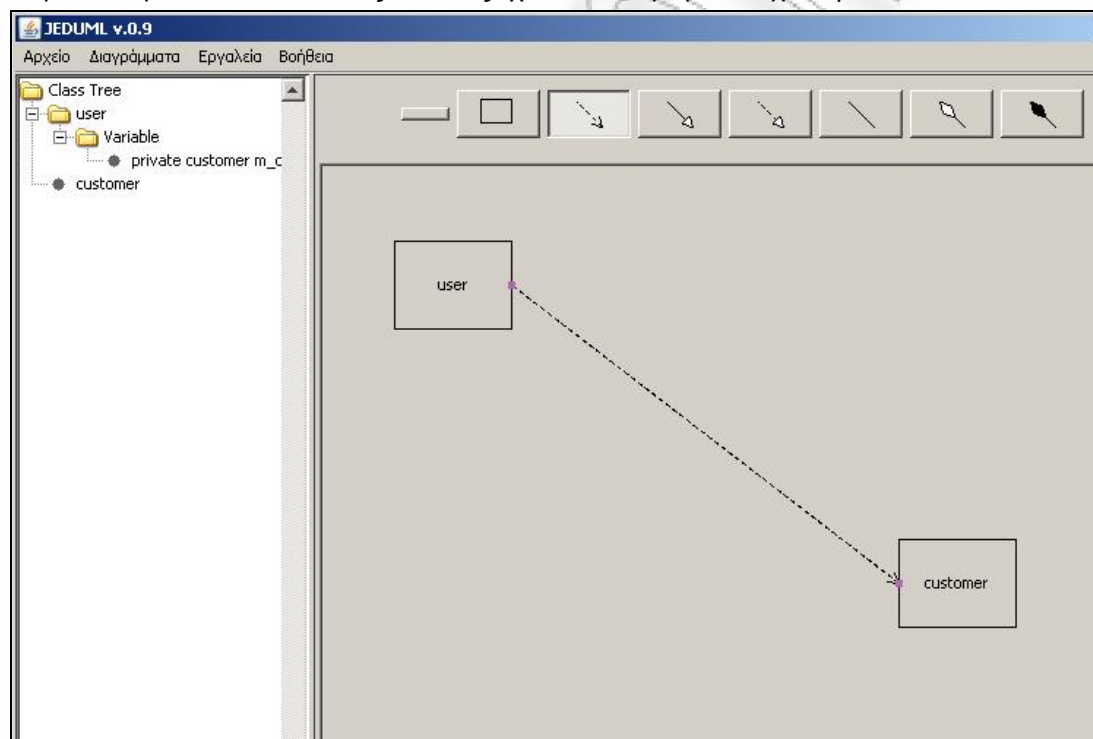


Συνένωση(Aggregation). Δηλώνει διαμοιραζόμενη ιδιοκτησία ownership κάποιων αντικειμένων πάνω σε ένα άλλο.



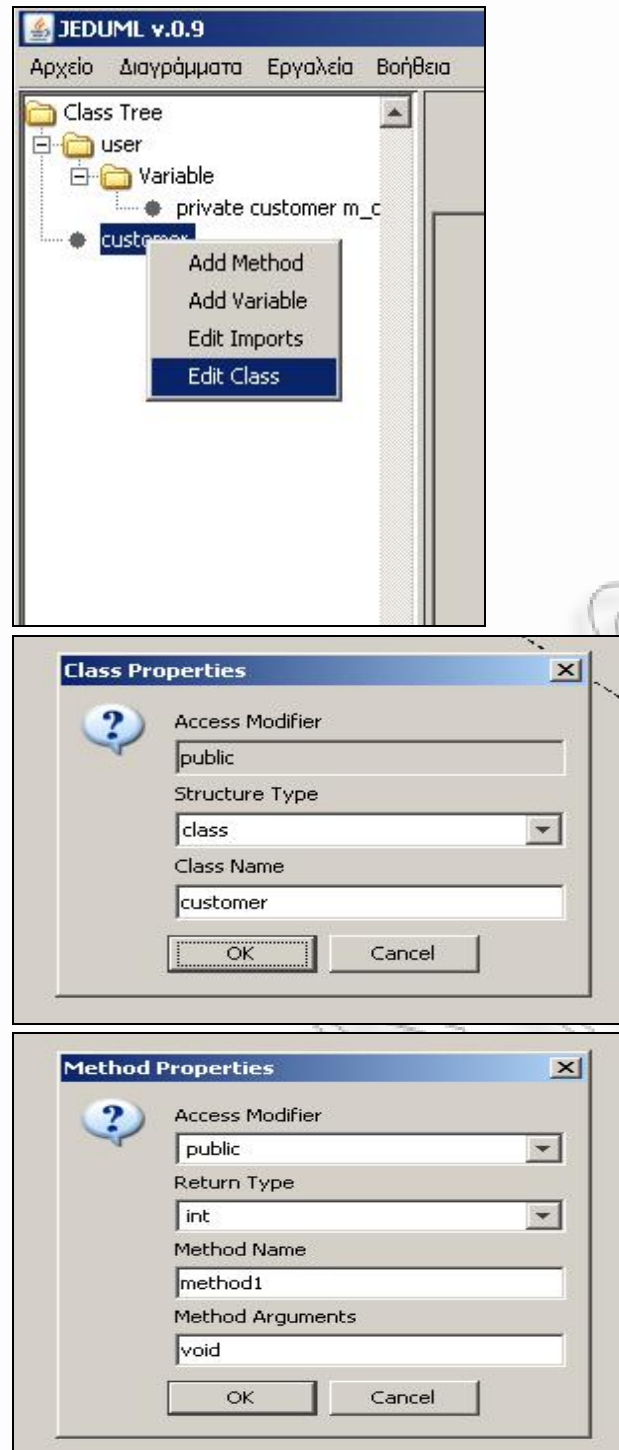
Σύνθεση(Composition). Κάποια αντικείμενα συναποτελούν ένα άλλο αντικείμενο.

Παρακάτω φαίνονται δύο κλάσεις οι οποίες έχουν ενωθεί με μια συσχέτιση.

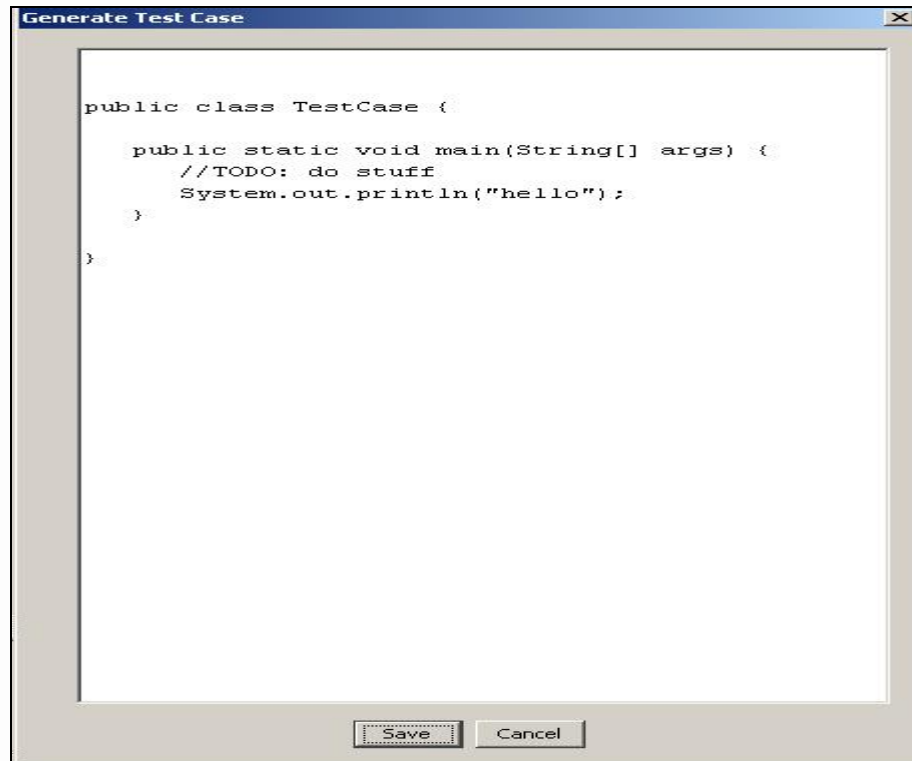


Εικόνα 41. Ένωση δύο κλάσεων με μια συσχέτιση

Εδώ μπορούμε να δούμε πως μπορούμε να ελέγξουμε ό,τι σχεδιάζουμε στο αριστερό μέρος του παραθύρου του σχεδιαστή. Βλέπουμε ότι μπορούμε να προσθέτουμε μεθόδους, μεταβλητές και άλλα στοιχεία. Στις επόμενες εικόνες βλέπουμε και με ποιον τρόπο μπορεί να γίνει αυτό.



Εικόνα 42. Προσθήκη μεθόδου και μεταβλητής στην κλάση



Εικόνα 43. Παραγωγή test case

Στο παρακάτω snippet μπορούμε να δούμε πως δομείται ένα XML αρχείο που χρησιμοποιείται για να αποθηκεύσει ένα project.

```
<umleditor.ClassDiagramGraph>  
<nodes>  
  <storage.ClassNode serialization="custom">  
    <umleditor.RectangularNode>  
      <default/>  
      <double>49.0</double>  
      <double>51.0</double>  
      <double>80.0</double>  
      <double>60.0</double>  
    </umleditor.RectangularNode>  
  <storage.ClassNode>  
    <default>  
      <botHeight>0.0</botHeight>  
      <midHeight>0.0</midHeight>  
    <extend/>  
    <includes>  
      <text></text>  
      <justification>1</justification>  
      <size>4</size>  
      <underlined>false</underlined>  
    </includes>  
  </interfaces/>
```

```

<methods/>
<name>
  <text>user</text>
  <justification>1</justification>
  <size>4</size>
  <underlined>false</underlined>
</name>
<showName>
  <text>user</text>
  <justification>1</justification>
  <size>4</size>
  <underlined>false</underlined>
</showName>
<type>
  <text>class</text>
  <justification>1</justification>
  <size>4</size>
  <underlined>false</underlined>
</type>
<variables>
  <storage.Variable>
    <mods>private</mods>
    <type>customer</type>
    <name>m_customer</name>
  </storage.Variable>
</variables>
</default>
</storage.ClassNode>
</storage.ClassNode>
<storage.ClassNode serialization="custom">
  <umleditor.RectangularNode>
    <default/>
    <double>394.0</double>
    <double>255.0</double>
    <double>82.0</double>
    <double>60.0</double>
  </umleditor.RectangularNode>
  <storage.ClassNode>
    <default>
      <botHeight>0.0</botHeight>
      <midHeight>0.0</midHeight>
    <extend/>
    <includes>
      <text></text>
    </includes>
  </storage.ClassNode>
  <justification>1</justification>

```

```
<size>4</size>  
<underlined>false</underlined>  
</includes>  
<interfaces/>  
<methods>  
  <storage.Method>  
    <mods>public</mods>  
    <type>int</type>  
    <name>method1</name>  
    <src>{  
.....
```

## Συμπεράσματα - Περίληψη

Οι συνεχώς μεταβαλλόμενες ανάγκες στο χώρο της εκπαίδευσης καθώς και η εξέλιξη στον τομέα της πληροφορικής καθιστούν επιτακτική την ανάγκη ανάπτυξης εκπαιδευτικού λογισμικού. Παραπάνω είδαμε το χώρο του εκπαιδευτικού λογισμικού, την εξέλιξη και την επίδρασή του στο χώρο της εκπαίδευσης καθώς και τεχνολογίες που μπορούν να χρησιμοποιηθούν για την ανάπτυξη τέτοιου είδους προγραμμάτων. Επιπλέον, είδαμε το λογισμικό JEDUML το οποίο είναι το εκπαιδευτικό λογισμικό που αναπτύχθηκε για την εκμάθηση της UML. Οι προτάσεις βελτίωσης οι οποίες προκύπτουν από την ανάλυση των παραπάνω ενοτήτων σε σχέση με το εκπαιδευτικό λογισμικό και το λογισμικό JEDUML είναι οι ακόλουθες:

- Το εκπαιδευτικό λογισμικό JEDUML θα μπορούσε να γίνει πιο προσαρμοστικό για να ανταποκρίνεται στις ανάγκες και το επίπεδο του κάθε χρήστη
- Θα μπορούσε να ενσωματωθεί ένα είδος wizard το οποίο να καθοδηγεί τον άπειρο χρήστη στην χρήση της εφαρμογής
- Θα μπορούσε να υπάρχει ένας μηχανισμός όπου όταν ο χρήστης σχεδιάζει κάτι να παρέχονται πληροφορίες, παρμένες από την θεωρία, σχετικά με το τι είναι αυτό που σχεδιάζει
- Θα μπορούσε να υπάρχει ένας χώρος όπου οι χρήστες να φτιάχνουν projects συνεργατικά ( πχ. web-based, σε module της εφαρμογής)
- Θα μπορούσε να προστεθεί περισσότερο εκπαιδευτικό υλικό και να βελτιωθεί η πλοήγηση σε αυτό.

## Επίλογος

Η ανάπτυξη και εξάπλωση των τεχνολογιών πληροφορικής και επικοινωνιών (ΤΠΕ) δεν θα μπορούσε να αφήσει ανεπηρέαστο και τον χώρο της εκπαίδευσης κάθε είδους και βαθμίδας. Έτσι δημιουργήθηκε το εκπαιδευτικό λογισμικό (educational software) και η εκπαίδευση με χρήση υπολογιστή (computer driven education). Στην παρούσα εργασία σχεδιάσαμε, αναπτύξαμε και αξιολογήσαμε ένα λογισμικό εκπαιδευτικού σκοπού, που αφορά της εκμάθησης της Unified Modelling Language (UML).

## ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

- [1] Χ. Παναγιωτακόπουλος, Χ.Πιερρακέας, Π.Πιντέλας, «Το εκπαιδευτικό λογισμικό και η αξιολόγησή του», 2003.
- [2] Wikipedia, 2010, <http://el.wikipedia.org/wiki/Wiki> .
- [3] Java Hellenic User Group, 2010, <http://www.jhug.gr/> .
- [4] Java, επίσημη ιστοσελίδα, 2010, <http://www.java.com/en/> .
- [5] Οδηγίες κατασκευής ιστοσελίδων με HTML, 2010, <http://www.eeei.gr/odhgos/htmlfaq.htm> .
- [6] Μαθήματα HTML, 2010, <http://www.it.uom.gr/project/html2/Lessons.html> .
- [7] EAITY, Πρόγραμμα «Οδύσσεια», <http://odysseia.cti.gr/> .
- [8] Geometer's Sketchpad, επίσημη ιστοσελίδα, [www.dynamicgeometry.com/](http://www.dynamicgeometry.com/) .
- [9] Modellus, επίσημη ιστοσελίδα, <http://modellus.fct.unl.pt/> .
- [10] <http://www.uml.org/>.
- [11] ACM portal, <http://www.acm.org/>.
- [12] Dirk Ohst, Michael Welle, Udo Kelter, (2003) Differences between Versions of UML Diagrams.
- [13] Brian Dobing, Jeffrey Parsons, (2006) How UML is used.
- [14] Quiz builder, επίσημη ιστοσελίδα, <http://www.quiz-builder.com/>