

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διδακτική του Προγραμματισμού για μαθητές Γυμνασίου

με χρήση του εργαλείου γραφικού
προγραμματισμού **Scratch**

Γιώργος Παπαευθυμίου ΜΕ 0688

22 Σεπτεμβρίου 2008

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Περιεχόμενα

Ευχαριστίες	7
Περίληψη	8
Εισαγωγή.....	9
Η Πληροφορική ως αντικείμενο και ως εργαλείο μάθησης.....	11
Η Πληροφορική ως εργαλείο μάθησης.....	11
Η Πληροφορική ως αντικείμενο μάθησης.....	12
Σκοπός της διπλωματικής	14
Δομή της παρούσας εργασίας.....	15
Κεφάλαιο 1 - Πλαίσιο σπουδών.....	17
Η Πληροφορική στο ελληνικό σχολείο – Ιστορικά στοιχεία.....	17
Πώς εντάσσεται σήμερα η διδασκαλία Πληροφορικής στο αναλυτικό πρόγραμμα δημοτικού-γυμνασίου	18
Οι ΤΠΕ στην πρωτοβάθμια εκπαίδευση.....	18
Οι ΤΠΕ στο γυμνάσιο	20
Η διδασκαλία Πληροφορικής ως γνωστικό αντικείμενο στο αναλυτικό πρόγραμμα δημοτικού-γυμνασίου	23
Κεφάλαιο 2 - Βασικές θεωρίες μάθησης και Πληροφορική	26
Skinner, οι θεωρίες μάθησης στον σχεδιασμό εκπαιδευτικού λογισμικού	26
Piaget, οι εφαρμογές του εποικοδομητισμού στην μάθηση μέσω του υπολογιστή.....	27
Εφαρμογές του εποικοδομητισμού στη μάθηση μέσω του ηλεκτρονικού υπολογιστή.....	29

Vygotsky, ζώνη της επικείμενης ανάπτυξης και η εφαρμογή της στη μάθηση μέσω του υπολογιστή.....	32
Εφαρμογές του κοινωνικοπολιτιστικού εποικοδομητισμού του Vygotsky στη διδασκαλία με υπολογιστή.....	34
Ο εποικοδομητισμός του Bruner.....	36
Εφαρμογές των απόψεων του Bruner στην διδασκαλία της Πληροφορικής.....	37
Τεχνοκεντρικές διδακτικές προσεγγίσεις	38
Δυσκολίες του τεχνοκεντρικού μοντέλου	41
Κεφάλαιο 3 - Διδακτική του προγραμματισμού.....	43
Μεταβλητή	45
Δομές ελέγχου	53
Δομές επανάληψης.....	58
Η έννοια της αναδρομικής διαδικασίας	63
Κεφάλαιο 4 - Προσεγγίσεις διδασκαλίας του προγραμματισμού και προγραμματιστικά περιβάλλοντα.....	67
Η κλασική προσέγγιση διδασκαλίας	67
Προγραμματιστικά περιβάλλοντα για αρχάριους	69
Τροποποίηση εμπορικών γλωσσών.....	70
Συστήματα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων.....	72
Περιβάλλοντα εναλλακτικών μορφών αναπαράστασης της σύνταξης εντολών.....	76
Περιβάλλοντα δημιουργίας προγραμμάτων μέσω μικρόκοσμων	85
Δυναμική ηχητική προσομοίωση εκτέλεσης προγραμμάτων (software auralisation)	88

Πίνακες ταξινόμησης των διδακτικών εργαλείων ανάλογα με τα χαρακτηριστικά τους	91
Κεφάλαιο 5 - Το περιβάλλον γραφικού προγραμματισμού Scratch	95
Αρχές σχεδιασμού	96
Βασικά χαρακτηριστικά	97
Υλοποίηση	99
Τρία τυπικά σενάρια χρήσης.....	99
Τα εκπαιδευτικά πλεονεκτήματα του Scratch	101
Δυνατότητες εφαρμογής στην εκπαίδευση.....	102
Περιγραφή του εργαλείου	103
Τα βασικά συστατικά ενός προγράμματος στο Scratch	103
Η κεντρική οθόνη του Scratch.....	104
Σύνοψη	112
Κεφάλαιο 6 - Ένα πρωτότυπο σενάριο.....	113
Θεωρία σύνταξης σεναρίων.....	113
Η δομή ενός σεναρίου	114
Προδιαγραφές σχεδιασμού δραστηριοτήτων	117
Σενάριο «Η δομή ελέγχου ΕΠΙΛΟΓΗ»	120
Κεφάλαιο 7 – Επισκόπηση – επίλογος	129
Επισκόπηση του εργαλείου.....	129
Αξιολόγηση του σεναρίου	130
Επεκτάσεις.....	132
Κατάλογος σχημάτων.....	133
Παράρτημα	134
Μετάφραση εγγράφων που σχετίζονται με το Scratch.....	134

Βιβλιογραφία 135

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Ευχαριστίες

Θεωρώ χρέος μου να εκφράσω τις ειλικρινείς ευχαριστίες μου στους διδάσκοντες-μέλη της τριμελούς επιτροπής κ. Σ. Ρετάλη, κ. Μ. Θεμιστοκλέους και κ. Δ. Σάμψων, που είχαν την επίβλεψη της παρούσας εργασίας και βοήθησαν στην υλοποίησή της. Ιδιαίτερες ευχαριστίες οφείλω να αποδώσω στον κ. Σ. Ρετάλη για την πολύτιμη αρωγή του, την καθοδήγηση, την εξαιρετικά θετική του διάθεση και το ενδιαφέρον που επέδειξε.

Θα αποτελούσε παράλειψη να μην αναφερθώ στη συνδρομή όλων των διδασκόντων του προγράμματος στην αναβάθμιση των γνώσεων μου στην Πληροφορική, αλλά και στους συμφοιτητές μου για τη δημιουργική συνεργασία μας.

Στη διπλωματική εργασία, ως καταληκτική του μεταπτυχιακού προγράμματος σπουδών και της φοιτητικής μου σταδιοδρομίας, δράττομαι της ευκαιρίας να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση και την αμέριστη υποστήριξή της στις προσπάθειές μου.

Περίληψη

Στην εργασία αυτή παρουσιάζεται μια εναλλακτική προσέγγιση εισαγωγής στον προγραμματισμό η οποία βασίζεται στην ιδέα μίας γραφικής γλώσσας προγραμματισμού που έχει ως στόχο να παρακάμψει τα προβλήματα και περιορισμούς της κλασικής μεθόδου (*classic approach*) διδασκαλίας του προγραμματισμού. Πρόκειται για το προγραμματιστικό περιβάλλον Scratch (scratch.mit.edu), το οποίο χρησιμοποιείται ως όχημα για τη διδασκαλία βασικών εννοιών του προγραμματισμού και της Πληροφορικής.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ

Εισαγωγή

2009, 60 χρόνια από τη δημιουργία του πρώτου ηλεκτρονικού υπολογιστή και η παρουσία του πια, είναι άρρηκτα συνδεδεμένη με τη καθημερινότητα των περισσότερων ανθρώπων του πλανήτη. Μιλώντας για την επιστήμη της Πληροφορικής, φυσικά δεν αναφερόμαστε μόνο στην κατασκευή του ηλεκτρονικού υπολογιστή αλλά σε έναν ευρύτερο τρόπο σκέψης, σύνθεσης και ανάλυσης πληροφοριών. Οι σύγχρονοι ρυθμοί ζωής και καθημερινότητας χαρακτηρίζονται από μια υπερπροσφορά πληροφοριών, ερεθισμάτων, εικόνων αλλά και υπερβολικών εξωγενών απαιτήσεων και εσωτερικών αναγκών. Ο άνθρωπος προκειμένου να καταφέρει να ανταποκριθεί στη σύγχρονη αυτή πραγματικότητα δημιούργησε μηχανές ώστε να ξεπεραστούν οι δικές του αδυναμίες. Τα μηχανικά αυτά συστήματα που κατά κανόνα ονομάζονται ηλεκτρονικοί υπολογιστές ακολουθούν σχεδόν σε κάθε ανθρώπινη έκφραση και ανάγκη.

- Η **επικοινωνία**, πρωταρχική ανάγκη και επιθυμία του κάθε ανθρώπου διευκολύνεται ιδιαίτερα με τη χρήση του ηλεκτρονικού ταχυδρομείου και των διαδικτυακών χώρων συζήτησης και επαφής. Η εκμηδένιση των αποστάσεων φέρνει τους ανθρώπους κοντά άμεσα.
- Η ανάγκη όλων για **γνώση** βρίσκει το διέξοδο της στο διαδίκτυο όπου κάθε δυνατή απορία μπορεί να απαντηθεί με επιστημονική πληρότητα και σαφήνεια.
- Η έλλειψη **χρόνου** μπορεί να νικηθεί αφού ο ηλεκτρονικός υπολογιστής έχει την ικανότητα να φέρνει σε πέρας χρονοβόρες δραστηριότητες ιδιαίτερα σύντομα και αποτελεσματικά.
- Η προσφορά του ηλεκτρονικού υπολογιστή (Η.Υ), είναι **καθολική, ανεξάρτητη** από διακρίσεις κοινωνικού, οικονομικού, πολιτισμικού επιπέδου. Ο υπολογιστής προσφέρει στο χρήστη του ό,τι ζητηθεί δίχως κριτική απαλλάσσοντας τον χειριστή από αισθήματα μειονεξίας ή φόβου τυχόν απόρριψης.

- Η δυνατότητα **οργάνωσης** των αναγκών μας μέσω του Η.Υ χαρίζει μεγαλύτερη αποτελεσματικότητα στη καθημερινότητά μας.

Αυτά και άλλα που θα αναφερθούν σταδιακά στη πορεία της εργασίας οδήγησαν στην καθολική εξάπλωση του Η.Υ σε κάθε σπίτι, εργασιακό χώρο, σχολείο, χώρο διασκέδασης.

Σε ένα ιδεατό κόσμο ο άνθρωπος θα κρατούσε τη δύναμη του ακέραιη χρησιμοποιώντας τα επιτεύγματά του ως συμπληρωματικά βοηθήματα στη ζωή του χαρίζοντας στο κόσμο του, νέους ορίζοντες γνώσεων και εμπειριών. Δυστυχώς σήμερα ο μέσος πολίτης συχνά γίνεται θύμα των τεχνολογικών «επιτευγμάτων», θέτοντας τον εαυτό του σε θέση παθητική απέναντι στη «βούληση» μιας μηχανής. Αποσβολωμένος από μηχανικές ικανότητες που ο ίδιος φυσικά δεν κατέχει ξεχνάει την ανθρώπινη φύση του λάθους διώχνοντας έτσι το χρώμα της ζωής του. Ο κίνδυνος είναι μεγάλος και ακουμπά κυρίως τις ευάλωτες κοινωνικές ομάδες, μια από τις οποίες είναι και τα παιδιά. Το καλύτερο όπλο απέναντι στο μύθο του Η.Υ αλλά και το καλύτερο εφόδιο απέναντι στη σημερινή κοινωνία δεν είναι άλλο από τη **παιδεία**.

Γνώση ουσιαστική και βαθιά που δεν αφορά απλά τις λειτουργίες του υπολογιστή αλλά το γιατί, το πώς αυτές οι λειτουργίες έρχονται εις πέρας και προέρχεται από την **σωστή εκπαίδευση**.

Η Πληροφορική ως αντικείμενο και ως εργαλείο μάθησης

Η χρήση των ηλεκτρονικών υπολογιστών ανοίγει νέους ορίζοντες στην εκπαιδευτική διαδικασία με αποτέλεσμα να δημιουργείται η ανάγκη για τη μελέτη της ίδιας της επιστήμης αλλά και η χρησιμοποίηση της για την μελέτη άλλων μαθημάτων. Έτσι έχουν αναπτυχθεί δύο διαφορετικές προσεγγίσεις ως αναφορά την Πληροφορική στην εκπαίδευση, η Πληροφορική ως αντικείμενο και ως εργαλείο μάθησης.

Η Πληροφορική ως εργαλείο μάθησης.

Σύμφωνα με αυτή τη προσέγγιση οι υπολογιστές αντιμετωπίζονται ως εργαλείο το οποίο μπορεί να χρησιμοποιηθεί σε όλα τα γνωστικά αντικείμενα προκειμένου να εξυπηρετήσει τους εκάστοτε σκοπούς τους. Η δυνατότητα αυτή του υπολογιστή οφείλεται στην ικανότητά του να αναπαριστά το φυσικό κόσμο που μας περιβάλλει και μαζί με τους κανόνες που τον διέπουν, όπως επίσης ιδέες και αφηρημένες έννοιες που δύσκολα περιγράφονται. Η λειτουργία αυτή εξυπηρετεί πολλούς παιδαγωγικούς στόχους και συχνά λειτουργεί ως καταλυτικός παράγοντας στην θετική εξέλιξη του μαθητή.

Πιο συγκεκριμένα, τα **διακριτά** στάδια προόδου στις εφαρμογές εκπαιδευτικού χαρακτήρα βοηθούν ενεργά στην εξατομικευμένη μάθηση. Το κάθε παιδί ακολουθώντας τους δικούς του προσωπικούς ρυθμούς δεν νιώθει πίεση, δεν συγκρίνεται άμεσα με τους υπόλοιπους και οδηγείται ανώδυνα στο πιο γρήγορο για εκείνο μονοπάτι γνώσης.

Ταυτόχρονα νομιμοποιείται το δικαίωμα του παιδιού στο λάθος αφού η τυχόν λανθασμένη του απάντηση εμπλέκεται σε μια επιθυμητή για το πρόγραμμα διαδικασία δοκιμής και πλάνης. Το παιδί άφοβα πειραματίζεται, δοκιμάζει,

συνειδητοποιεί πως δεν υπάρχει μόνο μια σωστή απάντηση ή μόνο ένας σωστός τρόπος σκέψης.

Παράλληλα, η εκάστοτε δραστηριότητα αξιολογείται άμεσα από τον υπολογιστή καθώς μια επιθυμητή αντίδραση (σωστή απάντηση) ακολουθείται άμεσα από το κατάλληλο ενισχυτικό ερέθισμα (έπαινος). Αντίθετα ένα τυχόν λάθος δεν απορρίπτεται αλλά δίνεται κίνητρο για επαναπροσδιορισμό των στόχων του χρήστη και ενίσχυση της προσπάθειάς του.

Τέλος, αναφέρουμε πως η συγκεκριμένη εκπαιδευτική δράση βοηθά ιδιαίτερα τους πιο αδύναμους μαθητές αφού η κατάκτηση της γνώσης έρχεται σταδιακά μέσα από μικρές επιτυχίες, οι οποίες είναι οργανωμένες σύμφωνα με το δικό τους επίπεδο με τρόπο τέτοιο ώστε να ενθαρρύνονται και να αποζητούν την συνέχεια της προσπάθειας.

Σημειώνουμε πως ο σχεδιασμός τέτοιων προγραμμάτων προκειμένου να εξυπηρετήσει τους παραπάνω στόχους πρέπει να είναι ιδιαίτερα προσεχτικός και μελετημένος. Η συνεργασία προγραμματιστών και εκπαιδευτικών καθώς και η συνεχή έρευνα και ενημέρωση τους τόσο σε τεχνολογικά όσο και σε παιδαγωγικά ζητήματα είναι απαραίτητη προϋπόθεση.

Η Πληροφορική ως αντικείμενο μάθησης.

Η προσέγγιση αυτή έχει σαν στόχο την απόκτηση γνώσεων για την λειτουργία των ηλεκτρονικών υπολογιστών, την εισαγωγή στον προγραμματισμό και γενικότερα τις λειτουργίες της λογικής διαδικασίας. Η Πληροφορική αντιμετωπίζεται ως σύνθεση τριών βασικών επιστημονικών όρων. Της **θεωρίας**, των **πειραματικών επιστημών** και της **τεχνολογίας**. Η θεωρία περιλαμβάνει το μαθηματικό υπόβαθρο της λειτουργίας των ηλεκτρονικών υπολογιστών, οι πειραματικές επιστήμες, θέματα που αφορούν την ανάπτυξη των υλικών ενώ η

τεχνολογία αφορά τις επιλογές εκείνες που σε συνδυασμό με τα παραπάνω οδηγούν στην επίλυση καθημερινών προβλημάτων και αναγκών.

Σύμφωνα με την προσέγγιση αυτή ο μαθητής μετατρέπεται από απλός χρήστη σε ουσιαστικό γνώστη της πληροφορικής επιστήμης αλλά και του υπολογιστή ως τεχνολογικού εργαλείου. Με αυτή την ιδέα ως καθοδηγητή οργάνωσης της εκπαιδευτικής διαδικασίας ο μαθητής μπορεί να αναπτύσσει την λογική, αναλυτική και συνθετική του ικανότητα αφού συνδέει αντικείμενα και ενέργειες των εννοιολογικών αντικειμένων με τα φαινόμενα στην οθόνη του υπολογιστή. Έτσι οι ενέργειες και οι αποφάσεις του χρήστη συνδέονται άμεσα με την ανατροφοδότηση που δέχεται από τον υπολογιστή του. Ταυτόχρονα μέσα από αυτούς τους χώρους της τυχαίας συχνά αναζήτησης οδηγούμαστε σε νέους λογικομαθηματικούς αναστοχασμούς που βοηθούν το μαθητή στην ανάπτυξη της κριτικής του ικανότητας καταγράφοντας τη διαδικασία ιδέα, ανάλυση, σύνθεση, υλοποίηση, καταγραφή, αξιολόγηση. Βέβαια αυτόνομο είναι πως μια τέτοια εκπαιδευτική πορεία που σκοπό έχει την δημιουργία ενός μαθητή γνώστη δε μπορεί παρά να δημιουργεί και ουσιαστικότερους χρήστες. Η γνώση έρχεται ως ανάδραση στον απλό χρήστη και τον οδηγεί σε πιο σωστούς και έγκυρους χειρισμούς.

Σήμερα έχει παρατηρηθεί πως οι γνώστες της πληροφορικής μπορούν να επικοινωνούν μεταξύ τους μέσω του διαδικτύου ανταλλάσσοντας τις προσωπικές τους «ανακαλύψεις» όχι σε προσωπικό αλλά επιστημονικό επίπεδο. Το γεγονός αυτό βοηθάει στην επίλυση προβλημάτων αλλά και στην ηθική ικανοποίηση του χρήστη που πια από απλός «πελάτης» του υπολογιστή έχει μετατραπεί σε ουσιαστικό «διαχειριστή».

Σκοπός της διπλωματικής

Σκοπός της διπλωματικής είναι να εξετάσει αν το Scratch αν βοηθά ώστε να ξεπεραστούν οι βασικές μαθησιακές δυσκολίες της διδακτικής του προγραμματισμού και έχει τα χαρακτηριστικά που χρειάζονται για την εισαγωγή του στην διδακτική της Πληροφορικής στην Δευτεροβάθμια Εκπαίδευση στη Ελλάδα, να το συγκρίνει με τα υπόλοιπα εργαλεία που υπάρχουν και τις διδακτικές προσεγγίσεις στις οποίες στηρίζονται και να παρουσιάσει ένα πρωτότυπο σενάριο.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ

Δομή της παρούσας εργασίας

Ακολουθώς παρουσιάζεται συνοπτικά η δομή της εργασίας και τα κεφάλαια από τα οποία αποτελείται:

Κεφάλαιο 1: Περιγράφεται το πλαίσιο σπουδών της ελληνικής εκπαίδευσης μέσα στο οποίο εντάσσεται η διδασκαλία του προγραμματισμού.

Κεφάλαιο 2: Αναφέρονται οι βασικές θεωρίες μάθησης και πώς η Πληροφορική μπορεί να χρησιμοποιηθεί σε κάθε μία.

Κεφάλαιο 3: Εντοπίζονται οι βασικοί άξονες της διδακτικής του προγραμματισμού και οι δυσκολίες για κάθε έναν.

Κεφάλαιο 4: Περιγράφονται οι λύσεις που δόθηκαν για να ξεπεραστούν οι δυσκολίες της διδακτικής του προγραμματισμού και σε θεωρητικό επίπεδο και σε επίπεδο εφαρμογών, με την περιγραφή των κυριότερων εξ' αυτών.

Με τα παραπάνω κεφάλαια τελειώνει η βιβλιογραφική αναφορά στο περιβάλλον που αναπτύχθηκε το Scratch και ξεκινάει η αναλυτική περιγραφή του εργαλείου.

Κεφάλαιο 5: Στην εισαγωγή παρατίθενται οι βασικοί λόγοι που οδήγησαν στη δημιουργία ενός νέου εργαλείου. Οι αρχές σχεδιασμού περιγράφονται τους βασικούς άξονες πάνω στους οποίους δημιουργήθηκε το Scratch ενώ τα βασικά χαρακτηριστικά περιγράφουν με ποιο τρόπο υλοποιήθηκαν οι αρχές σχεδιασμού. Στην συνέχεια έχουμε τρία παραδείγματα χρήσης του εργαλείου και ακολουθεί η θέση του Scratch στην εκπαιδευτική διαδικασία.

Κεφάλαιο 6: Δίνεται μια αναλυτική περιγραφή της χρήσης και της διεπαφής χρήστη.

Κεφάλαιο 7: Θεωρία σχεδιασμού δραστηριοτήτων

Κεφάλαιο 8: Ένα πρωτότυπο σενάριο

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Κεφάλαιο 1 - Πλαίσιο σπουδών

Η Πληροφορική στο ελληνικό σχολείο – Ιστορικά στοιχεία

Πρώτη ένταξη Πληροφορικής στην ελληνική εκπαίδευση

- 1983-1985: Πληροφορική στα τεχνικά επαγγελματικά και πολυκλαδικά λύκεια
 - δημιουργία κλάδου Πληροφορικής
 - ντε φάκτο καθιέρωση μοντέλου ένταξης ΤΠΕ με στόχο τον αλφαριθμητισμό στους υπολογιστές χωρίς πειραματική φάση
 - λαθεμένη η θεώρηση της Πληροφορικής ως μαθήματος επαγγελματικής κατάρτισης
- 1990: μάθημα Πληροφορικής στο γυμνάσιο
- 1997: Ενιαίο Πλαίσιο Προγράμματος Σπουδών (ΕΠΠΣ)
- 1998: Πληροφορική στο γενικό λύκειο
- 2003: Πληροφορική στο δημοτικό

Ενιαίο Πλαίσιο Προγράμματος Σπουδών (ΕΠΠΣ)

- ενιαίος τρόπος θεώρησης της ένταξης των ΤΠΕ στην ελληνική εκπαίδευση
 - γενικό πλαίσιο
 - προγράμματα σπουδών
 - μεθοδολογία διδασκαλίας
 - προδιαγραφές σχολικών εργαστηρίων
- 2001-2003: Διαθεματικό Ενιαίο Πλαίσιο Προγράμματος Σπουδών (ΔΕΠΠΣ)
- ένταξη ΤΠΕ στην ελληνική εκπαίδευση

- πρωτοβάθμια εκπαίδευση: κυρίως ολοκληρωμένο με στοιχεία τεχνοκεντρικού μοντέλου
- δευτεροβάθμια εκπαίδευση: κυρίως πραγματολογικό με στοιχεία τεχνοκεντρικού μοντέλου
- κύκλος «Πληροφορικής και Υπηρεσιών» του ενιαίου λυκείου: τεχνοκεντρικό μοντέλο
- το πλαίσιο σπουδών Πληροφορικής ως αυτόνομο γνωστικό αντικείμενο
 - ο σύγχρονος ορισμός της γνώσης πρέπει να περιλαμβάνει και την ικανότητα να κατανοούμε και να χρησιμοποιούμε την τεχνολογία
 - η αξιοποίηση των εφαρμογών της Πληροφορικής συνδέεται με ένα σύνολο δεξιοτήτων που θα είναι απαραίτητες στο σημερινό μαθητή-αυριανό πολίτη για να εξελιχθεί επαγγελματικά και να επιβιώσει σε ένα κόσμο συνεχώς μεταβαλλόμενο

Πώς εντάσσεται σήμερα η διδασκαλία Πληροφορικής στο αναλυτικό πρόγραμμα δημοτικού-γυμνασίου

Οι ΤΠΕ στην πρωτοβάθμια εκπαίδευση

Ώρα Πληροφορικής

- χρήση της Πληροφορικής αρχικά στη διδασκαλία των επιστημών και της τεχνολογίας
- συνάντηση με τις ΤΠΕ
- σφαιρική προσέγγιση διάφορων χρήσεων ΤΠΕ σε καθημερινές δραστηριότητες
 - πληροφορία και επεξεργασία της
 - επικοινωνία
 - ψυχαγωγία
 - νέες δυνατότητες προσέγγισης της γνώσης

Σκοπός της Πληροφορικής στο δημοτικό σύμφωνα με το ΠΙ

Γενικός σκοπός (ΕΠΠΣ)

«οι μαθητές χρησιμοποιούν με (ή χωρίς) τη βοήθεια του εκπαιδευτικού τον υπολογιστή ως «γνωστικό-διερευνητικό εργαλείο», αναζητούν πληροφορίες, επικοινωνούν και προσεγγίζουν βασικές αρχές που διέπουν τη χρήση της υπολογιστικής τεχνολογίας»

Ειδικός σκοπός (ΔΕΠΠΣ)

- Να εξοικειωθούν οι μαθητές και οι μαθήτριες με τις βασικές λειτουργίες του υπολογιστή
- Να έλθουν σε μια πρώτη επαφή με διάφορες χρήσεις του
 - ο ως εποπτικού μέσου διδασκαλίας
 - ο ως γνωστικού - διερευνητικού εργαλείου
 - ο ως εργαλείου επικοινωνίας και αναζήτησης πληροφοριών στο πλαίσιο των καθημερινών σχολικών τους δραστηριοτήτων
- Με τη χρήση κατάλληλου λογισμικού και ιδιαίτερα ανοικτού λογισμικού διερευνητικής μάθησης
- Σε καμία περίπτωση δεν νοείται η διδασκαλία της Πληροφορικής ως διδασκαλία γνωστικού αντικείμενου (λαμβανομένου, επιπλέον, υπόψη ότι δεν διατίθεται χρόνος στο αντίστοιχο Ωρολόγιο Πρόγραμμα)
- Σκοπός είναι ο μαθητής να μαθαίνει με τη χρήση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας (ΤΠΕ) παρά για τη χρήση τους

Άξονες υλοποίησης του σκοπού της Πληροφορικής στο δημοτικό

Ο υπολογιστής γνωστικό – διερευνητικό εργαλείο

- κύριος άξονας ένταξης ΤΠΕ
- χρήση ανοικτού λογισμικού διερευνητικής μάθησης
 - ο αλληλεπιδραστικά πολυμέσα, προσομοιώσεις, εκπαιδευτικά παιχνίδια, μοντελοποίηση, ...
- διερεύνηση πραγματικών ή φανταστικών καταστάσεων

- ανάπτυξη δημιουργικής και ανακαλυπτικής μάθησης
- ανάπτυξη δραστηριοτήτων
- οργάνωση γνώσεων και δεξιοτήτων

Ο υπολογιστής εποπτικό μέσο διδασκαλίας σε βασικά γνωστικά αντικείμενα

- λογισμικό ευρείας χρήσης
 - ζωγραφική, επεξεργασία κειμένου, λογιστικά φύλλα, ...
- διδασκαλία
 - γλώσσα, γραπτή έκφραση, μαθηματικά, καλλιτεχνικά, ...

Ο υπολογιστής εργαλείο επικοινωνίας και αναζήτησης πληροφοριών

- βάσεις δεδομένων για αναζήτηση, δίκτυα για επικοινωνία και αναζήτηση

Πληροφορικός αλφαριθμητισμός

- πλαίσιο της “Ευέλικτης Ζώνης”
 - βασικές λειτουργίες (αποθήκευση, επεξεργασία δεδομένων, επικοινωνία)
 - διανοητική – γνωστική πτυχή (κατανόηση της χρήσης πληροφορικών αντικειμένων)
 - ηθική – πολιτισμική πτυχή (κατανόηση των πληροφοριακών εργαλείων μέσα από την προοπτική της κατάρτισης του αυριανού πολίτη)

Οι ΤΠΕ στο γυμνάσιο

- Η 1^η σχολική βαθμίδα εισαγωγής αυτόνομου μαθήματος Πληροφορικής (σταδιακά για 10 χρόνια από το 1992)
- Εργαστήριο Πληροφορικής σε κάθε γυμνάσιο (10 – 20 PC, Windows)
- Ωριαίο μάθημα Πληροφορικής κάθε εβδομάδα

Σκοπός διδασκαλίας της Πληροφορικής στο γυμνάσιο

Συνολική θεώρηση Πληροφορικής

- γνώσεις
 - βασικές έννοιες και όροι της Πληροφορικής
- δεξιότητες
 - αυτονομία χειρισμού υπολογιστή
- ευαισθητοποίηση στις κοινωνικές επιπτώσεις των ΤΠΕ
 - στάσεις και αξίες

Η σύγχρονη κουλτούρα οφείλει να έχει ένα ισχυρό τεχνικό και επιστημονικό συστατικό.

Σκοπός διδασκαλίας της πληροφορικής στο γυμνάσιο (ΔΕΠΠΣ)

- Βασικές έννοιες και όροι της ΤΠΕ, δηλαδή τα μέσα και οι τεχνικές που χρησιμοποιούνται για
 - επεξεργασία, μετάδοση και λήψη πληροφοριών σε ψηφιακή μορφή
- Βασικές απλές έννοιες της γενικής δομής των υπολογιστικών συστημάτων και τις διαχρονικές αρχές που τα διέπουν
 - αρχιτεκτονική υπολογιστών, διαφορετικότητα υπολογιστικών συστημάτων, πρόγραμμα, οργάνωση και διαχείριση αρχείων ...
- Απαραίτητες δεξιότητες χειρισμού και κριτικής επεξεργασίας, καθώς και δεξιότητες μεθοδολογικού χαρακτήρα, σε ένα σύστημα υπολογιστών και στα βασικά εργαλεία που το συνοδεύουν
- διαδικασία επίλυσης απλών προβλημάτων με τη χρήση του υπολογιστή.
 - μια απλή μηχανή ελέγχεται και προγραμματίζεται από τον άνθρωπο

- Εφαρμογές πολυμέσων, πλοήγηση και αλληλεπίδραση, Διαδίκτυο, χρήση κατάλληλου λογισμικού για αξιοποίηση του υπολογιστή, αρχικά, στο πλαίσιο διαφόρων μαθημάτων αλλά και σε μετέπειτα δραστηριότητες
- Ανακάλυψη, επιλογή, ανάλυση και αξιολόγηση πληροφοριών για αξιοποίηση στις εκπαιδευτικές δραστηριότητες αλλά και στην καθημερινή ζωή
- Κώδικες δεοντολογίας στο πλαίσιο της συνεργασίας με άλλους, του σεβασμού της εργασίας και της διαφορετικότητάς
- Τρέχουσες και μελλοντικές επιπτώσεις των ΤΠΕ σε ατομικό και κοινωνικό επίπεδο αλλά και στους διάφορους τομείς της ανθρώπινης δραστηριότητας

Άξονες υλοποίησης του σκοπού της Πληροφορικής στο γυμνάσιο

Γνωρίζω – επικοινωνώ με τον υπολογιστή

- Σύνολο βασικών απλών εννοιών που αφορούν τη γενική δομή των υπολογιστών συστημάτων
 - ο αρχιτεκτονική υπολογιστών, διαφορετικότητα υπολογιστικών συστημάτων, πρόγραμμα, οργάνωση και διαχείριση αρχείων, ...
- Ανάπτυξη διαχρονικής κουλτούρας σε βασικές έννοιες Πληροφορικής (ορθολογική προσέγγιση)

Χρήση εργαλείων έκφρασης, επικοινωνίας, ανακάλυψης και δημιουργίας

- Βασικό λειτουργικό σύστημα ευρείας χρήσης
 - ο εφαρμογές γραφείου, λογισμικό πλοήγησης στο διαδίκτυο, ...
- Κατηγορίες λογισμικού, ικανότητες μεθοδολογίας
- Ορθολογικές πρακτικές ασκήσεις
- Μείζονος σημασίας άξονας (συμμετοχικός και συνεργατικός χαρακτήρας της μάθησης, αξιοποίηση λογισμικού για έκφραση και επικοινωνία, δεξιότητες μοντελοποίησης, διαχείριση πληροφοριών, ...)

Ο υπολογιστής στο σχολείο και στην κοινωνία

- Δραστηριότητες της καθημερινής ζωής, της κοινωνίας και του πολιτισμού
 - ο Επιπτώσεις των ΤΠΕ , προστασία πνευματικών δικαιωμάτων, ασφάλεια πληροφοριών, συμπεριφορά στο Διαδίκτυο, ...

Ελέγχω – προγραμματίζω τον υπολογιστή

Η διδασκαλία Πληροφορικής ως γνωστικό αντικείμενο στο αναλυτικό πρόγραμμα δημοτικού-γυμνασίου

Για την ένταξη την διδασκαλίας της Πληροφορικής στο αναλυτικό πρόγραμμα των ηλικιών που εξετάζουμε (12-14) βασιστήκαμε σε συγκεκριμένα τμήματα του πλαισίου σπουδών Πληροφορικής του Παιδαγωγικού Ινστιτούτου. Πιο συγκεκριμένα, η Πληροφορική ως γνωστικό αντικείμενο εμφανίζεται στην 6η τάξη του δημοτικού και στην 3η γυμνασίου.

Αναλυτικά:

ΤΑΞΗ ΣΤ' Δημοτικού

Ελέγχω και προγραμματίζω	Χρήση μιας απλής γλώσσας προγραμματισμού (Logo like) για τον έλεγχο και τον προγραμματισμό του υπολογιστή.	Πρόβλημα Οργάνωση, Διάκριση Μεταβολή, Προσαρμογή Επικοινωνία Αλληλεπίδραση
--------------------------	--	--

Ειδικοί σκοποί στον στόχο "Ελέγχω και προγραμματίζω"

Να κατανοήσουν ότι ο υπολογιστής εκτελεί οδηγίες που παίρνει από τον άνθρωπο σε μια κωδικοποιημένη μορφή. Να χρησιμοποιούν απλές εντολές για τη δημιουργία σχημάτων ή τη λύση απλών	Ελέγχω και προγραμματίζω Χρήση μιας απλής γλώσσας προγραμματισμού (Logo like) για τον έλεγχο και προγραμματισμό του υπολογιστή. Καλούνται να δημιουργήσουν απλά	Μέσα από επιλεγμένα παραδείγματα, όπου υπάρχουν σκόπιμα «λάθη», τα παιδιά κατανοούν ότι ο υπολογιστής εκτελεί τις οδηγίες που ο άνθρωπος του δίνει.
--	---	---

προβλημάτων.	γεωμετρικά σχήματα δίνοντας κατάλληλες εντολές μετακίνησης ή στροφής στη χελώνα.	
--------------	--	--

Από το Αναλυτικό πρόγραμμα πληροφορικής για το δημοτικό

ΤΑΞΗ Γ΄ Γυμνασίου

Γνωρίζω τον υπολογιστή ως ενιαίο σύστημα	Γλώσσες προγραμματισμού. Βασικά στάδια επίλυσης προβλήματος με τη χρήση υπολογιστή. Δημιουργία και εκτέλεση προγράμματος.	Πρόβλημα, Λύση Αξιολόγηση, Οργάνωση Διαδοχή, Διάκριση Μεταβολή, Προσαρμογή Επικοινωνία Αλληλεπίδραση
--	---	--

Ειδικοί σκοποί στον στόχο "Γνωρίζω τον υπολογιστή ως ενιαίο σύστημα"

<p>Οι μαθητές επιδιώκεται:</p> <ol style="list-style-type: none"> 1. Να αναγνωρίζουν την έννοια της γλώσσας προγραμματισμού και την αναγκαιότητα της χρήσης της. 2. Να σχεδιάζουν τη λύση ενός απλού προβλήματος και να την υλοποιούν σε ένα προγραμματιστικό περιβάλλον. 	<p><i>Γνωρίζω τον υπολογιστή ως ενιαίο σύστημα</i></p> <ul style="list-style-type: none"> • Γλώσσες προγραμματισμού. • Βασικά στάδια επίλυσης προβλήματος με τη χρήση υπολογιστή. • Περιγραφή και κατανόηση του προβλήματος. • Σχεδίαση της λύσης του προβλήματος. • Περιγραφή αλγορίθμου. • Κωδικοποίηση. • Δημιουργία και εκτέλεση προγράμματος 	<p>Οι μαθητές περιγράφουν τη διαδικασία επίλυσης ενός προβλήματος (σύλληψη, σκέψη, λύση – εκτέλεση πράξεων, αποτέλεσμα). Προβληματίζονται και διαπιστώνουν ότι ο υπολογιστής παρεμβαίνει στα δύο τελευταία στάδια (λύση – εκτέλεση πράξεων, αποτέλεσμα). Κατανοούν ότι η σύλληψη και η σκέψη είναι χαρακτηριστικά μόνον του ανθρώπου.</p> <p>Εισάγονται στην έννοια του αλγορίθμου αναλύοντας τα βήματα που ακολουθούν για την εκτέλεση μιας διαίρεσης (αλγόριθμος Ευκλείδη). Στη συνέχεια οι μαθητές που έχουν αναζητήσει (π.χ. στο Διαδίκτυο, σε βιβλία μαγειρικής, περιοδικά κ.ά.) και έχουν φέρει μαζί τους μια συνταγή μαγειρικής, τη μελετούν και διαπιστώνουν ότι για να παραχθεί ένα φαγητό ή ένα γλυκό πρέπει να ακολουθηθούν μια σειρά από</p>
---	--	--

		<p>συγκεκριμένα βήματα που αν δεν τηρηθούν δεν θα υπάρξει το επιθυμητό αποτέλεσμα. Τέλος, καταγράφουν τα βήματα που ακολουθούν για την αντιμετώπιση καθημερινών καταστάσεων όπως, για παράδειγμα, την πρωινή τους προετοιμασία στο σπίτι, τη σειρά των ενεργειών που ακολουθούν για να τοποθετήσουν μια μαγνητοταινία στο κασετόφωνο για να ακούσουν μουσική κλπ.</p> <p>Προβληματίζονται για τον τρόπο με τον οποίο μπορούν να χρησιμοποιήσουν τον υπολογιστή για την επίλυση ενός προβλήματος (π.χ. την πρόσθεση δύο αριθμών). Διαπιστώνουν την ανάγκη ύπαρξης μιας γλώσσας επικοινωνίας με τον υπολογιστή και γνωρίζουν το προγραμματιστικό περιβάλλον που θα χρησιμοποιήσουν.</p> <p>Επιλύουν αλγοριθμικά απλά προβλήματα τα οποία υλοποιούν σε προγραμματιστικό περιβάλλον.</p>
--	--	--

Από το Αναλυτικό πρόγραμμα Πληροφορικής για το γυμνάσιο

Κεφάλαιο 2 - Βασικές θεωρίες μάθησης και Πληροφορική

Skinner, οι θεωρίες μάθησης στον σχεδιασμό εκπαιδευτικού λογισμικού

Η μάθηση κατά τον Skinner συντελείται κυρίως:

1. Με την ενίσχυση της επιθυμητής συμπεριφοράς του ατόμου μέσω της αμοιβής και της ψυχολογικής ενίσχυσης.
2. Με την απάλειψη μιας ανεπιθύμητης συμπεριφοράς εφαρμόζοντας την (άμεση ή) έμμεση τιμωρία (έμμεση τιμωρία είναι η διακοπή ευχάριστων καταστάσεων) ή με την απόσβεσή της, δηλαδή, χωρίς την παροχή ενίσχυσης ή την επιβολή τιμωρίας, αλλά με τη αγνόηση της.

Ένας βασικός άξονας που απορρέει από τη θεωρία του Skinner και γενικότερα του συμπεριφορισμού είναι ότι αν ορισμένη επιθυμητή αντίδραση του εκπαιδευόμενου ακολουθείται από κάποιο ενισχυτικό ερέθισμα, η πιθανότητα να επαναληφθεί σε ανάλογες περιπτώσεις αυξάνεται. Αντίθετα, αν μια ορισμένη συμπεριφορά δεν συνοδεύεται από κάποια ενίσχυση, παύει σιγά-σιγά να εκδηλώνεται, γίνεται δηλαδή "απόσβεσή" της. Πρακτικά, σε ένα λογισμικό αυτό σημαίνει πως κατά την αξιολόγηση η σωστή απάντηση συνοδεύεται από ένα επαινετικό σχόλιο, ήχο ή εφέ ενώ η αρνητική **δεν** ακολουθείται από κάποια τιμωρία (κι άλλη άσκηση) αλλά ερέθισμα προσδοκώμενης θετικής συμπεριφοράς.

Μία ακόμη συμβολή του Skinner έχει σχέση με τον σχεδιασμό της προγραμματισμένης διδασκαλίας που οδηγεί σε τρεις βασικές θέσεις:

1. Οι παιδαγωγικοί και διδακτικοί στόχοι πρέπει να είναι διατυπωμένοι με πολύ συγκεκριμένο τρόπο και με σαφή περιγραφή των επιδιωκόμενων αλλαγών στη συμπεριφορά των μαθητών

2. Με αυτόν τον τρόπο έχουμε και ένα δευτερεύων αποτέλεσμα. Πετυχαίνουμε η σχολική ατμόσφαιρα να γίνει περισσότερο ενισχυτική και αισιόδοξη και -κατ' αυτόν τον τρόπο- λιγότερο ανασταλτική για την δραστηριοποίηση των μαθητών σε επιθυμητά έργα.

Πiaget, οι εφαρμογές του εποικοδομητισμού στην μάθηση μέσω του υπολογιστή.

Η επιστημονική και παιδαγωγική παραδοχή ότι τόσο η γνώση, όσο και η μάθηση δεν «αποκτιέται» ούτε μεταβιβάζεται, αλλά κατασκευάζεται με την ενεργό συμμετοχή των ίδιων των διδασκόμενων από την αλληλεπίδρασή τους τεκμηριώθηκε αρχικά από τον J.Piaget και τη θεωρία του εποικοδομητισμού.

Βασικές αρχές της θεωρίας του εποικοδομητισμού είναι οι παρακάτω:

Κάθε άτομο δημιουργεί τις δικές του **γνωστικές αναπαραστάσεις** οι οποίες δεν είναι παρά εμπειρίες βασιζόμενες σε προϋπάρχουσες δομές του ατόμου. Η αρχή αυτή έχει σαν αποτέλεσμα

- οι μαθητές να μαθαίνουν με το δικό τους τρόπο και ρυθμό μέσω μιας **ενεργούς γνωστικής εξερεύνησης** όπου ο μαθητής ερχόμενος αντιμέτωπος με την αντίφαση των δεδομένων για εκείνον γνώσεων και των νέων που λαμβάνει εμπειρικά, να καταλήγει στην ανάπτυξη ενός γνωστικού σχήματος πλήρους αλλά και ανοιχτού σε τυχόν αλλαγές και διαμορφώσεις. Η διαδικασία αυτή βοηθά τον μαθητή
- να είναι **ανοιχτός** σε νέους γνωστικούς ορίζοντες
- να **αγαπά** την μάθηση και την εξερεύνηση αυτής και ερχόμενος σε αντίθεση με το παραδοσιακό κατεστημένο
- να γνωρίζει πως η γνώση και η αλήθεια της **δεν είναι μια και μοναδική**, αλλά πρόκειται για κάτι ζωντανό το οποίο ακολουθεί το ρεύμα της ανάπτυξης και της προόδου και άρα δεν μπορεί παρά να μεταβάλλεται συνεχώς.

Σημαντικό ρόλο στη πορεία της γνώσης παίζει και το **περιβάλλον** στο οποίο εκφράζεται, ψάχνει και αλληλεπιδρά ο μαθητής. Έτσι το περιβάλλον πρέπει να είναι διαμορφωμένο με όσο το δυνατό περισσότερα ερεθίσματα, που καθιστούν δυνατή την πλούσια αλληλεπίδραση μεταξύ του δεδομένου και του καινούργιου, γεγονός που θα οδηγήσει στη γνώση.

Μία άλλη αρχή που βασίζεται σε ο,τι προαναφέρθηκε είναι εκείνη της **συγκεκριμενοποιημένης μάθησης**. Σύμφωνα με την αρχή αυτή η μάθηση είναι περισσότερο αποτελεσματική όταν λαμβάνει χώρα σε ένα *συναφές πλαίσιο*, το οποίο δίνει νόημα στη γνώση και αποτελεί μέρος της γνωστικής βάσης των μαθητευομένων. Η εκπαίδευση λοιπόν είναι καλό να παρέχεται σε σχετικό με τις εμπειρίες του μαθητευόμενου και τις κοινωνικές καταβολές του πλαισίου, που σύμφωνα με τον Piaget θα κάνει τον μαθητευόμενο να αισθάνεται «σαν το σπίτι του».

Παράλληλα η συγκεκριμένη θεωρία ενθαρρύνει τον μαθητή στην **αιτιολόγηση και τεκμηρίωση** των ερμηνειών του, στην κριτική και δημιουργική του σκέψη, στην ενεργητική χρήση και μεταφορά της γνώσης.

Η **γνώση αντιμετωπίζεται ως όλο**, με στοιχεία αλληλεξαρτώμενα τα οποία αφορούν άμεσα την πράξη και τη ζωή. Γι' αυτό πολλοί από τους μαθησιακούς στόχους απαιτούν εμπλοκή των μαθητών σε παιδαγωγικά σχεδιασμένες σύνθετες καταστάσεις που στοχεύουν στην ενεργοποίηση της κριτικής σκέψης και στις βιωματικές διαδικασίες μάθησης. Στη βασική αυτή εποικοδομιστική παιδαγωγική αρχή βασίζονται τα σύγχρονα αιτήματα για περισσότερη ενιαιοποίηση της γνώσης (*integration*) στο σχολείο, κάτι που συχνά αποκαλείται και διαθεματικότητα ή διεπιστημονικότητα.

Ο μαθητής προσεγγίζεται επίσης, ως όλον, που αναζητεί νοήματα, έχει κίνητρα, συναισθήματα, στόχους και **προσωπική αξία μοναδική**, πέραν των κριτηρίων του σωστού-λάθους που έχει θεσπίσει το σχολείο. Το λάθος απενεργοποιείται, αφού εκλαμβάνεται απλώς ως ένα γνωστικό σχήμα ή μια «θεωρία» μαθητή, όπως ακριβώς υπήρξαν και οι θεωρίες των έμπειρων επιστημών, πριν

διαψευστούν από άλλες, βιωσιμότερες θεωρίες. Αναπτυξιακή πορεία επίσης, δεν έχει μόνον το αποτέλεσμα, αλλά και η διαδικασία της μάθησης. Το αυτοσυναίσθημα των μαθητών είναι κεντρικής σημασίας για την ανάπτυξη της διδακτικής σχέσης ανάμεσα στην επίδοση των μαθητών και της σχολικής γνώσης και μάθησης.

Εφαρμογές του εποικοδομητισμού στη μάθηση μέσω του ηλεκτρονικού υπολογιστή

Η παιδαγωγική φιλοσοφία του εποικοδομητισμού έχει γνωρίσει ιδιαίτερη άνθηση τα τελευταία χρόνια στο τομέα της διδακτικής και της εκπαιδευτικής τεχνολογίας με τη βοήθεια του υπολογιστή. Εύκολες στη χρήση τους, ευέλικτες και πολυδύναμες, οι νέες τεχνολογίες των υπολογιστών παρέχουν αρκετές δυνατότητες για την εξασφάλιση εποικοδομιστικών περιβαλλόντων στις σχολικές τάξεις, όπως μας δείχνουν οι μέχρι τώρα δοκιμασμένες διδασκαλίες με τη χρήση και την παιδαγωγική αξιοποίηση των διαφόρων υπολογιστικών εργαλείων και του διαδικτύου. Ενδεικτικό είναι το γεγονός πως ο χαρακτηρισμός και μόνο ενός εκπαιδευτικού λογισμικού ως «εποικοδομιστικού» αποτελεί θετικό κριτήριο για την ποιότητά του.

Ένα άρτια οργανωμένο λογισμικό, ενημερωμένο για την εκπαιδευτική επικαιρότητα μπορεί να παρέχει εξαιρετικές εκπαιδευτικές δυνατότητες εποικοδομιστικής διδασκαλίας μέσω της προσομοίωσης φαινομένων και πραγματικών καταστάσεων, της μοντελοποίησης προβλημάτων καθώς και της δημιουργίας γνωστικών «μικρόκοσμων» και άλλων «ανοιχτών» περιβαλλόντων μάθησης, που ευνοούν την εφαρμογή σημαντικών παιδαγωγικών θεωριών όπως εκείνη του Piaget.

Αυτό συμβαίνει χάρις στην δυνατότητα ενός παιδαγωγικού σχεδιασμού βασισμένου σε ένα «ανοιχτό» μαθησιακό περιβάλλον που παρέχει στο μαθητή χειροπιαστά αντικείμενα, οικείους χώρους εξερεύνησης, εργαλεία σκέψης και όχι έτοιμες απαντήσεις. Έτσι ο διδασκόμενος μπορεί να χειρίζεται τον

υπολογιστή και τα ερεθίσματά του αυτόνομα και δυναμικά, να δρα πάνω σ' αυτά, να πειραματίζεται, να αυτοελέγχεται και να αυτοδιορθώνεται.

Ταυτόχρονα εφόσον υιοθετηθεί ένα ομαδοσυνεργατικό πλαίσιο εργασίας γύρω από τον υπολογιστή το παιδί μπορεί να διαπραγματευτεί με τους υπόλοιπους, να συζητήσει, να ανακαλύψει τη νέα γνώση χτίζοντάς την πάνω στα δικά του νοητικά σχήματα με την απαραίτητη διευκόλυνση εκ μέρους των συμμαθητών και των δασκάλων του.

Οι δάσκαλοι τώρα πια μπορούν μέσω του υπολογιστή να δημιουργήσουν ένα αυθεντικό, έστω και εικονικό, μαθησιακό περιβάλλον μέσα στο οποίο οι μαθητές θα λειτουργούν ενεργά και άμεσα ξεφεύγοντας από την ρουτίνα της θεωρίας. Οι μαθητές μπορούν να μπουν σε ρόλους να βιώσουν λειτουργικά τη ζωή και τα ενδιαφέροντα ενός χημικού για παράδειγμα. Έτσι μέσα από μια τέτοια βιωματική διαδικασία το παιδί θα οδηγηθεί στη γνώση των χημικών ουσιών, της χρησιμότητας ή της επικινδυνότητάς τους. Ευνοείται έτσι και η ανάπτυξη της κριτικής σκέψης αφού ο διδασκόμενος βρίσκεται αντιμέτωπος με ένα καταιγισμό πληροφοριών, τις οποίες, με τη σωστή πάντα καθοδήγηση πρέπει να κρίνει και να αναπαράγει.

Παράλληλα ο υπολογιστής μπορεί να αξιοποιηθεί απ' όλα τα γνωστικά αντικείμενα ως εργαλείο αναπαριστώντας τη γνώση διαθεματικά. Χρησιμοποιώντας τις δικτυακές υπηρεσίες, τα λογισμικά πακέτα διαφορετικών γνωστικών τομέων(πηγές, αρχεία, ιστορικό υλικό, χάρτες, εικονικά μουσεία και εκθέσεις κτλ) μέσα στο ίδιο εκπαιδευτικό πλαίσιο μπορούν ταυτόχρονα τα παιδιά να διδαχθούν φυσική, ιστορία, μουσική, ακόμη και καλλιτεχνικά καλλιεργώντας τους μέσω των εκάστοτε γραφικών την επιθυμητή αισθητική.

Έχοντας στα χέρια του ο δάσκαλος ένα πλούσιο σε διδακτικό υλικό περιβάλλον, μπορεί να βοηθήσει τους μαθητές να μεταμορφωθούν σε μικροί επιστήμονες. Έτσι όλοι οι μαθητές ψάχνοντας και αναζητώντας το καινούργιο είναι βέβαιο πως θα ανακαλύψουν κάτι που τους αρέσει πολύ, κάτι που το κάνουν καλά. Ακόλουθο είναι να μειωθούν οι τάσεις περιθωριοποίησης ορισμένων μαθητών

που δεν κατάφεραν να ανταποκριθούν στο ένα και μοναδικό επιζητούμενο για χρόνια πρότυπο του καλού μαθητή παπαγάλου.

Η πρόσβαση στη πληροφορία μέσω του internet καθώς η ανάπτυξη του παγκόσμιου ιστού world wide web, της γλώσσας HTML, για τη δημιουργία ιστοσελίδων επιτρέπει στους χρήστες μαθητές την μεταξύ τους αλληλεπίδραση αλλά και έκφραση με ένα τρόπο τελείως πρωτόγνωρο και διαφορετικό. Ο μαθητής βρίσκεται στο κέντρο της μάθησης, ενώ ο δάσκαλος έχει ένα ρόλο συντονιστικό και διαπραγματευτικό. Πρόκειται για ένα πεδίο ανοιχτό και πολλά υποσχόμενο.

Συνολικές εκτιμήσεις

Με τη χρήση ενός αλληλεπιδραστικού, δυναμικού και ανοιχτού λογισμικού και με την κατάλληλη παρέμβαση των διδασκόντων είναι δυνατό να έχουμε καλύτερα αποτελέσματα, σε σύγκριση με άλλους παραδοσιακούς τρόπους μάθησης (πχ αντικειμενισμός), όσον αφορά:

- την ενεργό γνωστική επεξεργασία πληροφοριών
- την ανάπτυξη ικανότητας επίλυσης προβλημάτων
- την εμπέδωση της γνώσης
- την ανάπτυξη θετικών στάσεων απέναντι στη μάθηση ενός γνωστικού φορέα
- την ανάπτυξη κινήτρου για τη διερεύνηση της γνώσης
- την αυξημένη συμμετοχή και ικανοποίηση των «αδύναμων» μαθητών στη διαδικασία
- την ανάπτυξη της αυτοεκτίμησης των μαθητών
- την αναβάθμιση της ποιότητας των διεργασιών της σκέψης.

Vygotsky, ζώνη της επικείμενης ανάπτυξης και η εφαρμογή της στη μάθηση μέσω του υπολογιστή.

Κατά τον Vygotsky η νοητική ανάπτυξη είναι μια διαδικασία αδιάρρηκτα συνδεδεμένη με την ιστορικοκοινωνική διάσταση και το πολιτισμικό πλαίσιο, μέσα στο οποίο αυτή συντελείται. Η ανάπτυξη και η παιδεία του ατόμου επιτυγχάνεται χάρις στον έμφυτο νοητικό «εξοπλισμό» του ανθρώπου αλλά και μέσω των κοινωνικών και πολιτισμικών εργαλείων όπως η γλώσσα, τα πρότυπα συμπεριφοράς, η θρησκεία κτλ. Η γνώση λοιπόν δεν αποτελεί παρά μετατροπή των κοινωνικών σχέσεων σε νοητικές λειτουργίες. Η μετατροπή αυτή πραγματοποιείται με τη σωστή χρήση εργαλείων μάθησης, ενός σωστά οργανωμένου μαθησιακά περιβάλλοντος και βέβαια με τη σωστή καθοδήγηση του δασκάλου. Ο Vygotsky επηρεασμένος από τον Piaget, θεωρεί ότι τα κοινωνικά δοσμένα μηνύματα μετασχηματίζονται σε γνώση κατά τρόπο διαλεκτικό, καθώς ενσωματώνονται σε προϋπάρχοντα νοητικά σχήματα και αναπτύσσονται ως αποτέλεσμα κριτικής, δημιουργικής και περισσότερο χειραφετημένης σκέψης, ιδιαίτερα κατά τα προχωρημένα στάδια της γνωστικής ανάπτυξης του ατόμου.

Τα **εργαλεία** για την καλλιέργεια της ψυχοκοινωνικής ανάπτυξης του ατόμου και μετατροπή της σε γνωστική, όπως αναφέρθηκε και παραπάνω μπορεί να είναι

- **υλικά**, δηλαδή να αναπαριστούν έννοιες με εικόνες.
- **ενδεικτικά**, υποδεικνύοντας σχέση αιτίας-αποτελέσματος.
- **συμβολικά** (πχ λέξεις, μαθηματικά σύμβολα, προφορική ομιλία κτλ).

Εργαλεία είναι και πολλά άλλα υλικά ή και κοινωνικά στοιχεία του πολιτισμού, τα οποία είναι ενταγμένα στην καθημερινή μας ζωή και εκπέμπουν διάφορα σήματα και σημασίες ή χρησιμοποιούνται για την επίλυση προβλημάτων.

Ζώνη επικείμενης ανάπτυξης

Ο Vygotsky, για να ολοκληρώσει το θεωρητικό του σχήμα που αφορά στη νοητική ανάπτυξη, πρότεινε μια έννοια, που θεωρείται πολύ σημαντική για την εκπαιδευτική διαδικασία, τη γνωστή ως ζώνη επικείμενης ανάπτυξης. Σύμφωνα με τη θεωρία αυτή κάθε άτομο μπορεί να αναπτυχθεί σε οποιοδήποτε τομέα με την κατάλληλη υποστήριξη του περιβάλλοντος. Το παιδί βρίσκεται σε θέση εν δυνάμει να αναπτυχθεί επιλύοντας προβλήματα με τη καθοδήγηση και συνεργασία έμπειρων ενηλίκων ή και συνομήλικων. Έτσι θα φτάσει στο σημείο όπου το παιδί θα μπορεί όχι μόνο να δραστηριοποιείται χωρίς βοήθεια, αλλά και να προωθείται κατά τρόπο αυτόνομο. Η διδασκαλία με άλλα λόγια θα πρέπει να οδηγεί το παιδί σε πιο προχωρημένο επίπεδο γνώσεων και γνωστικών λειτουργιών από αυτό στο οποίο ήδη βρίσκεται, αξιοποιώντας το δυναμικό και την ετοιμότητά του. Η αξιολόγηση των μαθητών θα πρέπει σύμφωνα με τη θεωρία της επικείμενης ανάπτυξης να αποτελεί μέρος της μάθησης και όχι χωριστή διαδικασία ελέγχου με «ποινικό» χαρακτήρα. Για λόγους μεθοδικούς ο Vygotsky τη **ζώνη επικείμενης ανάπτυξης** την προσδιορίζει σε **τρία** επίπεδα:

- I. Στο **ατομικό** επίπεδο που είναι η απόσταση μεταξύ των ικανοτήτων επίλυσης προβλημάτων, τις οποίες αναπτύσσει ο μαθητής, όταν δουλεύει μόνος του και των ικανοτήτων εκείνων που για να τις αναπτύξει ο ίδιος παραπέρα, χρειάζεται την υποστηρικτική αλληλεπίδραση ή την συνεργασία με πιο έμπειρα άτομα.
- II. Στο επίπεδο του **πολιτισμικού περιβάλλοντος** μέσα στο οποίο η εκπαίδευση θα πρέπει να παρέχει συγκεκριμένες επιστημονικές έννοιες, ώστε το παιδί να τις κάνει κτήμα του συνθέτοντας τις με εκείνες της καθημερινής του ζωής.
- III. Στο ευρύτερο **κοινωνικό επίπεδο** όπου οι καθημερινές πράξεις του ατόμου πρέπει να τοποθετηθούν μέσα στο ευρύτερο κοινωνικό σύνολο.

Ρόλος του δασκάλου

Αντίθετα από τον Piaget ο Vygotsky υποστηρίζει πως η μάθηση προηγείται της ανάπτυξης η οποία είναι οδηγεί στη γνώση και την εξέλιξη της προσωπικότητας. Ο εκπαιδευτικός πρέπει να δημιουργήσει ένα είδος σκαλωσιάς για τους μαθητές του ώστε να μπορέσουν να προχωρήσουν στα επόμενα στάδια ανάπτυξης. Η παρέμβαση του δασκάλου είναι καθοριστική αφού πρέπει να κάνει τους μαθητές να νιώθουν το σχολικό περιβάλλον οικείο, ζεστό και υποστηρικτικό. Το παιδί πρέπει να δομεί και όχι να αντιγράφει τη γνώση. Γι' αυτό και ο δάσκαλος δεν θα πρέπει να ακολουθεί μια αυστηρά δομημένη και προκαθορισμένη για όλους τους μαθητές πορεία αλλά να είναι ανοιχτός σε απρόσμενες κατευθύνσεις και αποτελέσματα. Εκείνο όμως που εμφανίζεται ως ευθύνη του δασκάλου, είναι μέσα από ένα σύστημα βιγκοτσκιανής φιλοσοφίας, όλοι οι εκπαιδευτικοί φορείς από το πιο χαμηλό ως το πιο υψηλό επίπεδο ιεραρχίας, να συνειδητοποιήσουν πως είναι μαθητευόμενοι και έχουν ανάγκη από τη μεταξύ τους αλληλεπίδραση προκειμένου να εξελιχθούν. Στις μέρες μας ιδιαίτερα, εποχή που η συνεχιζόμενη εκπαίδευση είναι πλέον αναγκαία για όλους τους εκπαιδευτικούς και κοινωνικούς φορείς, το λειτουργικό αυτό μοντέλο καθίσταται περισσότερο επίκαιρο και αναγκαίο από την εποχή του Vygotsky.

Εφαρμογές του κοινωνικοπολιτιστικού εποικοδομητισμού του Vygotsky στη διδασκαλία με υπολογιστή.

Τα προγράμματα των υπολογιστών μπορούν να σχεδιαστούν έτσι ώστε να βοηθήσουν τους μαθητές να βγάλουν στην επιφάνεια το επόμενο στάδιο ανάπτυξης τους συντομεύοντας αλλά κάνοντας και πιο δημιουργικό το δρόμο τους προς την ολοκλήρωση και την βαθύτερη γνώση. Ο υπολογιστής έχει και μπορεί να εκμεταλλευτεί προς τη συγκεκριμένη κατεύθυνση τις εξής δυνατότητες :

- Μπορεί να προγραμματιστεί ώστε να **ελέγχει** διάφορες ζώνες. Για παράδειγμα, ειδικά τεστ μπορούν να εκτιμήσουν την ικανότητα των μαθητών να επιλύουν συγκεκριμένα προβλήματα από μόνοι τους

(ατομικά), με την βοήθεια του δασκάλου τους(συνεργατικά με καθοδήγηση) ή με τη βοήθεια των συμμαθητών τους (ομαδοκεντρικά) αξιολογώντας έτσι την πρόοδο του μαθητή αλλά ταυτόχρονα ενημερώνοντας τους εμπλεκόμενους ποιος τρόπος διδασκαλίας ταιριάζει καλύτερα στο συγκεκριμένο μαθητή.

- Επίσης ο υπολογιστής είναι σε θέση να παρακινήσει τους μαθητές και να τους υποστηρίξει στο να **μάθουν να λύνουν** προβλήματα. Λειτουργεί ως ένας ιδιόμορφος δάσκαλος που παρέχει στους μαθητές του «έξυπνα» εργαλεία τα οποία εκτός από ερεθίσματα προσφέρουν συχνή ανατροφοδότηση για το αποτέλεσμα των ενεργειών τους αλλά και ευκαιρίες για συζήτηση σε διάφορα στάδια της «ζώνης επικείμενης ανάπτυξης». Τα «έξυπνα» εργαλεία δημιουργούν «έξυπνους» μαθητές, ικανούς για κριτική και δημιουργική σκέψη.
- Ταυτόχρονα είναι δυνατή η λειτουργία εκπαιδευτικών προγραμμάτων τα οποία **αξιολογούν** με τρόπο προσαρμοσμένο στην εκπαιδευτική διαδικασία. Έτσι ακόμη και οι πιο αδύναμοι μαθητές θα λαμβάνουν την αξιολόγηση ως μια μη θεμιτή εξέλιξη της πράξης τους και όχι ως μηχανισμό περιθωριοποίησης και στιγματίσής τους ως κακούς μαθητές, μη έξυπνους κτλ.
- Παράλληλα η προσαρμογή της θεωρίας του Vygotsky στην εκπαίδευση συμβάλει στο να **μειωθεί η ανισότητα**, που υπάρχει ανάμεσα σε ένα δάσκαλο, ως μοναδική πηγή γνώσης μέσα στη τάξη, και τους μαθητές του.
- Τέλος ο υπολογιστής μπορεί να προγραμματιστεί ώστε να μαθαίνουν τα παιδιά **παίζοντας**. Μέσα από την ευχαρίστηση και τους κανόνες του παιχνιδιού μαθαίνουν διάφορες μορφές αυτοελέγχου. Πειθαρχώντας οικειοθελώς σε κανόνες παιχνιδιού, τα παιδιά όχι μόνο αποκτούν κάποιες γνώσεις, αλλά αναπτύσσουν και τη βάση της

μελλοντικής τους κοινωνικής ωριμότητας και ηθικής. Γι' αυτό είναι σημαντικό, οι κανόνες, οι ενισχύσεις, οι όροι και τα κίνητρα του παιχνιδιού να είναι παιδαγωγικά μελετημένα, ώστε να προάγουν την κοινωνική, την ηθική και προσωπική υγεία των παιδιών.

Ο εποικοδομητισμός του Bruner.

Ο J.Bruner ανήκει στη κατηγορία των γνωστικών ψυχολόγων μάθησης που δίνουν έμφαση στην διευκόλυνση της μάθησης μέσα από την ανακάλυψη. Ο ίδιος βέβαια επισημαίνει ότι *‘η ανακάλυψη χρησιμοποιήθηκε από μερικούς παιδαγωγούς σαν να ήταν αξία από μόνη της, άσχετα με το τι και για ποιόν ανακαλύπτεται’* (Bruner,1973a). Έτσι η ανακάλυψη δεν πρέπει να γίνεται αυτοσκοπός αλλά να οργανώνεται με βάση τους σκοπούς και τους στόχους του δασκάλου και του διδασκόμενου. Σύμφωνα με τον Bruner οι τρόποι σκέψης που χρησιμοποιεί ο μαθητευόμενος για να κατανοεί τις πληροφορίες και να αναπτύσσεται γνωστικά είναι:

- Το σύστημα της **πραξιακής αναπαράστασης** (enactive representation), αντίστοιχο προς εκείνο της αισθησιοκινητικής νοημοσύνης του Piaget, μάθηση δηλαδή μέσω της εμπειρίας.
- Το σύστημα της **εικονικής αναπαράστασης**, γνώση μέσω εσωτερικών πνευματικών εικόνων, αισθήσεων, συναισθημάτων κτλ.
- Το σύστημα της **συμβολικής αναπαράστασης**, όπου ο μαθητής μπορεί να κάνει συσχετισμούς ανάμεσα σε θεωρίες και σύμβολα καταλήγοντας στα δικά του συμπεράσματα.

Το παιδί χρησιμοποιώντας έναν από τους παραπάνω τρόπους σκέψης μπορεί να οδηγηθεί στη γνώση. Απαραίτητη προϋπόθεση είναι το παιδί να διδαχθεί στη «γλώσσα» που το παιδί καταλαβαίνει. Μιλώντας για γλώσσα ο Bruner δεν αναφερόταν στη λεκτική επικοινωνία αλλά στο σύνολο των κοινωνικοπολιτιστικών ερεθισμάτων με τα οποία το παιδί έχει γαλουχηθεί από την οικογένεια του και με τα οποία ως σύνολο έχει μάθει να λειτουργεί, να συμπεριφέρεται, να σκέφτεται, να κατανοεί τον κόσμο. Ο καλός δάσκαλος θα

πρέπει να ξεφύγει από το στατικό μοντέλο της παραδοσιακής μάθησης και να καταφέρει να φτιάξει ένα τέτοιο περιβάλλον μέσα στο οποίο κάθε παιδί θα μπορεί να βρει μια ανταπόκριση του κόσμου του.

Υποστηρίζει ακόμη ότι πρέπει να δίνεται η ευκαιρία στο μαθητή να έρχεται αντιμέτωπος με προβληματικές καταστάσεις και να ενεργεί κατά τρόπο παρόμοιο με εκείνο που θα ενεργούσε ένας επιστήμονας σε ανάλογη περίπτωση. Ο βαθμός δυσκολίας θα διαφέρει αλλά η στάση απέναντι στην απορία, το πρόβλημα και κατ' επέκταση τη λύση και τη γνώση πρέπει να είναι η ίδια. Η γνώση που έχει κατακτηθεί με το σωστό για το παιδί τρόπο αλλά και τη σωστή χρονική στιγμή, έχει περισσότερες πιθανότητες να γίνει κτήμα του παιδιού.

Εφαρμογές των απόψεων του Bruner στην διδασκαλία της Πληροφορικής

Μέσω της χρησιμοποίησης της θεωρίας του Bruner στο μάθημα της Πληροφορικής σήμερα μπορούμε να βοηθήσουμε τόσο τους μαθητές όσο και τους διδάσκοντες στον δημιουργικότερο δρόμο προς την γνώση και τη χρήση του ηλεκτρονικού υπολογιστή.

- Αντιμετωπίζοντας τα τυχόν προβλήματα και αδιέξοδα των μαθητών με επιστημονικό τρόπο και έρευνα, τα παιδιά μαθαίνουν να επιλύουν ουσιαστικά τα προβλήματά τους, ερευνώντας την καταλληλότερη λύση, ψάχνοντας και προσπαθώντας. Έτσι αναπτύσσουν θετική στάση απέναντι στην ακαδημαϊκή εργασία ενώ ταυτόχρονα ανυψώνεται το ηθικό και η αυτοπεποίθησή τους. Μέσω της προσπάθειάς τους για εξερεύνηση βρίσκονται αντιμέτωπα με νέα πεδία γνώσης τα οποία δεν είναι παρά καινούργιοι γνωστικοί ορίζοντες.
- Τα παιδιά μέσω του υπολογιστή και ενός *μπρουνερικού συστήματος* εκμάθησης Πληροφορικής έχουν την δυνατότητα να διαλέξουν μόνα τους το γνωστικό πεδίο στο οποίο θα αναπτυχθούν. Πολλοί

υποστηρίζουν πως τα παιδιά διαλέγουν απλά ο,τι είναι εύκολο για εκείνα, και άρα δεν οδηγούνται στη γνώση, αλλά σε μια απλή επανάληψη των ήδη γνωστών τους δομών. Ο Bruner απαντά πως η άνεση με το γνωστικό αντικείμενο που θα αναπτύξει ένα παιδί είναι απαραίτητο συναισθηματικό εφόδιο για την μετέπειτα πορεία του. Όπως είπαμε και προηγουμένως το παιδί θα πρέπει να αναγνωρίζει ένα κομμάτι του δικού του κόσμου σε ο,τι μελετά , προκειμένου να το βλέπει με ενδιαφέρον.

- Ο δάσκαλος θα πρέπει να έχει οργανώσει τέτοιου είδους προγράμματα που να οδηγούν τους μαθητές στην ανακάλυψη του επόμενου βήματος. Έτσι είτε μέσω συνεργασίας, είτε ατομικά τα παιδιά θα ικανοποιούν την ανάγκη τους για ανακάλυψη και πειραματισμό γεγονός που θα τα οδηγήσει στη γνώση.

Τεχνοκεντρικές διδακτικές προσεγγίσεις

Οι τεχνοκεντρικές διδακτικές προσεγγίσεις φαίνεται ότι κυριαρχούν στη διδασκαλία των μαθημάτων Πληροφορικής κυρίως στο Γυμνάσιο και στο Λύκειο. Η φιλοσοφία του Προγράμματος Σπουδών (ιδιαίτερα του μαθήματος Πληροφορικής στο Γυμνάσιο και του μαθήματος επιλογής στο Λύκειο), δεν εφαρμόζεται σε μεγάλη κλίμακα και δεν υπάρχει ιδιαίτερη διασύνδεση του μαθήματος με άλλα γνωστικά αντικείμενα. Είναι προφανές ότι η Πληροφορική δε μπορεί να προσεγγισθεί αποτελεσματικά με βάση τα παραδοσιακά διδακτικά μοντέλα, όπου ο δάσκαλος αποτελεί το φορέα γνώσης και ο μαθητής τον αποδέκτη.

Όπως αναφέρεται και στο ΔΕΠΠΣ η διδασκαλία δεν πρέπει να έχει γνωσιοκεντρικό χαρακτήρα. Η διδακτική πορεία θα βασίζεται στην **έμφυτη περιέργεια του μαθητή και στην αυτενέργειά του**. Θα πρέπει να παρακινείται από τον εκπαιδευτικό στο να προσδιορίζει και να **αξιοποιεί διάφορες πηγές** και μέσα πληροφόρησης. Να συνδυάζει τη θεωρία με την πράξη και να στοχεύει κυρίως στην απόκτηση κριτικών δεξιοτήτων και **δεξιοτήτων μεθοδολογικού**

χαρακτήρα. Κατά το σχεδιασμό των διδακτικών ενεργειών θα πρέπει να λαμβάνεται υπόψη ότι η κατανόηση των εννοιών και η απόκτηση ουσιαστικής γνώσης επιτυγχάνεται αν στηρίζεται σε προηγούμενες γνώσεις, εμπειρίες και βιώματα των μαθητών. Ο εκπαιδευτικός θα πρέπει να δίνει ευκαιρίες συμμετοχής και να προσφέρει υψηλής ποιότητας μορφωτικό αγαθό σε όλους τους μαθητές: αγόρια και κορίτσια, άτομα με ειδικές εκπαιδευτικές ανάγκες ή δυσκολίες, ανεξάρτητα από την κοινωνική ή την εθνική τους προέλευση και το πολιτισμικό τους υπόβαθρο. Μέσα από μια ενιαία και συνεχή δημιουργική διαδικασία, η διδασκαλία θα βοηθά τους μαθητές να ανακαλύπτουν οι ίδιοι τη γνώση, προτρέποντας και εθίζοντάς τους να αναπτύσσουν πρωτοβουλίες. Γενικότερα η διδασκαλία θα πρέπει να στοχεύει στην ολόπλευρη ανάπτυξη των μαθητών και να τους βοηθά προσαρμόζοντας και αξιοποιώντας τα σημερινά δεδομένα να οραματίζονται το δικό τους κόσμο και ένα καλύτερο αύριο.

Η διδασκαλία επίσης θα πρέπει να στηρίζεται στη **συμμετοχική μέθοδο**. Η ουσία του συμμετοχικού τρόπου βρίσκεται στην ανάπτυξη της συλλογικότητας και της πρωτοβουλίας των εκπαιδευομένων οι οποίοι αντιμετωπίζονται ως αυτόνομες προσωπικότητες. Συμμετέχουν ενεργά στη μαθησιακή διαδικασία και βρίσκονται σε συνεχή (διαρκή) σχέση αλληλεπίδρασης με τον εκπαιδευτικό. Επιδιώκεται με τον τρόπο αυτό η ανάπτυξη της δημιουργικότητας, της συνεργατικότητας και της ικανότητας επικοινωνίας.

Τα παραπάνω προϋποθέτουν τη χρήση μεθόδων που να προωθούν, να ενισχύουν και να ενθαρρύνουν:

- Την ενεργοποίηση του μαθητή και την εμπλοκή του σε διαδικασίες μέσα από τις οποίες θα κατακτά ο ίδιος τη γνώση
- Τη δημιουργική δράση και τον πειραματισμό
- Τη συνεργατική και ανακαλυπτική μάθηση
- Την ανάπτυξη ικανοτήτων και δεξιοτήτων μεθοδολογικού χαρακτήρα
- Τη συζήτηση, τον προβληματισμό και την καλλιέργεια κριτικής σκέψης
- Την καλλιέργεια ελεύθερης σκέψης και έκφρασης

- Τη μάθηση πάνω στο πώς μαθαίνουμε

Παράλληλα, με τις δραστηριότητες θα πρέπει να διευκολύνεται και να ενισχύεται:

- Η ανάπτυξη της ικανότητας του μαθητή να δημιουργεί.
- Ο συμμετοχικός-συνεργατικός χαρακτήρας της μάθησης.
- Η δυνατότητα αναλυτικής και συνθετικής σκέψης
- Η ανάπτυξη δεξιοτήτων μοντελοποίησης και τεχνικών επίλυσης προβλημάτων.
- Η ικανότητα στη χρήση συμβολικών μέσων έκφρασης και διερεύνησης.
- Η καλλιέργεια διαχρονικών δεξιοτήτων και δεξιοτήτων μεθοδολογικού χαρακτήρα.

Για να επιτευχθούν τα παραπάνω θα πρέπει, τα προβλήματα οι εργασίες και οι δραστηριότητες κατά το δυνατό να επιλέγονται από το χώρο των ενδιαφερόντων των μαθητών και να δίνουν ιδιαίτερη έμφαση στην ανάλυση και το σχεδιασμό της λύσης. Οι εργασίες θα πρέπει να είναι δομημένες κατά τέτοιο τρόπο ώστε να μπορούν να ολοκληρώνονται τμηματικά, να απαιτούν την ταυτόχρονη ενασχόληση των μαθητών της ομάδας και για την ολοκλήρωσή τους να απαιτούν πολύπλευρη προσέγγιση της γνώσης. Η ανάλυση και ο σχεδιασμός θα είναι η βάση της δουλειάς εκτός εργαστηρίου, στο σπίτι ή στην αίθουσα. Στην αίθουσα θα γίνεται επίσης, η εισαγωγή και η ανάπτυξη των διαφόρων εννοιών, παράλληλα με την πρακτική άσκηση στο εργαστήριο. Η ανάπτυξη της ύλης θα πρέπει να γίνεται με σπειροειδή τρόπο και να κατευθύνεται: από το ειδικό στο γενικό, από το απλό στο σύνθετο, από το εύκολο στο δύσκολο.

Ο τρόπος δόμησης και παρουσίασης των εννοιών θα πρέπει επίσης να στηρίζεται στη σύγχρονη μεθοδολογία διδασκαλίας κατά την οποία οι μαθητές δεν πρέπει να αντιμετωπίζονται ως παθητικοί δέκτες γνώσης, αλλά ως άτομα που αναπτύσσουν δικό τους κώδικα επικοινωνίας. Κατά τη διδασκαλία θα πρέπει να εφαρμόζονται διάφορες μέθοδοι και τεχνικές, οι οποίες έχουν ως

σκοπό να δημιουργούν κίνητρα για μάθηση, να διεγείρουν το ενδιαφέρον των μαθητών, να δημιουργούν κλίμα συνεργασίας, να προωθούν τη συμμετοχή.

Ενδεικτικότερες μέθοδοι διδασκαλίας αλλά και διδακτικές ενέργειες που σε σημαντικό βαθμό ικανοποιούν τους παραπάνω στόχους είναι:

- Κατευθυνόμενη ανακάλυψη,
- Καταιγισμός ιδεών
- Παιγνίδι ρόλων
- Μελέτες περιπτώσεων
- Εργασία σε ομάδες
- Προσχεδιασμένες εργασίες
- Επιστημονική μέθοδος

Στην όλη διαδικασία της μάθησης επιλέγεται κατά περίπτωση η προσφορότερη ή συνδυασμός των προσφορότερων.

Σημαντικό στοιχείο που πρέπει να λαμβάνεται υπόψη είναι ότι, η διδασκαλία γίνεται αποτελεσματικότερη αν πριν από την εισαγωγή των εννοιών και της σχετικής ορολογίας εξασφαλίζεται η εμπειρική γνώση και αναδεικνύεται η αναγκαιότητα της εισαγωγής τους. Θα πρέπει συνεπώς με κατάλληλα παραδείγματα ή προβλήματα να αναδεικνύεται η αναγκαιότητα της εισαγωγής των εννοιών και κατόπιν να εισάγονται οι έννοιες, η δε επεξεργασία τους να στηρίζεται στις προηγούμενες γνώσεις και εμπειρίες ώστε ο μαθητής να εντάσσει σταδιακά τη νέα γνώση στις ήδη υπάρχουσες. Οι γενικεύσεις επίσης θα πρέπει να υποστηρίζονται από παραδείγματα τα οποία θα αντλούν ιδέες από το περιβάλλον και τις προσωπικές εμπειρίες των μαθητών.

Δυσκολίες του τεχνοκεντρικού μοντέλου

Θα πρέπει να αναγνωρίσουμε ότι η αλλαγή αυτή συνιστά ένα δύσκολο διδακτικό εγχείρημα, στο οποίο παρεμβαίνουν ανασταλτικά:

- I. το συνολικό εκπαιδευτικό πλαίσιο στη ελληνική εκπαίδευση στο οποίο κυριαρχεί η διδασκαλία μετωπικού χαρακτήρα, ενώ απουσιάζουν οι διερευνητικές και συνεργατικές δραστηριότητες
- II. οι εμπειρικές διδακτικές προσεγγίσεις οι οποίες κυριαρχούν συνολικά στη σχολική πρακτική
- III. η ελλιπής προετοιμασία των καθηγητών της Πληροφορικής, οι οποίοι αναπαράγουν τα παραδοσιακά εκπαιδευτικά πρότυπα που ανακαλούν εύκολα από την εμπειρία των σπουδών τους (στη δευτεροβάθμια και στην τριτοβάθμια εκπαίδευση)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ

Κεφάλαιο 3 - Διδακτική του προγραμματισμού

Μέσα στο εκπαιδευτικό σύστημα, η Πληροφορική εμφανίζεται ταυτόχρονα ως αντικείμενο, που ανήκει σε ένα επιστημονικό χώρο με τις ιδιαίτερες έννοιές του, και ως εργαλείο που συμβάλει στην επίλυση προβλημάτων σε άλλους χώρους (Rogalski, 1988). Αυτή η διπλή ιδιότητα της Πληροφορικής ενέχει κινδύνους ως προς τον στόχο που έχει η διδακτική του προγραμματισμού. Έτσι, πρέπει να ξεκαθαρίσουμε πως η διδασκαλία του προγραμματισμού δεν αποσκοπεί στην προετοιμασία και εκπαίδευση μελλοντικών προγραμματιστών αλλά στην δημιουργία μιας κουλτούρας στην Πληροφορική ώστε οι εκπαιδευόμενοι να αποκτήσουν μια πιο βαθιά γνώση του πως λειτουργεί ένας υπολογιστής, πράγμα που θα τους βοηθήσει στην βελτιστοποίηση της χρήσης του.

Οι έρευνες των τελευταίων χρόνων στην διδασκαλία και την εκμάθηση του προγραμματισμού κατέληξαν πως οι βασικές δυσκολίες εκδηλώνονται στα εξής πεδία:

- Κατασκευή ενός αλγορίθμου ή ενός προγράμματος
- Περιγραφή-αιτιολόγηση ενός αλγορίθμου
- Πρόβλεψη συμπεριφοράς ενός αλγορίθμου
- Διόρθωση ενός αλγορίθμου

Η διεθνής εκπαιδευτική κοινότητα έχει διερευνήσει εκτεταμένα το θέμα και σήμερα γνωρίζουμε αρκετά καλά τα προβλήματα που αντιμετωπίζουν οι αρχάριοι προγραμματιστές στους τομείς αυτούς. Υπάρχουν πολλές θεωρίες που προσπαθούν να εξηγήσουν τι είναι αυτό που καθιστά την εκμάθηση του προγραμματισμού τόσο δύσκολη (Brusilovsky 1997, Du Boulay 1989, Brooks 1977).

Τα θέματα που κυρίως έχουν απασχολήσει την σχετική βιβλιογραφία είναι τα εξής:

- Οι βασικές έννοιες που αφορούν:
 - ο Μεταβλητές- σταθερές
 - ο Αριθμητικές εκφράσεις-Λογικές εκφράσεις
 - ο Ακολουθιακή δομή
 - ο Δομή ελέγχου
 - ο Δομή επανάληψης / βρόχου
 - ο Διαδικασία
 - ο Αναδρομική διαδικασία
- Η εισαγωγή στο λογικό και τον παράλληλο προγραμματισμό.
- Ο κύκλος ανάπτυξης των προγραμμάτων
- Συναρτήσεις – διαδικασίες
- Δομές δεδομένων

Τα διδακτικά προβλήματα εντοπίζονται στις πιο κάτω δραστηριότητες:

- Αναγνώριση και χρήση μεταβλητών και σταθερών.
- Δηλώσεις και τύποι μεταβλητών.
- Εύρεση μετά από εκχώρηση ή ανάθεση περιεχομένων των μεταβλητών.
- Παρακολούθηση δομών απλής, πολύπλοκης και πολλαπλής διακλάδωσης (IF THEN, IF THEN ELSE και CASE OF).
- Αναγνώριση συνθηκών (OR, AND και NOT).
- Αναπαράσταση δομής επανάληψης.
- Υλοποίηση κώδικα σε δομές επανάληψης με συνθήκη (repeat until - do while).
- Αρχικοποίηση μεταβλητών σε δομή επανάληψης με συνθήκη
- Δομή επανάληψης με συνθήκη, καταμετρητής και αθροιστής.
- Αναγνώριση και διάκριση μεταξύ τυπικών παραμέτρων τιμών και τυπικών παραμέτρων αναφοράς.
- Νοητικό τρέξιμο κώδικα.
- Σύγκριση κώδικα (εννοιολογικές δομές).
- Συμπλήρωση και συγγραφή μέρους κώδικα (Διαδικασία – Procedure).

- Εύρεση αριθμητικών, συντακτικών και λογικών λαθών.
- Μεταβλητές εισόδου και εξόδου σε συναρτήσεις.
- Εντοπισμός λογικού λάθους σε διαδικασίες
- Αναγνώριση και επεξεργασία δομής δεδομένων (Πίνακας-Array)
- Φωλιασμένες (nested loops) δομές επανάληψης με καταμετρητή και τεχνικές ταξινόμησης σε δομές δεδομένων τύπου πίνακα

Κατά την διδασκαλία του προγραμματισμού πρέπει, επίσης, να αποσαφηνιστεί η δομή ενός προγράμματος, δομή που θεωρείται με σπονδυλωτή και ιεραρχική μορφή. Για να το πετύχουμε απαιτείται μια μεθοδολογική προσέγγιση από τα πρώτα βήματα εκμάθησης του προγραμματισμού, με στόχο να μην αποκτηθούν εμπειρικές και μη δομημένες προγραμματιστικές συνήθειες και ταυτόχρονα να αναπτυχθούν τεχνικές επίλυσης προβλημάτων.

Στην συνέχεια του κεφαλαίου θα δούμε τα παραπάνω θέματα λεπτομερώς.

Μεταβλητή

Η έννοια της μεταβλητής είναι ένα από τα πρώτα και πιο σημαντικά προβλήματα που συναντούν οι μαθητές στη διδασκαλία του προγραμματισμού. Πολλοί ερευνητές έχουν ασχοληθεί με την έννοια της μεταβλητής και το ρόλο της στην αλγοριθμική και τον προγραμματισμό, όπως οι (Schneiderman, 1980), (Mayer, 1981), (Samurçay, 1989), (Dagdilelis, 1986) και (Rogalski, 1989).

Η οικοδόμηση της έννοιας της μεταβλητής από τους μαθητές έχει πρωταρχική σημασία στον προγραμματισμό, καθώς συμβάλλει καθοριστικά στην οικοδόμηση πιο σύνθετων δομών (Τζιμογιάννης, 2000).

Η πλειονότητα των γλωσσών προγραμματισμού βασίζεται στην ίδια θεμελιώδη τεχνική, αυτή της «διαχείρισης τιμών που περιέχονται σε μεταβλητές» (Delanoy, 1996). Η μεταβλητή ορίζεται ως μια ποσότητα που μπορεί να έχει διαφορετικές τιμές όταν εκτελείται το πρόγραμμα (Knuth, 1968) και χαρακτηριστικά της είναι

έκτος από την προαναφερόμενη τιμή, ένα όνομα, ένα σύνολο ιδιοτήτων και μια αναφορά, δηλαδή η θέση μνήμης που την προσδιορίζει (Horowitz, 1984).

Όνομα μεταβλητής

Το ζήτημα της αποθήκευσης των δεδομένων με τη μορφή μεταβλητών οι οποίες γίνονται αντιληπτές με τη χρήση συμβολικών κωδικών (όνομα μεταβλητής) συνιστά ένα δύσκολο πρόβλημα (Dufoyer, 1988) το οποίο γίνεται ακόμα πιο δύσκολο, αφού η έννοια της αποθήκευσης εμπερικλείει τόσο την εκχώρηση τιμής όσο και την εμφάνισή της, δηλαδή λειτουργικά χαρακτηριστικά της έννοιας που δεν μπορούν εύκολα να διαχωριστούν. Η επιλογή κατάλληλου ονόματος μεταβλητής μπορεί να διευκολύνει τη κατανόηση της ίδιας της έννοιας της μεταβλητής στον προγραμματισμό ως ονομασία μιας θέσης μνήμης (Rogalski, 1987). Η χρήση ενός ονόματος αντί για μια διεύθυνση μνήμης παρέχει μεγαλύτερη ελευθερία χειρισμού και κυρίως διευκολύνει την αναγνωσιμότητα των προγραμμάτων. Η χρήση κανόνων ονοματοδοσίας (coding standards) αποτελεί την καλύτερη προγραμματιστική τακτική για την αποσαφήνιση των μεταβλητών του προγράμματος.

Τύπος μεταβλητής

Σύμφωνα με τον Schneiderman υπάρχουν τέσσερις διαφορετικοί τύποι μεταβλητών (Schneiderman, 1980), ανάλογα με την χρήση τους:

- Μεταβλητές μετρητές, για χρήση σε βρόχους
- Μεταβλητές υπολογισμού
- Μεταβλητές αθροισμάτων
- Μεταβλητές γενικής χρήσης

Σύμφωνα με μια άλλη κατηγοριοποίηση, οι μεταβλητές χωρίζονται σε δύο κατηγορίες (Samurçay, 1989):

- Εξωτερικές
Οι εξωτερικές μεταβλητές αντιστοιχούν σε τιμές που ελέγχονται από τους χρήστες του προγράμματος όπως για παράδειγμα οι μεταβλητές

που αντιστοιχούν σε εισόδους (input) και εξόδους (output) του προβλήματος και κατά κανόνα παρουσιάζουν τα λιγότερα διδακτικά προβλήματα. Εκφράζουν διασαφηνισμένα δεδομένα του προβλήματος και είναι υπό τον έλεγχο του χρήστη, όταν εκτελεί το πρόγραμμα.

- Εσωτερικές

Οι εσωτερικές μεταβλητές χρησιμοποιούνται μέσα στον κώδικα ενός προγράμματος, δεν είναι ορατές από τον χρήστη, είναι αναγκαίες μόνο στη (προγραμματιστική) λύση ενός προβλήματος και δεν έχουν σημασιολογικό αντίστοιχο όταν το πρόβλημα επιλύεται με το χέρι. Είναι υπό τον έλεγχο του προγραμματιστή. Παράδειγμα αποτελούν οι μεταβλητές που απαιτούνται για την αλλαγή θέσης (αντιμετάθεση) δύο τιμών (η τιμή του x να τοποθετηθεί στο y και η τιμή του y στο x).

Η δεύτερη κατηγορία προξενεί μεγάλες δυσκολίες στους αρχάριους προγραμματιστές αφού, για να τις διαχειριστούν, πρέπει να μπορούν να κατανοήσουν και να ακολουθήσουν την εκτέλεση του προγράμματος και την εσωτερική λειτουργία του υπολογιστή. Συνεπώς, η απουσία κατάλληλων αναπαραστάσεων του συστήματος «υπολογιστής – γλώσσα προγραμματισμού» αποτελεί μία από τις βασικές αδυναμίες των αρχάριων (Lagrange, 1992).

Η έννοια του τύπου μιας μεταβλητής απαιτεί ιδιαίτερη προσοχή όταν προσεγγίζεται διδακτικά, αφού ο τύπος προσδιορίζει το είδος των πληροφοριών που αναπαρίστανται από τη μεταβλητή, την κωδικοποίηση, τους περιορισμούς που αφορούν στις τιμές που μπορούν να αναπαρασταθούν και τις δυνατές πράξεις που μπορούν να πραγματοποιηθούν από τις αντίστοιχες μεταβλητές. Οι αρχάριοι προγραμματιστές φαίνεται να προτιμούν τη χρήση μόνο των προκαθορισμένων από τη χρησιμοποιούμενη γλώσσα τύπων (Lagrange, 1990).

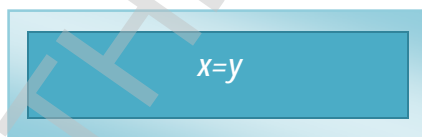
Εκχώρηση τιμών σε μεταβλητές

Γενικότερα, μπορούμε να διακρίνουμε τέσσερις διαφορετικούς τύπους εκχώρησης τιμής μιας μεταβλητής, οι οποίοι ενέχουν διαφορετικού επιπέδου διδακτικά προβλήματα (Samurçay, 1989):

1. Ανάθεση σταθερής τιμής (constant value)
π.χ. $x=10$, $list="class"$, $test=false$
2. Ανάθεση τιμής που προκύπτει από υπολογισμό (calculated value)
π.χ. $a=10+5$, $y=2*4+3$
3. Αντιγραφή (duplication)
π.χ. $x=y$
4. Συσσώρευση (accumulation)
π.χ. $x=x+1$, $sum=sum+n$, $exprn=exprn*n$

Οι τρεις πρώτες περιπτώσεις δεν εμφανίζουν ιδιαίτερα διδακτικά προβλήματα αφού οι κατάλληλες αναπαραστάσεις που απαιτούνται έχουν ήδη οικοδομηθεί από τον χώρο των μαθηματικών. Πρέπει, παρόλ' αυτά, να δοθεί προσοχή στην ανάγνωση της εντολής από το τέλος προς την αρχή έτσι ώστε να έχουμε την σωστή σειρά των πράξεων.

Για παράδειγμα, στην εντολή:


$$x=y$$

το πρόβλημα εντοπίζεται στην τιμή «πηγή» και στην τιμή «στόχο». Στην άσκηση «Βάλε την τιμή του y στο x » πολλές φορές η απάντηση των μαθητών είναι $y=x$ που είναι βεβαίως λανθασμένη.

Ένα πιο σύνθετο πρόβλημα είναι αυτό της αντιμετάθεσης τιμών δύο μεταβλητών που συνιστά ένα συχνό πρόβλημα στον προγραμματισμό (Τζιμογιάννης, 2000). Η περίπτωση αυτή μελετάται επιπρόσθετα, επειδή απαιτεί τη χρήση βοηθητικής μεταβλητής. Ο αλγόριθμος που συχνά χρησιμοποιούν οι μαθητές είναι ο παρακάτω (γλώσσα παραδείγματος Pascal):


```
Program Algorithm1;  
Var  
X, Y: Integer;  
Begin  
X:=5; Y:=10;  
X:=Y; Y:=X;  
Writeln('X= ', X, ' Y= ', Y)  
end.
```

Το σκεπτικό του μαθητή περιγράφεται απλά ως εξής: «βάλε όπου A το B και όπου B το A». Το λάθος έγκειται στο ότι δεν κρατήθηκε σε κάποια προσωρινή θέση μνήμης η αρχική τιμή του A. Έτσι οι τιμές των μεταβλητών στο τέλος της εκτέλεσης του προγράμματος θα είναι $X = 10$ και $Y = 10$.

Παρακάτω, παρατίθεται το σωστό πρόγραμμα χρησιμοποιείται η εσωτερική μεταβλητή Temp.

```
Program Algorithm2;  
Var  
X, Y, Temp: Integer;  
Begin  
X:=5; Y:=10;  
Temp:=Y;  
Y:=X;  
X:= Temp;  
Writeln('X= ', X, ' Y= ', Y)  
end.
```

Η αναπαράσταση της μνήμης του υπολογιστή με τις εκάστοτε τιμές των μεταβλητών του προγράμματος και η εκτέλεση του αλγορίθμου επίλυσης του προβλήματος στο χαρτί θα βοηθήσουν τους μαθητές στην κατανόηση και αποφυγή του συγκεκριμένου λάθους (Αλεβίζος, 1988).

Οι Bayman και Mayer (1983) καταλήγουν στο συμπέρασμα ότι δύο είδη εκχώρησης τιμών σε μεταβλητές μπορεί να οδηγήσουν σε παρερμηνείες. Η πρώτη περίπτωση είναι η αρχικοποίηση ($LET X=0$) και η άλλη είναι η εξίσωση ($LET X=Y+1$). Στις περιπτώσεις αυτές οι μαθητές συχνά θεωρούν ότι ο υπολογιστής έχει καταγράψει κάπου την πληροφορία ή την έχει εκτυπώσει στην οθόνη ενώ αντίθετα η πληροφορία έχει αποθηκευθεί σε συγκεκριμένη θέση στη μνήμη. Πολλοί μαθητές θεωρούν ότι αποθηκεύεται η εξίσωση και όχι η τιμή. Οι συγγραφείς καταλήγουν στο συμπέρασμα ότι οι αρχάριοι προγραμματιστές χρειάζονται ειδική εκπαίδευση στα θέματα που αφορούν θέσεις μνήμης και κάτω από ποιες συνθήκες οι τιμές που αποθηκεύονται στις θέσεις αυτές μπορούν να αντικατασταθούν.

Ειδικότερα, η συσώρευση είναι η περίπτωση όπου η έννοια της μεταβλητής προκαλεί τη μεγαλύτερη σύγχυση. Στην περίπτωση αυτή πρέπει να γίνει σαφής διάκριση μεταξύ αριστερού και δεξιού τμήματος της εντολής ανάθεσης, και να κατανοήσουν οι μαθητές ότι το δεξί τμήμα σχετίζεται με την τιμή που θα πάρει η μεταβλητή ενώ το αριστερό τμήμα με τη θέση μνήμης. Εδώ παρουσιάζεται η πλήρης λειτουργικότητα της μεταβλητής στην Πληροφορική και η διάκριση της από τα μαθηματικά. Το εμπόδιο αυτό ανακύπτει κυρίως σε αρχάριους προγραμματιστές οι οποίοι χρησιμοποιούν έγκυρες αναπαραστάσεις της μεταβλητής που έχουν οικοδομήσει στο χώρο των μαθηματικών, οι οποίες στον προγραμματισμό δεν είναι πάντα έγκυρες.

Το πιο συνηθισμένο λάθος είναι η χρήση της εντολής εκχώρησης ως ισότητας. Για παράδειγμα, αν η εντολή:

```
X=X+Y;
```

εκληφθεί ως μαθηματική ισότητα ο μαθητής οδηγείται σε λογικό αδιέξοδο. Σ' αυτό το παράδειγμα επανέρχεται το ζήτημα της ανάγνωσης της εντολής από το τέλος προς την αρχή που αναφέραμε πιο πάνω έτσι ώστε να έχουμε την σωστή χρονική σειρά των πράξεων. Δηλαδή, αρχικά προσθέτουμε τις μεταβλητές X και Y (εσωτερικά το αποτέλεσμα αποθηκεύεται σε μια ενδιάμεση μεταβλητή) και στη συνέχεια εκχωρούμε το αποτέλεσμα της πρόσθεσης στην μεταβλητή X. Ουσιαστικά έχουμε δύο πράξεις που γίνονται σε δύο διαφορετικές χρονικές στιγμές

Πρέπει να αναφερθεί επίσης ότι στις γλώσσες υψηλού επιπέδου η προσωρινή αποθήκευση κάποιας τιμής γίνεται απευθείας από το σύστημα, οπότε ο προγραμματιστής δεν χρειάζεται να ασχοληθεί με την συγκεκριμένη λειτουργία. Όταν γράφουμε στην Pascal:

```
X:=X+Y;
```

ορισμένες αρχές της έννοιας της διατήρησης παραβιάζονται. Σε αυτό το παράδειγμα, η μεταβλητή X βρίσκεται και στις δύο πλευρές της σχέσης. Υπάρχει συνεπώς κάποια στιγμή όπου η τιμή X+Y θα έπρεπε να βρίσκεται κάπου πριν εισαχθεί στην Sum. Ο μαθητής θα κατανοούσε καλύτερα τη σύνταξη εάν είχαμε γράψει:

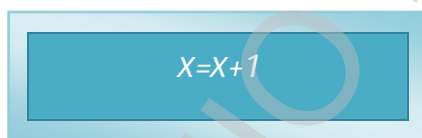
```
Temp:=X+Y;  
X:= Temp;
```

Σύμβολο ανάθεσης τιμής

Ένα ακόμη πρόβλημα είναι η μη ταύτιση της έννοιας της μαθηματικής μεταβλητής με αυτήν της μεταβλητής στον προγραμματισμό. Έχει αποδειχθεί ότι οι μαθητές διατηρούν και μεταφέρουν τη μαθηματική αναπαράσταση για την έννοια της μεταβλητής, ακόμη και μετά από πολλές ώρες διδασκαλίας στον προγραμματισμό (Τζιμογιάννης, 2000).

Σε πολλές γλώσσες προγραμματισμού για την εκχώρηση τιμής σε μια μεταβλητή χρησιμοποιείται το σύμβολο "=", πράγμα που δημιουργεί σύγχυση σε σχέση με το αντίστοιχο σύμβολο στα μαθηματικά.

Έτσι, η έκφραση


$$X=X+1$$

στον προγραμματισμό, παριστάνει μια εκχώρηση τιμής στην μεταβλητή X , ενώ στα μαθηματικά δεν έχει νόημα, αφού δεν είναι δυνατό ένας αριθμός X να είναι ίσος με τον εαυτό του συν τη μονάδα. Λύση στο πρόβλημα είναι η χρησιμοποίηση άλλου συμβολισμού (π.χ. το $:=$ στην PASCAL) ώστε να γίνεται εμφανής η διάκριση.

Από διδακτική σκοπιά, όταν προσεγγίζεται η έννοια της εκχώρησης τιμής μεταβλητής, είναι σκόπιμο, σε επίπεδο ψευδογλώσσας, να χρησιμοποιούνται πιο ουδέτεροι συμβολισμοί όπως για παράδειγμα το σύμβολο \leftarrow . Με τη χρήση ενός τέτοιου συμβολισμού γίνεται πιο ευδιάκριτη η έννοια εντολής εκχώρησης τιμής μεταβλητής (Arsac, 1980) (Du Boulay, 1986).

Παρόμοιο πρόβλημα είναι η σύγχυση του συμβόλου "=" της εκχώρησης τιμής σε μια μεταβλητή και η χρήση του ως σύμβολο στον έλεγχο ισότητας δύο μεταβλητών, αντί για το σωστό "==".

Στο παράδειγμα (γλώσσα παραδείγματος C):

```
If (x=1)  
then printf("Equal");
```

το αποτέλεσμα του ελέγχου είναι πάντα αληθές, αφού εκχωρείται η τιμή 1 στην μεταβλητή x. Για να αποφευχθεί αυτό το λάθος προτείνεται, στην σύνταξη του ελέγχου ισότητας, να γράφεται στο πρώτο μέρος η τιμή και στο δεύτερο η μεταβλητή ώστε να έχει τη μορφή:

τιμή σταθεράς = μεταβλητή

Με αυτόν τον τρόπο η χρήση του λάθος συμβόλου θα επιφέρει συντακτικό λάθος.

Διδακτικά προβλήματα

Ο (Du Boulay, 1989) θεωρεί ότι οι παρερμηνείες στην κατανόηση των μεταβλητών βασίζονται στα παραδείγματα (παρομοιώσεις-αναλογίες) που χρησιμοποιούν οι εκπαιδευτικοί στην τάξη. Για παράδειγμα η παρομοίωση της μεταβλητής με ένα κουτί ή ένα συρτάρι με μια ετικέτα, μπορεί να παρερμηνευτεί από τους μαθητές και να θεωρήσουν ότι η μεταβλητή μπορεί να έχει περισσότερες από μία τιμές. Έτσι ορισμένοι μαθητές θεωρούν ότι η μεταβλητή «θυμάται» κάθε τιμή που της δίνεται, εκτός αν σβηστούν τα περιεχόμενα της μνήμης. Δεν κατανοούν ότι η νέα τιμή καταχωρείται στη θέση της παλιάς η οποία και χάνεται.

Μια άλλη περίπτωση, που αναφέρει ο Du Boulay, είναι η παρομοίωση της μεταβλητής με μια πλάκα (slate) όπου πάνω μπορούν να γραφούν τιμές. Και αυτή η αναλογία μπορεί να παρερμηνευτεί από μαθητές, που δεν μπορούν να αντιληφθούν ότι η υπάρχουσα τιμή επικαλύπτεται από τη νέα. Οι μαθητές αυτοί θεωρούν τη μεταβλητή σαν μια λίστα που περιέχει όλες τις τιμές που έχουν εκχωρηθεί στη μεταβλητή, οι οποίες και μπορούν να ανακτηθούν.

Μια συνηθισμένη παρερμηνεία της εντολής ανάθεσης τιμής $A = B$ είναι η ακόλουθη: Το B συνδέεται με το A και έτσι όποιες αλλαγές πρόκειται να γίνουν στο A θα επηρεάσουν άμεσα και το B . Παρατηρείται επίσης ότι ένα συχνό λάθος είναι η μη απόδοση αρχικής τιμής (λ.χ. 0 ή άλλη τιμή) σε μετρητές βρόχων. Στην περίπτωση αυτή οι μαθητές μπερδεύονται με την αναλογία μεταβλητής – κουτιού, όπου το κουτί είναι άδειο έως ότου τοποθετήσουμε κάτι μέσα του, οπότε ανάλογα θεωρούν ως 0 και το περιεχόμενο της μεταβλητής.

Δομές ελέγχου

Οι βασικές εντολές χειρισμού μεταβλητών (εκχώρηση, ανάγνωση και εγγραφή τιμής) δεν αρκούν ώστε να επιλυθεί ένα πρόβλημα με προγραμματισμό. Απαιτούνται επίσης εντολές (ακολουθία, επιλογή, επανάληψη) που αφορούν στην αυστηρή δόμηση των παραπάνω βασικών εντολών (Τζιμογιάννης, 1999). Έτσι, το πρόγραμμα διαθέτει μια αυστηρή αλληλουχία εντολών και για κάθε είσοδο δεδομένων το σύνολο των περιγραφόμενων υπολογισμών πραγματοποιείται με τη σειρά γραφής των εντολών. Οι δομές ελέγχου διακόπτουν την γραμμική εξέλιξη της εκτέλεσής του προγράμματος σε σχέση με το κείμενο του προγράμματος (Rogalski, 1987). Στην περίπτωση των δομών ελέγχου, μία είσοδο δεδομένων θα ακολουθηθεί μόνο μία από τις δυνατές διαδρομές κατά την εκτέλεση.

Στο πλαίσιο όπου μια δομή ελέγχου οφείλει να διαχειριστεί μια τροποποίηση των σχέσεων ανάμεσα στη γραμμική γραφή του προγράμματος και στο σειριακό τρόπο (ακολουθιακή δομή) της εκτέλεσης, οι αναπαραστάσεις που έχει ο μαθητής πάνω στη σειριακή αυτή σχέση θα αλληλεπιδράσουν με τις σημασιολογικές και τις συντακτικές ιδιότητες του προγράμματος. Συνεπώς, ιδιαίτερες γνωστικές δυσκολίες θα προστεθούν στις γενικές γνωστικές δυσκολίες που αφορούν στην πρόσκτηση του προγραμματισμού.

Ο χώρος προβλημάτων που απαιτούν μια δομή ελέγχου, μπορεί να οριστεί με πολύ γενικό τρόπο ως τα προβλήματα στα οποία μία επεξεργασία εξαρτάται

από την τιμή μιας παραμέτρου (Rogalski, 1988). Η εξάρτηση μπορεί να έχει τη μορφή μιας απλής υπό συνθήκη εντολής:

```
IF συνθήκη  
THEN επεξεργασία 1  
ELSE επεξεργασία 2
```

Ή μπορεί να είναι πολλαπλή με πολλές συνθήκες και πολλές αποφάσεις:

```
IF συνθήκη 1 THEN επεξεργασία 1  
IF συνθήκη 2 THEN επεξεργασία 2  
...  
IF συνθήκη N THEN επεξεργασία N  
ELSE επεξεργασία N+1
```

Ανάλογα με τη φύση των συνθηκών από τις οποίες εξαρτώνται οι έλεγχοι μπορούμε τους διαχωρίσουμε σε αυτούς που στηρίζονται σε:

- ενδογενείς συνθήκες
δηλαδή, ορισμένη από το αποτέλεσμα ενός εσωτερικού υπολογισμού
- εξωγενείς συνθήκες
ορισμένη από τα δεδομένα εισόδου που δόθηκαν από το χρήστη

Οι έρευνες με μαθητές έχουν δείξει ότι ο ενδογενής ή εξωγενής χαρακτήρας των συνθηκών παράγει παρόμοια αποτελέσματα πάνω στην επεξεργασία από αρχάριους προγραμματιστές όπως με αυτά της μεταβλητής (Rogalski, 1988). Οι αρχάριοι προγραμματιστές συναντούν μεγαλύτερες δυσκολίες στις ενδογενείς παρά στις εξωγενείς συνθήκες.

Η φύση των «πληροφορικών αντικειμένων» που εμπλέκονται με σαφή τρόπο μέσα στη συνθήκη μπορεί να ποικίλει: σχέσεις που αφορούν μεταβλητές όπως

οι αριθμοί ή οι χαρακτήρες οδηγούν σε οικεία νοητικά σχήματα, η χρήση λογικών (Boolean) μεταβλητών απαιτεί νέες αναπαραστάσεις, η παρέμβαση συναρτήσεων οδηγεί σε αλλαγή γνωστικού επιπέδου. Έτσι, έχει παρατηρηθεί ότι κατά τη συγγραφή ελέγχων (απλή υπό συνθήκη εντολή) από μαθητές 15-16 ετών σπάνια γίνονται λάθη τα οποία αφορούν απλούς αριθμητικούς ελέγχους (όπως $M > 0$). Σχετικά περισσότερα λάθη γίνονται με πιο σύνθετες αριθμητικές δομές (που αναπαριστούν διάρκεια) ή δεδομένα τύπου χαρακτήρων (string) για τα οποία οι μαθητές δεν διαθέτουν τις ίδιες πρότερες γνώσεις (Rogalski, 1988) (Τζιμογιάννης, 1999).

Ανάλογα με την υλοποίηση του κώδικα μέσα σε μια γλώσσα προγραμματισμού τα προβλήματα που μπορεί να προκύψουν είναι τα εξής:

- Η χρησιμοποιούμενη σύνταξη μπορεί να απομακρύνεται από τη φυσική γλώσσα, γεγονός που οδηγεί σε ανάλογα διδακτικά εμπόδια.
- Γλωσσολογικές γνώσεις μπορούν να διευκολύνουν την πρόσβαση στη συντακτική φόρμα ή μπορεί, κατά περίπτωση, να αποτελέσουν εμπόδιο στο μέτρο που οι κανόνες λειτουργίας της φυσικής γλώσσας και αυτοί κάποιων γλωσσών προγραμματισμού δεν συνάδουν απόλυτα.

Στα παραπάνω πρέπει να προστεθεί η πολυπλοκότητα που δημιουργεί η αλληλουχία και το εμφώλιασμα δομών ελέγχου μέσα σε άλλες δομές ελέγχου ή επανάληψης. (Rogalski, 1987).

Διδακτικό παράδειγμα: Σύγκριση και εκτύπωση του μεγαλύτερου από δύο αριθμούς (Καβακλή, 2001).

Δίνονται τρεις δυνατές λύσεις που κάνουν χρήση

- ενός IF ... THEN
- δύο IF ... THEN και
- ενός IF ... THEN ... ELSE

(Αλεβίζος και Καμπουρέλης, 1988).

1^η περίπτωση:

```
program megalyteros1;  
var  
X, Y, Meg: Real;  
begin  
Writeln('Δώσε δύο αριθμούς');  
ReadLn(X, Y);  
Meg:= Y;  
if X > Y then Meg:=X;  
Writeln('Μεγαλύτερος είναι ο αριθμός: ' Meg)  
end.
```

Γίνεται χρήση μιας εντολής εκχώρησης `Meg:= Y;` που κάνει την υπόθεση ό,τι ο μεγαλύτερος αριθμός είναι ο δεύτερος. Στη συνέχεια εκτελείται η εσωτερική εντολή `IF ... THEN` και στην περίπτωση που η υπόθεση δεν αληθεύει τότε διορθώνεται η αρχική τιμή της μεταβλητής `Meg`.

2^η περίπτωση:

```
program megalyteros2;  
var  
X, Y, Meg: Real;  
begin  
Writeln('Δώσε δύο αριθμούς');  
ReadLn(X, Y);  
if X <= Y then Meg:=Y;  
if X > Y then Meg:=X;  
Writeln('Μεγαλύτερος είναι ο αριθμός: ' Meg)  
end.
```

Στην περίπτωση αυτή δεν γίνεται αυθαίρετη χρήση της εντολής $Meg:=Y$ αλλά χρησιμοποιείται ως εσωτερική της εντολής $IF \dots THEN$. Και στις δύο περιπτώσεις οι χρησιμοποιούμενοι αλγόριθμοι δεν είναι κοντά στη φυσική γλώσσα αλλά χρησιμοποιούν κυρίως τις ιδιότητες του πληροφορικού συστήματος: καταχωρείται μία τιμή στη μεταβλητή Meg και αλλάζει μόνο όταν δεν ισχύει η λογική συνθήκη (για την πρώτη περίπτωση). Ο επόμενος αντίθετα αλγόριθμος που υλοποιείται με τη χρήση της εντολής $IF \dots THEN \dots ELSE$ είναι πιο κοντά στη φυσική γλώσσα.

3^η περίπτωση:

```
program megalyteros3;  
var  
X, Y, Meg: Real;  
begin  
Writeln ('Δώσε δύο αριθμούς');  
ReadLn (X, Y);  
if X > Y then Meg:=X else Meg:=Y;  
Writeln('Μεγαλύτερος είναι ο αριθμός: ' Meg)  
end.
```

Ο αλγόριθμος αυτός όχι μόνο συνάδει με τη φυσική γλώσσα αλλά είναι και πιο οικονομικός αφού απαιτεί μία ακριβώς σύγκριση και μία εκχώρηση. Αντίθετα, στην πρώτη περίπτωση απαιτείται μία σύγκριση και 1,5 εκχώρηση κατά μέσο όρο, αφού η πρώτη εκχώρηση θα γίνει οπωσδήποτε και η δεύτερη (εάν $X>Y$) έχει 50% πιθανότητα. Στην δεύτερη περίπτωση θα γίνουν δύο συγκρίσεις ($X=<Y$ και $X>Y$) και μόνο μία εντολή εκχώρησης.

Προϋπάρχουσες γνώσεις λογικής και μαθηματικών

Σημαντικό ρόλο στην πρόσκτηση της έννοιας της υπό συνθήκη εντολής διαδραματίζουν και οι προϋπάρχουσες μαθηματικές και γνώσεις λογικής

(Bonar, 1985). Για την κατανόηση της υπό συνθήκη εντολής χρειάζονται ικανότητες στη χρήση των λογικών τελεστών (and, or, not) της άλγεβρας Boole. Σε γενικό επίπεδο, οι έρευνες δείχνουν ότι οι μαθητές που έχουν ένα ανεπτυγμένο μαθηματικό υπόβαθρο, οικοδομούν αυτού του τύπου τις δομές πιο γρήγορα.

Σημαντικό ρόλο παίζει, επίσης, η κατανόηση της μη γραμμικής εκτέλεσης ενός προγράμματος από το πληροφορικό μέσο. Οι αναπαραστάσεις των μαθητών πάνω στη σειριακή εκτέλεση του προγράμματος από το πληροφορικό μέσο δεν φαίνεται να ανασχηματίζονται από τις απλές αλληλεπιδράσεις ανάμεσα στη χρήση από τα υποκείμενα των δομών ελέγχου. Για το λόγο αυτό απαιτούνται κατάλληλες διδακτικές παρεμβάσεις. χρησιμοποιούνται συντακτικά σύμβολα (BEGIN ... END) που ορίζουν το χώρο αναφοράς των IF και THEN, συντελούν στην ευκολότερη χρήση της δομής ελέγχου σε σχέση με τη δομή άλματος και διευκολύνουν στην κατανόηση των προγραμμάτων.

Δομές επανάληψης

Επαναληπτική διαδικασία (βρόχος) ονομάζεται η επανάληψη μίας επεξεργασίας για ορισμένες φορές ή μέχρι να γίνει αληθής κάποια συνθήκη. Όπως και στη δομή ελέγχου, η επανάληψη διακόπτει την ακολουθιακή εκτέλεση του προγράμματος με όλες τις συνέπειες που επιφέρει αυτό το γεγονός σε γνωστικό και διδακτικό επίπεδο. Η επαναληπτική δομή αποτελεί μια από τις βασικές προγραμματιστικές δομές και σύμφωνα με διάφορες έρευνες παρουσιάζει ιδιαίτερα διδακτικά προβλήματα (Kessler, 1986) (Spohrer, 1989) (Dagdillelis, 1990).

Σε μία επανάληψη ο αριθμός των επαναλήψεων μπορεί (Soloway, 1983):

- να είναι γνωστός εκ των προτέρων ή
- να εξαρτάται από μια κατάσταση που εμφανίζεται κατά τη διάρκεια της επεξεργασίας

Στην πρώτη περίπτωση ο έλεγχος που τερματίζει το βρόχο είναι μια σταθερά. Όταν είναι εκ των προτέρων γνωστός ο αριθμός των επαναλήψεων χρησιμοποιείται η **εντολή FOR**, η οποία απαντάται πολύ συχνά στον προγραμματισμό και αποτελεί δομή όλων πρακτικά των γλωσσών.

Διδακτικό παράδειγμα: Εκτύπωση στην οθόνη 10 φορές του string "Hello".
(γλώσσα παραδείγματος Pascal)

```
program Loop;  
var  
  i: Integer;  
begin  
  for i := 1 to 10 do  
    Writeln('Hello');  
end.
```

Στην γενικευμένη μορφή της η εντολή περιγράφεται (σε συμβολισμό BNF) ως εξής:

for <μεταβλητή>:=<αρχική-τιμή> step <βήμα> until <τελική-τιμή> do <εντολή>

Σύμφωνα με τον Horowitz (1984) ερωτήσεις που πρέπει να απαντηθούν ώστε να κατανοηθεί ο τρόπος με τον οποίο δουλεύει η FOR είναι οι παρακάτω:

- ποιος είναι ο τύπος των τιμών που παίρνει η <μεταβλητή>,
- πόσο σύνθετες μπορεί να είναι οι παραστάσεις <αρχική-τιμή>, <τελική-τιμή>, <βήμα>,
- ποιος ο τύπος των αποτελεσμάτων τους,
- πότε ελέγχεται η <μεταβλητή> σε σχέση με την <τελική-τιμή>,
- μπορεί η <μεταβλητή> να αλλάξει από μια ανάθεση τιμής μέσα στο βρόχο,
- ποια είναι η τιμή της <μεταβλητής> μετά τον τερματισμό του βρόχου

Στη δεύτερη περίπτωση ο τερματισμός του βρόχου προέρχεται από το υπολογισμό μιας μεταβλητής κατά τη διαδικασία της επανάληψης. Εδώ φαίνεται και η πλήρης ισχύς της επαναληπτικής διαδικασίας που ο Goldschlager περιέγραψε ως «έναν αλγόριθμο πεπερασμένου μήκους που περιγράφει μια ακαθόριστης διάρκειας διεργασία» (1996).

Η επαναληπτική διαδικασία, της οποίας ο αριθμός επαναλήψεων δεν είναι εκ των προτέρων γνωστός μπορεί να εκφραστεί με δύο διαφορετικούς τρόπους, ανάλογα με τη θέση της συνθήκης ελέγχου σε σχέση με το σώμα του βρόχου, ο καθένας εκ των οποίων επηρεάζει τη συμπεριφορά του προγραμματιστή και τη σημασία ή την πολυπλοκότητα του προγράμματος (Laborde, 1985):

Σώμα βρόχου / συνθήκη ελέγχου (που εκφράζεται π.χ. με την εντολή REPEAT ... UNTIL).

Συνθήκη ελέγχου / σώμα βρόχου (που εκφράζεται π.χ. με την εντολή WHILE ... DO).

Διδακτικό παράδειγμα: Πρόσθεση των 100 πρώτων ακεραίων αριθμών και εκτύπωση του αποτελέσματος. (γλώσσα παραδείγματος Pascal)

<p>1^η περίπτωση (με χρήση της εντολής REPEAT ... UNTIL)</p>	<p>2^η περίπτωση (με χρήση της εντολής WHILE...DO)</p>
<pre> program Loop1; var sum, i: Integer; begin Sum:=0; i:=1; Repeat sum:=sum+i; i:=i+1 Until i>100; Writeln(sum) end. </pre>	<pre> program Loop1; var sum, i: Integer; begin Sum:=0; i:=1; While i<=100 do begin sum:=sum+i; i:=i+1 end; Writeln(sum); end. </pre>

Τα παραπάνω παραδείγματα δείχνου δύο διαφορετικούς σχεδιασμούς για την επίτευξη του ίδιου στόχου. Στο πρώτο παράδειγμα πρέπει να περιγραφεί η

δράση (action) και στη συνέχεια να γίνει ο έλεγχος (test) ενώ στη δεύτερη περίπτωση ισχύει το αντίθετο (Rogalski, 1987):

Μεταβλητές διαδικασίας / μεταβλητή ελέγχου

Μεταβλητή ελέγχου / μεταβλητές διαδικασίας

Οι έρευνες που έχουν γίνει πάνω στα νοητικά μοντέλα των μαθητών δείχνουν ότι οι αυθόρμητοι συλλογισμοί τους και οι αναπαραστάσεις που διαθέτουν σχετικά με την επανάληψη έχουν κάποια συγκεκριμένη δομή και σταθερή σειρά έκφρασης (Samurçay, 1987):

- Περιγραφή της δράσης
- Μετρητής επαναλήψεων
- Προσδιορισμός της επανάληψης
- Τέλος συνθήκη ελέγχου

Από τα προηγούμενα συνάγεται ότι ο τύπος επαναληπτικής δομής που ταιριάζει περισσότερο στις αρχικές ιδέες των μαθητών είναι ο REPEAT ... UNTIL ενώ αντίθετως η δομή WHILE ... DO ακολουθεί αντίθετη πορεία συλλογισμού και για αυτό το λόγο εμφανίζει περισσότερα προβλήματα κατανόησης..

Η δημιουργία μιας δομής επανάληψης (iteration)

Για την δημιουργία ενός βρόχου τρεις είναι οι βασικές παράμετροι που απαιτούνται (Rogalski, 1990):

1. σχεδιασμός της επεξεργασίας, δηλαδή των εντολών που θα επαναληφθούν (σώμα του βρόχου)
2. προσδιορισμός της συνθήκης ελέγχου για τον τερματισμό του βρόχου και της θέσης του θέσης του μέσα σε στη διαδικασία
3. προσδιορισμός της αρχικής κατάστασης των μεταβλητών του βρόχου (αρχικοποίηση)

Μία διδακτική πρόταση για την κατανόηση της δομής επανάληψης είναι η καταγραφή της κατάστασης του συνόλου των μεταβλητών του προγράμματος στο τέλος κάθε επανάληψης (Κόμης, 2001).

Με αυτόν τον τρόπο μπορούν να δημιουργηθούν οι βάσεις ώστε:

1. να γίνει η διάκριση των μεταβλητών που κρατούν την τιμή τους σε όλη τη διάρκεια της εκτέλεσης και αυτών που αλλάζει η τιμή τους κατά την εκτέλεση του βρόχου
2. να αρχίσει μια εργασία αφενός, πάνω στον εμπειρικό προσδιορισμό των αναλλοίωτων σχέσεων ανάμεσα σε τιμές και μεταβλητές και αφετέρου, στην εγκυρότητα (validation) της παραδοχής μέσω ενός συλλογισμού που αφορά τους μετασχηματισμούς των μεταβλητών στο σώμα ενός βρόχου.

Η έννοια της αναδρομικής διαδικασίας

Η έννοια της Διαδικασίας

Η διαδικασία είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελεί όλες τις λειτουργίες ενός προγράμματος. Είναι ένας τύπος νοητικής αφαίρεσης (abstraction) στις γλώσσες προγραμματισμού. Κύριος λόγος της ύπαρξης αυτής αφαίρεσης με χρήση διαδικασιών (procedural abstraction) είναι ο διαχωρισμός του προβλήματος σε υποπροβλήματα και αυτά με τη σειρά τους σε μικρότερα έτσι ώστε αυτές οι μικρές ανεξάρτητες ενότητες να μπορούν λυθούν χωριστά. Ο συγκεκριμένος τρόπος σχεδίασης είναι γνωστός ως "top down design".

Η αφαίρεση με χρήση διαδικασιών (procedural abstraction) είναι «μια αντιστοίχιση ενός συνόλου εισόδων σε ένα σύνολο εξόδων που μπορεί να περιγραφεί φορμαλιστικά». Η περιγραφή, ενώ δείχνει πως σχετίζονται οι είσοδοι με τις εξόδους, δεν αποκαλύπτει, ούτε καν υπονοεί, τον τρόπο με τον οποίο υπολογίζονται οι έξοδοι (Horowitz, 1984).

Μια διαδικασία περιλαμβάνει τέσσερα στοιχεία (Horowitz, 1984):

- όνομα,

- λίστα αναγνωριστικών που ονομάζονται παράμετροι,
- σώμα και
- περιβάλλον.

procedure ONOMA (λίστα παραμέτρων)

δηλώσεις

σώμα

end ONOMA

Υπάρχουν δύο διαφορετικά είδη διαδικασιών:

- Η υπορουτίνα (subroutine)
Μια διαδικασία που δρα με τρεις τρόπους. Είτε εκχωρεί τα αποτελέσματά της σε μια ή περισσότερες από τις παραμέτρους της, είτε αλλάζει το περιβάλλον της, είτε συνδυάζει τα δύο παραπάνω.
- Η συνάρτηση (function)
Η συνάρτηση είναι μια διαδικασία που επιστρέφει μια τιμή, το αποτέλεσμα της επεξεργασίας της.

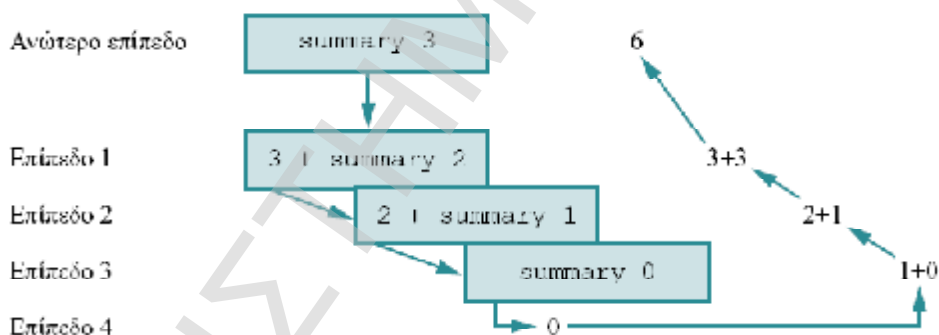
Η αναδρομική διαδικασία

Αναδρομή ονομάζεται η δυνατότητα ενός υποπρογράμματος να καλεί τον εαυτό του και είναι μία βασική έννοια του προγραμματισμού. Όπως είπε ο McCracken (1987): «Η αναδρομική διαδικασία είναι θεμελιώδης στην Πληροφορική, είτε ως μαθηματική έννοια, είτε ως προγραμματιστική τεχνική, είτε ως μέθοδος επίλυσης προβλημάτων». Οι αναδρομικές διαδικασίες προϋποθέτουν μεγάλη εξοικείωση με βασικές έννοιες του προγραμματισμού και θέτουν σημαντικά προγραμματιστικά αλλά και διδακτικά προβλήματα (Anzai, 1982) (Mendelsohn, 1985) (Pirolli, 1986) (Rouchier, 1987) (Komis, 1991).

Διδακτικό παράδειγμα σύνθετης αναδρομικής διαδικασίας με παράμετρο (Κόμης, 2001): Πρόσθεση των n πρώτων ακεραίων αριθμών και εκτύπωση του αποτελέσματος, με $n=3$. (γλώσσα παραδείγματος C)

```
function summary (n)
{
  if n==0
  then return 0
  else return (n + summary (n-1));
}
```

Κατά την εκτέλεση της `summary 3` η διαδικασία εκτελείται 4 φορές και συμβαίνουν τα εξής (σχήμα 2):



Η εκτέλεση της `summary 3` καλεί τη `summary 2`, που καλεί τη `summary 1`, η οποία με τη σειρά της καλεί την `summary 0`. Η κλήση αυτή αποτιμάται απευθείας αφού η τιμή της δίνεται από το πρόγραμμα ($n==0$ (return 0)). Στη συνέχεια η αποτίμηση της προηγούμενης κλήσης είναι εφικτή κ.ο.κ. μέχρι και την `summary 3`.

Διδακτικά προβλήματα σχετικά με την έννοια της αναδρομικότητας

Η αναδρομή είναι ένα χρήσιμο, δυνατό και αποτελεσματικό εργαλείο στα χέρια ενός προγραμματιστή. Παρ' όλα αυτά, η εγγενής πολυπλοκότητα της αναδρομικής διαδικασίας δημιουργεί σημαντικά διδακτικά προβλήματα, τα

οποία απαντώνται κυρίως από αρχάριους προγραμματιστές (Anzai, 1982) και προέρχονται κυρίως από:

- Απειρία στην κατανόηση, χρήση και υλοποίηση διαδικασιοστρεφούς προγραμματισμού
- Χαμηλή εκτίμηση της λειτουργικότητας και χρηστικότητας της αναδρομής
- Ελλιπής κατάρτιση για την καταγραφή μιας λύσης με αναδρομή

Βασικό στοιχείο του σκεπτικού μιας λύσης με αναδρομή είναι ο διαχωρισμός του προβλήματος σε υπο-προβλήματα κατά κανόνα πιο απλά στη λύση τους. Στην συνέχεια ακολουθούνται τα παρακάτω βήματα:

1. Ανάλυση και παραμετροποίηση του προβλήματος έτσι ώστε να βρεθούν τα στοιχεία από τα οποία εξαρτάται η λύση
2. Αναζήτηση, αρχικά, μιας λύσης χωρίς αναδρομή
3. Γενίκευση της προηγούμενης λύσης και δημιουργία της αναδρομικής διαδικασίας

Στα παραπάνω βήματα απαντώνται τα εξής προβλήματα:

- Ποια η συνθήκη τερματισμού της αναδρομικής κλήσης, πως τροποποιείται κατά την εκτέλεση και που τοποθετείται
- Που τοποθετείται η κλήση της αναδρομικής διαδικασίας

Κεφάλαιο 4 - Προσεγγίσεις διδασκαλίας του προγραμματισμού και προγραμματιστικά περιβάλλοντα

Η δημιουργία ενός προγράμματος εμπεριέχει τρία στοιχεία:

- Την γνώση της αλγοριθμικής λογικής
Η ανάλυση και επεξεργασία μιας ιδέας και η μετατροπή της σε ένα πρόγραμμα που χρησιμοποιεί τις αρχές του προγραμματισμού (διαχωρισμός σε μεθόδους και αντικείμενα, κληρονομικότητα κλπ.)
- Την γνώση μιας γλώσσας προγραμματισμού
Το συντακτικό και τις εντολές μιας συγκεκριμένης γλώσσας προγραμματισμού
- Την γνώση ενός περιβάλλοντος προγραμματισμού

Το πιο σημαντικό απ' αυτά είναι η αλγοριθμική λογική η οποία ξεπερνά τα όρια μιας γλώσσας και βοηθά στην βαθιά κατανόηση του προγραμματισμού. Παρ' όλα αυτά, για πολλά χρόνια, η διδακτική στα πρώτα βήματα του αρχάριου προγραμματιστή έδινε μεγάλο βάρος στην εκμάθηση του συντακτικού και των εντολών μιας γλώσσας και στον τρόπο αποφυγής κάποιων λαθών που σχετίζονταν με ιδιορρυθμίες της συγκεκριμένης γλώσσας. Η αρχές του προγραμματισμού έρχονταν σε ένα επόμενο βήμα και κτίζονταν εμμέσως μέσα από τις ασκήσεις που έπρεπε να φέρει εις πέρας ο μαθητής (Sajaniemi, 2005).

Στη συνέχεια του κεφαλαίου αυτό παρουσιάζουμε όλες τις διαφορετικές προσεγγίσεις στη διδασκαλία του προγραμματισμού. Μελετούμε την κλασική προσέγγιση καθώς και μια σειρά από εναλλακτικές προσεγγίσεις.

Η κλασική προσέγγιση διδασκαλίας

Μέχρι σήμερα, η πιο διαδεδομένη μέθοδος για τη διδασκαλία των αρχών του προγραμματισμού ήταν η αποκαλούμενη «κλασική προσέγγιση». Η διδασκαλία βασιζόταν σε μία επαγγελματική γλώσσα γενικού σκοπού όπως οι Pascal, C, LISP και Java, και ένα εμπορικό προγραμματιστικό περιβάλλον για την

συγκεκριμένη γλώσσα. Όπως δείχνουν πρόσφατες έρευνες, δεν εναρμονίζεται με τις απαιτήσεις των μαθητών και θεωρείται μια από τις σημαντικότερες πηγές των προβλημάτων που δημιουργούνται κατά τη διδασκαλία του προγραμματισμού. Αυτή η προσέγγιση δημιουργούσε πολλών ειδών προβλήματα για τους αρχάριους προγραμματιστές (Brusilovsky, 1997):

- Οι γλώσσες γενικού σκοπού περιέχουν μεγάλο αριθμό εντολών και ιδιομορφιών που δεν διευκολύνουν τίποτα τον αρχάριο προγραμματιστή, αντιθέτως τον δυσχεραίνουν. Αντιθέτως δυσχεραίνουν την κατανόηση των βασικών σημείων στα οποία πρέπει ο μαθητής να επικεντρωθεί. Το παραπάνω πρόβλημα οδηγεί εμμέσως σε ένα ακόμη. Ο μαθητής προσπαθώντας να μάθει την συγκεκριμένη γλώσσα, δεν κατανοεί σε βάθος τις βασικές έννοιες της αλγοριθμικής λογικής. Αποτέλεσμα είναι ο μαθητής, αντί να δημιουργεί μια υποδομή πάνω στις προγραμματιστικές αρχές, να μαθαίνει τις ιδιορρυθμίες τις συγκεκριμένης γλώσσας και πώς να τις αποφεύγει.
- Η δημιουργία ενός απλού προγράμματος προϋποθέτει γνώσεις ενός μεγάλου μέρους της γλώσσας, πράγμα που κουράζει τους αρχάριους προγραμματιστές και τους αποθαρρύνει πριν ακόμη ξεκινήσουν. Στη συνέχεια, για τη δημιουργία μεγαλύτερων προγραμμάτων, χρειάζεται μεγάλη εξοικείωση με το προγραμματιστικό περιβάλλον. Αυτό το περιβάλλον, μπορεί να διαθέτει πολλά χρήσιμα εργαλεία για τους έμπειρους χρήστες αλλά για τους αρχάριους η πολυπλοκότητά τους αποτελεί ένα ακόμη πρόβλημα. Στόχος πρέπει να είναι η εκμάθηση προγραμματισμού και όχι του συντακτικού της γλώσσας.
- Η γλώσσες γενικού σκοπού και τα εμπορικά περιβάλλοντα προγραμματισμού παρέχουν ελάχιστη βοήθεια στην κατανόηση των βασικών εντολών και δομών ελέγχου. Αποκρύβουν τις βασικές λειτουργίες μέσα σε μεθόδους που υλοποιούν περισσότερες λειτουργίες, δεν παρέχουν καμιά οπτικοποίηση ούτε της εκτέλεσης, ούτε του τελικού αποτελέσματος του προγράμματος. Έτσι οι μαθητές

αδυνατούν να παρακολουθήσουν την εξέλιξη του προγράμματος, να ελέγξουν τις μεταβλητές εισόδου-εξόδου και τις τιμές των ενδιάμεσων μεταβλητών.

- Οι μεταγλωττιστές των εμπορικών γλωσσών είναι πολυσύνθετοι, με μεγάλο πλήθος παραμέτρων και βιβλιοθηκών τις οποίες όμως οι μαθητές δεν μπορούν να χρησιμοποιήσουν. Επιπλέον, τα μηνύματα λάθους που παράγονται δεν γίνονται κατανοητά από τους αρχάριους αφού έχουν δυσνόητες πληροφορίες και δεν προτείνουν τρόπο επίλυσης του προβλήματος.
- Όλες οι εμπορικές γλώσσες και τα περιβάλλοντα ανάπτυξης είναι κατεξοχήν στα αγγλικά και χρησιμοποιούν όρους που δεν περιγράφουν ακριβώς την αντίστοιχη λειτουργία τους. Έτσι, ακόμη και μαθητές που έχουν γνώση της αγγλικής γλώσσας παρανοούν εντολές και οδηγούνται σε λάθος χρησιμοποίησή τους.
- Το ενσωματωμένο σύστημα βοήθειας κάνει μια γρήγορη καταγραφή του συντακτικού της γλώσσας και στοχεύει κατά κύριο λόγο στην εξοικείωση του χρήστη με το περιβάλλον και τις λειτουργίες του χωρίς να αναφέρεται σε βασικά σημεία όπως η εκχώρηση τιμών σε μεταβλητές, οι δομές ελέγχου κ.α.

Προγραμματιστικά περιβάλλοντα για αρχάριους

Τα προβλήματα τα οποία προέκυψαν από την κλασική μέθοδος διδασκαλίας του προγραμματισμού οδήγησαν τους επιστήμονες να προτείνουν νέες μεθόδους. Σκοπός τους ήταν η απαλλαγή της διδασκαλίας από την περιττή πολυπλοκότητα που ενείχαν τα εμπορικά περιβάλλοντα, είτε με τον σχεδιασμό νέων γλωσσών προγραμματισμού, είτε ολοκληρωμένων περιβαλλόντων που να είναι προσανατολισμένα στους αρχάριους προγραμματιστές. Με αυτόν τον τρόπο οι μαθητές θα γνωρίσουν τις βασικές αρχές της προγραμματιστικής λειτουργίας και αφού τις κατανοήσουν θα μπορέσουν εύκολα να μεταβούν σε

εμπορικές γλώσσες προγραμματισμού που προορίζονται για έμπειρους προγραμματιστές (Pascal, C, Java).

Οι λύσεις που προτάθηκαν οδήγησαν σε πολλές διαφορετικές κατευθύνσεις. Συνοπτικά θα μπορούσαν να ταξινομηθούν ως ακολούθως (Εφόπουλος, 2005):

Τροποποίηση εμπορικών γλωσσών

Η απλοποίηση κάποιων εμπορικών γλωσσών και των αντίστοιχων προγραμματιστικών περιβαλλόντων έχει ως σκοπό την εξοικείωση του μαθητή με τις αρχές του προγραμματισμού χρησιμοποιώντας όμως μια γλώσσα που θα τον διευκολύνει στην μετέπειτα μετάβαση του σε ένα πλήρες επαγγελματικό περιβάλλον. Η τροποποίηση αυτή οδήγησε σε γλώσσες που είχαν απλοποιημένη σύνταξη και περιορισμένο ρεπερτόριο εντολών με μειωμένη πολυπλοκότητα και περιβάλλοντα που από τη μία απέκρυπταν τις μη χρήσιμες επεκτάσεις της γλώσσας, από την άλλη βοηθούσαν τον χρήστη με έναν τροποποιημένο μεταγλωττιστή που παρήγαγε εύληπτα μηνύματα λάθους και πρότεινε διορθώσεις και έναν αποσφαλματωτή που βοηθούσε στην παρακολούθηση της εξέλιξης του προγράμματος και των επιμέρους μεταβλητών.

ΔΕΛΥΣ (Dagdilelis, 2003), Turing (Holt, 1988), DrJava (Allen, 2002), Thetis (Freund & Roberts, 1996, BlueJ (Kölling, 1999), Junior Java (Motil & Epstein, 2000)

ΔΕΛΥΣ

Το πρόγραμμα ΔΕΛΥΣ, αποτελεί ένα περιβάλλον για τη υποστήριξη της διδασκαλίας της Πληροφορικής στη δευτεροβάθμια εκπαίδευση. Αποτελείται από τέσσερις μικρόκοσμους διερεύνησης και εργασίας καθώς και ένα εικονικό εργαστήριο στο οποίο οι μαθητές μπορούν να εργαστούν στα πλαίσια διαφόρων δραστηριοτήτων και διδακτικών σεναρίων. Ο μικρόκοσμος του προγραμματισμού περιλαμβάνει μια απλή Pascal-like γλώσσα, τη γλώσσα X, που είναι υποσύνολο της Pascal και είναι προσανατολισμένη στον αρχάριο χρήστη. Σημαντικό πλεονέκτημα θεωρείται η επιλογή για βηματική εκτέλεση του πηγαίου κώδικα με ταυτόχρονη προβολή των τιμών των μεταβλητών του

προγράμματος, του assembly κώδικα και η προβολή των περιεχομένων των καταχωρητών του συστήματος.



Εικόνα 1 - Το περιβάλλον του ΔΕΛΥΣ (X-Compiler)

Turing

Η γλώσσα Turing αναπτύχθηκε από τον Ric Holt ως μια εκπαιδευτική γλώσσα με απλή σύνταξη. Βασίστηκε στις προϋπάρχουσες προγραμματιστικές γλώσσες Pascal, Euclid and SP/k και χρησιμοποιήθηκε ευρέως στην δευτεροβάθμια εκπαίδευση και σε πανεπιστήμια. Χαρακτηριστικά της αποτελούν ο δυναμικός ορισμός πινάκων και τα αλφαριθμητικά μεταβλητού μήκους. Επιπλέον, η γλώσσα δεν απαιτεί επικεφαλίδα (program), έχει καλή διαχείριση αλφαριθμητικών, έχει απλοποιημένη σύνταξη ιδιαίτερα στις δηλώσεις δεικτών και δεν απαιτεί τη χρήση του ερωτηματικού στο τέλος κάθε εντολής. Από το 2007 αποτελεί προϊόν ελεύθερου λογισμικού.

DrJava

Η DrJava αποτελεί ένα εκπαιδευτικό προγραμματιστικό περιβάλλον που αποσκοπεί στην συγκέντρωση του μαθητή στην δημιουργία προγραμμάτων χωρίς να χρειάζεται πρώτα να εξοικειωθεί με το interface της εφαρμογής. Γι' αυτό το λόγο το περιβάλλον διεπαφής της DrJava είναι απλό και διαισθητικό με ευδιάκριτες επιλογές. Βασίζεται στο μοντέλο βρόχου «Είσοδος – Υπολογισμός – Έξοδος» ("read-eval-print loop") και οδηγεί τον χρήστη να υλοποιήσει, να ελέγξει το πρόγραμμα για λάθη και να κάνει τις απαραίτητες διορθώσεις σε ένα

ατέρμονο κύκλο μέχρι το τελικό ζητούμενο. Η DrJava διευκολύνει τους εκπαιδευόμενους να γνωρίσουν τη σύνταξη της γλώσσας με χρήση διαφόρων τρόπων, όπως λ.χ. η δυνατότητα αυτόματης συμπλήρωσης κώδικα. Υποστηρίζει χρωματισμό εντολών, αρίθμηση εντολών, αυτόματη προσαρμογή κώδικα (indentation), έλεγχο παρενθέσεων-αγκυλών (brace matching) και διαχειρίζεται πολλά έγγραφα ταυτόχρονα.

Thetis

Το Thetis αποτελεί την πρόταση του Πανεπιστήμιου του Στάνφορντ για την εκμάθηση της γλώσσας ANSI C. Αποτελείται από έναν μεταγλωττιστή της C και ένα απλό, διαδραστικό περιβάλλον που βοηθάει τον αρχάριο χρήστη σε όλα τα βήματα του προγραμματισμού, από την καταγραφή του πηγαίου κώδικα μέχρι την αποσφαλμάτωση. Αυτό επιτυγχάνεται με οπτικά βοηθήματα (visualization capabilities) και προσφέροντας δυνατότητες ανίχνευσης και αναφοράς λαθών μέσα από μηνύματα που είναι προσανατολισμένα στον αρχάριο χρήστη. Επίσης, ενσωματώνει δυνατότητες εκτέλεσης του κώδικα βήμα-βήμα (step execution), υποστηρίζει την εισαγωγή breakpoints και την παρακολούθηση των μεταβλητών του προγράμματος.

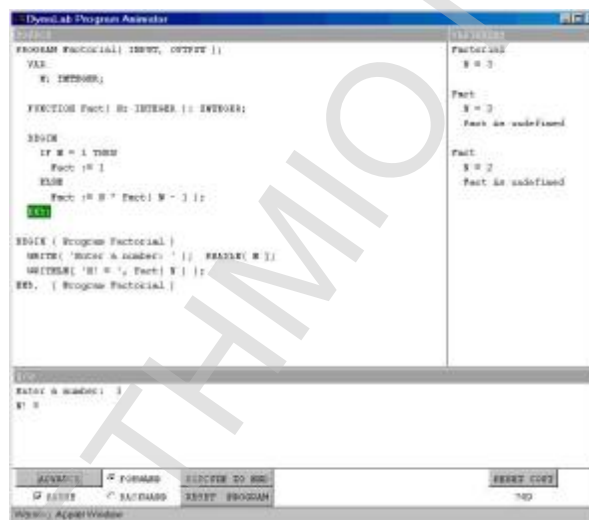
Συστήματα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων

Τα συστήματα δυναμικής προσομοίωσης της εκτέλεσης ενός προγράμματος (Program Animation Systems) δίνουν τη δυνατότητα οπτικοποίησης της ροής εκτέλεσης του πηγαίου κώδικα, των δομών, των τιμών των μεταβλητών, των δεδομένων εισόδου και των αποτελεσμάτων σε όλη τη διάρκεια εκτέλεσης του προγράμματος. Με αυτόν τον τρόπο, ο αρχάριος προγραμματιστής κατανοεί την λειτουργία των εντολών του πηγαίου κώδικα βήμα-βήμα και μπορεί πιο εύκολα να υλοποιήσει αυτό που χρειάζεται.

Dynalab (Birch, 1995), AnimPascal (Σατρατζέμη, 2000) και Weblab (Boroni, 1997), Διερμηνευτής της ΓΛΩΣΣΑΣ - Μέρος του Έργου Πλειάδες/Ενότητα Νηρηίδες «Αλγοριθμική-Προγραμματισμός» (Δικ09)

Dynalab Program Animation System

Το Dynalab (πλήρες όνομα: dynamic computer science laboratory infrastructure featuring program animation) πρόκειται για μια εφαρμογή η οποία υποστηρίζει την βήμα προς βήμα, αλληλεπιδραστική με τους χρήστες εκτέλεση ενός προγράμματος, παρέχοντας ταυτόχρονα δυνατότητα δυναμικής θέασης των περιεχομένων της μνήμης και της τυπικής εξόδου (οθόνης) του υπολογιστή. Διαθέτει μια εικονική μηχανή την E-Machine, που τρέχει τον, χαμηλού επιπέδου, κώδικα E-code. Περιλαμβάνει πρωτοποριακά χαρακτηριστικά όπως η ανάστροφη εκτέλεση μίας ή περισσότερων γραμμών κώδικα του προγράμματος, η μετατροπή από έναν πλήρη μεταγλωττιστή της γλώσσας Pascal σε Ecode.



Εικόνα 2 - Το περιβάλλον του Dynalab

Δύο χρόνια μετά την εμφάνιση του Dynalab, έγινε μια προσπάθεια για μεταφορά του περιβάλλοντος στο διαδίκτυο. Το νέο διαδικτυακό Dynalab ονομάστηκε Weblab αλλά οι δημιουργοί του δε συνέχισαν να υποστηρίζουν την ανάπτυξη του και το Weblab εγκαταλείφθηκε.

AnimPascal

Το AnimPascal περιλαμβάνει τα όλα τα τυπικά χαρακτηριστικά ενός συστήματος δυναμικής προσομοίωσης εκτέλεσης προγράμματος το οποίο ενσωματώνει επιπλέον τη δυνατότητα καταγραφής των ενεργειών των σπουδαστών (recordability). Το AnimPascal στην προσπάθεια να βοηθήσει τους αρχάριους

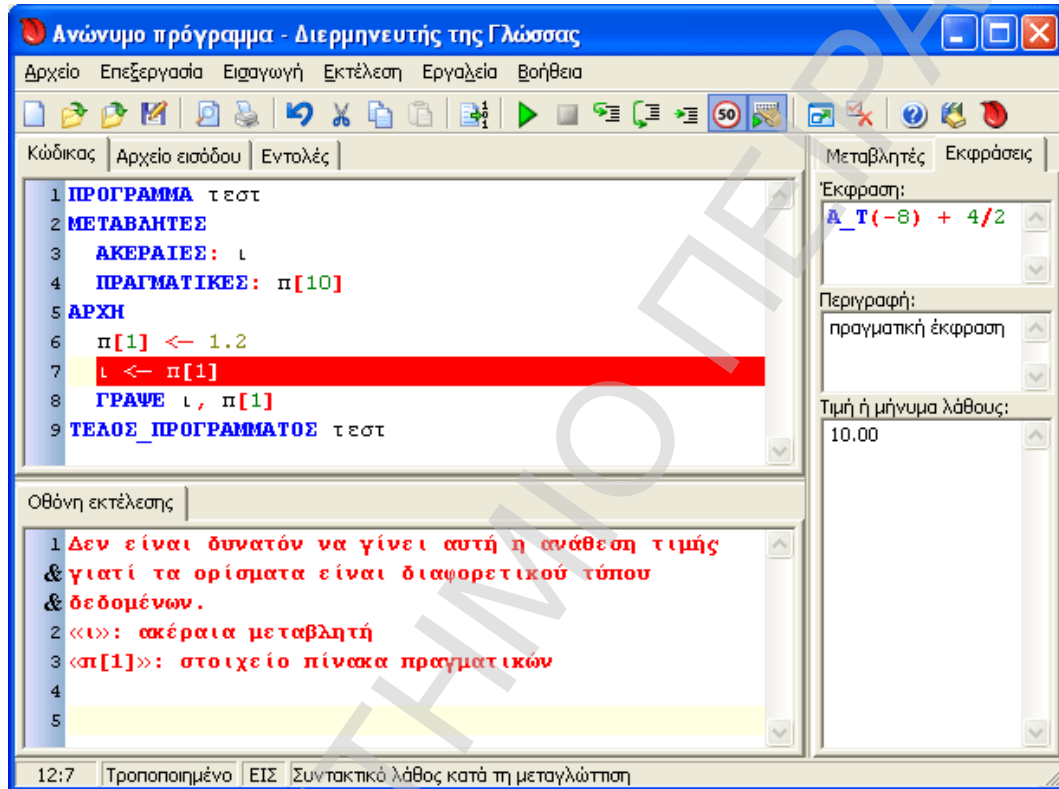
στη φάση της ανάπτυξης, της αποσφαλμάτωσης, της εκτέλεσης και ελέγχου ενός προγράμματος προσφέρει εκτέλεση του κώδικα βήμα-βήμα ή μαζικής εκτέλεσης πολλών γραμμών, μορφοποίησης της τρέχουσας γραμμής κώδικα του εκτελέσιμου αρχείου (αλλαγή χρώματος υποβάθρου και γραμματοσειράς) και εμφάνιση των τιμών των μεταβλητών, εμφάνιση της εισόδου & εξόδου του προγράμματος. Επίσης, έχει την ικανότητα να καταγράψει και να παρουσιάσει το ιστορικό των μεταγλωττίσεων τους ώστε να βοηθήσει τον διδάσκοντα στον εντοπισμό των λανθασμένων αντιλήψεων των μαθητών.

Διερμηνευτής της ΓΛΩΣΣΑΣ (Μέρος του Έργου Πλειάδες/Ενότητα Νηρηίδες «Αλγοριθμική-Προγραμματισμός»)

Ο Διερμηνευτής της ΓΛΩΣΣΑΣ (Γεωργόπουλος) είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης αλγορίθμων σε μορφή ψευδοκώδικα, ειδικά σχεδιασμένο για τη ΓΛΩΣΣΑ προγραμματισμού που διδάσκεται στα πλαίσια του μαθήματος «Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον» (ΑΕΠΠ) της Γ΄ Ενιαίου Λυκείου. Περιλαμβάνει χαρακτηριστικά όπως παρακολούθηση της ροής του προγράμματος με βήμα-προς-βήμα εκτέλεση, των μεταβλητών, συνθηκών και εκφράσεων, σημεία διακοπής(breakpoints). Βασικό πλεονέκτημα θεωρείται η υλοποίηση του στην ελληνική γλώσσα και η σύμπνοια με το ισχύον Ενιαίο Πρόγραμμα Προγραμμάτων Σπουδών Πληροφορικής. Ο διερμηνευτής στοχεύει στους αρχάριους προγραμματιστές, όχι στους επαγγελματίες, και προσπαθεί να υποβοηθήσει την ανάπτυξη της αλγοριθμικής σκέψης. Έτσι, για παράδειγμα, αντίθετα με τους επαγγελματικούς compilers, σε περίπτωση συντακτικών λαθών ή λαθών χρόνου εκτέλεσης εμφανίζει αναλυτικά μηνύματα εξηγώντας πού ακριβώς υπήρξε πρόβλημα.

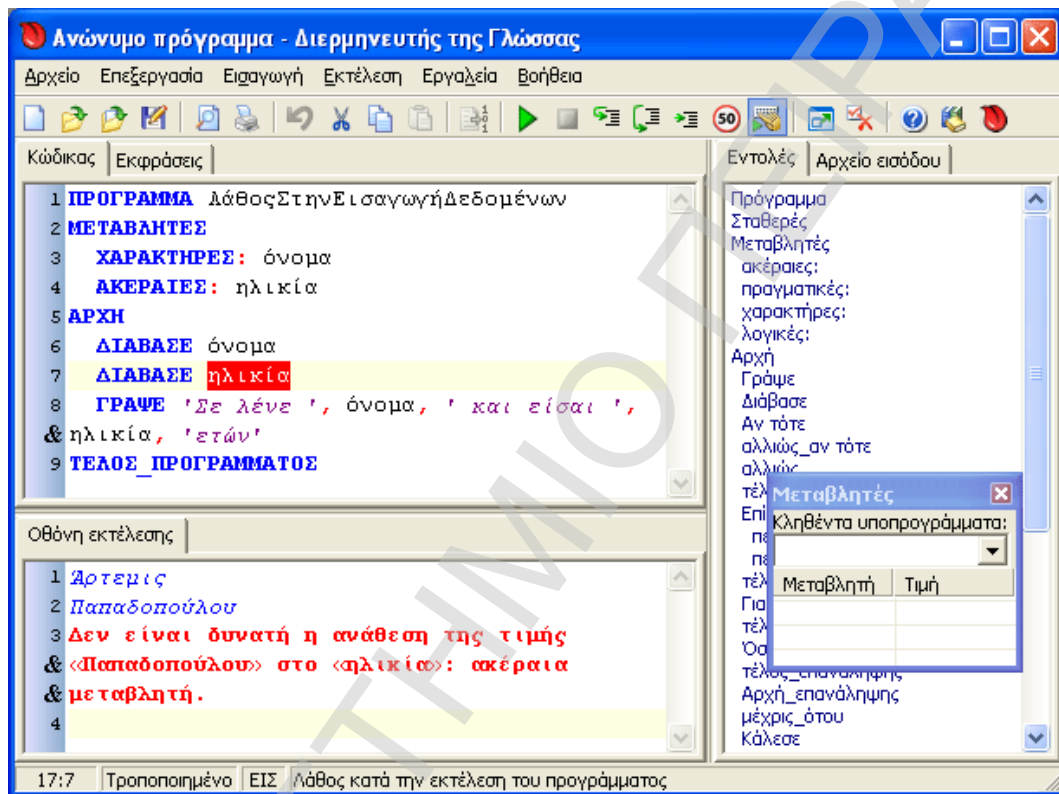
Εικόνες λειτουργίας του Διερμηνευτή:

Όταν γίνεται προσπάθεια ανάθεσης τιμής από πραγματικό αριθμό σε ακέραιο, εμφανίζεται το μήνυμα της παρακάτω εικόνας.



Εικόνα 3 - Διερμηνευτής της ΓΛΩΣΣΑΣ – Στιγμιότυπο 1

Ένα από τα μεγάλα πλεονεκτήματα των διερμηνευτών (interpreters) σε σχέση με τους μεταφραστές (compilers) είναι ότι μπορούν να εμφανίζουν επεξηγηματικά μηνύματα ακόμα και κατά το χρόνο εκτέλεσης. Κατά την εκτέλεση αυτού του προγράμματος ο χρήστης εισήγαγε «Παπαδοπούλου» ενώ έπρεπε να εισάγει την ηλικία, δηλαδή αριθμό.



Εικόνα 4 - Διερμηνευτής της ΓΛΩΣΣΑΣ – Στιγμιότυπο 2

Περιβάλλοντα εναλλακτικών μορφών αναπαράστασης της σύνταξης εντολών

Περιλαμβάνονται εργαλεία διαφορετικών κατηγοριών που έχουν ένα κοινό βασικό χαρακτηριστικό. Ο χρήστης τους δεν γράφουν κώδικα αλλά χρησιμοποιούν έτοιμες γραφικές μονάδες που συνενώνουν για το τελικό παραγόμενο. Αυτές οι γραφικές μονάδες, ανάλογα με την προσέγγιση, μπορεί να είναι μονάδες κώδικα όπως εντολές, μεταβλητές ή δομές ελέγχου και επανάληψης ή εικονίδια σχεδίασης αλγορίθμων ή διαγραμμάτων ροής. Με

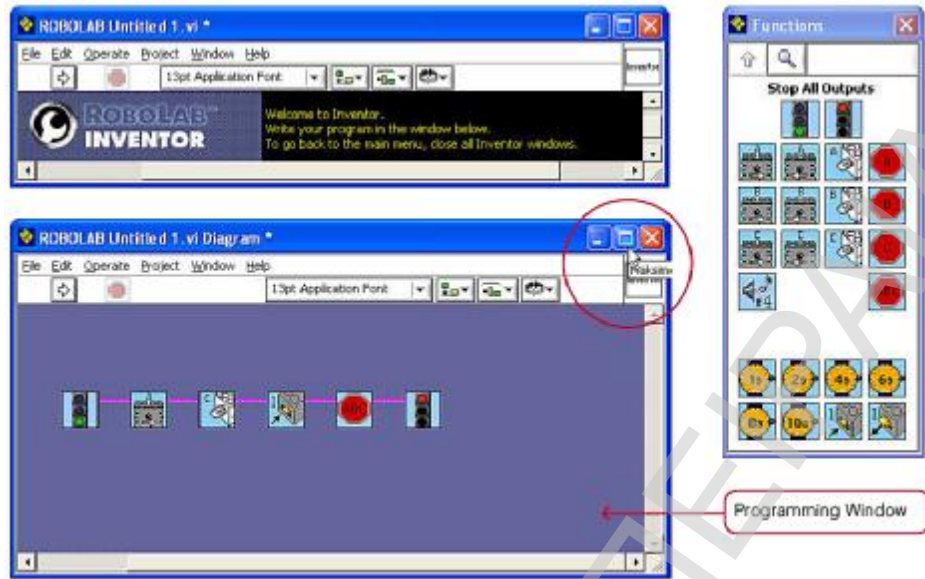
αυτόν τον τρόπο οι χρήστες κατανοούν τις έννοιες του προγραμματισμού και υλοποιούν προγράμματα χωρίς να δημιουργούνται περαιτέρω δυσκολίες λόγω συντακτικών λαθών ή λαθών στην πληκτρολόγηση.

Γραφικός προγραμματισμός: Scratch (Resnick, 2003) (Kafai, 2004), Lego Robolab (Mindell, 2000), LogoBlocks (Begel, 1996)

Αλγόριθμοι και διαγράμματα ροής: BACCII (Calloni B., 1994), FLINT (Crews, 1998), Δημιουργός Διαγραμμάτων Ροής - Μέρος του Έργου Πλειάδες/Ενότητα Νηριίδες «Αλγοριθμική-Προγραμματισμός» (Δικ09), ΓλωσσοΜάθεια (Spi09), ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ – Ελπήνωρ, LENS (Mukherjea, 1993)

Lego Robolab

Το Lego Robolab, ένα λογισμικό ρομποτικής βασισμένο στο Lego RCX τσιπ της ομώνυμης εταιρίας, χρησιμοποιεί ένα γραφικό περιβάλλον με «τουβλάκια» κώδικα για την δημιουργία προγραμμάτων που θα χειρίζονται ένα ρομπότ φτιαγμένο από lego. Οι μαθητές πρέπει να ενώσουν τα «τουβλάκια» κώδικα για να δημιουργήσουν κάποιο πρόγραμμα ακολουθώντας τις βασικές αρχές του προγραμματισμού χωρίς να χρειαστεί να μάθουν κάποια συγκεκριμένη γλώσσα προγραμματισμού ούτε να προβληματιστούν για την σύνταξη των εντολών. Περιλαμβάνει τρία επίπεδα προγραμματισμού ανάλογα με το επίπεδο του χρήστη. Ένα, επίσης, σημαντικό πλεονέκτημα είναι πως το αποτέλεσμα του προγράμματος δεν εμφανίζεται απλά στην οθόνη αλλά μεταφέρεται στον τρισδιάστατο «πραγματικό» κόσμο με την βοήθεια του ρομπότ της Lego.



Εικόνα 5 - Το περιβάλλον του Lego Robolab

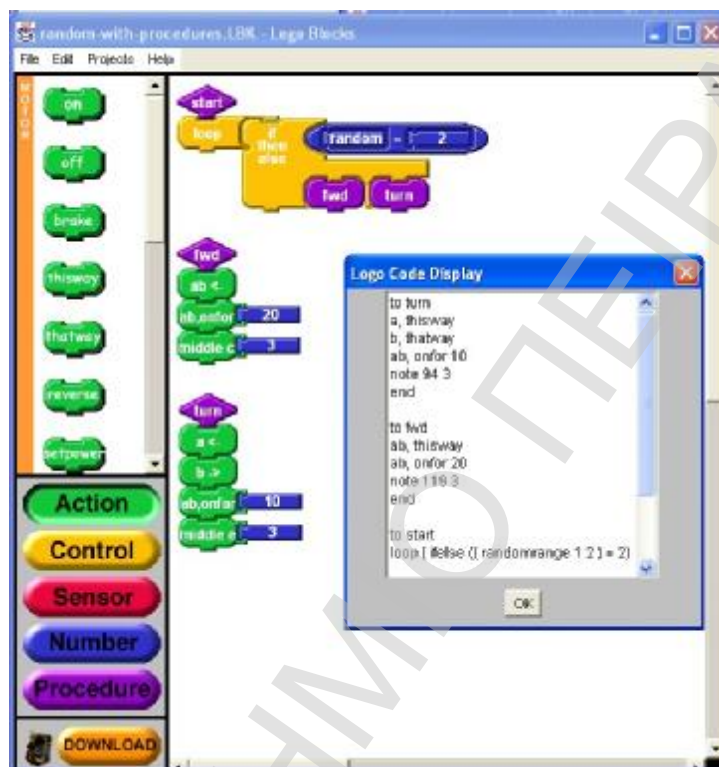


Εικόνα 6 - Lego RCX

LogoBlocks

Το LogoBlocks υλοποιήθηκε στα εργαστήρια του MIT Media Lab ως μια γλώσσα γραφικού προγραμματισμού για το Programmable Brick. Το τελευταίο είναι ένα μικρός υπολογιστής χειρός που, προσαρτώμενο σε ένα ρομπότ φτιαγμένο από τούβλα Lego, μπορεί να ελέγξει την κίνησή του με τέσσερις κινητήρες (μηχανές), να διαβάσει δεδομένα από έξι αισθητήρες και να πράξει ανάλογα με τις τιμές και το πρόγραμμα που έχει γράψει ο χρήστης. Το LogoBlocks χρησιμοποιώντας ένα διαισθητικό γραφικό περιβάλλον βοηθάει τους μικρούς προγραμματιστές να συγκεντρωθούν στην υλοποίηση της ιδέας τους με ευχάριστο τρόπο. Η γλώσσα που χρησιμοποιείται είναι η BrickLogo, μια διαφοροποιημένη έκδοση της Logo. Οι εντολές της γλώσσας στο LogoBlocks αναπαριστώνται με γραφικά

σχήματα που μπορούν να συρθούν (με χρήση drag & drop) από την παλέτα των εργαλείων στην περιοχή δημιουργίας, δίπλα σε άλλα σχήματα, και να σχηματίσουν το πρόγραμμα.

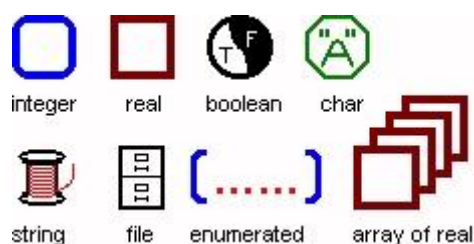


Εικόνα 7 - Το περιβάλλον προγραμματισμού LogoBlocks

BACCII και BACCII++

Οι δημιουργοί του BACCII (Calloni B., 1994), βασισμένοι στο εγνωσμένο πλεονέκτημα (Scanlan, Sept. 1989) που έχουν τα περιβάλλοντα γραφικού προγραμματισμού σε σύγκριση με τον παραδοσιακό προγραμματισμό (text-based), υλοποίησαν ένα παραθυρικό περιβάλλον γραφικού σχεδιασμού προγραμμάτων για Microsoft Windows βασισμένο σε διαγράμματα ροής. Το BACCII επιτρέπει στο χρήστη τον σχεδιασμό αλγορίθμων με τη χρήση γραφικών που αναπαριστούν όλες τις βασικές δομές που χρησιμοποιούνται στον προγραμματισμό, όπως βρόχους και δομές ελέγχου, μέσα σε ένα υποβοηθούμενο περιβάλλον. Με αυτόν τον τρόπο ο αρχάριος προγραμματιστής μπορεί να δημιουργήσει συντακτικά σωστούς αλγορίθμους που στη συνέχεια το

περιβάλλον μπορεί να τους μετατρέψει σε γνωστές γλώσσες προγραμματισμού (C, C++, FORTRAN, Pascal).

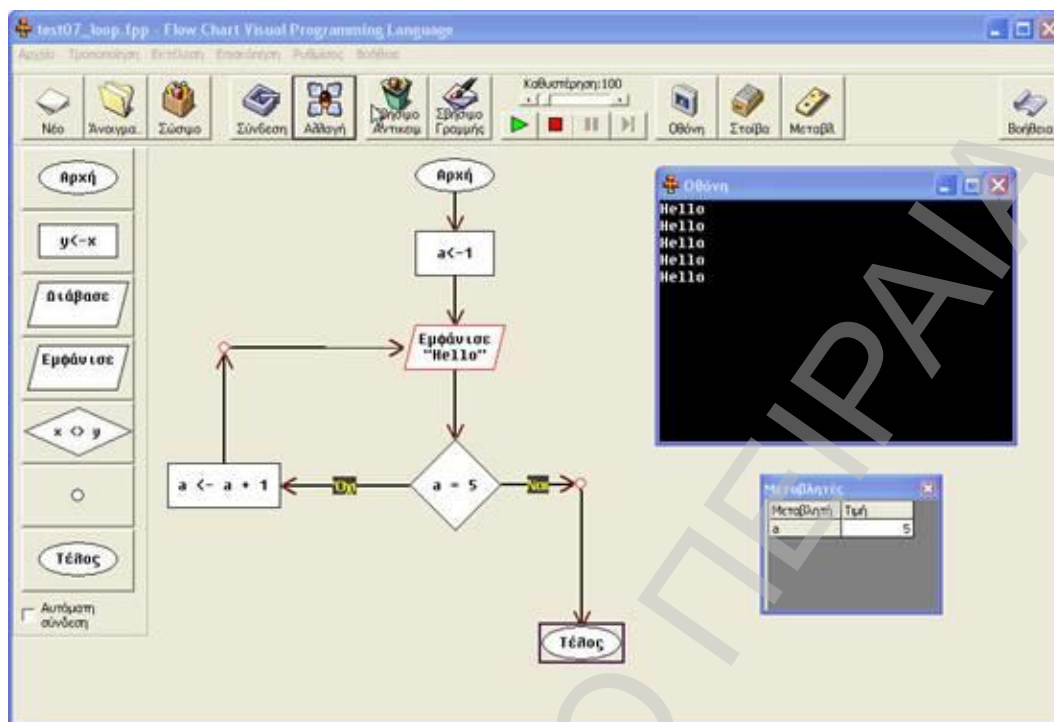


Εικόνα 8 - Οι τύποι δεδομένων που περιλαμβάνει το BACCII

Σε νεότερες εκδόσεις συμπεριελήφθη η επιλογή προγραμματισμού με αντικείμενα, που τελικά οδήγησε στη δημιουργία του BACCII++ για αντικειμενοστραφή προγραμματισμό.

Δημιουργός Διαγραμμάτων Ροής (Μέρος του Έργου Πλειάδες/Ενότητα Νηρηίδες «Αλγοριθμική-Προγραμματισμός»)

Ο Δημιουργός Διαγραμμάτων Ροής (Τσέλιος Ν.) είναι ένα περιβάλλον σχεδίασης και εκτέλεσης αλγορίθμων με τη χρήση διαγραμμάτων ροής. Δημιουργός του ήταν ο Yuri Margolin, ενώ ο εξελληνισμός και η βελτίωση υλοποιήθηκαν από τον Δρ. Τσέλιο Νικόλαο. Η ανάπτυξη αλγορίθμων γίνεται με την τεχνική drag 'n' drop και είναι δυνατή η βήμα-προς-βήμα/αργή εκτέλεση και η παρακολούθηση των τιμών των μεταβλητών. Πλεονεκτήματα θεωρούνται το πλήρως εξελληνισμένο περιβάλλον και η παρακολούθηση της εκτέλεσης του αλγορίθμου με οπτικό τρόπο.

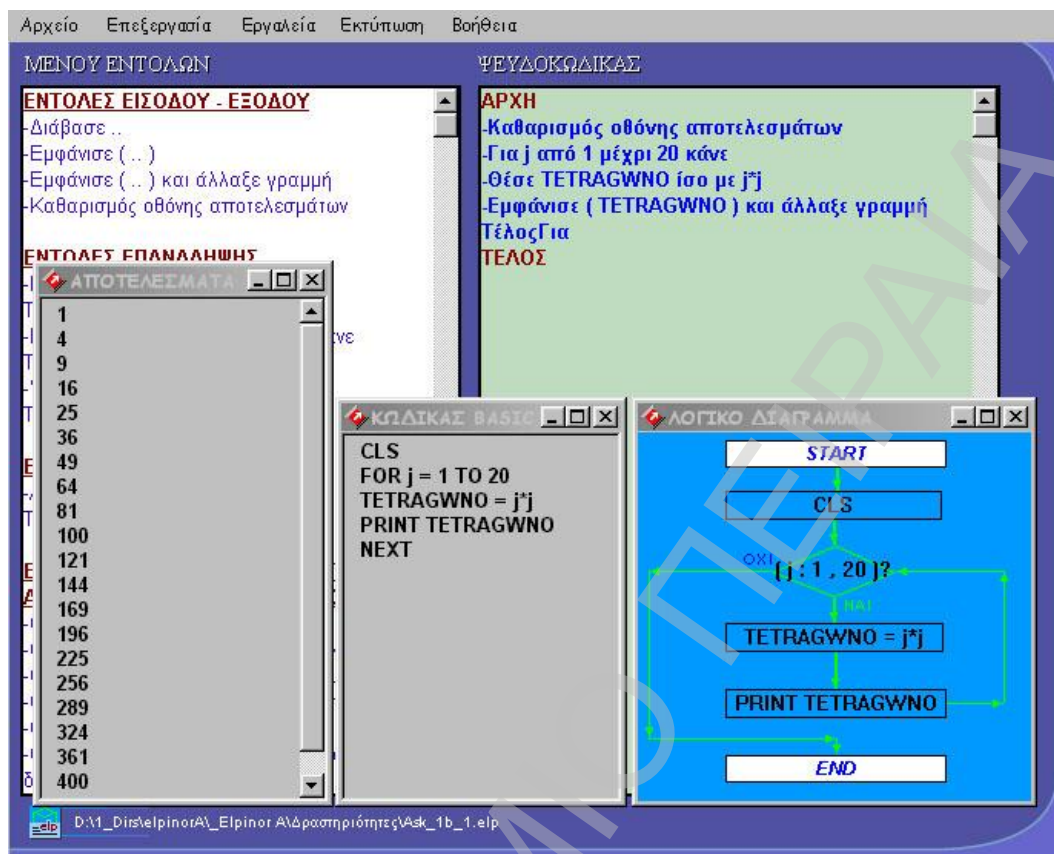


Εικόνα 9 - Δημιουργός Διαγραμμάτων Ροής

ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ - Ελπήνωρ

(Μέρος του Έργου Οδύσσεια / Ενέργεια Ελπήνωρ)

Το έργο E13-Ελπήνωρ είναι ένα από τα πιλοτικά έργα της Οδύσσειας που αφορά στην αξιοποίηση υπολογιστικών και δικτυακών τεχνολογιών στο Ενιαίο Λύκειο. Βασικός στόχος του έργου είναι ο σχεδιασμός και η ανάπτυξη εξειδικευμένων εργαλείων λογισμικού για 12 γνωστικά αντικείμενα (16 μαθήματα) του Ενιαίου Λυκείου με κοινό γνώμονα την χρήση ενός ενιαίου εκπαιδευτικού περιβάλλοντος. Με το ειδικά σχεδιασμένο λογισμικό του προγραμματιστικού περιβάλλοντος που σχεδιάστηκε για το μάθημα της Πληροφορικής, οι μαθητές εισάγονται στη μεθοδολογία του δομημένου προγραμματισμού. Με τη χρήση των βασικών δομών προγραμματισμού, δημιουργούν προγράμματα σε μορφή ψευδοκώδικα τα οποία μετατρέπονται σε λογικά διαγράμματα & κωδικοποιούνται αυτόματα σε μία procedural γλώσσα(Basic). Ο ψευδοκώδικας αποτελείται από εντολές που προσεγγίζουν πολύ τη φυσική γλώσσα.



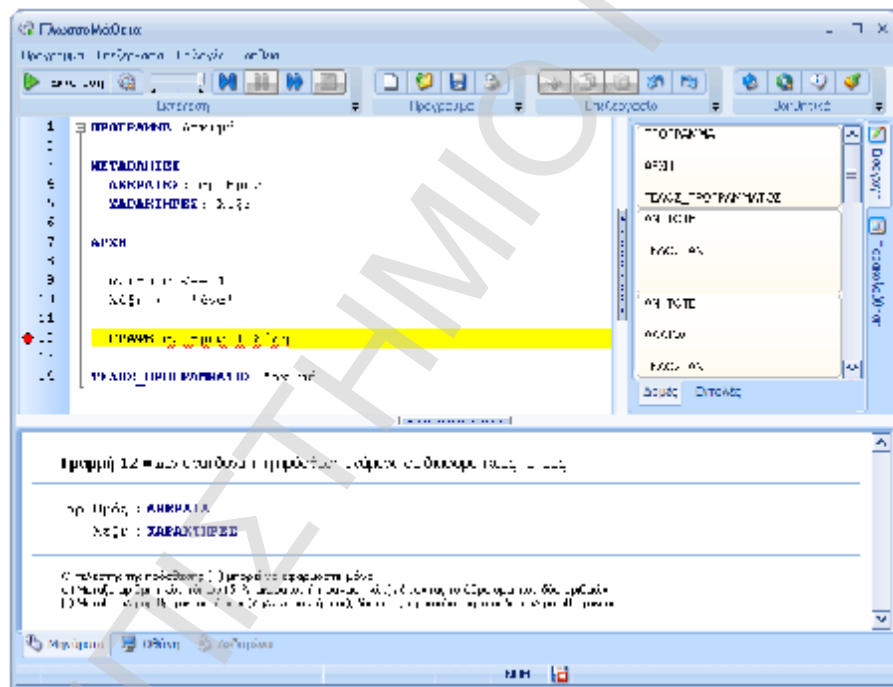
Εικόνα 10 - ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ - Ελήνηωρ

LENS

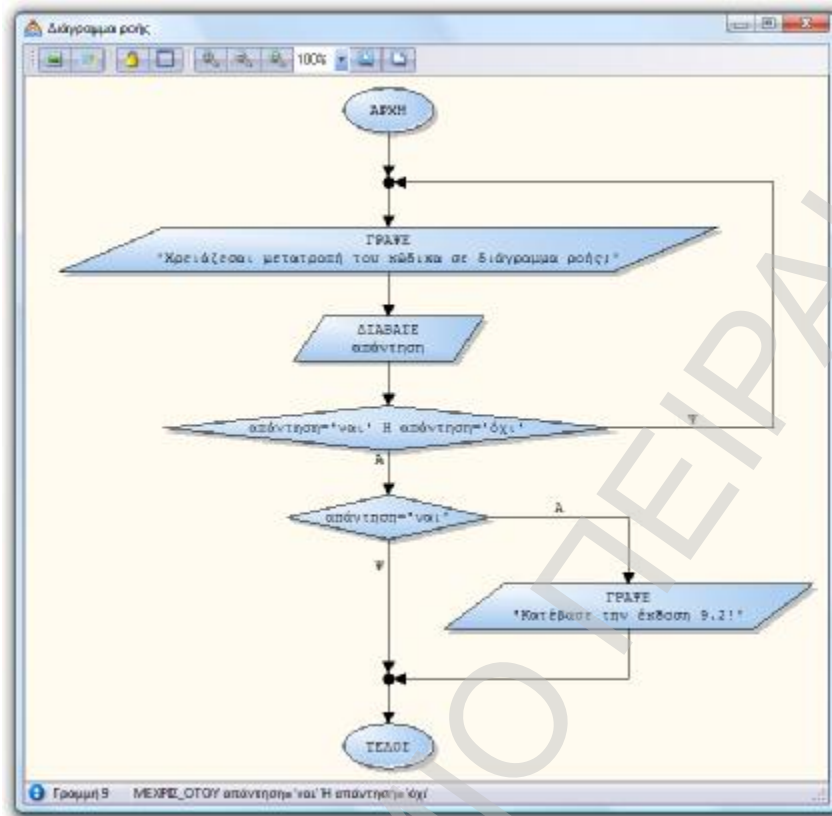
Οι δημιουργοί του συστήματος LENS βασίστηκαν στο ρητό “Πιστεύω μόνο ότι βλέπω” (“Seeing is believing”), δηλαδή, αν μπορεί κάποιος να οπτικοποιήσει κάτι, τότε μπορεί και να το κατανοήσει. Το LENS επιτρέπει στον χρήστη να υλοποιήσει γρήγορα, σε λίγα μόλις λεπτά, αλγόριθμους και να τους οπτικοποιήσει με τη χρήση γραφικών. Και όλα αυτά χωρίς να είναι προαπαιτούμενες οι εκτενείς γνώσεις ούτε στον προγραμματισμό, ούτε στα γραφικά. Το συγκεκριμένο εργαλείο έχει ως τελικό σκοπό να προσδώσει στην αφηρημένες οντότητες του προγραμματισμού μια μορφή και πάνω σ’ αυτήν, ο χρήστης, να χτίσει ένα ισχυρό οικοδόμημα προγραμματιστικής γνώσης. Περιέχεται επίσης ένας αποσφαλματωτή που χρησιμοποιείται για την λεπτομερή εξερεύνηση του κώδικα.

ΓλωσσοΜάθεια

Η ΓλωσσοΜάθεια είναι ένα ολοκληρωμένο εκπαιδευτικό περιβάλλον προγραμματισμού που υλοποιεί τη γλώσσα που διδάσκεται στα πλαίσια του μαθήματος "Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον" (ΑΕΠΠ), του Γενικού Λυκείου. Περιλαμβάνει την συγγραφή προγραμμάτων μέσω της σχεδίασης διαγραμμάτων ροής με γραφικό τρόπο και κατά την εκτέλεση δίνει έμφαση στην εμφάνιση αναλυτικών μηνυμάτων σφάλματος, κατανοητών ακόμα και από αρχάριους χρήστες. Το πρόγραμμα είναι δυνατόν να εκτελεστεί με επιλεγόμενη ταχύτητα με ταυτόχρονη παρακολούθηση των τιμών μεταβλητών και εκφράσεων.



Εικόνα 11 - ΓλωσσοΜάθεια – Προβολή βασικής οθόνης



Εικόνα 12 - ΓλωσσοΜάθεια - Προβολή διαγράμματος ροής

```
! -----  
! Το πρόγραμμα ζητάει έναν τετραψήφιο αριθμό και  
! αποθηκεύει τα ψηφία του σε 4 ανεξάρτητες μεταβλητές.  
! -----  
  
ΠΡΟΓΡΑΜΜΑ Απομόνωση_ψηφίων_τετραψήφιου_αριθμού  
  
ΜΕΤΑΒΛΗΤΕΣ  
  ΑΚΕΡΑΙΕΣ: x, ψ1, ψ2, ψ3, ψ4, υπόλοιπο  
  
ΑΡΧΗ  
  
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ  
    ΓΡΑΨΕ 'Δώσε έναν τετραψήφιο αριθμό:'  
    ΔΙΑΒΑΣΕ x  
    ΑΝ x<1000 Η x>9999 ΤΟΤΕ  
      ΓΡΑΨΕ 'Ο αριθμός αυτός δεν είναι τετραψήφιος'  
    ΤΕΛΟΣ_ΑΝ  
  ΜΕΧΡΙΣ_ΟΤΟΥ x>=1000 ΚΑΙ x<=9999  
  
  ψ1 <-- x DIV 1000  
  υπόλοιπο <-- x MOD 1000  
  ψ2 <-- υπόλοιπο DIV 100  
  υπόλοιπο <-- υπόλοιπο MOD 100  
  ψ3 <-- υπόλοιπο DIV 10  
  ψ4 <-- υπόλοιπο MOD 10  
  
  ΓΡΑΨΕ 'Τα ψηφία του είναι:', ψ1, ψ2, ψ3, ψ4  
  
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Απομόνωση_ψηφίων_τετραψήφιου_αριθμού
```

Εικόνα 13 - Γλωσσομάθεια - Παράδειγμα σε ΓΛΩΣΣΑ

Περιβάλλοντα δημιουργίας προγραμμάτων μέσω μικρόκοσμων

Περιβάλλοντα που επικεντρώνονται στη εισαγωγή στην αντικειμενοστραφή λογική μέσω μικρόκοσμων (microworlds) που βασίζονται σε μινι-γλώσσες (mini-languages). Σε πολλές περιπτώσεις ο μικρόκοσμος έχει επικρατήσει να δηλώνει το συνδυασμό ενός πρωταγωνιστή (actor), ο οποίος ενεργεί στον μικρόκοσμο, και μιας γλώσσας, που συνήθως αναφέρεται σε μικρό σύνολο εντολών, οι οποίες ελέγχουν τη λειτουργία του πρωταγωνιστή. Οι μαθητές μαθαίνουν να προγραμματίζουν καθοδηγώντας τις ενέργειες μιας ή περισσότερων ψηφιακών οντοτήτων (π.χ. γάτας, χελώνας) που ζει σε έναν εικονικό κόσμο (μικρόκοσμο). Τα συγκεκριμένα περιβάλλοντα προγραμματισμού αν και είναι απλά και

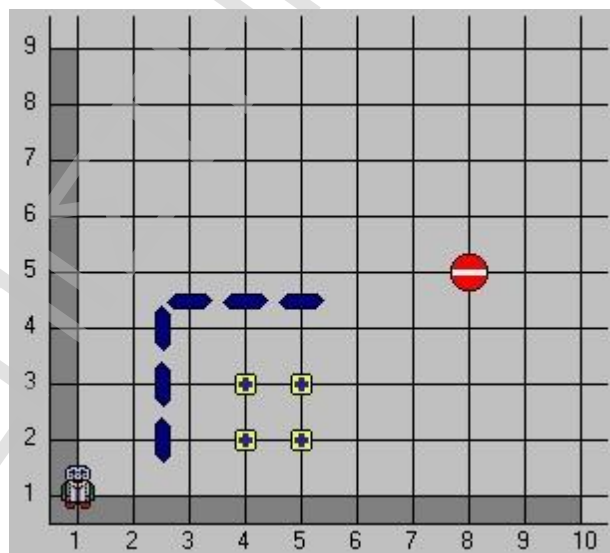
διαισθητικά, αποτελούν ταυτόχρονα και μια ισχυρή μέθοδο διδασκαλίας του προγραμματισμού.

Kara, Playground, Karel++, Karel J Robot, J Karel, Karel the Robot (Pattis, 1995), ObjectKarel (Satratzemi, 2003 και Xinogalos, 2002), Squeak (Inga, 1997)

Microworlds Logo, το οποίο έχει σαν βάση τη γλώσσα προγραμματισμού Logo (Papert, 1980), Stagecast Creator (Smith & Cypher, 1999) και Alice (Burnette T., 1995)

Οικογένεια εργαλείων βασισμένα στο **Karel the Robot**

Ο Karel είναι ένα προγραμματιζόμενο ρομπότ το οποίο ζει σε έναν κόσμο δύο διαστάσεων που αποτελείται από δρόμους (streets), λεωφόρους (avenues), beepers και εμπόδια (οριζόντια και κάθετα). Αποτελεί έναν ευρύτατα χρησιμοποιούμενο μικρόκοσμο που σχεδιάστηκε για την εισαγωγή των νέων χρηστών στον προγραμματισμό. Το πρώτο Karel (1985) έχει ως στόχο την εκμάθηση της γλώσσας Pascal.



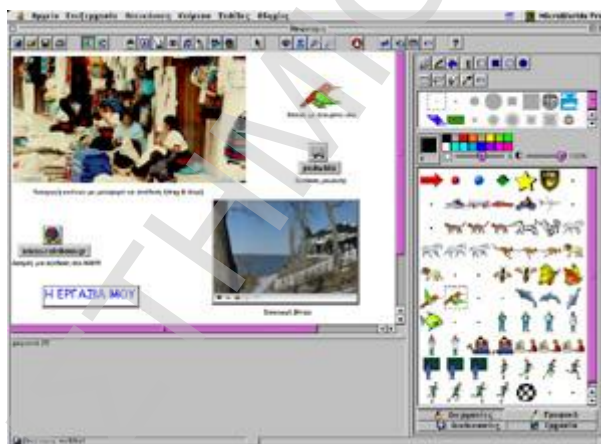
Εικόνα 14 - Ο κόσμος του Karel

Μία εξέλιξη του, το Karel++, εισάγει τον αρχαίο προγραμματιστή στην ιδέα του αντικειμενοστραφούς προγραμματισμού (object-oriented programming - OOP). Σκοπός του Karel++ ήταν να διατηρήσει την απλότητα του αρχικού εργαλείου,

ενώ παράλληλα θα έδειχνε τον τρόπο εργασίας μέσα σε ένα αντικειμενοστραφές περιβάλλον. Για να το πετύχει, βασίστηκε σε μία σύνταξη της γλώσσας προγραμματισμού που προσομοιάζει στην C++ και την Java.

Microworlds Logo

Το MicroWorlds Pro είναι ένα περιβάλλον ανάπτυξης εφαρμογών πολυμέσων και δημιουργίας προσομοιώσεων, με χρήση της γλώσσας προγραμματισμού Logo (Papert, 1980). Μέσα από την δημιουργία δυναμικών, αλληλεπιδραστικών εργασιών οι μαθητές αναπτύσσουν την δημιουργικότητα και την κριτική σκέψη τους. Για την κατασκευή των προσομοιώσεων οι χρήστες δίνουν στους χαρακτήρες εντολές/οδηγίες ώστε να κινούνται στο δυσδιάστατο χώρο της οθόνης. Οι χαρακτήρες αλληλεπιδρούν μεταξύ τους σύμφωνα με τις εντολές των χρηστών.

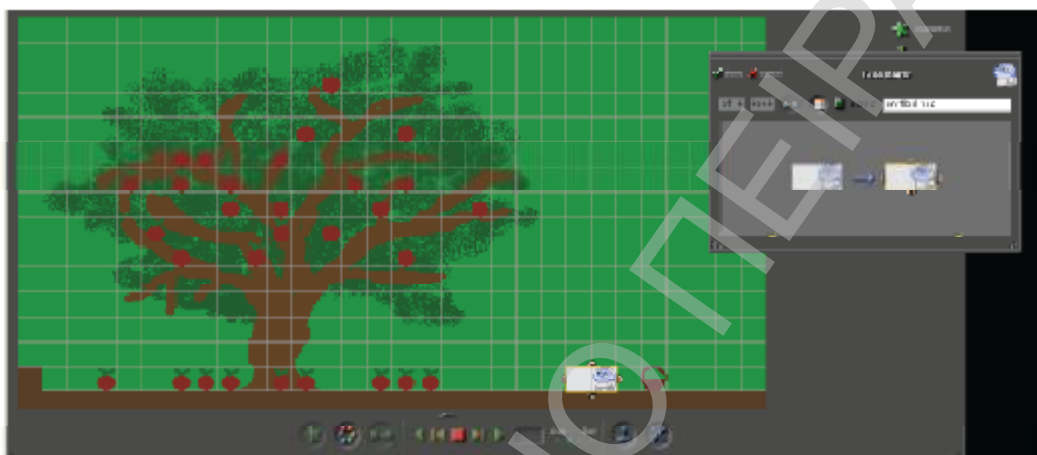


Εικόνα 15 - Το περιβάλλον MicroWorlds

Stagecast Creator

Το Stagecast Creator είναι ένα προγραμματιστικό περιβάλλον σχεδιασμένο για νεαρούς προγραμματιστές. Χρησιμοποιεί ως βάση έναν μικρόκοσμο μέσα στον οποίο οι χαρακτήρες αλληλεπιδρούν σύμφωνα με τους κανόνες που έχει γράψει ο χρήστης. Για την εισαγωγή των κανόνων δεν χρειάζεται η χρησιμοποίηση του πληκτρολογίου. Όταν εκτελείται το πρόγραμμα το Stagecast Creator προσπαθεί να ταιριάζει τον κόσμο γύρω από ένα αντικείμενο με τον όρο ενός κανόνα. Όταν βρεθεί μια τέτοια αντιστοιχία, ο κόσμος γύρω από τον χαρακτήρα αλλάζει όπως

καθορίζεται από τον κανόνα. Σκοπός του εργαλείου είναι να εισάγει τον αρχάριο προγραμματιστή σε βασικές έννοιες όπως οι μεταβλητές, η επανάληψη, η υπό όρους εκτέλεση μιας εντολής και οι διαδικασίες. Ταυτόχρονα αποτελεί έναν ιδανικό χώρο για την κατασκευή μοντέλων στη φυσική και σε άλλες θετικές επιστήμες.



Εικόνα 16 - Το περιβάλλον Stagecast Creator

Alice

Το περιβάλλον Alice είναι ιδανικό για μια πρώτη επαφή των μαθητών με τον αντικειμενοστραφή προγραμματισμό. Μέσα από την δημιουργία μικρών ταινιών ή απλών παιχνιδιών, επιτρέπει στον χρήστη να κατανοήσει τις βασικές αρχές του προγραμματισμού. Ο χρήστης ελέγχει την συμπεριφορά τρισδιάστατων αντικειμένων και χαρακτήρων, χωρίς να χρειάζεται να γράψει σε κάποια γλώσσα προγραμματισμού. Αντίθετα, επιλέγει τις κινήσεις κάθε αντικείμενου μέσα από μια λίστα από επιτρεπόμενες κινήσεις. Με αυτόν τον τρόπο δημιουργεί ένα σενάριο. Οι μαθητές διδάσκονται εκτός των άλλων, πώς να διαχωρίζουν ένα πρόβλημα σε μικρά κομμάτια που είναι εύκολο να λυθούν.

Δυναμική ηχητική προσομοίωση εκτέλεσης προγραμμάτων (software auralisation)

Η αποσφαλμάτωση προγραμμάτων βασίζεται παραδοσιακά στην προσεκτική ανάγνωση του κώδικα και στην εκτέλεσή του. Νεότερα μοντέλα αποσφαλμάτωσης περιλαμβάνουν την οπτικοποίηση του προγράμματος μέσω

γραφικών. Η κατηγορία της ηχητικής προσομοίωσης προτείνει μία ριζοσπαστική ιδέα: την σύνδεση της διαδικασίας της αποσφαλμάτωσης με την ακοή. Βασίζεται στις μοναδικές ικανότητες της ακοής που δεν συναντώνται στις άλλες μας αισθήσεις.

Ο εγκέφαλος, μέσω της ακοής, έχει την ικανότητα:

- να κατανοεί ηχητικά μοτίβα και να αντιλαμβάνεται τυχόν διαφοροποιήσεις
- να κατανοεί και να επεξεργάζεται ταυτόχρονα πληθώρα μελωδιών που προέρχονται από διαφορετικές πηγές (όργανα)
- να αντιλαμβάνεται τις χρησιμοποιούμενες κλίμακες/συγχορδίες και να τις κατηγοριοποιεί (ματζόρε , μινόρε)

Η έρευνες που διεξήχθησαν (Brown, 1997) έδειξαν ότι, παρότι ο ήχος χωρίς ομιλία (non-speech sound) δεν είναι το κατάλληλο εργαλείο για να μεταφέρει στον χρήστη ακριβή ποσοτικά δεδομένα, είναι ιδανικός για την απεικόνιση ποιοτικών διαφοροποιήσεων (π.χ. η μεταβλητή counter αυξάνει) και γενικότερα της συμπεριφοράς, δομής και λειτουργίας του κώδικα.

CAITLIN: Computer Audio Interface to Locate Incorrect Nonsense

Το project CAITLIN (Vickers, 1999) είναι ένα από τα λίγα παραδείγματα αυτής της κατηγορίας. Είναι μία εφαρμογή που μεταφράζει σε πραγματικό χρόνο τον κώδικα της Pascal σε μελωδία χρησιμοποιώντας μουσικά μοτίβα. Κάθε δομή και εντολή μεταφράζεται σε ένα συγκεκριμένο ήχο ή μουσικό θέμα σύμφωνα με ένα ιεραρχικό αυστηρά ορισμένο σύνολο κανόνων. Η αλληλουχία των ήχων δίνει δημιουργεί μια τελική μελωδία. Το εξασκημένο αυτί ακούγοντας την τελική μελωδία μπορεί να αντιληφθεί τα λάθος σημεία του προγράμματος.

Παράδειγμα:

```
Program Algorithm1;  
Var  
A: Integer;  
Begin  
A := 1 ;  
IF A = 10 THEN  
Writeln ('A is 10')  
ELSE  
Writeln ('A is not 10') ;  
end.
```

Η εκκίνηση του ελέγχου IF είναι ο ήχος μιας καμπάνας. Ακολουθεί η μελωδία της εκτέλεσης του ελέγχου (πίانو). Αν ο έλεγχος είναι αληθής, θα ακουστεί μία ματζόρε/μείζονα μελωδία ενός αν είναι ψευδής μία μινόρε/ελάσσονα. Ο ήχος μιας καμπάνας επαναλαμβάνεται για να δείξει τον τερματισμό του IF. Έτσι, ο προγραμματιστής καταλαβαίνει την ροή του προγράμματος.

Πίνακες ταξινόμησης των διδακτικών εργαλείων ανάλογα με τα χαρακτηριστικά τους

Στην συνέχεια ακολουθούν δύο πίνακες ταξινόμησης των παραπάνω διδακτικών εργαλείων σε σχέση με τα χαρακτηριστικά τους. Ο Πίνακας 1, με τίτλο «Χαρακτηριστικά εργαλείων», εκτείνεται σε δύο σελίδες και περιλαμβάνει (οριζόντια) τα εργαλεία που περιγράφηκαν και (κάθετα) τα χαρακτηριστικά που υποστηρίζει το κάθε εργαλείο. Ο εκάστοτε εκπαιδευτικός μπορεί να ανατρέξει στον συγκεκριμένο πίνακα και να βρει το εργαλείο που ταιριάζει στις ανάγκες του μαθήματός του. Παρ' ότι το προτεινόμενο εργαλείο, το Scratch, δεν έχει ακόμη περιγραφεί, περιέχεται στον πίνακα για λόγους πληρότητας.

Ο Πίνακας 2, με τίτλο «Χαρακτηριστικά διδακτικών προσεγγίσεων», αποτελεί μια συγκεντρωτική παρουσίαση των χαρακτηριστικών που υποστηρίζονται από τα εργαλεία που ανήκουν στην κάθε διδακτική προσέγγιση. Στηρίζεται στο γεγονός πως τα εργαλεία κάθε διδακτικής προσέγγισης έχουν, ως επί το πλείστον, κοινά χαρακτηριστικά. Όπως και στον Πίνακα 1, περιλαμβάνεται και το προτεινόμενο εργαλείο Scratch, παρ' ότι δεν έχει ακόμη περιγραφεί, για λόγους πληρότητας. Βρίσκεται σε ξεχωριστή στήλη για να φανούν τα συγκριτικά πλεονεκτήματά του, αφού συνενώνει χαρακτηριστικά από τρεις προσεγγίσεις (Συστήματα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων, Περιβάλλοντα εναλλακτικών μορφών αναπαράστασης της σύνταξης εντολών, Περιβάλλοντα δημιουργίας προγραμμάτων μέσω μικρόκοσμων). Τυπικά, ανήκει στην διδακτική προσέγγιση που κάνει χρήση των Μικροκόσμων.

Διδακτική του Προγραμματισμού για μαθητές Γυμνασίου

		Περιβάλλοντα εναλλακτικών μορφών αναπαράστασης της σύνταξης εντολών			Περιβάλλοντα δημιουργίας προγραμμάτων μέσω μικρόκοσμων					Scratch
		BACCI+ +	Δημιουργός Διαγραμμάτων Ροής	Γλωσσο Μάθεια	Karel the Robot	Micro worlds Logo	Stagecast Creator	Alice	CAITLIN	
Περιβάλλον	Ελληνικό περιβάλλον		ü	ü						ü
Παραγόμενο	Παραγωγή συντακτικά ορθού προγράμματος	ü	ü	ü						ü
	Εξαγωγή σε εμπορική γλώσσα									
Τρόπος σύνταξης προγράμματος	Χρήση γραφικών μονάδων κώδικα									ü
	Καταγραφή του προγράμματος σε κειμένο									
	Χρήση διαγραμμάτων ροής	ü	ü	ü						
Μεταγλωττιστής	Εκτέλεση βήμα-προς-βήμα	ü	ü	ü						ü
	Ανάστροφη εκτέλεση μίας ή περισσότερων γραμμών κώδικα									
	Μεταγλωττιστής που παρήγαγε εύληπτα μηνύματα		ü	ü						
	Μεταγλωττιστής που παρήγαγε ελληνικά μηνύματα		ü	ü						
	Χρήση μικρόκοσμων				ü	ü	ü	ü		ü
	Καθοδήγηση ενός ήρωα				ü	ü		ü		
	Διασύνδεση με τον τρισδιάστατο κόσμο									ü
	Ηχητική προσομοίωση εκτέλεσης								ü	
Υποστήριξη πολυμέσων	Υποστήριξη πολυμέσων									
Γλώσσα	Απλή εκπαιδευτική γλώσσα (όνομα γλώσσας)		ΓΛΩΣΣΑ	ΓΛΩΣΣΑ	ü	Logo	ü			
	Βασισμένο σε εμπορική γλώσσα								ü	ü
	Βοήθεια στη μετάβαση σε ένα επαγγελματικό περιβάλλον									ü
	Αντικειμενοστραφής προγραμματισμός	ü			ü	ü		ü		ü

Πίνακας 2 - Χαρακτηριστικά διδακτικών προσεγγίσεων

	Τροποποίηση εμπορικών γλωσσών	Συστήματα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων	Περιβάλλοντα εναλλακτικών μορφών αναπαράστασης της σύνταξης εντολών	Περιβάλλοντα δημιουργίας προγραμμάτων μέσω μικρόκοσμων	Scratch
Ελληνικό περιβάλλον	ü	ü	ü		ü
Παραγωγή συντακτικά ορθού προγράμματος			ü	ü	ü
Χρήση γραφικών μονάδων κώδικα			ü		ü
Διαγράμματα ροής			ü		
Εκτέλεση βήμα-προς-βήμα	ü	ü	ü	ü	ü
Ανάστροφη εκτέλεση κώδικα		ü			
Εύληπτα μηνύματα αποσφαλματωτή	ü	ü	ü		
Χρήση μικρόκοσμων				ü	ü
Διασύνδεση με τον πραγματικό κόσμο			ü		ü
Υποστήριξη πολυμέσων				ü	ü

Κεφάλαιο 5 - Το περιβάλλον γραφικού προγραμματισμού Scratch

Το Scratch είναι ένα **πολυμεσικό προγραμματιστικό περιβάλλον** σχεδιασμένο για νεαρές ηλικίες που έχουν πρόσβαση σε «φτωχούς» τεχνολογικά πόρους και έχει ως σκοπό την ανάπτυξη της **ευχέρειας στην τεχνολογία** (*technological fluency*). Ο αδόκιμος αυτός όρος προσπαθεί να περιγράψει την εξοικείωση με τις νέες τεχνολογίες και την ενστικτώδη κατανόηση της χρήσης και των δυνατοτήτων τους. Σήμερα, αναφερόμενοι σε τέτοιου είδους τεχνολογίες αναφερόμαστε σε υπολογιστές (φορητούς, σταθερούς, παλάμης), κινητά τηλέφωνα, μικροσυσκευές όπως MP3 players, συσκευές GPS κ.α.

Από την υπερπροσφορά των νέων επιτευγμάτων της τεχνολογίας, εκείνα που προσελκύουν τους νέους περισσότερο είναι τα πολυμέσα και το διαδίκτυο. Γι' αυτό το Scratch στοχεύει στην ανάπτυξη των ικανοτήτων των χρηστών στον προγραμματισμό μέσα από την χρήση των πολυμέσων για την δημιουργία προγραμμάτων και το διαμοιρασμό τους σε ένα παγκόσμιο ελεύθερο δίκτυο. Σύμφωνα με την αρχική ιδέα των δημιουργών του Scratch, όσο οι νέοι δουλεύουν, με ισχυρό κίνητρο, πάνω σε εργασίες στις οποίες βρίσκουν ένα προσωπικό ενδιαφέρον (κινούμενα σχέδια, παιχνίδια, καλλιτεχνικές δραστηριότητες), τόσο θα αναπτύσσουν ικανότητες στην επίλυση μαθηματικών, λογικών και προγραμματιστικών προβλημάτων. Παράλληλα θα ενισχύουν την αυτοπεποίθησή τους στη χρήση των νέων τεχνολογιών.

Συνοπτικά, το Scratch δημιουργήθηκε ώστε:

- Η επαφή των νέων με την τεχνολογία στην εκπαίδευση να μην καταναλώνεται στην απλή χρήση και εκμάθηση του υπολογιστή αλλά να αποσκοπεί στην βαθιά κατανόηση των νέων μέσων και στην απόκτηση μιας γενικής ευχέρειας στις νέες τεχνολογίες.

- Οι νέοι που προέρχονται από χαμηλά κοινωνικά στρώματα να αποκτήσουν ίσες ευκαιρίες ως προς την επαφή τους με τις νέες τεχνολογίες.
- Να επιτευχθεί διάχυση των προηγμένων τεχνολογιών σε ευρύτερη κλίμακα (κοινωνική, οικονομική, πολιτισμική, εθνική, γλωσσική).
- Να επιτευχθεί ευρεία συνεργασία που ξεπερνά κι αυτή με τη σειρά της κοινωνικά, οικονομικά, πολιτισμικά και γλωσσικά σύνορα.

Αρχές σχεδιασμού

Πολλοί ερευνητές συντονίστηκαν για να υλοποιηθεί το Scratch. Μέλη της ομάδας είχαν εργαστεί σε πολλά προγράμματα για τη δημιουργία εκπαιδευτικών εργαλείων προγραμματισμού όπως το LogoBlocks (η γλώσσα προγραμματισμού του εργαλείου LEGO MindStorms) και το Design By Numbers (χρησιμοποιείται από τουλάχιστον 25 πανεπιστήμια για την εισαγωγή στον προγραμματισμό φοιτητών σε σπουδές σχετικά με την τέχνη). Τα χρόνια ενασχόλησής τους στον συγκεκριμένο τομέα ανέδειξαν τις βασικές αρχές πάνω στις οποίες χτίστηκε το νέο εργαλείο και οι οποίες, στοχεύοντας πάντα στο νεανικό κοινό, είναι οι εξής:

- Οι χρήστες θα αποδεχτούν το εργαλείο εφόσον συμβαδίζει με τα ενδιαφέροντα και την αισθητική τους
- Οι χρήστες πρέπει να δουν την δυναμική του εργαλείου με την πρώτη επαφή, γι' αυτό πρέπει να μπορούν να δημιουργήσουν τα πρώτα προγράμματα γρήγορα και εύκολα
- Οι χρήστες πρέπει να μπορούν να επιδείξουν το δημιούργημά τους για να τονιστεί η αυτοπεποίθησή τους
- Το εργαλείο θα πρέπει να υποστηρίζει πολλά είδη δραστηριοτήτων
- Οι προαναφερόμενες δραστηριότητες πρέπει να ενδιαφέρουν διαφορετικά προφίλ χρηστών (π.χ. διαφορετική μητρική γλώσσα, κοινωνικές ομάδες)
- Οι χρήστες πρέπει να μπορούν να «ανακαλύπτουν» τις δυνατότητες του εργαλείου σταδιακά

- Οι χρήστες πρέπει να μπορούν να χρησιμοποιούν το εργαλείο για να φτιάξουν, εξελικτικά, προγράμματα αυξανόμενης δυσκολίας και πολυπλοκότητας

Βασικά χαρακτηριστικά

Σύμφωνα με τις αρχές σχεδιασμού, οι ερευνητές κατέληξαν ότι το εργαλείο θα πρέπει να περιλαμβάνει τα εξής βασικά χαρακτηριστικά:

- **Οπτικός προγραμματισμός**
Ένα γραφικό περιβάλλον που επιτρέπει την δημιουργία δομών κώδικα ενώνοντας γραφικά αρθρώματα κώδικα (blocks), δηλαδή εικόνων, με την απλή λειτουργία της μεταφοράς και απόθεσης (Drag & Drop). Οι διαφορετικοί τύποι δεδομένων και δομών αναπαρίστανται με αρθρώματα διαφορετικών σχημάτων και χρωμάτων έτσι ώστε τα αρθρώματα να μπορούν να κολλήσουν μεταξύ τους μόνο αν διαμορφώνουν συντακτικά ορθές σύνθετες δομές. Έτσι αποκλείονται τα συντακτικά λάθη, ένα από τα σημαντικότερα μειονεκτήματα της κλασικής προσέγγισης στη διδασκαλία του προγραμματισμού, αφήνοντας τον χρήστη να επικεντρωθεί στην υλοποίηση του προγράμματος.
- **Υποστήριξη αντικειμενοστραφούς προγραμματισμού**
Κάθε αντικείμενο στηρίζεται σε μια βιβλιοθήκη λειτουργιών βασισμένη στην κλάση του αντικείμενου. Χρησιμοποιώντας την απλή λειτουργία Drag & Drop, ο χρήστης μεταφέρει αρθρώματα κώδικα και δημιουργεί στοιβάδες αρθρωμάτων, τις διαδικασίες, που ελέγχουν την συμπεριφορά των αντικειμένων. Για την ομαλή προσαρμογή σε πιο σύνθετα προγράμματα, το Scratch βασίζεται σε μία text-based γλώσσα προγραμματισμού. Πρέπει να τονιστεί πως η λέξη «αντικείμενο» έχει την έννοια ενός πραγματικού αντικείμενου για τον χρήστη, π.χ. μια μπάλα, ακόμη κι αν τον εισάγει εμμέσως και στην προγραμματιστική έννοια της λέξης.

- Διαχείριση πολυμέσων μέσω του προγραμματισμού
Όλα τα προγράμματα που τραβάνε την προσοχή του νεανικού κοινού περιλαμβάνουν ευρεία χρήση εικόνων, ήχων, βίντεο και μουσικής. Αντιθέτως η κλασική διδασκαλία του προγραμματισμού περιλαμβάνει μονάχα την διαχείριση αριθμών και κειμένου. Το Scratch είναι το εργαλείο που επιτρέπει τη δημιουργία απλών δραστηριοτήτων που όμως περιλαμβάνουν εκτεταμένη χρήση πολυμέσων. Ένα παράδειγμα αποτελεί η χρήση των φίλτρων για τα γραφικά τα οποία μπορεί ο χρήστης να ενεργοποιήσει και να ελέγξει σε συνάρτηση με το χρόνο εξέλιξης του προγράμματος. Έτσι, δημιουργεί ελκυστικά εφέ ενώ ταυτόχρονα κατανοεί μαθηματικολογικές διεργασίες.
- Εγγενής ελεύθερη διανομή και επίδειξη των προγραμμάτων
Είναι σημαντικό να μπορούν οι χρήστες να μοιράζονται ιδέες, προγράμματα και τεχνικές επίλυσης. Έτσι, το Scratch σχεδιάστηκε για να μπορεί ο καθένας να μοιράσει από απλές διαδικασίες ή αντικείμενα, μέχρι πλήρεις εργασίες. Επιπλέον, αυτός ο διαμοιρασμός διαπερνάει διαφορετικές τεχνολογίες, υπολογιστές, κινητά ή άλλες συσκευές όπως για παράδειγμα, το ρομπότ της LEGO. Η ομάδα υλοποίησης προσβλέπει στην δημιουργία μιας μεγάλης βάσης προγραμμάτων μέσω της οποίας οι χρήστες θα μπορούν διαρκώς να ανταλλάσσουν ιδέες και να βρίσκουν έτοιμες λύσεις.
- Μεταφορά των προγραμμάτων στον πραγματικό τρισδιάστατο χώρο
Οι δημιουργοί του Scratch, βασιζόμενοι στην πρότερη εμπειρία από ερευνητικά προγράμματα όπως το LEGO Mindstorm, το σχεδίασαν έτσι ώστε οι χρήστες να μπορούν να προγραμματίσουν πραγματικά αντικείμενα (μικρο-κινητήρες, LEDs, συσκευές ήχου MIDI) με τον ίδιο τρόπο που προγραμματίζουν εικονικά αντικείμενα στην οθόνη. Η διαδικασία της ανάδρασης (feedback) γίνεται με την χρήση αισθητήρων (απόστασης, κίνησης, επιτάχυνσης, ήχου). Με αυτόν τον τρόπο μπορεί ο χρήστης να ελέγξει πλήρως την συμπεριφορά ενός αντικειμένου.
- Υποστήριξη πολυγλωσσικότητας

Για να επιτευχθεί ένα παγκόσμιο δίκτυο διαμοιρασμού ιδεών και προγραμμάτων θα πρέπει το Scratch να υποστηρίζει διαφορετικές γλώσσες έτσι ώστε ο καθένας να μπορεί να εκφραστεί στην μητρική του γλώσσα. Η υποστήριξη του οπτικού προγραμματισμού επιτρέπει την απρόσκοπτη υποστήριξη διαφορετικών γλωσσών και αλφάβητων. Το εργαλείο επιτρέπει την δυναμική αλλαγή γλώσσας ακόμη και την στιγμή που τρέχει ένα πρόγραμμα.

Υλοποίηση

Το Scratch έχει γραφτεί σε Squeak, μια υλοποίηση της γλώσσας Smalltalk-80 με άδεια ελεύθερου λογισμικού. Η Squeak τρέχει σε πολλές συσκευές και σε πολλά διαφορετικά περιβάλλοντα. Από πλήρης υπολογιστές με Windows, Macintosh, Linux/Unix, Acorn, BeOS, υπολογιστές παλάμης με Windows CE, Zaurus OS, Compaq "Itsy", μέχρι παιχνιδιομηχανές όπως το Sony Playstation. Επιπλέον, τρέχει σε συσκευές χωρίς κάποιο λειτουργικό σύστημα, όπως ο επεξεργαστής της Mitsubishi M32R/D και το developer board StrongARM. Η φορητότητα της Squeak επιτρέπει στους χρήστες να χρησιμοποιήσουν το Scratch μέσα από ένα πλήθος συσκευών. Τέλος, το Scratch τρέχει σε οποιαδήποτε συσκευή υποστηρίζει Java J2ME και έτσι πετυχαίνει την είσοδό του στα κινητά τηλέφωνα.

Ο διαμοιρασμός τμημάτων ή ολοκληρωμένων προγραμμάτων επιτυγχάνεται με έναν συνδυασμό από web servers και ενός server για την αποθήκευσή τους, που ονομάζεται "Scratch Object Library". Ο τελικός χρήστης μπορεί να ανεβάσει τα προγράμματά του, να τρέξει και να κατεβάσει τα προγράμματα άλλων χρηστών χωρίς να φεύγει από το περιβάλλον του Scratch.

Ο κώδικας του Scratch υπόκειται στην άδεια ελεύθερου λογισμικού και μπορεί να χρησιμοποιηθεί και να αλλαχθεί ανάλογα με τις ανάγκες του καθενός.

Τρία τυπικά σενάρια χρήσης

Για την κατανόηση των δυνατοτήτων του εργαλείου παρατίθενται τρία τυπικά σενάρια χρήσης τα οποία δείχνουν επιπλέον πως με το Scratch η διδασκαλία δεν στηρίζεται σε ένα σφιχτό και αυστηρό πλαίσιο μαθήματος αλλά στο

πραγματικό ενδιαφέρον των μαθητών να δημιουργήσουν κάτι που έχει αξία γι' αυτούς.

Επεξεργασία εικόνας μέσω προγραμματισμού

Η εξοικείωση με το Photoshop ή συναφείς εφαρμογές είναι πολύ συνηθισμένη στις νεανικές ηλικίες. Χρησιμοποιούνται κυρίως φίλτρα/εφέ (blur, distort, pixelate, sharpen) για την επεξεργασία εικόνων και φωτογραφιών. Το Scratch επιδιώκει να εφοδιάσει τους χρήστες με πιο δημιουργικά εφέ και καλύτερο έλεγχο των φίλτρων με στόχο την βελτίωση της καλλιτεχνικής έκφρασης τους. Επίσης, εισάγει την παράμετρο του χρόνου ξεπερνώντας τα όρια της σταθερής εικόνας και φτάνει στο βίντεο. Πιο συγκεκριμένα, επιτρέπει στον χρήστη να χρησιμοποιήσει πολλές διαφορετικές τεχνικές:

- Ο χρήστης δουλεύοντας με την φωτεινότητα (brightness) σε σχέση με το χρόνο μπορεί να ελέγξει πως θα εφαρμόζεται το φίλτρο (γράφοντας μαθηματικές μεθόδους που ελέγχουν τις παραμέτρους του φίλτρου) έτσι ώστε να υλοποιήσει το φαινόμενο fade in για να εμφανίζεται σταδιακά η εικόνα.
- Μπορεί να γράψει κώδικα για να ελέγξει τον κορεσμό (saturation) της εικόνας και με αυτόν τον τρόπο να δημιουργήσει βίντεο με αλλαγές ανάμεσα στην έγχρωμη και ασπρόμαυρη έκδοσή της.
- Μπορεί να επεξεργαστεί το χρωματικό μοντέλο RGB έτσι ώστε να αλλάξει συγκεκριμένα χρώματα της εικόνας.
- Μπορεί να επεξεργαστεί το κανάλι της διαφάνειας (alpha channel) έτσι ώστε να συνθέσει διαφορετικές εικόνες.

Συνδυάζοντας τις παραπάνω τεχνικές μπορεί ο καθένας να δημιουργήσει ένας πλήρες βίντεο ως αναφορά το περιεχόμενο και την αισθητική του.

Μουσική και παιχνίδι σε τρισδιάστατο χώρο

Το "Dance Dance Revolution" (DDR) είναι ένα μουσικό ηλεκτρονικό παιχνίδι που χρησιμοποιεί ένα ειδικό «χαλί» με αισθητήρες όπου ο παίκτης ελέγχει το πρόγραμμα με τις κινήσεις του πάνω στο χαλί. Με το Scratch ο καθένας μπορεί

να δημιουργήσει πολλές εκδοχές του DDR. Χρειάζεται να σχεδιάσει ένα «χαλί» με τέσσερις αισθητήρες-κουμπιά, να τους συνδέσει στον υπολογιστή και χρησιμοποιώντας MIDI αρχεία να ελέγξει το συγχρονισμό των αισθητήρων με τη μουσική. Ο σχεδιασμός του ειδικού «χαλιού» μπορεί να χρησιμοποιηθεί σε πολλές άλλες περιπτώσεις, όπως στο να δημιουργήσει ο χρήστης μουσική αντί να την ακολουθεί (είτε με νότες είτε με έτοιμους ήχους-samples) ή να το χρησιμοποιήσει σαν χειριστήριο.

Ανταλλαγή χαρακτήρων

Η δημιουργία χαρακτήρων ήταν πάντα ένα αγαπημένο θέμα. Με το Scratch η δημιουργικότητα φτάνει σε νέα επίπεδα. Η δημιουργία ενός χαρακτήρα μέσω του σχεδιασμού εικόνων και την επιλογή των ήχων που τον αντιπροσωπεύουν είναι στο Scratch μια απλή διαδικασία. Επιπλέον, είναι πολλή εύκολη η ανταλλαγή χαρακτήρων έτσι ώστε να χρησιμοποιηθούν σε διαφορετικά προγράμματα και από διαφορετικούς χρήστες. Ακόμα, το Scratch καταγράφει σε ένα δενδροειδές γράφημα που και πως χρησιμοποιήθηκε ο κάθε χαρακτήρας του και τι αλλαγές υπέστη.

Τα εκπαιδευτικά πλεονεκτήματα του Scratch

Συνοπτικά, το περιβάλλον του Scratch προσφέρει:

- Εισαγωγή στις αρχές προγραμματισμού για άτομα χωρίς προηγούμενη προγραμματιστική εμπειρία
- Ένα πλήρως εξελληνισμένο περιβάλλον
- Χρήση παραστάσεων από την καθημερινή ζωή
- Εύκολο έλεγχο μίας οντότητας και διαδραστικότητα με το περιβάλλον
- Εμφανή αλληλεπίδραση μεταξύ των διαφορετικών αντικειμένων και έλεγχός της με απλό τρόπο
- Άμεση ανταπόκριση στις εντολές του χρήστη

Δυνατότητες εφαρμογής στην εκπαίδευση

Το γραφικό περιβάλλον του Scratch θα μπορούσε να χρησιμοποιηθεί για μια ευχάριστη και παιγνιώδη εισαγωγή στον προγραμματισμό στις τελευταίες τάξεις του Γυμνασίου. Επίσης, θα μπορούσε να αξιοποιηθεί στο Λύκειο ως ένα καλό προπαρασκευαστικό εργαλείο για την εισαγωγή των μαθητών σε μια ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού, την Java, την οποία μαθητές της βαθμίδας αυτής μπορούν να οικειοποιηθούν αν την προσεγγίσουν μέσα από κατάλληλες εκπαιδευτικές δραστηριότητες.

Το Scratch έχει χρησιμοποιηθεί με επιτυχία σε διάφορες βαθμίδες εκπαίδευσης στο εξωτερικό, κυρίως στην Γερμανία, την Αυστρία και την Ελβετία. Όσον αφορά στη δευτεροβάθμια εκπαίδευση των παραπάνω χωρών, έρευνες που έγιναν σε μαθητές Γυμνασίου που χρησιμοποίησαν το περιβάλλον Scratch στα σχολεία τους και οι οποίοι είχαν μηδενική ή περιορισμένη προηγούμενη εμπειρία στον προγραμματισμό, έδειξαν ότι:

1. Οι αντιδράσεις των μαθητών ήταν πολύ θετικές.
2. Το περιβάλλον κέντρισε το ενδιαφέρον των μαθητών και έδρασε παρακινητικά, ιδιαίτερα για τους μαθητές χωρίς προηγούμενη προγραμματιστική εμπειρία.
3. Οι μαθητές βρήκαν τη διεπαφή της εφαρμογής πολύ απλή και εύχρηστη.
4. Οι μαθητές εκτίμησαν ιδιαίτερα με το γεγονός ότι μετά την σύντομη φάση της αρχικής εξοικείωσης με το περιβάλλον άρχισαν πολύ γρήγορα να επιλύουν ενδιαφέροντα προβλήματα.
5. Η έλλειψη τεχνικών δυσκολιών, για παράδειγμα τα συντακτικά λάθη, επέτρεψε στους μαθητές να εστιάσουν στην επίλυση των προβλημάτων.

Επίσης, η απλή και διαισθητική διεπαφή χρήστη του Scratch ελαχιστοποιεί το χρόνο εκμάθησης της χρήσης του, πράγμα που έχει ιδιαίτερη σημασία για τη σχολική εκπαίδευση όπου οι ώρες διδασκαλίας της Πληροφορικής είναι συνήθως περιορισμένες.

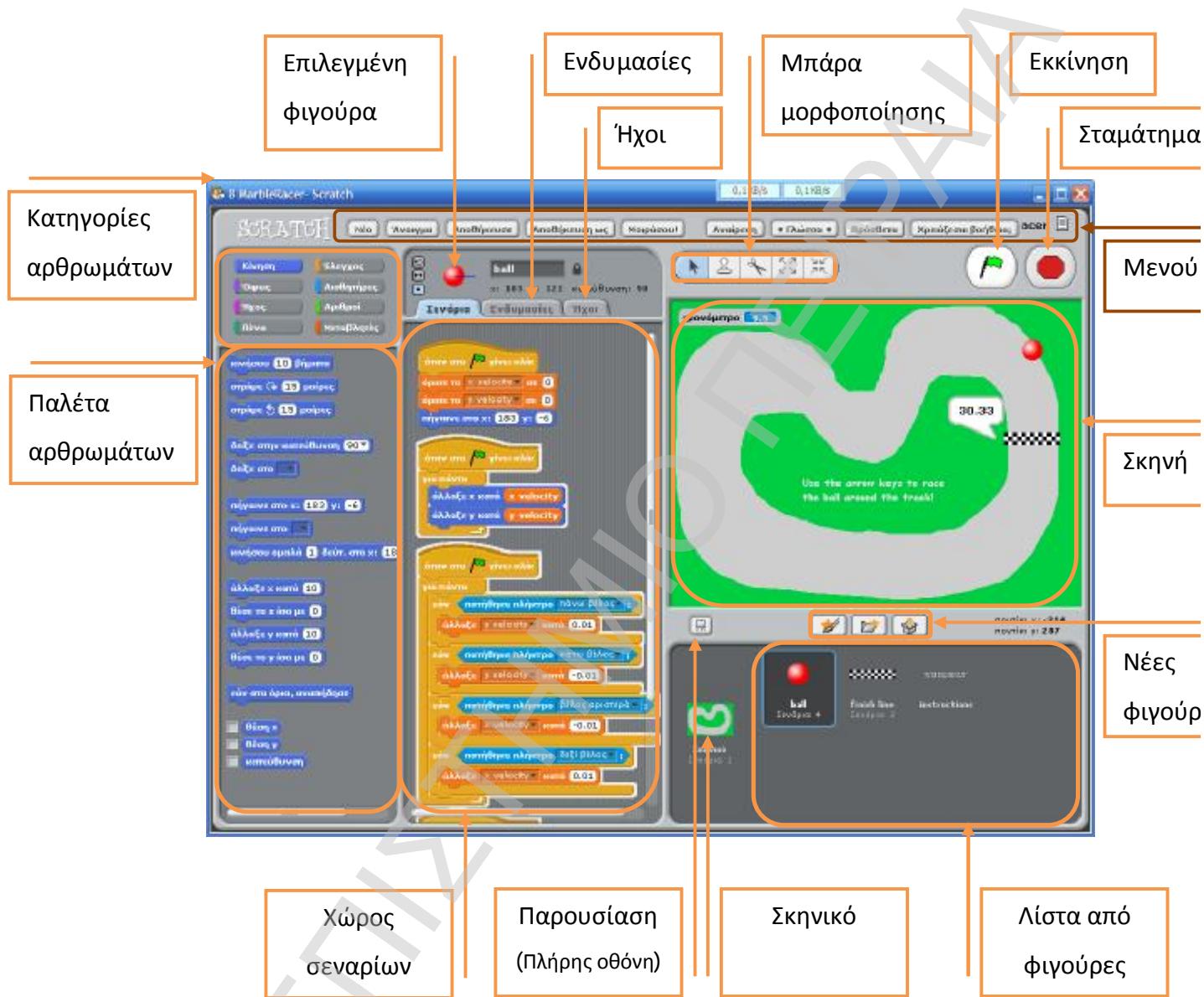
Περιγραφή του εργαλείου

Τα βασικά συστατικά ενός προγράμματος στο Scratch

Το βασικό συστατικό του Scratch είναι η **φιγούρα** (sprite), δηλαδή σε προγραμματιστικούς όρους το αντικείμενο. Μία φιγούρα περιέχει διαφορετικές **ενδυμασίες** (costumes), δηλαδή εικόνες, και μια συλλογή από **ήχους**.

Ο χρήστης μπορεί να δώσει **εντολές** σε μία φιγούρα για να κινηθεί, να αλλάξει ενδυμασία, να παίξει κάποιον ήχο ή να αλληλεπιδράσει με μια άλλη φιγούρα. Για να ορίσει ο χρήστης την συμπεριφορά της φιγούρας, ενώνει αρθρώματα από εντολές σε στοιβάδες που ονομάζουμε **σενάρια** (scripts). Κάνοντας διπλό κλικ σε ένα σενάριο, **ενεργοποιείται** και τρέχουν οι εντολές σειριακά από πάνω προς τα κάτω.

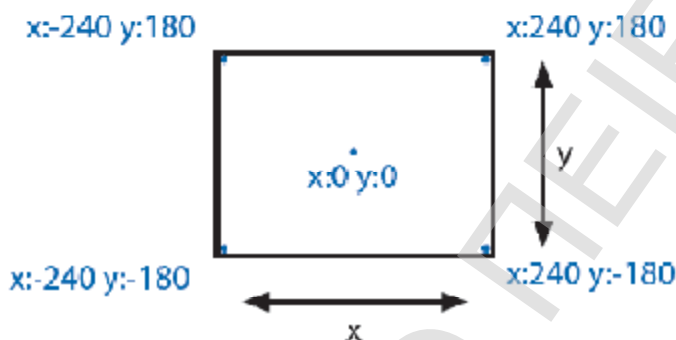
Η κεντρική οθόνη του Scratch



Εικόνα 17- Κεντρική οθόνη

Σκηνή

Η οθόνη εξόδου του Scratch. Στην σκηνή λαμβάνουν χώρα όλα τα σενάρια. Αποτελείται από ένα ορθογώνιο 480 επί 360 μονάδων που χωρίζεται σε ένα δισδιάστατο πλέγμα με κέντρο το κέντρο του ορθογωνίου και συντεταγμένες των ακραίων σημείων όπως φαίνονται στο παρακάτω σχήμα:






Εικόνα 18 - Σκηνή

Πατώντας το κουμπί «Παρουσίαση», η σκηνή καταλαμβάνει όλη την οθόνη. Πατώντας το πλήκτρο Esc επιστρέφουμε στην κεντρική οθόνη.

Νέες φιγούρες

Ξεκινώντας ένα νέο πρόγραμμα πρέπει ο χρήστης να εισάγει ή να δημιουργήσει τις φιγούρες που θέλει να χρησιμοποιήσει.

-  Δημιουργία μιας νέας φιγούρας μέσω του Επεξεργαστή ζωγραφικής
-  Επιλογή νέας φιγούρας από το αρχείο
-  Εισαγωγή μιας τυχαίας φιγούρας

Λίστα από φιγούρες

Εμφανίζει μία λίστα από μικρογραφίες των φιγούρων του προγράμματος. Για κάθε φιγούρα εμφανίζει, επίσης, το όνομά της και τον αριθμό των σεναρίων

που περιέχει. Για να την επεξεργασία μίας φιγούρας πρέπει πρώτα ο χρήστης να την επιλέξει από την λίστα φιγούρων.



Εικόνα 19 - Λίστα από φιγούρες

Σκηνικό

Η εικόνα που αντιπροσωπεύει την ίδια τη σκηνή, εμφανίζεται δίπλα στις φιγούρες και ονομάζεται σκηνικό. Αποτελεί ένα ειδικό αντικείμενο με περιορισμένες ιδιότητες.



Εικόνα 20 - Σκηνικό

Παλέτα αρθρώσεων και χώρος σεναρίων

Καταλαμβάνουν μαζί το μεγαλύτερο μέρος της κεντρικής οθόνης και είναι ο χώρος που ο χρήστης δημιουργεί το πρόγραμμα. Για να ελέγξει ένας χρήστης μία φιγούρα, μεταφέρει αρθρώματα από την παλέτα στον χώρο σεναρίων. Για να ενεργοποιήσει ένα άρθρωμα αρκεί ένα διπλό κλικ πάνω του. Για την δημιουργία μιας σειράς από κινήσεις, ο χρήστης ενώνει αρθρώματα και δημιουργεί σεσάρια. Τα αρθρώματα είναι χωρισμένα σε κατηγορίες ανάλογα με τον τύπο τους.

Οι κατηγορίες είναι οι εξής:

Κίνηση	Έλεγχος της κίνησης της φιγούρας
Όψεις	Αλλαγή ενδυμασιών, μεγέθους φιγούρας και πλαίσια διαλόγου
Ήχος	Έλεγχος της αναπαραγωγής των ήχων
Πένα	Έλεγχος της πένας <i>Η πένα είναι ένα εργαλείο καταγραφής της πορείας μίας φιγούρας</i>

Έλεγχος	Έλεγχος της ροής με δομές ελέγχου και επανάληψης
Αισθητήρες	Έλεγχος του χρόνου και των συσκευών εισόδου (ποντίκι, πληκτρολόγιο, αισθητήρες)
Αριθμοί	Μαθηματικές και λογικές συναρτήσεις
Μεταβλητές	Έλεγχος των μεταβλητών

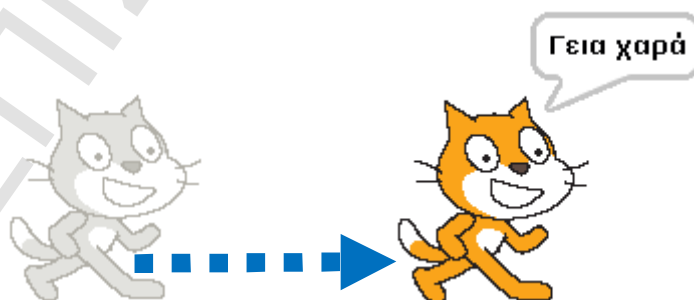
Παράδειγμα

Η φιγούρα «Γάτα» πρέπει να κινηθεί 10 βήματα και μετά να πει «Γεια χαρά». Ο χρήστης εισάγει την εντολή/άρθρωμα «κινήσου 10 βήματα» και στην συνέχεια την εντολή «πες Γεια χαρά». Την δεύτερη εντολή την ενώνει στο κάτω μέρος της πρώτης και έχουμε το παρακάτω αποτέλεσμα:



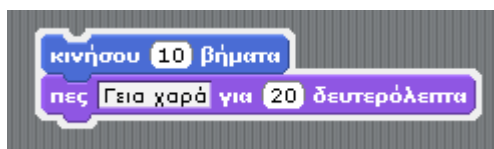
Εικόνα 21 - Ένα απλό σενάριο

Η δομή που δημιουργήθηκε είναι μια πολύ απλή μορφή σεναρίου. Κάνοντας διπλό κλικ πάνω του, η φιγούρα πραγματοποιεί τις εντολές του χρήστη τη μία μετά την άλλη, ξεκινώντας από πάνω προς τα κάτω.



Εικόνα 22 - Το αποτέλεσμα του σεναρίου

Για όση διάρκεια εκτελείται το σενάριο, υπάρχει ένα λευκό πλαίσιο γύρω του που υποδεικνύει ότι είναι ενεργοποιημένο.



Εικόνα 23 - Ενεργοποιημένο σενάριο

Διαχείριση σεναρίων

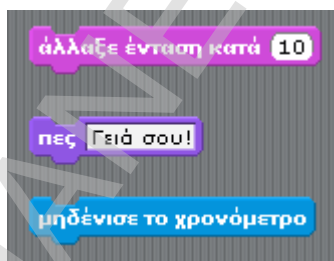
Η θέση των σεναρίων στον χώρο δεν παίζει ρόλο στην σειρά εκτέλεσης τους. Για την μετακίνηση ενός σεναρίου, ο χρήστης κάνει κλικ στο πρώτο άρθρωμα και κρατώντας πατημένο το ποντίκι το μετακινεί στην επιθυμητή θέση. Αντίθετα, για να διαχωρίσει τις εντολές ενός σεναρίου, κάνει κλικ και μετακινεί το άρθρωμα που θέλει να αποκοπεί και το μετακινεί σε νέα θέση. Μαζί με το επιλεγμένο άρθρωμα θα μετακινηθούν και αυτά που βρίσκονται από κάτω. Για να δημιουργήσει ένα αντίγραφο ενός σεναρίου στην ίδια ή σε άλλη φιγούρα, το επιλέγει και το αποθέτει πάνω σε μία φιγούρα από την Λίστα φιγούρων.

Αρθρώματα

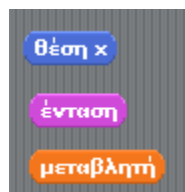
Τα αρθρώματα που έχουν εσοχές και εξοχές, σαν κομμάτια παζλ, είναι εντολές.

Τα αρθρώματα που είναι ορθογώνια με λειασμένες γωνίες είναι μεταβλητές. Χωρίζονται σε μεταβλητές συστήματος και χρήση.

Τα αρθρώματα με γωνίες στις πλάγιες πλευρές ελέγχουν τιμές και πηγές εισόδου. Επίσης, συγκρίνουν μεταβλητές και σταθερές τιμές.



Εικόνα 24 – Εντολές



Εικόνα 25 - Μεταβλητές



Εικόνα 26 – Έλεγχοι

Ορισμένα αρθρώματα έχουν ελεύθερα πεδία που δέχονται εμφωλευμένα άλλα αρθρώματα ή σταθερές τιμές, όπως αριθμούς ή αλφαριθμητικά. Τα ελεύθερα πεδία με λειασμένες γωνίες δέχονται σταθερές τιμές ή αρθρώματα μεταβλητών, ενώ εκείνα με γωνίες στις πλάγιες πλευρές δέχονται αρθρώματα ελέγχου. Με αυτόν τον τρόπο, κάθε δυνατός συνδυασμός είναι ταυτόχρονα και έγκυρος συντακτικά.

Ενδυμασίες

Οι ενδυμασίες είναι οι εικόνες που παίρνει η κάθε φιγούρα. Οι εικόνες αυτές μπορούν να εισαχθούν από προϋπάρχοντα αρχεία εικόνων ή να δημιουργηθούν με τον Επεξεργαστή ζωγραφικής. Τα κουμπιά των αντίστοιχων λειτουργιών φαίνονται στην παρακάτω εικόνα.

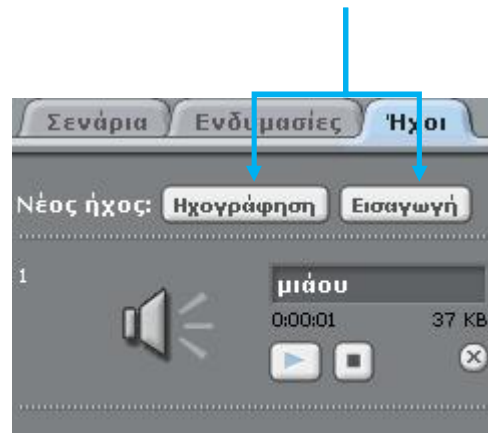


Εικόνα 27 – Ενδυμασίες

Στην παραπάνω εικόνα φαίνεται ότι η συγκεκριμένη φιγούρα περιέχει δύο ενδυμασίες. Η ενδυμασία που χρησιμοποιεί αυτή τη στιγμή η φιγούρα είναι η πρώτη, η οποία φαίνεται τονισμένη. Για να αλλάξει ενδυμασία η φιγούρα, αρκεί ο χρήστης να κάνει κλικ σε κάποια άλλη ενδυμασία.

Ήχοι

Είναι οι ήχοι που μπορεί να αναπαράγει η κάθε φιγούρα. Ένας ήχος μπορεί να εισαχθεί από ένα προϋπάρχον αρχείο ήχου ή να ηχογραφηθεί εκείνη τη στιγμή. Τα κουμπιά των αντίστοιχων λειτουργιών φαίνονται στην παρακάτω εικόνα.




Εικόνα 28 - Ήχοι

Μπάρα μορφοποίησης

Η μπάρα μορφοποίησης φιγούρων αφορά τις παρακάτω λειτουργίες.

	Βέλος	Επιλογή και μετακίνηση φιγούρων και σεναρίων
	Διπλότυπο	Δημιουργία αντιγράφου για φιγούρες, ενδυμασίες, ήχους, σεσάρια ή αρθρώματα
	Διαγραφή	Διαγραφή φιγούρων, ενδυμασιών, ήχων, σεναρίων ή αρθρωμάτων
	Μεγέθυνση	Μεγέθυνση φιγούρων
	Σμίκρυνση	Σμίκρυνση φιγούρων


Κουμπιά εκκίνησης και σταματήματος

Το κουμπί εκκίνησης  είναι ο ιδανικός τρόπος για την εκκίνηση πολλών σεναρίων ταυτόχρονα. Όσα σεσάρια ξεκινούν σε το ακόλουθο άρθρωμα ελέγχου

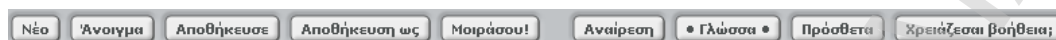


θα ξεκινήσουν την εκτέλεση τους ταυτόχρονα με το πάτημα του κουμπιού εκκίνησης. Είναι εύκολα κατανοητό ότι η εκκίνηση ενός

πλήρους προγράμματα ξεκινά με αυτόν τον τρόπο. Αντίστοιχα, το κουμπί

σταματήματος  ακυρώνει κάθε σενάριο που εκτελείται.

Γραμμή μενού



Εικόνα 29 - Γραμμή μενού

Νέο, Άνοιγμα, Αποθήκευση, Αποθήκευση ως : διαχειρίζονται τα αρχεία προγραμμάτων του Scratch ακριβώς όπως το Microsoft Word τα δικά του έγγραφα.

Μοιράσου! : Ανοίγει τη φόρμα διαμοιρασμού προγραμμάτων του Scratch. Μόλις συμπληρωθούν τα απαραίτητα στοιχεία το πρόγραμμα «ανεβαίνει» στον server του Scratch (<http://scratch.mit.edu>) και είναι ελεύθερο για χρήση μέσω διαδικτύου.



Εικόνα 30 - Φόρμα διαμοιρασμού

Undo : Επιτρέπει την επαναφορά του τελευταίου αρθρώματος ή σεναρίου που διαγράφηκε.

Γλώσσα : Επιτρέπει την αλλαγή γλώσσας του περιβάλλοντος του Scratch χωρίς να χρειαστεί επανεκκίνηση της εφαρμογής.

Πρόσθετα : Ανοίγει ένα νέο μενού που περιέχει τις εξής λειτουργίες

Εισαγωγή έργου : Εισάγει όλες τις φιγούρες και τα σκηνικά από ένα άλλο έργο στο ήδη ανοικτό.

Ξεκίνησε απλό βηματισμό: Εκτέλεση του προγράμματος βήμα-βήμα. Χρήσιμο είτε για την εύρεση λαθών (debugging) είτε για την κατανόηση της ροής του προγράμματος από αρχάριους χρήστες.

Συμπίεσε ήχους και συμπίεσε εικόνες : Συμπίεση των αντίστοιχων πόρων για την ελαχιστοποίηση του όγκου του προγράμματος.

Σύνοψη

Μέσα από αυτήν την εκτενή αναφορά στη λειτουργικότητα και τα χαρακτηριστικά του Scratch φαίνεται και η χρησιμότητα του συγκεκριμένου εργαλείου στην διδακτική του Προγραμματισμού στο ελληνικό σχολείο. Για πρώτη φορά ένα προγραμματιστικό εργαλείο περιλαμβάνει ταυτόχρονα χαρακτηριστικά όπως το ελληνικό περιβάλλον, η παραγωγή συντακτικά ορθού προγράμματος μέσω της χρήσης γραφικών μονάδων κώδικα, η χρήση μικρόκοσμων και η υποστήριξη πολυμέσων.

Επίσης, η πιλοτική του εφαρμογή σε σχολεία δευτεροβάθμιας εκπαίδευσης σε πολλές χώρες και η επιτυχία αυτών των προγραμμάτων αποτελεί ένα ακόμα θετικό δείγμα προς την καθιέρωση αυτού του νέου, πρωτοποριακού εργαλείου.

Στο επόμενο κεφάλαιο αφού περιγραφούν τα βασικά στοιχεία ενός σεναρίου, θα παρουσιαστεί ένα πλήρες σενάριο για την διδασκαλία μιας διδακτικής ενότητας του ελληνικού Αναλυτικού Προγράμματος στην Πληροφορική.

Κεφάλαιο 6 - Ένα πρωτότυπο σενάριο

Θεωρία σύνταξης σεναρίων

Εκπαιδευτικά Σενάρια

Εκπαιδευτικό σενάριο (educational scenario) είναι η περιγραφή ενός μαθησιακού πλαισίου με εστιασμένο γνωστικό(ά) αντικείμενο(α), συγκεκριμένους εκπαιδευτικούς στόχους, παιδαγωγικές αρχές και σχολικές πρακτικές. Ένα τέτοιο σενάριο ενέχει (υλοποιείται μέσα από) σειρά εκπαιδευτικών δραστηριοτήτων (educational activities). Η δομή και ροή κάθε δραστηριότητας καθώς και οι ρόλοι του διδάσκοντα-μαθητών και η αλληλεπίδρασή τους με τα όποια χρησιμοποιούμενα μέσα και υλικό, περιγράφονται σε ένα πλάνο δραστηριότητας (activity plan). Μέρος ενός τέτοιου πλάνου προβλέπει τη χρήση λογισμικού.

Έτσι λοιπόν η περιγραφή των σεναρίων δεν αφορά μόνο στο σχεδιασμό εργαλείων και δραστηριοτήτων βασισμένων σε κάποιο επιλεγμένο λογισμικό ή στην συνέργια δύο ή περισσότερων λογισμικών. Αντίθετα το σενάριο αποτελεί μια εκ βαθέων διείσδυση στη διδακτική πρακτική. Μέσω αυτού γίνεται ανάλυση της μεθόδου και της στρατηγικής σύμφωνα με την οποία προτείνουμε να εφαρμοστεί το σύνολο των σχεδιασμένων δραστηριοτήτων στην τάξη και στους προτεινόμενους ρόλους που καλούνται να παίξουν οι συμμετέχοντες (μαθητές, καθηγητές, επιμορφωτές και διοίκηση του σχολείου), αλλά και στη δομή της συνεργασίας μεταξύ διαφορετικών ομάδων (η τάξη σαν σύνολο, μικρές ομάδες μαθητών στην ίδια τάξη ή σε διαφορετικές τάξεις). Το σενάριο πρέπει να είναι σχεδιασμένο με τέτοιο τρόπο ώστε να επιτρέπει παρέμβαση και αλλαγή από μέρους του καθηγητή και δημιουργική χρήση του σεναρίου στην τάξη. Επίσης πρέπει να έχει τη δυνατότητα επέκτασης ανάλογα με τον αριθμό των μαθητών και τάξεων που παίρνουν μέρος.

Η συγγραφή των σεναρίων λαμβάνει υπόψη τα παρακάτω:

- Την πολυπλοκότητα της παιδαγωγικής, διδακτικής και γνωστικής προσέγγισης της πραγματευόμενης ενότητας με χρήση της συγκεκριμένης τεχνολογίας και του προτεινόμενου σεναρίου.
- Την εμπλοκή διαφόρων γνωστικών περιοχών και διαφορετικής φύσης παραγόντων στην εκμάθηση και διδασκαλία μιας διδακτικής ενότητας και την ανάγκη συνέργιας μεταξύ τους.
- Την ανάγκη συνεργασίας μεταξύ τεχνικών (ανθρώπων που έχουν εμπειρία στην δημιουργία και χρήση τεχνολογικών εργαλείων) και παιδαγωγών (ανθρώπων που έχουν εμπειρία στην διδακτική και μάθηση γνωστικών εννοιών) για την επιτυχή δημιουργία και συγγραφή ρεαλιστικών σεναρίων.
- Την άμεση ανταπόκριση των σεναρίων στην καθημερινή σχολική πραγματικότητα και την συμβολή τους στην αναβάθμιση της διδακτικής πρακτικής, της εξομάλυνσης δυσκολιών κατανόησης και της μαθησιακής επικοινωνίας.
- Την ανάγκη επικοινωνίας των σεναρίων (δραστηριότητες, εργασίες, ασκήσεις, μέθοδος εφαρμογής στην τάξη και στο σχολείο) με μαθητές, καθηγητές, διεύθυνση σχολείων και άλλων σχετικών φορέων.

Η δομή ενός σεναρίου

Ένα τυπικό εκπαιδευτικό σενάριο περιέχει τα παρακάτω:

- Γνωστικό αντικείμενο
- Τίτλος μαθήματος
- Ένταξη μαθήματος στο αναλυτικό πρόγραμμα
- Εκτιμώμενη διάρκεια
- Στόχοι Διδασκαλίας
 - Μαθησιακοί
 - Παιδαγωγικοί
- Κεντρική ιδέα

- Εμπλεκόμενοι ρόλοι
- Δραστηριότητες εμπλεκόμενων ρόλων
- Προαπαιτούμενη γνώση
- Στρατηγικές εφαρμογής
- Τεχνολογικό υλικό και λογισμικό
- Συμπληρωματικό υλικό
- Προετοιμασία
- Διδακτική διαδικασία
- Δραστηριότητες
- Προτάσεις για επέκταση / διαφοροποίηση

Στην συνέχεια θα περιγράψουμε αναλυτικότερα τις πιο βασικές υποενότητες. Πρέπει να αναφέρουμε ότι ανάλογα με τη φύση του σεναρίου κάποιες υποενότητες μπορεί να παραλειφθούν.

Περιγραφή ενοτήτων σεναρίου

Ένταξη μαθήματος στο αναλυτικό πρόγραμμα

Το σενάριο πρέπει να έχει ξεκάθαρους επιδιωκόμενους διδακτικούς στόχους, να βασίζεται σε καλά τεκμηριωμένη παιδαγωγική θεώρηση, και να γίνεται σαφής η σύνδεση του με το αναλυτικό πρόγραμμα.

Μαθησιακοί και παιδαγωγικοί διδακτικοί στόχοι

Αναλυτική περιγραφή των χαρακτηριστικών του σεναρίου σε σχέση με το γνωστικό και παιδαγωγικό προσδοκώμενο και την κοινωνική και πολιτισμική επιρροή που έχει στη σχολική κοινότητα.

Σε περιπτώσεις διαθεματικότητας, διάκριση των διαφόρων εμπλεκόμενων γνωστικών περιοχών και σαφής στοχοθέτηση για κάθε μία ξεχωριστά.

Κεντρική ιδέα

Σύντομη περιγραφή της ιδέας που διέπει το σενάριο, των τεχνολογικών εργαλείων που προτείνονται προς χρήση και στις προβλεπόμενες διδακτικές μεθόδους

Στρατηγικές εφαρμογής

Αναφορά των διδακτικών μεθόδων που εμπλέκονται και απαιτούνται στα πλαίσια εφαρμογής του σεναρίου

Αναφορά του τρόπου εργασίας των μαθητών

Διδακτική διαδικασία

Αναλυτική ανάπτυξη της διδακτικής διαδικασίας που απαιτείται για την επίτευξη των παραπάνω προσδοκώμενων μαθησιακών στόχων.

- Περιγραφή των διδακτικών μεθόδων που εμπλέκονται και απαιτούνται στα πλαίσια εφαρμογής του σεναρίου.
- Ανάλυση του τρόπου εργασίας των μαθητών (μόνοι, σε ομάδες, μετωπική) και του βαθμού αυτενέργειας και αυτονομίας που παρέχεται.
- Αναφορά και επεξήγηση της τυχόν ανάγκης για αλλαγή του ρόλου των καθηγητών και των αντιλήψεων τους για την διδασκαλία του συγκεκριμένου γνωστικού αντικείμενου μέσα από το προτεινόμενο σενάριο.
- Πιθανή χρήση αξιολόγησης και αναλυτική περιγραφή της ποιότητας, είδους και στόχων της αξιολόγησης αυτής.

Δραστηριότητες

Περιγραφή των επιμέρους εκπαιδευτικών δραστηριοτήτων και των αντίστοιχων «πλάνων» εφαρμογής τους (σε ξεχωριστά υποκεφάλαια, ένα για κάθε δραστηριότητα).

Συγκεκριμένα:

- Στόχοι του Προγράμματος Σπουδών που θίγονται/πραγματεύονται από τη δραστηριότητα (μπορεί να αφορούν περισσότερα του ενός γνωστικά αντικείμενα).
- Περιγραφή της ροής της δραστηριότητας με διάκριση των διαδοχικών βημάτων εφαρμογής της διαδικασίας. Για καθένα απ' αυτά

περιγράφεται μεταξύ άλλων η αλληλεπίδραση μαθητών και διδασκόντων με τα εργαλεία λογισμικού.

- Πιθανές επιπτώσεις από την εφαρμογή της δραστηριότητας στη διεύθυνση / διοίκηση του σχολείου και σχετικά μέτρα που πρέπει να ληφθούν.
- Χρονισμός εφαρμογής της δραστηριότητας και εναρμονισμός της με το ωρολόγιο πρόγραμμα (εκτιμώμενη διάρκεια, συνεργασία με διδακτικές ώρες μαθημάτων, κλπ).
- Περιγραφή των σημείων που μένουν αμετάβλητα καθώς και εκείνων που δέχονται περαιτέρω αλλαγές, επιπροσθέσεις και γενικότερα είναι περισσότερο ευλύγιστα για μετατροπές στα χέρια των καθηγητών και μαθητών τους.
- Αναφορά σε συγκεκριμένους στόχους ή διαδικασίες που δεν μπορούν να επιτευχθούν μέσω της συγκεκριμένης μορφής και εφαρμογής του σεναρίου.

Προδιαγραφές σχεδιασμού δραστηριοτήτων

Συνδυάζοντας τα τρία προηγούμενα κεφάλαια παρήχθησαν οι προδιαγραφές σχεδιασμού δραστηριοτήτων σύμφωνα με τις σύγχρονες θεωρίες μάθησης για τη μάθηση βασικών εννοιών του προγραμματισμού σε συνδυασμό με την εικονική γλώσσα προγραμματισμού του Scratch.

Οι δραστηριότητες θα πρέπει να..
<ul style="list-style-type: none">• είναι ανοικτού τύπου, να δέχονται δηλαδή πολλαπλές επιλύσεις ώστε να επιτρέπουν την έκφραση των ατομικών διαφορών των μαθητών• αποτελούν προβλήματα των οποίων τα δεδομένα μπορούν εν μέρει ή στο σύνολο να καθορίζονται από το μαθητή
<ul style="list-style-type: none">• έχουν ενδιαφέρον για το μαθητή ώστε να του δημιουργούν κίνητρο (Cognitive knowledge)

<ul style="list-style-type: none">• ξεφεύγουν από τα συνήθη αριθμητικά ή αλφαριθμητικά παραδείγματα
<ul style="list-style-type: none">• επιτρέπουν στο μαθητή να έχει έλεγχο της μάθησής του• μπορούν να αναπαρασταθούν σε εικονικά συστήματα όπως η ζωγραφική ώστε να είναι δυνατή η εικονική ανατροφοδότηση και αυτοδιόρθωση του μαθητή
<ul style="list-style-type: none">• είναι προβλήματα δηλαδή να επιδέχονται επίλυση με απαίτηση ανάλυσης σύνθεσης και να απαιτούν για την επίλυσή τους τη σύνθεση εννοιών• αποτελούν προβλήματα με απαιτήσεις γνώσεων ολιστικού τύπου όπως η χρήση περισσότερων της μιας εντολών ή αλγοριθμικών δομών
<ul style="list-style-type: none">• δίνουν την ευκαιρία στο μαθητή για πειραματισμό με κάποιου είδους πρακτική-χειρωνακτική εμπειρία με πρωταρχικά δεδομένα• πραγματοποιούνται με χρήση εικονικών εργαλείων και σχεδίασης βασικών γεωμετρικών σχημάτων
<ul style="list-style-type: none">• επικεντρώνονται στην κατανόηση βασικών εννοιών του αντικειμένου μάθησης• επικεντρώνονται στην επίλυση προγραμματιστικού προβλήματος με χρήση της αλγοριθμικής λογικής και όχι στην εκμάθηση των συντακτικών κανόνων της γλώσσας
<ul style="list-style-type: none">• επικεντρώνονται στα σημεία που οι μαθητές δυσκολεύονται• δίνουν έμφαση στην κατανόηση των μεταβλητών, των βασικών αλγοριθμικών δομών

- μπορούν να προσφέρουν οπτική ανατροφοδότηση για να δίνουν ευκαιρίες αυτοδιόρθωσης του μαθητή
- επιτρέπουν μια αρχική επίλυση του προβλήματος με χρήση ενός εικονικού περιβάλλοντος και σύγκριση με τη γραφική έξοδο του προγράμματος για αυτοδιόρθωση

- επιτρέπουν στους μαθητές να εκφράσουν χωρίς ιδιαίτερη δυσκολία την πρότερη γνώση τους σχετικά με άλλα γνωστικά αντικείμενα
- επιτρέπουν τη χρήση στοιχειωδών εννοιών για την επίλυση προγραμματιστικού προβλήματος όπως π.χ. γεωμετρικές έννοιες

- απαιτούν από τους μαθητές στοιχειώδεις γνώσεις από άλλα γνωστικά αντικείμενα και έτσι να τους απαλλάσσουν από το επιπλέον νοητικό φορτίο του να εξοικειωθούν με κάποιο άλλο γνωστικό αντικείμενο δύσκολο σε αυτούς
- απαιτούν γνώση βασικών στοιχειωδών όπως π.χ. βασικών εννοιών γεωμετρικών σχημάτων και κατασκευών

Σενάριο «Η δομή ελέγχου ΕΠΙΛΟΓΗ»

ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Γνωστικό αντικείμενο	Πληροφορική - Γ' τάξη γυμνασίου
Τίτλος	Η δομή ελέγχου ΕΠΙΛΟΓΗ. <i>Εντάσσεται στο κεφάλαιο "Γνωρίζω τον υπολογιστή ως ενιαίο σύστημα" του αναλυτικού προγράμματος της Γ' γυμνασίου, και συγκεκριμένα στον στόχο "Ελέγχω και προγραμματίζω"</i>
Μαθησιακοί στόχοι	Η κατανόηση της δομής επιλογή, η συνθήκη στην οποία βασίζεται μία επιλογή και οι λογικές πράξεις ΚΑΙ-Η
Προσδοκώμενες παιδαγωγικές και μαθησιακές κατακτήσεις	<ul style="list-style-type: none"> • Η ανάγνωση και κατανόηση του κώδικα ενός προγράμματος • Η εύρεση και αντιμετώπιση απλών προβλημάτων στον κώδικα ενός προγράμματος • Ο σχεδιασμός, η υλοποίηση και η επεξεργασία προγραμματιστικών λύσεων • Η αξιολόγηση στρατηγικών επίλυσης ενός προβλήματος • Η μάθηση κάποιων σημαντικών στρατηγικών επίλυσης • Η συνειδητοποίηση της δύναμης και των ορίων του προγραμματισμού • Η συνεργασία σε ομάδες για την ανάλυση και επίλυση προβλημάτων • Η ανάπτυξη της κριτικής σκέψης και της δημιουργικότητας • Η ανάληψη πρωτοβουλιών και η ενεργή συμμετοχή στη διαδικασία της μάθησης
Εκτιμώμενη διάρκεια	4 διδακτικές ώρες <i>(Δύο μαθήματα των δύο διδακτικών ωρών)</i>

Στρατηγικές εφαρμογής	<ul style="list-style-type: none">• Διερευνητικό-αποκαλυπτικό μοντέλο• Ομαδο-συνεργατική διδασκαλία• Half-baked programs (Μισο-ψημένα προγράμματα, δηλαδή απόκρυψη λεπτομερειών από τον κώδικα προγράμματος)• Γραφική παρουσίαση με χρήση PowerPoint• Αξιολόγηση με διαδραστικό λογισμικό

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΙΑ

<p>Κεντρική ιδέα</p>	<p>Μάθημα 1^ο</p> <p>Οι μαθητές θα πειραματιστούν με ένα "παιχνίδι" στο Scratch. Θα εξερευνήσουν το περιβάλλον του μικρόκοσμου που θα χρησιμοποιηθεί και τους κανόνες που το διέπουν. Στην συνέχεια χρησιμοποιώντας την μέθοδο των "μισοψημένων" παιχνιδιών θα τους δοθεί ο κώδικας του παιχνιδιού με επεξηγήσεις και το πρόβλημα που πρέπει να λύσουν.</p> <p>Μάθημα 2^ο</p> <p>Η τάξη θα κρίνει και θα αξιολογήσει τις διαφορετικές λύσεις που δόθηκαν και θα βρουν τα μειονεκτήματα και τα πλεονεκτήματα κάθε μίας. Θα δοθεί ένα νέο πρόβλημα που θα επεκτείνει τις προηγούμενες λειτουργίες. Στην συνέχεια θα χρησιμοποιηθεί διαδραστικό λογισμικό αξιολόγησης.</p>
<p>Εμπλεκόμενοι ρόλοι</p>	<p>Οι μαθητές μιας τάξης της 3ης γυμνασίου και ο καθηγητής πληροφορικής.</p>
<p>Δραστηριότητες εμπλεκόμενων ρόλων</p>	<p>Οι μαθητές: αναλύουν / προβληματίζονται / εξηγούν / εφαρμόζουν / συνθέτουν / αξιολογούν</p> <p>Ο καθηγητής: ρωτά / απαντά / εξηγεί / σχολιάζει / συζητά / βοηθά / αξιολογεί</p>
<p>Προαπαιτούμενη γνώση</p>	<p>Οι μαθητές πρέπει σε προηγούμενα μαθήματα να έχουν κατανοήσει την έννοια και τη χρήση των μεταβλητών όπως και των εντολών εκχώρησης τιμής σε μεταβλητή.</p>

Τεχνολογικό υλικό και λογισμικό	Ένας υπολογιστής για κάθε δύο μαθητές με σύνδεση στο internet και εγκατεστημένο το λογισμικό Scratch Ένας υπολογιστής για τον καθηγητή με σύνδεση στο internet και εγκατεστημένο το λογισμικό Scratch και το λογισμικό εκμάθησης-εκγύμνασης Hot Potatoes Βιντεοπρωτόκολλο
---------------------------------	---

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ

Διδακτική διαδικασία

Μάθημα 1^ο

Οι μαθητές θα χωριστούν σε ομάδες των δύο ατόμων.

Θα χρησιμοποιηθεί αρχικά η μέθοδος του "μαύρου κουτιού" σύμφωνα με την οποία, οι μαθητές θα πειραματιστούν με ένα "παιχνίδι" γραμμένο στο Scratch χωρίς να έρθουν σε επαφή με τον κώδικά. Αυτό θα έχει ως σκοπό να εξερευνήσουν το περιβάλλον του μικρόκοσμου που θα χρησιμοποιηθεί και θα γίνει , επίσης, μια συζήτηση για να εκμαιευθούν οι κανόνες που διέπουν το συγκεκριμένο περιβάλλον. Στην συνέχεια χρησιμοποιώντας την μέθοδο των "μισοψημένων" παιχνιδιών θα τους δοθεί ο κώδικας του παιχνιδιού με επεξηγήσεις και το πρόβλημα που πρέπει να λύσουν. Η λύση του προβλήματος θα προέλθει από κοινή συζήτηση στην τάξη που στη συνέχεια θα υλοποιήσει κάθε μαθητής ξεχωριστά στον υπολογιστή του.

Μάθημα 2^ο

Στο δεύτερο μάθημα η τάξη θα κρίνει και θα αξιολογήσει τις διαφορετικές προσεγγίσεις και θα βρουν τα μειονεκτήματα και τα πλεονεκτήματα κάθε μίας. Στη συνέχεια θα δοθεί ένα νέο πρόβλημα που θα επεκτείνει τις προηγούμενες λειτουργίες, το οποίο θα πρέπει να λυθεί από κάθε ομάδα ξεχωριστά. Στη συνέχεια θα χρησιμοποιηθεί λογισμικό αξιολόγησης. Η αξιολόγηση θα στοχεύει στην ποιοτική αξιολόγηση των βασικών εννοιών που διδάχτηκαν.

Δραστηριότητες (Βήματα)

Μάθημα 1^ο

- Οι μαθητές χωρίζονται σε ομάδες των δύο ατόμων.
- Εκκινούν τον φυλλομετρητή ιστού, εισάγουν την διεύθυνση του προγράμματος [\(Γίνε πιλότος σε πύραυλο\)](#) που βρίσκεται στον δικτυακό τόπο του Scratch και χρησιμοποιούν το πρόγραμμα ακολουθώντας τις οδηγίες.

Ο πύραυλος "Πολύχρωμος εξερευνητής" έχει τα τελευταία συστήματα αυτόματου ελέγχου. Ο πιλότος του πρέπει να κάνει μόνο ένα πράγμα, να αποφεύγει τους κομήτες!

Άλλαξε την πορεία του πυραύλου όταν η απόσταση από τον κομήτη γίνει πολύ μικρή και χτυπήσει συναγερμός!

- Αφού γίνει η πρώτη συζήτηση για το περιβάλλον του προγράμματος και τους κανόνες που το διέπουν (Σε ποια απόσταση χτυπάει συναγερμός;), ανοίγουν την εφαρμογή του Scratch και φορτώνουν τον κώδικα του προγράμματος.
- Μετά τις απαραίτητες εξηγήσεις και τη σύνδεση κώδικα και αποτελέσματος, δίνεται το πρόβλημα και ακολουθεί συζήτηση πάνω στον τρόπο επίλυσης χωρίς τεχνικούς όρους.

Κάντε το διαστημόπλοιο να αλλάζει **αυτόματα** πορεία όταν βρίσκεται σε μικρή απόσταση από τον κομήτη.

- Παρουσιάζεται με τη χρήση του PowerPoint η δομή επιλογή μέσω γραφικής

απεικόνισης και γίνεται συζήτηση.

- Οι ομάδες υλοποιούν το ζητούμενο με συνεργασία των δύο μελών της ομάδας και το αποθηκεύουν στον κοινόχρηστο φάκελο του υπολογιστή τους

Μάθημα 2^ο

- Γίνεται συνοπτική παρουσίαση των διαφορετικών λύσεων που προτάθηκαν από τις ομάδες και η τάξη βρίσκει τα πλεονεκτήματα και τα μειονεκτήματα της κάθε μίας.
- Ο καθηγητής συνοψίζει τις καλές τεχνικές που χρησιμοποιήθηκαν έτσι ώστε όλοι να έχουν το ίδιο επίπεδο γνώσης.

Δίνεται ένα νέο πρόβλημα που αποτελεί επέκταση του προηγούμενου (εμφάνιση έκρηξης και τέλος παιχνιδιού σε περίπτωση σύγκρουσης) και ζητείται η υλοποίηση του.

Αν αποτύχει η αυτόματη αλλαγή πορείας του πυραύλου και επέλθει σύγκρουση τότε πρέπει το παιχνίδι να τελειώσει.

- Ακολουθεί αξιολόγηση μέσω διαδραστικού λογισμικού με ερωτήσεις πολλαπλών επιλογών, αντιστοίχισης και συμπλήρωσης κενών.

Προετοιμασία	<p>Μάθημα 1ο</p> <ul style="list-style-type: none">• Αναζήτηση - υλοποίηση προγραμμάτων για την αφόρμηση και την πρώτη άσκηση στο Scratch• Δημιουργία γραφικής παρουσίασης της δομής επιλογής στο PowerPoint <p>Μάθημα 2ο</p> <ul style="list-style-type: none">• Συγκέντρωση και εξέταση της λύσης κάθε ομάδας• Δημιουργία online ερωτηματολογίου αξιολόγησης στο Hot Potatoes
Προτάσεις για επέκταση / διαφοροποίηση	

Κεφάλαιο 7 – Επίλογος



Επισκόπηση του εργαλείου

Το Scratch, όπως φαίνεται και στο σχήμα, δεν είναι απλά ένα ακόμη εργαλείο που βασίζεται στην **διδακτική προσέγγιση της χρήσης μικροκόσμων**. Χωρίς να στερείται των πλεονεκτημάτων αυτής της κατηγορίας (εγγενής υποστήριξη του αντικειμενοστραφούς προγραμματισμού, απλό και διαισθητικό περιβάλλον, αυξημένο ενδιαφέρον απ' την μεριά του χρήστη, υποστήριξη πολυμέσων) επιτυγχάνει να συμπεριλάβει τα σημαντικότερα χαρακτηριστικά από τα **συστήματα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων** (χρήση γραφικών μονάδων κώδικα για την παραγωγή συντακτικά ορθού προγράμματος) και τα **περιβάλλοντα εναλλακτικών μορφών αναπαράστασης της σύνταξης εντολών** (εκτέλεση κώδικα βήμα-προς-βήμα).

Ο συνδυασμός των τριών διδακτικών προσεγγίσεων, σε συνάρτηση με το πλήρως εξελληνισμένο περιβάλλον του Scratch, την πολύ μεγάλη γκάμα δραστηριοτήτων που υποστηρίζει και τις μικρές απαιτήσεις σε πόρους, δημιουργεί ένα εργαλείο που θα μπορούσε να υποστηρίξει πλήρως την διδασκαλία της Πληροφορικής σύμφωνα με το ελληνικό Αναλυτικό Πρόγραμμα.

Αξιολόγηση του σεναρίου

Το σενάριο «**Η δομή ελέγχου ΕΠΙΛΟΓΗ**» που περιγράφηκε στο προηγούμενο κεφάλαιο έθεσε ως στόχο «την κατανόηση της δομής επιλογή, των συνθηκών στις οποίες βασίζεται μία επιλογή και τις λογικές πράξεις ΚΑΙ-Η».

Στο επίπεδο των **γενικών γνωστικών και διδακτικών δυσκολιών** στην διδασκαλία του προγραμματισμού και γενικότερα στην επίλυση προβλημάτων, οι μαθητές, μέσα από την δομή του σεναρίου και κυρίως την τεχνική των *Half-baked programs*, μαθαίνουν:

- § να διαβάζουν και να κατανοούν τον κώδικα ενός προγράμματος
- § να αντιμετωπίζουν απλα προβλήματα στον κώδικα ενός προγράμματος
- § να κάνουν αλλαγές σε κώδικα που προϋπάρχει
- § να σχεδιάζουν και να υλοποιούν προγραμματιστικές λύσεις
- § να αξιολογούν στρατηγικές επίλυσης ενός προβλήματος
- § να συνειδητοποιούν την δύναμη και τα ορία του προγραμματισμού

Ως προς τα **ειδικά γνωστικά και διδακτικά προβλήματα** που δημιουργούνται κατά τη διδασκαλία της δομής επιλογής, οι μαθητές ανακαλύπτουν μέσα από τον πειραματισμό (διερευνητικό-αποκαλυπτικό μοντέλο), τις αρχές που διέπουν την δομή της επιλογής και πιο συγκεκριμένα:

- § το ρόλο και τη χρησιμότητα των ενδογενών συνθηκών και των διαχωρισμό τους από τις εξωγενείς

§ την πολυπλοκότητα που δημιουργεί η αλληλουχία και το εμφώλιασμα δομών ελέγχου μέσα σε άλλες δομές (ελέγχου ή επανάληψης)

Επίσης, μαθαίνουν πως δομές ελέγχου, όπως η δομή επιλογή, διακόπτουν την γραμμική εξέλιξη της εκτέλεσής του προγράμματος σε σχέση με το κείμενο του προγράμματος.

Σε **παιδαγωγικό επίπεδο**, μέσω της χρήσης της ομαδο-συνεργατικής διδασκαλίας, της παρουσίασης των προτεινόμενων λύσεων και της συζήτησης που διεξάγεται μέσα στην τάξη, οι μαθητές:

§ ανάπτυξουν την κριτική σκέψη τους και τη δημιουργικότητά

§ μαθαίνουν να συνεργάζονται σε ομάδες για την ανάλυση και επίλυση προβλημάτων

§ ανάλαμβάνουν πρωτοβουλίες και έχουν ενεργή συμμετοχή στη διαδικασία της μάθησης

Το σενάριο, σε σχέση με τα ήδη γνωστά διδακτικά μέσα, **εισάγει καινοτομίες** όπως η χρήση ενός παιχνιδιού κατά την αφόρμηση και η εκτεταμένη χρήση πολυμέσων τόσο κατά την παρουσίαση του προβλήματος, όσο και στην προσπάθεια για την λύση του. Με αυτό το τρόπο διατηρείται ζωντανό το ενδιαφέρον των μαθητών καθ' όλη τη διάρκεια της διδασκαλίας.

Επεκτάσεις

Στα πλαίσια της διπλωματικής δημιουργήθηκε ένα πρωτότυπο σενάριο για την δομή ελέγχου ΕΠΙΛΟΓΗ. Για να αξιολογηθεί πλήρως χρειάζεται η πιλοτική εφαρμογή του σε ένα σχολείο δευτεροβάθμιας εκπαίδευσης και η αξιολόγηση των αποτελεσμάτων.

Στην συνέχεια, αφού αξιολογηθούν τα αποτελέσματα της πιλοτικής εφαρμογής του πρώτου σεναρίου, και αφού γίνει σαφής ο τρόπος που μπορεί να χρησιμοποιηθεί το εργαλείο στον ελληνική εκπαίδευση χρειάζεται να δημιουργηθούν σεναρία που να καλύπτουν κι άλλες διδακτικές ενότητες, μέχρι την πλήρη κάλυψη του Αναλυτικού Προγράμματος και την εφαρμογή του σε πιλοτικό επίπεδο.

Κατάλογος σχημάτων

Εικόνα 1 - Το περιβάλλον του ΔΕΛΥΣ (X-Compiler)	71
Εικόνα 2 - Το περιβάλλον του Dynalab	73
Εικόνα 3 - Διερμηνευτής της ΓΛΩΣΣΑΣ – Στιγμιότυπο 1	75
Εικόνα 4 - Διερμηνευτής της ΓΛΩΣΣΑΣ – Στιγμιότυπο 2	76
Εικόνα 5 - Το περιβάλλον του Lego Robotlab	78
Εικόνα 6 - Lego RCX	78
Εικόνα 7 - Το περιβάλλον προγραμματισμού LogoBlocks	79
Εικόνα 8 - Οι τύποι δεδομένων που περιλαμβάνει το BACCII	80
Εικόνα 9 - Ο κόσμος του Karel	86
Εικόνα 10 - Το περιβάλλον MicroWorlds	87
Εικόνα 11 - Το περιβάλλον Stagecast Creator	88
Εικόνα 12- Κεντρική οθόνη	104
Εικόνα 13 - Σκηνή	105
Εικόνα 14 - Λίστα από φιγούρες	106
Εικόνα 15 - Σκηνικό	106
Εικόνα 16 - Ένα απλό σενάριο	107
Εικόνα 17 - Το αποτέλεσμα του σεναρίου	107
Εικόνα 18 - Ενεργοποιημένο σενάριο	108
Εικόνα 19 – Εντολές	108
Εικόνα 20 - Μεταβλητές	108
Εικόνα 21 – Έλεγχοι	108
Εικόνα 22 – Ενδυμασίες	109
Εικόνα 23 - Ήχοι	110
Εικόνα 24 - Γραμμή μενού	111
Εικόνα 25 - Φόρμα διαμοιρασμού	111

Παράρτημα

Μετάφραση εγγράφων που σχετίζονται με το Scratch

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Βιβλιογραφία

Anzai, Y. and Uesato, Y. 1982. *Learning recursive procedures by middle school children*. Ann Harbour : MI, 1982. σσ. 100-102. Τόμ. Proceedings of the Fourth Annual Conference of the Cognitive Science Society.

Arsac, J. 1980. *Premières Leçons de Programmation*. Paris : Cedic/Nathan, 1980.

Bayman, P., Mayer, R. E. 1983. A Diagnosis of Beginning Programmers' Misconceptions of BASIC Programming Statements. *Communications of the ACM*. 1983, Τόμ. vol. 26, n° 9, σσ. 677-679.

Begel, A. 1996. *LogoBlocks: A Graphical Programming Language for*. s.l. : MIT, 1996.

Birch, M., Boroni, C., Goosey, F., Patton, S., Poole, D., Pratt, C. & Ross, R. 1995. DYNALAB: A Dynamic Computer Science Laboratory Infrastructure Featuring Program Animation. *ACM*. 1995, Τόμ. 27, 1, σσ. 29-33.

Bonar, J. and Soloway, E. 1985. Preprogramming Knowledge: a major source of misconceptions in novice programmers. *Human-Computer Interaction*. 1985, 1, σσ. 133-161.

Brown, M. και Hershberger, J. 1997. Program Auralization. [συγγρ. βιβλίου] J., Domingue, J., Brown, M. and Price, B., Eds. Stasko. *Software Visualization: Programming as a multimedia Experience*. Cambridge : MIT Press, 1997, σσ. 139-144.

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., and Miller, P. 1997. Mini-languages: A Way to Learn Programming Principles. *Education and Information Technologies*. 1997, Τόμ. 1, 2, σσ. 65-83.

Burnette T., Capeheart A.C., Conway M., Cosgrove D., DeLine R., Durbin J. 1995. *Alice: Rapid Prototyping System for Virtual Reality Randy Pausch*. s.l. : IEEE Computer Graphics and Applications, 1995.

Calloni B., Bagert D. 1994. Iconic programming in BACII vs. Textual programming: which is a better learning environment?. *SIGCSE '94* . ACM, 1994, 3/94, σσ. 188-192.

Crews, T. & Ziegler, U. 1998. The Flowchart Interpreter for Introductory Programming Courses. *Proceedings of FIE '98 Conference*. 1998, σσ. 307-312.

Dagdilelis, V. 1986. *Conceptions des eleves a propos des notions fondamentales de la programmation informatique en classe de Troisieme*. Grenoble : Universite Joseph FOURIER, 1986.

Dagdilelis, V., Balacheff, N. et Capponi, B. 1990. L'apprentissage de l'itération dans deux environnements informatiques, , , . *ASTER*. 1990, 11, σσ. 45-66.

Dagdilelis, V., Evangelidis, G., Satratzemi, M., Efopoulos, V. & Zagouras, C. 2003. DELYS: a novel microworld-based educational software for teaching computer science subjects. *Computers & Education*. Elsevier Publishers, 2003, 40, σσ. 307-325.

Delanoy, C. 1996. *Initiation à la programmation*. Paris : Eyrolles, 1996.

Du Boulay, B. 1986. Some Difficulties in Learning to Program. *Journal of Educational Computing Research*. 1986, 2, σσ. 57-73.

—. 1989. Some Difficulties of Learning to Program. [συγγρ. βιβλίου] E. Soloway and J. C. Spohrer. *Studying the Novice Programmer*. Hillsdale, NJ : Lawrence Erlbaum Associates, 1989, σσ. 283-299.

Dufoyer, J. P. 1988. *Informatique, éducation et psychologie de l'enfant*. Paris : PUF, 1988.

Goldschlager, L. and Lister, A. 1996. *Εισαγωγή στη σύγχρονη Επιστήμη των Υπολογιστών*. Αθήνα : Δίαυλος, 1996.

Hoc, M. 1989. *Do we really have conditional statements in our brains?* s.l. : Lawrence Erlbaum Associates, 1989. σσ. 179-190. Τόμ. Studying the Novice Programmer.

Horowitz, E. 1984. *Fundamental of programming languages*. s.l. : Science Press, 1984.

Kafai, Y., Maloney, J., Burd, L., Rusk, N., Silverman, B., and Resnick, M. 2004. *Scratch: A Sneak Preview. Second International Conference on Creating, Connecting, and Collaborating through Computing*. s.l. : Kyoto, Japan, 2004. σσ. 104-109.

Kessler, C.M. and Anderson, J.R. 1986. Learning Flow of Control: Recursive and Iterative Procedures. *Human Computer Interaction*. 1986, σσ. 135-166.

Knuth, D.E. 1968. *The Art of Computer Programming, Fundamental Algorithms*. s.l. : Addison-Wesley, 1968.

Komis, V. 1991. *La récursivité : une méthode de programmation*. s.l. : Département des Mathématiques, Université de Salonique, 1991. σσ. 127-137. Τόμ. Cahiers de Didactique des Mathématiques.

Laborde, C., Balacheff, N. et Mejias, B. 1985. *Genèse du concept d'itération : une approche expérimentale*. s.l. : Enfance, 1985. σσ. 223-239.

Lagrange, J.B. 1990. *Des situations connues aux traitements sur des données codifiées : une étude des difficultés d'élèves débutant en informatique, In Actes du deuxième colloque francophone sur la didactique de l'Informatique*. Paris : EPI, 1990. σσ. 55-79.

—. 1992. *Les objets et les types dans un enseignement de la programmation s'adressant aux débutants, In Actes de la troisième rencontre francophone de la Didactique de l'Informatique*. Sion : EPI, 1992. σσ. 115-120.

Mayer, R. 1981. Computing Surveys. *The Psychology of How Novices Learn Computer Programming*. 1981, Τόμ. Vol. 13, n. 1, σσ. 121-141.

McCracken, D. 1987. Ruminations on Computer Science Curricula. *Communications of the ACM*. ACM, 1987, 30.

Mendelsohn, P. 1985. *Learning recursive procedures through Logo programming*. s.l. : University of London, 1985. Τόμ. Proceedings of the second Logo and Mathematics Education Conference.

Mendelson, P., Green, T. R. G., and Brna, P. 1990. Programming languages in education: the search for an easy start. [συγγρ. βιβλίου] T. R. G. Green, D. Gilmore and R. Samurçay (eds) J.-M. Hoc. *In Psychology of programming*. London : Academic Press, 1990, σσ. 175-200.

Mindell, D., Beland, C., Wesley, C., Clarke, D., Park, R., Trupiano, M. 2000. *Lego mindstorms the structure of an engineering (r)evolution, 6.933J* . s.l. : 6.933J Structure of Engineering Revolutions, 2000.

Mukherjea, Stasko. 1993. *Applying algorithm animation techniques for program tracing, debugging, and understanding*. Georgia : s.n., 1993.

Pair, C. 1990. *Programming Languages and Programming Methods*. s.l. : Academic Press, 1990. σσ. 9-20. Τόμ. Psychology of Programming.

Pirolli, P. 1986. A Cognitive Model and Computer Tutor for Programming Recursion. *Human-Computer Interaction*. 1986, 2, σσ. 319-355.

Resnick, M., Kafai, Y., Maeda, J. 2003. *A Networked, Media-Rich Programming Environment to Enhance Technological Fluency at After-School Centers in Economically-Disadvantaged Communities*. s.l. : Proposal to National Science Foundation, 2003.

Rogalski, J. and Samurçay, R. 1990. *Acquisition of Programming Knowledge and Skills*. s.l. : Academic Press, 1990. σσ. 157-174. Τόμ. Psychology of Programming.

Rogalski, J. 1988. *Enseignement de méthodes de programmation dans l'initiation à l'informatique*. Paris : EPI, 1988. σσ. 61-74.

Rogalski, J. et Vergnaud, G. 1987. Didactique de l'Informatique et acquisitions cognitives en programmation. *Psychologie Française*. 1987, 32-34, σσ. 267-273.

—. 1987. *Didactique de l'Informatique et acquisitions cognitives en programmation*. s.l. : Psychologie Française (32-34), 1987. σσ. 267-273.

Rogalski, J. 1989. *Problem-Solving in Mathematics and in Informatics: Differencies and Invariants*. 1989. σσ. 236-247.

Rouchier, A. 1987. L'écriture et l'interprétation de procédures récursives en Logo. *Psychologie Française*. 1987, 32, σσ. 281-285.

Sajaniemi, J. 2005. Roles of Variables and Learning to Program. [συγγρ. βιβλίου] A. Jimoyiannis. *Proceedings of the 3rd Panhellenic Conference "Didactics of Informatics"*. Korinthos, Greece : University of Peloponnese, 2005.

Samurçay, R. 1987. *Modèles cognitifs dans l'acquisition des concepts informatiques*. Grenoble : La Pensée Sauvage, 1987. σσ. 215-223. Τόμ. Actes du premier colloque franco-allemand de didactique.

—. 1989. *The concept of variable in programming: Its meaning and use in problem-solving by novice programmers*, In E. Soloway and J. C. Spohrer (Eds), *Studying the Novice Programmer*. Hillsdale : NJ: Erlbaum, 1989. σσ. 161-178.

Scanlan, D. Sept. 1989. Structured flowcharts outperform pseudocode: an experimental comparison. *IEEE Software*. Sept. 1989, Τόμ. Vol 6, No 5, σσ. 28-36.

Schneiderman, B. 1980. *Software Psychology: Human Factors in Computer and Information Systems*. Cambridge : Winthrop Publishers Inc., 1980.

Soloway, E., Bonar, J. and Ehrlich, K. 1983. Cognitive Strategies and looping constructs: An empirical study. *Communication of the ACM*. 1983, Τόμ. 26, σσ. 853-860.

SpiNet - ΓλωσσοΜάθεια. <http://spinet.gr/glossomatheia/>. [Ηλεκτρονικό]
[Παραπομπή: 21 05 2009.]

Spoehrer, J. and Soloway, E. 1989. *Novice Mistakes: are the folk wisdoms correct?* s.l. : Lawrence Erlbaum Associates, 1989. σσ. 401-416. Τόμ. *Studying the Novice Programmer*.

Vickers, P. & Alty, J. 1999. *CAITLIN: Implementation of a Musical Program Auralisation System to Study the Effects on Debugging Tasks as Performed by Novice Pascal Programmers*. s.l. : Loughborough University, Dept. of Computer Science, 1999.

Αλεβίζος, Θ. και Καμπουρέλης, Α. 1988. *Μαθήματα Προγραμματισμού: εισαγωγή με τη γλώσσα Pascal*. Αθήνα : Θυμέλη, 1988.

Γεωργόπουλος, Άλκης. Διερμηνευτής της ΓΛΩΣΣΑΣ . www.alkisg.com.
[Ηλεκτρονικό] [Παραπομπή: 20 05 2009.]

Δικτυακός Τόπος Αλγοριθμικής - Έργο Πλειάδες/ Ενότητα Νηρηίδες:
«Αλγοριθμική-Προγραμματισμός».

<http://www.ecedu.upatras.gr/algorithmics/Skopos.htm>. [Ηλεκτρονικό]
[Παραπομπή: 20 05 2009.]

Εφόπουλος, Β. 2005. *Διαδικτυακό Περιβάλλον υποστηριζόμενο από Σύστημα Διαχείρισης Βάσεων Δεδομένων για την εισαγωγή στη διδασκαλία των αρχών του προγραμματισμού*. Θεσσαλονίκη : Πανεπιστήμιο Μακεδονίας, 2005.

Καβακλή, Β. 2001. *ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ ΚΑΙ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ*. s.l. : ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ, 2001.

Κόμης, Βασίλειος. 2001. *Διδακτική της Πληροφορικής*. Πάτρα: ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ, 2001. 1.

Σατρατζέμη, Μ., Χατζηαθανασίου, Κ. & Δαγδιλέλης, Β. 2000. AnimPascal: Ένα Εκπαιδευτικό Περιβάλλον για τη Στήριξη Εισαγωγικών Μαθημάτων Προγραμματισμού, – οι Τεχνολογίες της Πληροφορίας και της Επικο. *Πρακτικά 2ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή -οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση*. 2000, σσ. 125-135.

Τζιμογιάννης, Α. 2000. *Η διδασκαλία του Προγραμματισμού Η/Υ στη Δευτεροβάθμια Εκπαίδευση. Δυσκολίες και αντιλήψεις των μαθητών για την έννοια της μεταβλητής*,. s.l. : Η Βάση, 2000. σσ. 35-42.

Τζιμογιάννης, Α. και Γεωργίου, Β. 1999. *Οι δυσκολίες μαθητών δευτεροβάθμιας εκπαίδευσης στην εφαρμογή της δομής ελέγχου για την ανάπτυξη αλγορίθμων. Μία μελέτη περίπτωσης*. s.l.: Σύλλογος Καθηγητών Πληροφορικής Ηπείρου, 1999. σσ. 183-192. Τόμ. Τζιμογιάννης, Α. (επιμ.), Πρακτικά Πανελληνίου Συνεδρίου «Πληροφορική και Εκπαίδευση».

Τζιμογιάννης, Α. και Κόμης, Β. 1999. *Επίλυση προβλημάτων σε προγραμματιστικό περιβάλλον: η οικοδόμηση της δομής ελέγχου από τους μαθητές του Ενιαίου Λυκείου*. Ρέθυμνο: s.n., 1999. Τόμ. Πρακτικά 4ου Πανελληνίου Συνεδρίου "Διδακτική των Μαθηματικών & Πληροφορική στην Εκπαίδευση".

—. 2000. *Η έννοια της μεταβλητής στον Προγραμματισμό: δυσκολίες και παρανοήσεις μαθητών του Ενιαίου Λυκείου*. Πάτρα: s.n., 2000. σσ. 103-114. Τόμ. Κόμης, Β. (επιμ.), Πρακτικά 2ου Πανελληνίου Συνεδρίου "Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση".

Τσέλιος Ν., Κόμης Β. Δημιουργός Διαγραμμάτων Ροής.
<http://www.ecedu.upatras.gr/flowchart/index.htm>. [Ηλεκτρονικό] [Παραπομπή: 20 05 2009.]

Χριστοπούλου, Αικατερίνη. 2000. *Γιατί αποτυγχάνουν οι μαθητές στην επίλυση προβλημάτων σε προγραμματιστικό περιβάλλον.* Αθήνα : Πανεπιστήμιο Αθηνών, 2000.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ