



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Διπλωματική Εργασία

**“Σύγχρονα λογισμικά για τη βελτίωση της
διδασκαλίας προγραμματισμού στη δευτεροβάθμια
εκπαίδευση”**

Σπαής Φ. Αλέξανδρος

Επιβλέπουσα Καθηγήτρια: Χούστη Αικατερίνη

Βόλος, Φεβρουάριος 2014

Περιεχόμενα

Ευχαριστίες	3
Περίληψη	4
Ενότητα 1: Εισαγωγή.....	5
1.1 Νέες Τεχνολογίες.....	5
1.2 Ιστορική Αναδρομή των ΤΠΕ στην Εκπαίδευση.....	8
1.2.1 Η Πληροφορική στο Γυμνάσιο	10
1.2.2 Η Πληροφορική στο Λύκειο	12
1.3 Η Συμβολή των ΤΠΕ στην Εκπαίδευση.....	15
Ενότητα 2: Το μάθημα ΑΕΠΠ	18
2.1 Περιγραφή του μαθήματος	18
2.2 Στόχοι του μαθήματος.....	21
2.3 Δομή του μαθήματος ΑΕΠΠ	23
2.4 Διδασκαλία του μαθήματος και προβλήματα	35
2.4.1 Διδακτικές προσεγγίσεις.....	35
2.4.2 Προβλήματα που αντιμετωπίζουν οι μαθητές.....	38
Ενότητα 3: Διδακτικές προσεγγίσεις και εκπαιδευτικά λογισμικά.....	41
3.1 Η έννοια της μάθησης.....	41
3.2 Θεωρίες μάθησης.....	45
3.3 Η γλώσσα Logo και Logo like εκπαιδευτικά περιβάλλοντα	51
Ενότητα 4: Το εκπαιδευτικό λογισμικό Scratch.....	55
4.1 Ιστορικά στοιχεία και περιγραφή.....	55
4.2 Δομή του προγράμματος	58
4.3 Παιδαγωγικά οφέλη του περιβάλλοντος Scratch.....	63
Ενότητα 5 : Σχέδιο Μαθήματος	66
5.1 ΑΕΕΠ: Η σημασία του, Παρανοήσεις, Δυσκολίες	67
5.2 Επιλογή Εκπαιδευτικού Λογισμικού.....	70
5.3 Διδακτικό Πλάνο	73
Ενότητα 6: Συμπεράσματα.....	103
Ενότητα 7: Βιβλιογραφία.....	106

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην επιβλέπουσα καθηγήτριά μου κα. Χούστη Αικατερίνη, για τη βοήθεια, το χρόνο και τις πολύτιμες συμβουλές που μου παρείχε, μέχρι την ολοκλήρωση της εργασίας αυτής. Ευχαριστώ επίσης την συνεπιβλέπουσα, κα. Δασκαλοπούλου Ασπασία για την καθοδήγησή της.

Ένα μεγάλο ευχαριστώ στην κα. Τσαλαπάτα Χαρίκλεια η οποία επέβλεψε στενά την προσπάθεια αυτή μέχρι το τέλος της, για τις συμβουλές και το επιστημονικό υλικό που μου διέθεσε.

Αφιερώνω αυτή την προσπάθεια στην οικογένειά μου, για την αμέριστη συμπαράσταση που μου παρείχε όλα αυτά τα χρόνια, μέχρι την ολοκλήρωση των σπουδών μου.

Στην οικογένειά μου...

Περίληψη

Η τεχνολογική ευχέρεια θεωρείται μια ανωτέρου επιπέδου δεξιότητα, καθώς με τη χρήση τεχνολογικών εργαλείων ο χρήστης δεν καταναλώνει απλά περιεχόμενο, αλλά μπορεί και να το κατασκευάσει. Ένας τρόπος για παράγει κανείς έργο στον ηλεκτρονικό υπολογιστή είναι και ο προγραμματισμός.

Η παρούσα διπλωματική εργασία περιγράφει πως κατατάσσεται η διδασκαλία του Προγραμματισμού στη δευτεροβάθμια εκπαίδευση. Συγκεκριμένα, μελετάται και αναλύεται το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον της Γ' τάξης Γενικού Λυκείου Τεχνολογικής Κατεύθυνσης, καθώς και τα λάθη και παρανοήσεις που παρατηρούνται στους μαθητές.

Στη συνέχεια, περιγράφεται το εκπαιδευτικό λογισμικό Scratch, τα οφέλη του στην εκπαίδευση, αλλά και πως μπορεί να συμβάλλει στη διδασκαλία του προγραμματισμού.

Τέλος, παρατίθεται διδακτικό πλάνο, στο οποίο παρουσιάζονται βασικές έννοιες του προγραμματισμού όπως μεταβλητή, δομή επιλογής, δομή επανάληψης, υποπρογράμματα και αναδρομή. Αναλύονται δυσνόητα σημεία της θεωρίας και προτείνεται μία σειρά από δραστηριότητες, όπως έτοιμες ασκήσεις και ασκήσεις προς επίλυση με τη χρήση του λογισμικού, με σκοπό την εμπλοκή των μαθητών σε διαδικασία πειραματισμού.

Λέξεις Κλειδιά : Προγραμματισμός, διδακτική, ΑΕΙΠΠ, Scratch.

Ενότητα 1: Εισαγωγή

1.1 Νέες Τεχνολογίες

Η σημερινή εποχή χαρακτηρίζεται από την έκρηξη της πληροφορίας. Ο όρος data (δεδομένα) θεωρείται συνώνυμο της πληροφορίας, ενώ ο τρόπος που αξιοποιούμε τα δεδομένα γίνεται με σχετική ευκολία χάρη στα νέα συστήματα επικοινωνίας.

Ο ηλεκτρονικός υπολογιστής έχει εισχωρήσει στην καθημερινότητά μας, με αποτέλεσμα να επηρεάζει τις κοινωνικές και επαγγελματικές μας σχέσεις, την ποιότητα ζωής μας καθώς και τη διαμόρφωση του χαρακτήρα μας. Συμβάλλει με καθοριστικό τρόπο στην εξέλιξη της επιστήμης, της εκπαιδευτικής διαδικασίας αλλά και κάθε άλλης παραγωγικής δραστηριότητας. Παρατηρούμε πως σε κάθε καθημερινή δραστηριότητα, κοινωνική και επαγγελματική, η χρήση Η/Υ γίνεται κατά κάποιο τρόπο αναγκαία. Χαρακτηριστικά παραδείγματα είναι τα “έξυπνα” κινητά τηλέφωνα (smartphones) και οι φορητοί υπολογιστές, που μπορούν να παρέχουν υπηρεσίες και πληροφορίες ανά πάσα στιγμή, με μικρό κόστος και σε ελάχιστο χρόνο.

Η τεχνολογία εξελίσσεται με ταχύτατους ρυθμούς και η εξέλιξη αυτή προβλέπεται αύξουσα και για τα επόμενα χρόνια. Ήδη παρατηρείται πως τα τεχνολογικά επιτεύγματα των τελευταίων ετών είναι πολύ περισσότερα και πιο σημαντικά από αυτά ολόκληρου του περασμένου αιώνα. Το αποτέλεσμα της εξέλιξης αυτής είναι να απαξιώνεται η γνώση. Καθίσταται επομένως αναγκαία η ικανότητα να μαθαίνουμε και να προσαρμοζόμαστε γρήγορα στις αλλαγές αυτές, καθώς θεωρείται τόσο σημαντική όσο το

διάβασμα πριν αρκετές δεκαετίες. Η ευχέρεια στη χρήση των νέων τεχνολογιών είναι προαπαιτούμενο για την εξάσκηση οποιουδήποτε επαγγέλματος, τη δια βίου μάθηση καθώς και την ουσιαστική συμμετοχή στην κοινωνία. Ο παραπάνω ισχυρισμός συνίσταται στη χρήση τεχνολογικών εργαλείων ώστε να δημιουργούνται σημαντικά αντικείμενα, αφού οι χρήστες πρέπει να είναι ικανοί όχι μόνο να καταναλώνουν περιεχόμενο, αλλά και να κατασκευάζουν (Resnick 2009).

Από τις νέες τεχνολογίες δεν θα μπορούσε να απέχει και η εκπαιδευτική διαδικασία. Η μεγάλη ανάπτυξη και ο ρόλος των **Τεχνολογιών Πληροφοριών και Επικοινωνιών (ΤΠΕ)**, τις έχουν καταστήσει συνιστώσα κάθε σύγχρονου προγράμματος σπουδών. Είναι εξάλλου πολύ σημαντικό οι μαθητές να μπορούν να εξοικειωθούν με τις ΤΠΕ και να κατανοούν βασικές έννοιες της πληροφορικής αλλά και να αναπτύσσουν δεξιότητες (τεχνολογικός αλφαριθμητισμός). Στο πλαίσιο ένταξης των ΤΠΕ στο σχολικό σύστημα διακρίνονται οι εξής προσεγγίσεις (Ν. Κόμης) :

- Η πληροφορική και οι νέες τεχνολογίες ως **αυτόνομο γνωστικό αντικείμενο**
- Η πληροφορική ως εργαλείο **γνώσης, έρευνας και μάθησης**, που χρησιμοποιείται σε άλλα γνωστικά αντικείμενα
- Η πληροφορική ως **κοινωνικό φαινόμενο**, που πρέπει να μελετηθεί

Ένας σημαντικός τομέας των ΤΠΕ θεωρείται και ο προγραμματισμός. Είναι ένας βασικός τρόπος με τον οποίο μπορεί κάποιος να κατασκευάσει

μια ατομική δημιουργία στον υπολογιστή. Δηλαδή, μια δεξιότητα που δίνει κίνητρα για ένα δομημένο τρόπο σκέψης, ο οποίος μπορεί να εφαρμοστεί σε όλα τα αντικείμενα (Papert , 1980). Από παιδαγωγικής άποψης, εφοδιάζει τον μαθητή με ένα δομημένο τρόπο σκέψης, εμπειρία σε πολύπλοκα συστήματα και παρέχει δεξιότητες που μπορούν να εφαρμοστούν σε ένα ευρύ φάσμα επιστημονικών κλάδων (Wing, 2006). Το ενδιαφέρον του προγραμματισμού έγκειται κυρίως στην ανάλυση ενός προβλήματος, στον τρόπο διδασκαλίας της αλγοριθμικής λογικής και στην επίλυσή του.

Τέλος , σύμφωνα με τις σύγχρονες εποικοδομητικές θεωρίες μάθησης (Piaget 1952, Bruner 1960, Jonassen 1999) είναι σημαντική η ενεργή συμμετοχή των μαθητών στην εκπαιδευτική διαδικασία. Οι μαθητές πρέπει να είναι ενεργοί και όχι παθητικοί δέκτες στη διαδικασία αυτή, ενώ η οικοδόμηση των γνώσεών τους, θα πρέπει να προέρχεται από ενέργειες και δραστηριότητες ενδιαφέρουσες σε αυτούς. Ένας τρόπος να επιτευχθεί αυτό είναι να υπάρξει ένας τρόπος “σύνδεσης” της καθημερινότητάς τους και της μαθησιακής διαδικασίας, της άτυπης και της τυπικής μάθησης.

1.2 Ιστορική Αναδρομή των ΤΠΕ στην Εκπαίδευση

Η πληροφορική είναι μια σχετικά νέα επιστήμη, που εδώ και μερικές δεκαετίες συνιστά έναν πανεπιστημιακό κλάδο (ΗΠΑ, Αγγλία, Γαλλία κλπ). Στην Ελληνική τριτοβάθμια εκπαίδευση ξεκίνησε τη δεκαετία του 80, ενώ θεωρείται μια ιδιαίτερα ελκυστική προοπτική για πολλούς μαθητές.

Η διδασκαλία της πληροφορικής σαν μάθημα δευτεροβάθμιας εκπαίδευσης μπορεί να διακριθεί σε δύο φάσεις-προσεγγίσεις.

Η πρώτη φάση ξεκινά με την εισαγωγή μαθημάτων στα Τεχνικά Επαγγελματικά Λύκεια (ΤΕΛ) το 1985, στα Ενιαία Πολυκλαδικά Λύκεια (ΕΠΛ) το 1986 και στα Γυμνάσια το 1992. Στις δύο πρώτες περιπτώσεις δίνεται έμφαση σε δραστηριότητες προγραμματισμού, όπως αλγοριθμική, μοντελοποίηση, οργάνωση δεδομένων και η διδασκαλία περιλαμβάνει την εκμάθηση γλωσσών προγραμματισμού (Basic, Pascal, Cobol). Στο Γυμνάσιο (από το 1992) γίνεται σταδιακά η εισαγωγή της Πληροφορικής ως αυτόνομο γνωστικό αντικείμενο, στα πλαίσια του μαθήματος Πληροφορική-Τεχνολογία.

Η προσέγγιση αυτή έχει χαρακτηριστεί ως **τεχνοκεντρική**, τόσο ως προς το σχεδιασμό όσο και ως προς τη διδασκαλία. Οι λόγοι που επαληθεύουν την παραπάνω θεώρηση βασίζονται στο γεγονός ότι αφενός τα Προγράμματα Σπουδών δεν είχαν καθορίσει σαφείς διδακτικούς στόχους, με αποτέλεσμα η Πληροφορική να είναι ασύνδετη με άλλα γνωστικά αντικείμενα, αφετέρου η διδασκαλία ήταν επικεντρωμένη στην εξοικείωση των μαθητών με τη χρήση υπολογιστών και την εκμάθηση γλωσσών προγραμματισμού και πακέτων χρήσης.

Η δεύτερη φάση ουσιαστικά ορίζεται το 1998, με την καθιέρωση του Ενιαίου Πλαισίου Προγραμμάτων Σπουδών Πληροφορικής (ΕΠΠΣΠ, ΥΠΕΠΘ 1998). Γίνεται η επέκταση μαθήματος πληροφορικής στο Ενιαίο Λύκειο, στα πλαίσια μιας προσέγγισης που θεωρείται *πραγματολογική* (Κόμης, Μικρόπουλος 2001). Η Πληροφορική εξακολουθεί να είναι αυτόνομο γνωστικό αντικείμενο, ενώ σχεδιάζεται η σταδιακή εφαρμογή των ΤΠΕ σε όλο το φάσμα του Προγράμματος Σπουδών ως μέσο διδασκαλίας, έρευνας και μάθησης. Στόχος της παραπάνω προσέγγισης είναι η προετοιμασία των μαθητών, ώστε να μπορέσουν να αξιοποιήσουν σωστά τις γνώσεις και την εμπειρία τους στις ΤΠΕ, τόσο στις σπουδές τους όσο και στη μετέπειτα επαγγελματική τους πορεία.

Τέλος, από το 2003, η Πληροφορική ξεκίνησε να διδάσκεται και στα Δημοτικά Ολοήμερα σχολεία, ενώ από το σχολικό έτος 2010-11 ενσωματώθηκε πιλοτικά και στο πρωινό τους πρόγραμμα.

1.2.1 Η Πληροφορική στο Γυμνάσιο

Στα πλαίσια του ΕΠΠΣΠ εντάσσεται στο γυμνάσιο το μάθημα της Πληροφορικής. Σύμφωνα με αυτό, οι μαθητές θα πρέπει να αποκτήσουν δεξιότητες χειρισμού ενός Η/Υ, να μπορούν να κατανοούν βασικές έννοιες και όρους με σκοπό να αποκτήσουν κριτική ικανότητα επεξεργασίας πληροφοριών και δεξιότητες μεθοδολογικού χαρακτήρα.

Η κατηγοριοποίηση των στόχων είναι :

- Γνώση
- Μεθοδολογία
- Συνεργασία
- Επικοινωνία
- Νέες Τεχνολογίες στην καθημερινότητα

Σύμφωνα με το ΕΠΠΣΠ οι στόχοι με τη διδασκαλία του μαθήματος στο γυμνάσιο είναι:

- “Οι μαθητές να αποκτήσουν την ικανότητα να αναλύουν βασικές έννοιες και όρους της Πληροφορικής, της Δικτυακής Τεχνολογίας και των Πολυμέσων”
- “Να γνωρίζουν τη λειτουργία των κύριων μονάδων ενός Η/Υ και να μπορούν να τον χρησιμοποιούν με ευχέρεια”
- “Να αποκτήσουν δεξιότητες συλλογής, επιλογής, ανάλυσης και αξιολόγησης πληροφοριών από διάφορες πηγές (π.χ. διαδίκτυο) και να τις αξιοποιούν για ατομικές ή ομαδικές εργασίες”
- “Να συνεργασθούν για την εκτέλεση συγκεκριμένης εργασίας, να αναπτύξουν πρωτοβουλίες, να αναγνωρίσουν τη συμβολή

της ομαδικής εργασίας στην παραγωγή έργου, να είναι σε θέση να κρίνουν τόσο τις δικές τους όσο και τις δημιουργίες των συμμαθητών τους”

- “Να αναπτύξουν κριτικές δεξιότητες για την αντιμετώπιση προβλημάτων με χρήση Η/Υ και να επιλύσουν απλά προβλήματα σε προγραμματιστικό περιβάλλον”
- “Να αξιοποιήσουν τις δυνατότητες που προσφέρουν οι ΤΠΕ για επικοινωνία, να ανταλλάξουν απόψεις, να προβληματισθούν, να διασκεδάσουν και να εφαρμόσουν απλές γνώσεις των ΤΠΕ στην καθημερινή τους ζωή”.

1.2.2 Η Πληροφορική στο Λύκειο

Στο Ενιαίο (πλέον Γενικό) Λύκειο διδάσκονται μαθήματα επιλογής για όλες τις τάξεις, αλλά και κατεύθυνσης για την Γ' τάξη.

Στα μαθήματα επιλογής ανήκουν:

Εφαρμογές Πληροφορικής (Α' τάξη) και Εφαρμογές Υπολογιστών (Β' ή Γ' τάξη).

Διδακτικοί Στόχοι :

- “Να επεκτείνουν οι μαθητές τη γενική πληροφορική παιδεία τους με έμφαση την ανάπτυξη δεξιοτήτων και ικανοτήτων στη χρήση και αξιοποίηση των υπολογιστικών και δικτυακών τεχνολογιών ως εργαλείων μάθησης και σκέψης”
- “Να ενημερωθούν για τις εφαρμογές της πληροφορικής στο σύγχρονο κόσμο, για τις δυνατότητες που προσφέρει και τις προοπτικές που δημιουργεί”
- “Να ευαισθητοποιηθούν, να προβληματισθούν και να αναπτύξουν κριτική ικανότητα στα κοινωνικά, ηθικά και πολιτισμικά ζητήματα που τίθενται με την είσοδο των υπολογιστικών τεχνολογιών στην ανθρώπινη δραστηριότητα”

Πολυμέσα-Δίκτυα (Γ' τάξη)

Διδακτικοί Στόχοι:

- “Να είναι σε θέση οι μαθητές να γνωρίζουν και να κατανοούν τα στάδια σχεδίασης, υλοποίησης και αξιολόγησης μιας εφαρμογής πολυμέσων”

- “Να αποκτήσουν την απαραίτητη γνώση και τεχνογνωσία σε θέματα δικτύων υπολογιστών και του τρόπου εφαρμογής τους σε πρακτικό επίπεδο”

Εφαρμογές Λογισμικού (Γ΄τάξη)

Διδακτικοί Στόχοι:

- “Να αναπτύξουν οι μαθητές δεξιότητες και τεχνικές επίλυσης προβλημάτων, καθώς και να προάγουν την αναλυτική και συνθετική τους σκέψη”
- “Να καλλιεργήσουν διαχρονικές δεξιότητες στη χρήση λογισμικού”
- “Να διευκολυνθεί η ανάπτυξη της ικανότητας να δημιουργούν”
- “Να ενισχυθεί ο συμμετοχικός χαρακτήρας μάθησης”

Τεχνολογία Υπολογιστικών Συστημάτων και Λειτουργικά Συστήματα (Γ΄τάξη)

Διδακτικοί Στόχοι:

- “Να αποκτήσουν οι μαθητές σωστές βάσεις, γνώσεις και εμπειρία στην αρχιτεκτονική, την οργάνωση και τη λειτουργία ενός υπολογιστικού συστήματος”
- “Να κατανοήσουν την έννοια και το ρόλο ενός λειτουργικού συστήματος”

Τέλος, στην Τεχνολογική Κατεύθυνση της Γ΄τάξης, προσφέρεται το μάθημα **Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον**.

Είναι μάθημα πανελλαδικώς εξεταζόμενο και οι διδακτικοί του στόχοι είναι:

- “Να αποκτήσουν οι μαθητές αναλυτική και συνθετική σκέψη καθώς και ικανότητες μεθοδολογικού χαρακτήρα”
- “Να μπορούν να επιλύουν απλά προβλήματα σε προγραμματιστικό περιβάλλον”
- “Να αναπτύξουν δεξιότητες αλγοριθμικής προσέγγισης, δημιουργικότητα, φαντασία, αναλυτικό πνεύμα και αυστηρότητα στην έκφραση ”
- “Να μπορούν να διακρίνουν ποια προβλήματα επιλύονται σε προγραμματιστικό περιβάλλον”

Συνολικά, κρίνοντας από τα παραπάνω, συμπεραίνουμε ότι γίνεται μια προσπάθεια για μετάβαση στην κοινωνία της γνώσης. Για την επίτευξη αυτού του στόχου θα χρειαστεί να υπάρξει Τεχνολογικός Αλφαριθμητισμός, καθώς η εξοικείωση των μαθητών με τις έννοιες της Πληροφορικής, τις νέες τεχνολογίες και η ανάπτυξη σχετικών δεξιοτήτων είναι εφάμιλλης σημασίας με το να γνωρίζει κάποιος ανάγνωση και γραφή(Unesco,2000).

1.3 Η Συμβολή των ΤΠΕ στην Εκπαίδευση

Σύμφωνα με τα νέα Προγράμματα Σπουδών, παρατηρείται μια όλο και αυξανόμενη εισαγωγή των ΤΠΕ στην εκπαιδευτική διαδικασία. Πέρα από τα μαθήματα σχετικά με το αντικείμενο της πληροφορικής, ο Η/Υ και τα εκπαιδευτικά λογισμικά αποτελούν συμπλήρωμα της διδασκαλίας και για άλλα μαθήματα, όπως τα μαθηματικά, η φυσική, η γεωγραφία κ.α.

Βάσει ερευνητικών μελετών, οι υπολογιστές παρουσιάζουν πολλές εκπαιδευτικές δυνατότητες. Η μάθηση αποκτά πλέον ενδιαφέρον, καθώς πληροφορίες και γεγονότα παρουσιάζονται με ποικίλους τρόπους (video, εικόνες, ήχοι) και οι μαθητές συμμετέχουν ενεργά σε αυτή τη διαδικασία. Βοηθούν στην επανατροφοδότηση και δημιουργούν συνθήκες συνεργατικής μάθησης. Τέλος, συνδέουν τη μαθησιακή διαδικασία με την καθημερινότητα των μαθητών, με αποτέλεσμα να γίνεται πιο ενδιαφέρουσα. Μέσα από αυτή την αλληλεπίδραση χρήστη-συστήματος, ο υπολογιστής γίνεται ένα ιδιαίτερα ενδιαφέρον και ελκυστικό “παιχνίδι”, καθώς και χρήσιμο εργαλείο.

Στο πλαίσιο εισαγωγής των ΤΠΕ στην εκπαίδευση, ενδιαφέρον έχει αποτελέσει η διδασκαλία της πληροφορικής και ιδιαίτερα του προγραμματισμού Η/Υ. Παιδαγωγικά θεωρείται μια σημαντική και ωφέλιμη ικανότητα, διότι βοηθά στην καλλιέργεια ανώτερων μορφών σκέψης (Papert, 1991).

Η σημασία και η συνεισφορά του προγραμματισμού Η/Υ στην ανάπτυξη ενός δομημένου τρόπου σκέψης, έχουν τεθεί για πρώτη φορά από τον Papert (1980). Με τον προγραμματισμό διδάσκονται βασικές έννοιες που έχουν εφαρμογή και σε άλλες επιστήμες, όπως τα

Μαθηματικά, η Λογική και η Φυσική (Papert 1980, Howe et al. 1989). Γίνεται χρήση θεμελιωδών εννοιών όπως **μεταβλητή**, **δομές επιλογής**, **δομές επανάληψης** με σκοπό την επίλυση ενός προβλήματος με τη χρήση προγραμματιστικών εργαλείων. Από διδακτική άποψη, το ενδιαφέρον επικεντρώνεται στον τρόπο που σκέφτονται και δρουν οι προγραμματιστές, ανεξάρτητα από το επίπεδο εμπειρίας, και στις μεθόδους που ακολουθούν για τη σύλληψη, σχεδιασμό και υλοποίηση προγραμμάτων.

Το υπολογιστικό σύστημα σε συνδυασμό με το περιβάλλον προγραμματισμού, αποτελούν έναν “μηχανισμό” για τη δημιουργία προγραμμάτων. Ο μαθητής έρχεται σε άμεση επαφή με τη μηχανή, **ενοιολογική μηχανή** (national machine) όπως χαρακτηριστικά αναφέρει ο Du Boulay, δημιουργεί, ανακαλύπτει και διαπιστώνει αποτελέσματα και σφάλματα. Ο υπολογιστής παίζει σημαντικό ρόλο στη διδασκαλία του προγραμματισμού, καθώς δεν αποτελεί απλά μια μηχανή που εκτελεί ένα πρόγραμμα. Δίνει τη δυνατότητα στο μαθητή να συντάξει, να ελέγξει και να διορθώσει τη δημιουργία του. Παρέχει τη δυνατότητα για εξάσκηση και πειραματισμό, ενώ αποτελεί και μέσο συνεργασίας και επικοινωνίας, πράγμα που τον καθιστά παιδαγωγικά ωφέλιμο.

Ωστόσο, παρατηρούνται και ορισμένες δυσκολίες σχετικά με τη διδασκαλία του προγραμματισμού. Όσο ωφέλιμη ικανότητα και αν θεωρείται, μελέτες έχουν δείξει ότι για την πλειονότητα των μαθητών αποτελεί μια δύσκολη και ελάχιστα ελκυστική δραστηριότητα. Το ενδιαφέρον των μαθητών εστιάζεται κυρίως στη χρήση των Η/Υ για παιχνίδια και πλοήγηση στο διαδίκτυο, κάνοντας έτσι δύσκολη τη διδασκαλία του προγραμματισμού από τους εκπαιδευτικούς.

Οι δυσκολίες, παρανοήσεις και η αντιμετώπιση των προβλημάτων στη διδασκαλία του προγραμματισμού θα αναλυθούν στις επόμενες ενότητες.

Ενότητα 2: Το μάθημα ΑΕΠΠ

2.1 Περιγραφή του μαθήματος

Στα πλαίσια του **Ενιαίου Πλαισίου Προγραμμάτων Σπουδών Πληροφορικής** (ΕΠΠΣΠ, 1998), γίνεται η εισαγωγή μαθημάτων πληροφορικής στο Ενιαίο Λύκειο. Σύμφωνα με την προσέγγιση που ακολουθήθηκε, η Πληροφορική αντιμετωπίζεται ως αυτόνομο γνωστικό αντικείμενο, ιδιαίτερης σημασίας, καθώς παρέχει στους μαθητές επαρκείς γνώσεις και δεξιότητες στον τομέα της πληροφορικής και τους προετοιμάζει κατάλληλα τόσο στην πορεία τους στο Λύκειο, όσο και μετά από αυτό. Η διδασκαλία της Πληροφορικής αποτελεί αντικείμενο γενικής παιδείας που απευθύνεται σε όλους τους μαθητές και έχει σαν σκοπό :

- Να κατανοήσουν οι μαθητές βασικές έννοιες της Πληροφορικής
- Να αποκτήσουν δεξιότητες μεθοδολογικού χαρακτήρα
- Να αποκτήσουν ευρύτερη παιδεία και κουλτούρα γύρω από την Πληροφορική.

Βάσει του ΕΠΠΣΠ (1998), ξεκινά να διδάσκεται το μάθημα **Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (ΑΕΠΠ)**, κατά το σχολικό έτος 1999-2000. Ανήκει στον κλάδο **Πληροφορικής και Υπηρεσιών της Τεχνολογικής Κατεύθυνσης Γ΄τάξης** Ενιαίου Λυκείου και διδάσκεται επίσης στην Δ΄τάξη Εσπερινού Λυκείου. Χαρακτηρίζεται ως εργαστηριακό μάθημα, διδάσκεται 2 ώρες την εβδομάδα και εξετάζεται σε Πανελλαδικό Επίπεδο, για την εισαγωγή σε ΑΕΙ και ΤΕΙ. Οι γραπτές

εξετάσεις περιλαμβάνουν συνολικά τέσσερα θέματα κλιμακούμενης δυσκολίας, ένα θέμα θεωρίας και τρία θέματα ασκήσεων ή προβλημάτων σχετικά με το περιεχόμενο του μαθήματος. Η συνολική βαθμολογία προκύπτει κατά 40% από το θέμα της θεωρίας και κατά 60% (ισοδύναμα) από τα θέματα των ασκήσεων.

Ο γενικός σκοπός του μαθήματος είναι “*οι μαθητές να αναπτύξουν αναλυτική και συνθετική σκέψη, να αποκτήσουν ικανότητες μεθοδολογικού χαρακτήρα και να μπορούν να επιλύουν απλά προβλήματα σε προγραμματιστικό περιβάλλον*”. Δίνεται κυρίως έμφαση στις ενότητες **Ανάλυση Προβλήματος** και **Σχεδίαση Αλγορίθμου** ώστε οι μαθητές “*να αναπτύξουν δεξιότητες αλγοριθμικής προσέγγισης, δημιουργικότητα, φαντασία, αναλυτικό πνεύμα και αυστηρότητα στην έκφραση και να μπορούν να διακρίνουν ποια προβλήματα αντιμετωπίζονται σε προγραμματιστικό περιβάλλον*” (ΑΠΣ, ΑΕΠΠ, 1998).

Αρχικά το μάθημα διέθετε δύο διδακτικά πακέτα. Το κάθε πακέτο περιελάμβανε βιβλίο και τετράδιο μαθητή καθώς και βιβλίο καθηγητή και ήταν στην ευχέρεια κάθε εκπαιδευτικού να επιλέξει το διδακτικό πακέτο που επιθυμούσε. Ωστόσο, επειδή τα δύο διδακτικά πακέτα διέφεραν αρκετά τόσο σε φιλοσοφία όσο και σε προσέγγιση, διατηρήθηκαν κατά τα δύο πρώτα σχολικά έτη. Από το 2001 υπάρχει στη διάθεση το διδακτικό πακέτο των Βακάλη Α., Γιαννόπουλος Η., Ιωαννίδης Ν., Κοίλιας Χ., Μάλαμας Κ., Μανολόπουλος Ι., Πολίτης Π.

Στο παραπάνω διδακτικό πακέτο, στα κεφάλαια 2 έως 5, παρουσιάζονται τρόποι επίλυσης προβλημάτων με χρήση διαγραμμάτων ροής και ψευδογλώσσας, ενώ στα κεφάλαια 7 έως 14, η ψευδογλώσσα

κωδικοποιείται με τη χρήση της ΓΛΩΣΣΑΣ. Δίνεται επιπλέον η επιλογή στους μαθητές να διατυπώνουν τις λύσεις των ασκήσεων τους σε οποιαδήποτε μορφή παράστασης αλγορίθμου (π.χ. ψευδογλώσσα) ή ΓΛΩΣΣΑ.

2.2 Στόχοι του μαθήματος

Σύμφωνα με τη συγγραφική ομάδα του διδακτικού πακέτου, στο βιβλίο του καθηγητή προσδιορίζονται οι στόχοι του μαθήματος.

“Ο γενικός σκοπός του μαθήματος Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον της Γ τάξης του κύκλου Πληροφορικής και Υπηρεσιών της Τεχνολογικής Κατεύθυνσης του Ενιαίου Λυκείου, δεν είναι αυτός στον οποίο θα κατέληγε κάποιος διαβάζοντας απλά και μόνο τον τίτλο του μαθήματος. Το μάθημα έχει σαν πρωταρχικό στόχο την ανάπτυξη δεξιοτήτων και ικανοτήτων σχετικών με την αλγοριθμική και την ορθολογική χρήση της στην καθημερινή ζωή. Πολλές βασικές έννοιες αλγοριθμικής, αλλά και προγραμματισμού (π.χ. συνθήκες ελέγχου, λογικές προτάσεις και συμπεράσματα, κ.α.), συνιστούν αναπόσπαστο τμήμα των γενικών γνώσεων και δεξιοτήτων που πρέπει να αποκτήσει ο μαθητής, οι οποίες δεξιότητες και γνώσεις -στην πλειονότητά τους- δεν προσεγγίζονται από άλλα μαθήματα. Το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον δεν έχει σαν στόχο τη διδασχή και την εκμάθηση κάποιου συγκεκριμένου προγραμματιστικού περιβάλλοντος, ούτε την καλλιέργεια προγραμματιστικών δεξιοτήτων από τη μεριά των μαθητών. Δεν αποσκοπεί στη λεπτομερειακή εξέταση της δομής, του ρεπερτορίου και των συντακτικών κανόνων κάποιας γλώσσας προγραμματισμού. Δεν προτίθεται να επιχειρήσει να δημιουργήσει προγραμματιστές. Το μάθημα δεν αφορά την εκμάθηση εξεζητημένων τεχνικών προγραμματισμού, αλλά ως εργαλείο δόμησης της σκέψης πρέπει να εστιάζει στις προσεγγίσεις και στις τεχνικές επίλυσης προβλημάτων”.

Το μάθημα ΑΕΠΠ έχει σαν γενικό σκοπό οι μαθητές:

- Να καλλιεργήσουν αναλυτική σκέψη και συνθετική ικανότητα
- Να αναπτύξουν τη δημιουργικότητα, τη φαντασία στο σχεδιασμό
- Να καλλιεργήσουν και να εθιστούν στην αυστηρότητα και σαφήνεια της έκφρασης και της διατύπωσης
- Να αναπτύξουν ικανότητες μεθοδολογικού χαρακτήρα
- Να αποκτήσουν δεξιότητες αλγοριθμικής προσέγγισης
- Να καταστούν ικανοί να υλοποιούν τις λύσεις απλών προβλημάτων με χρήση βασικών προγραμματιστικών γνώσεων.

2.3 Δομή του μαθήματος ΑΕΠΠ

Όπως αναφέρθηκε και προηγουμένως, το μάθημα ΑΕΠΠ περιλαμβάνει 14 κεφάλαια, τα οποία διασπώνται σε δύο κατηγορίες:

- 1^ο Κεφάλαιο έως 5^ο Κεφάλαιο, τρόποι επίλυσης προβλημάτων με τη χρήση διαγραμμάτων ροής ή ψευδογλώσσα
- 6^ο Κεφάλαιο έως 14^ο Κεφάλαιο, κωδικοποίηση της ψευδογλώσσας με τη χρήση της ΓΛΩΣΣΑΣ

Σύμφωνα με το ΦΕΚ 2156/2012, η εξεταστέα ύλη για το μάθημα ΑΕΠΠ για το σχολικό έτος 2012-2013 αφορά τα κεφάλαια 1^ο – 3^ο και 6^ο – 10^ο, όπως και για το προηγούμενο έτος.

Στη συνέχεια, περιγράφεται το περιεχόμενο κάθε κεφαλαίου καθώς και οι διδακτικοί του στόχοι, όπως αυτοί αναφέρονται στο βιβλίο του καθηγητή.

1^ο Κεφάλαιο : Ανάλυση Προβλήματος

Στο κεφάλαιο αυτό παρουσιάζεται η έννοια του προβλήματος και γίνεται σαφής η ανεξαρτησία της λύσης του από τον υπολογιστή. Επισημαίνονται βασικά στοιχεία στην ανάλυση και σύνταξη προβλημάτων που αφορούν τη σαφήνεια της διατύπωσης και κατ' επέκταση στην κατανόησή τους. Στη συνέχεια παρουσιάζεται η έννοια της δομής ενός προβλήματος, που ουσιαστικά ανάγει την ανάλυσή του σε άλλα απλούστερα. Τέλος γίνεται σαφής καθορισμός των δεδομένων και ζητουμένων ενός προβλήματος.

Ενότητες Κεφαλαίου

- 1.1 Η έννοια πρόβλημα
- 1.2 Κατανόηση προβλήματος
- 1.3 Δομή προβλήματος
- 1.4 Καθορισμός απαιτήσεων
- 1.5 Κατηγορίες προβλημάτων
- 1.6 Πρόβλημα και υπολογιστής

Γενικός διδακτικός σκοπός

Ο γενικός διδακτικός σκοπός του κεφαλαίου είναι οι μαθητές να αναπτύξουν αναλυτικές και συνθετικές ικανότητες στην αντιμετώπιση προβλημάτων και να διατυπώνουν με σαφήνεια τις σκέψεις τους.

Με την ολοκλήρωση του κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- Να κατανοούν πλήρως τα προβλήματα που τους τίθενται
- Να προσδιορίζουν τα συστατικά μέρη ενός προβλήματος
- Να αναλύουν ένα πρόβλημα σε απλούστερα
- Να προσδιορίζουν τα δεδομένα που παρέχονται για την αντιμετώπιση ενός προβλήματος
- Να προσδιορίζουν τα ζητούμενα αποτελέσματα και τη μορφή απόδοσής τους
- Να θέτουν οι ίδιοι προβλήματα διατυπώνοντάς τα με ακρίβεια και πληρότητα

2^ο Κεφάλαιο : Βασικές έννοιες αλγορίθμων

Στο κεφάλαιο αυτό γίνεται η πρώτη γνωριμία με την έννοια του αλγορίθμου. Δίνονται οι απαραίτητοι ορισμοί, οι οποίοι συνοδεύονται με αρκετά παραδείγματα. Αλγόριθμος θεωρείται η διαδικασία της λύσης ενός προβλήματος και η παράσταση ενός αλγορίθμου μπορεί να γίνει με διάφορους τρόπους, ωστόσο δίνεται έμφαση στην παράσταση με χρήση ψευδογλώσσας. Τέλος, αναπτύσσονται οι κύριες αλγοριθμικές δομές, δηλαδή η ακολουθία, η επιλογή και η επανάληψη.

Ενότητες Κεφαλαίου

- 2.1 Τι είναι αλγόριθμος
- 2.2 Σπουδαιότητα αλγορίθμου
- 2.3 Περιγραφή και αναπαράσταση αλγορίθμων
- 2.4 Βασικές συνιστώσες/εντολές ενός αλγορίθμου
 - 2.4.1 Δομή ακολουθίας
 - 2.4.2 Δομή επιλογής
 - 2.4.3 Διαδικασίες πολλαπλών επιλογών
 - 2.4.4 Εμφωλευμένες διαδικασίες
 - 2.4.5 Δομή επανάληψης

Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να κατανοήσουν οι μαθητές την έννοια του αλγορίθμου, να αναγνωρίσουν την σπουδαιότητα των αλγορίθμων και να εκτιμήσουν την αναγκαιότητα της αλγοριθμικής προσέγγισης για την επίλυση προβλημάτων με σταδιακή προσέγγιση

αλγοριθμικών εννοιών, χρησιμοποιώντας συγκεκριμένες τεχνικές και συνιστώσες επίλυσης προβλημάτων.

Με την ολοκλήρωση του κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση :

- Να αναγνωρίζουν τη σπουδαιότητα και την αναγκαιότητα των αλγορίθμων
- Να αποτυπώνουν ένα πρόβλημα σε βήματα αλγορίθμου με συγκεκριμένη δομή
- Να διακρίνουν τα είδη των αλγοριθμικών συνιστωσών που χρησιμοποιούνται για την επίλυση προβλημάτων
- Να χρησιμοποιούν σωστά τις συνιστώσες της ακολουθίας, της επιλογής και των πολλαπλών επιλογών
- Να εφαρμόζουν στις κατάλληλες περιπτώσεις τα είδη των συνιστωσών της επανάληψης
- Να χρησιμοποιούν εμφωλευμένες αλγοριθμικές δομές και να συνδυάζουν διάφορες συνιστώσες μεταξύ τους

3^ο Κεφάλαιο : Δομές δεδομένων και αλγόριθμοι

Στο κεφάλαιο αυτό δίνεται ο ορισμός της δομής δεδομένων και εξετάζονται οι δομές πίνακα, στοίβας και ουράς. Στη συνέχεια παρουσιάζονται προβλήματα, η λύση των οποίων εντάσσεται στις κατηγορίες αναζήτησης και ταξινόμησης. Η τεχνική της σειριακής/γραμμικής αναζήτησης στοιχείων από πίνακα, δίνεται με χρήση σχετικών αλγορίθμων και προσδιορίζονται οι περιπτώσεις όπου η μέθοδος αυτή είναι αποτελεσματική.

Ενότητες Κεφαλαίου

- 3.1 Δεδομένα
- 3.2 Αλγόριθμοι + Δομές δεδομένων = Προγράμματα
- 3.3 Πίνακες
- 3.4 Στοίβα
- 3.5 Ουρά
- 3.6 Αναζήτηση
- 3.7 Ταξινόμηση

Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να γίνει κατανοητό στο μαθητή ότι τα δεδομένα αποθηκεύονται στη μνήμη του υπολογιστή (κύρια ή δευτερεύουσα) με κάποια συγκεκριμένη μέθοδο, δηλαδή με τη βοήθεια κάποιας δομής. Επίσης κάθε δομή δεδομένων συνοδεύεται απαραίτητως από κάποιες συγκεκριμένες λειτουργίες (δηλαδή πράξεις).

Με την ολοκλήρωση του παρόντος κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- Να διακρίνουν τις διάφορες δομές δεδομένων, όπως πίνακας, στοίβα, ουρά
- Να αντιλαμβάνονται ποιες είναι οι πράξεις που συνοδεύουν κάθε δομή
- Να εφαρμόσουν μεθόδους ταξινόμησης σε τυχαία σύνολα δεδομένων

6^ο Κεφάλαιο : Εισαγωγή στον προγραμματισμό

Στο κεφάλαιο αυτό, ο μαθητής έρχεται για πρώτη φορά σε επαφή με την έννοια του προγράμματος. Κατανοεί ότι η δημιουργία ενός προγράμματος είναι η μετατροπή ενός αλγορίθμου που επιλύει ένα πρόβλημα σε εντολές προγράμματος. Οι εντολές αυτές γράφονται σε κάποια γλώσσα προγραμματισμού, ενώ η επιλογή της καταλληλότερης γλώσσας εξαρτάται από το είδος της εφαρμογής, το υπολογιστικό περιβάλλον που θα εκτελεστεί καθώς και τις γνώσεις και προτιμήσεις του προγραμματιστή.

Ενότητες Κεφαλαίου

- 6.1 Η έννοια του προβλήματος
- 6.2 Ιστορική αναδρομή
- 6.3 Φυσικές και τεχνητές γλώσσες
- 6.4 Τεχνικές σχεδίασης προγραμμάτων
- 6.7 Προγραμματιστικά περιβάλλοντα

Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι οι μαθητές να καταστούν ικανοί να αναγνωρίζουν τις βασικές τεχνικές προγραμματισμού και να χειρίζονται το προγραμματιστικό περιβάλλον του σχολείου τους ώστε να εκτελούν απλά έτοιμα προγράμματα.

Μετά την ολοκλήρωση του κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση :

- Να ορίζουν τι είναι πρόγραμμα
- Να κατατάσσουν τις γλώσσες προγραμματισμού
- Να αναγνωρίζουν τα κυριότερα είδη προγραμματισμού
- Να περιγράφουν τα βασικά χαρακτηριστικά των τεχνικών που χρησιμοποιούνται στον προγραμματισμό
- Να περιγράφουν τη διαδικασία εκτέλεσης ενός προγράμματος
- Να αναφέρουν τα βασικά προγράμματα που περιέχει ένα προγραμματιστικό περιβάλλον
- Να χρησιμοποιούν το προγραμματιστικό περιβάλλον του εργαστηρίου

7^ο Κεφάλαιο : Βασικές έννοιες προγραμματισμού

Στο κεφάλαιο αυτό παρουσιάζονται τα βασικά χαρακτηριστικά της ΓΛΩΣΣΑΣ. Συγκεκριμένα το αλφάβητό της, οι τύποι δεδομένων που υποστηρίζει, οι κανόνες για τα ονόματα που χρησιμοποιούνται, οι αριθμητικές πράξεις, η εντολή εκχώρησης, οι εντολές εισόδου και εξόδου καθώς και η δομή που πρέπει να έχει ένα πρόγραμμα.

Ενότητες Κεφαλαίου

- 7.1 Το αλφάβητο της ΓΛΩΣΣΑΣ
- 7.2 Τύποι δεδομένων
- 7.3 Σταθερές
- 7.4 Μεταβλητές
- 7.5 Αριθμητικοί τελεστές
- 7.6 Συναρτήσεις

- 7.7 Αριθμητικές εκφράσεις
- 7.8 Εντολή εκχώρησης
- 7.9 Εντολές εισόδου-εξόδου
- 7.10 Δομή προγράμματος

Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να καταστούν ικανοί οι μαθητές να συντάσσουν και να εκτελούν σε δομημένη γλώσσα προγραμματισμού απλά προγράμματα τα οποία να δέχονται διαφόρων τύπων δεδομένα, στη συνέχεια να τα επεξεργάζονται και τελικά να παρέχουν τα αποτελέσματα.

Με την ολοκλήρωση του κεφαλαίου, οι μαθητές θα πρέπει να είναι σε θέση:

- Να διακρίνουν τις σταθερές από τις μεταβλητές ενός προγράμματος
- Να αναγνωρίζουν τους διάφορους τύπους των μεταβλητών
- Να χρησιμοποιούν σωστά την εντολή εκχώρησης
- Να μετατρέπουν τις αριθμητικές πράξεις σε εντολές προγράμματος
- Να χρησιμοποιούν τις εντολές εισόδου και εξόδου
- Να διατυπώνουν τη δομή ενός προγράμματος

8^ο Κεφάλαιο : Επιλογή και επανάληψη

Στο κεφάλαιο αυτό παρουσιάζονται οι εντολές που χρησιμοποιεί η ΓΛΩΣΣΑ για τις δομές επιλογής και επανάληψης.

Ενότητες Κεφαλαίου

- 8.1 Εντολές Επιλογής
- 8.2 Εντολές Επανάληψης

Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι οι μαθητές να καταστούν ικανοί να συντάσσουν και να εκτελούν σε δομημένη γλώσσα προγραμματισμού προγράμματα τα οποία χρησιμοποιούν και τις τρεις βασικές δομές του δομημένου προγραμματισμού (ακολουθία, επιλογή, επανάληψη).

Με την ολοκλήρωση του κεφαλαίου οι μαθητές θα πρέπει να είναι σε θέση:

- Να σχηματίζουν λογικές εκφράσεις, απλές και σύνθετες
- Να διατυπώνουν τις μορφές εντολών επιλογής (ελέγχου)
- Να διακρίνουν τις διαφορές των μορφών των εντολών επιλογής
- Να επιλέγουν την καλύτερη μορφή των εντολών επιλογής
- Να διατυπώνουν τις εντολές επανάληψης
- Να διακρίνουν τις διαφορές των διαφορετικών εντολών επανάληψης
- Να επιλέγουν την καλύτερη δομή επανάληψης και να χρησιμοποιούν την κατάλληλη εντολή

9^ο Κεφάλαιο : Πίνακες

Στο κεφάλαιο αυτό παρουσιάζονται οι πίνακες και πώς η ΓΛΩΣΣΑ χειρίζεται τους πίνακες. Ορίζεται η έννοια του πίνακα, ως μία ομάδα μεταβλητών ίδιου τύπου με κοινό όνομα και αποθήκευση σε διαδοχικές θέσεις στη μνήμη. Καθορίζονται οι διαστάσεις του πίνακα καθώς και ο αριθμός των δεικτών. Τέλος, μελετώνται οι επεξεργασίες που γίνονται σε πίνακες (αναζήτηση, ταξινόμηση, συγχώνευση) καθώς και οι αλγόριθμοι που έχουν αναπτυχθεί για αυτές.

Ενότητες Κεφαλαίου

- 9.1 Μονοδιάστατοι πίνακες
- 9.2 Πότε πρέπει να χρησιμοποιούνται πίνακες
- 9.3 Πολυδιάστατοι πίνακες
- 9.4 Τυπικές επεξεργασίες πινάκων

Γενικός διδακτικός σκοπός

Ο γενικός σκοπός είναι να καταστούν ικανοί οι μαθητές να χρησιμοποιούν πίνακες στα προγράμματά τους.

Με την ολοκλήρωση του κεφαλαίου οι μαθητές θα πρέπει να είναι σε θέση :

- Να επιλέγουν το είδος του πίνακα
- Να ορίζουν τους πίνακες σε ένα πρόγραμμα
- Να εισάγουν, να επεξεργάζονται και να τυπώνουν τα στοιχεία ενός πίνακα
- Να αποφασίζουν αν είναι απαραίτητη η χρήση ενός πίνακα

- Να χρησιμοποιούν πολυδιάστατους πίνακες
- Να αναφέρουν τις βασικές επεξεργασίες ενός πίνακα
- Να αναζητούν και να ταξινομούν τα στοιχεία ενός πίνακα

10^ο Κεφάλαιο : Υποπρογράμματα

Στο κεφάλαιο αυτό παρουσιάζονται οι αρχές και τα χαρακτηριστικά του τμηματικού προγραμματισμού καθώς και ο τρόπος που χειρίζεται τα υποπρογράμματα η ΓΛΩΣΣΑ. Διακρίνει τα δύο είδη υποπρογραμμάτων, συναρτήσεις και διαδικασίες και γίνεται λόγος για τον τρόπο με τον οποίο επικοινωνούν τα υποπρογράμματα με το κυρίως πρόγραμμα.

Ενότητες Κεφαλαίου

- 10.1 Τμηματικός προγραμματισμός
- 10.2 Χαρακτηριστικά των υποπρογραμμάτων
- 10.3 Πλεονεκτήματα του τμηματικού προγραμματισμού
- 10.4 Παράμετροι
- 10.5 Διαδικασίες και συναρτήσεις

Γενικός διδακτικός σκοπός

Ο γενικός σκοπός του κεφαλαίου είναι να καταστούν οι μαθητές ικανοί να χρησιμοποιούν υποπρογράμματα για τη δημιουργία σύνθετων προγραμμάτων.

Μετά την ολοκλήρωση του κεφαλαίου οι μαθητές θα πρέπει να είναι σε θέση:

- Να αναλύουν ένα πρόβλημα σε απλούστερα υποπροβλήματα
- Να διακρίνουν τις συναρτήσεις από τις διαδικασίες
- Να γράφουν τη δομή των υποπρογραμμάτων
- Να επιλέγουν τη χρήση διαδικασίας ή συνάρτησης για την υλοποίηση ενός προγράμματος
- Να χρησιμοποιούν παραμέτρους για την επικοινωνία των υποπρογραμμάτων

2.4 Διδασκαλία του μαθήματος και προβλήματα

2.4.1 Διδακτικές προσεγγίσεις

Όπως αναφέρθηκε και στην αρχή, το μάθημα ΑΕΠΠ χαρακτηρίζεται ως εργαστηριακό και η αρχική πρόταση της συγγραφικής ομάδας περιελάμβανε τρεις (3) διδακτικές ώρες για τη διδασκαλία του μαθήματος, τόσο στην αίθουσα όσο και στο εργαστήριο.

Σαν μεθοδολογία διδασκαλίας του μαθήματος οι συγγραφείς προτείνουν την **σπειροειδή προσέγγιση**. Ουσιαστικά, το μάθημα χωρίζεται σε δυο “κατηγορίες” όπου επαναλαμβάνονται τα ίδια θέματα, την πρώτη φορά με αλγοριθμική προσέγγιση και τη δεύτερη με προγραμματιστική, με στόχο να βοηθήσει το έργο του καθηγητή σε θεωρητικό και εργαστηριακό επίπεδο. Στο βιβλίο του καθηγητή αναφέρεται το εξής :

“Μέσα από το πρίσμα της σπειροειδούς προσέγγισης του μαθήματος οι έννοιες προσεγγίζονται επαναληπτικά με διαφορετικό τρόπο και/ή βάθος ανά περίπτωση. Με βάση κάθε φορά προκαθορισμένο προς επίλυση πρόβλημα, ο διδάσκων επανέρχεται σε θεματολογία και έννοιες που έχουν ήδη εν μέρει αναπτυχθεί. Δεν αποκλείονται οι επαναλήψεις περιεχομένων, με έμφαση κατά την πρώτη φορά παρουσίασης. Επιπρόσθετο λόγο για αυτού του είδους την προσέγγιση, αποτελεί η ανάγκη επανάληψης θεμάτων και εννοιών (π.χ. δομές ελέγχου κ.α.) ,λόγω διττής αναφοράς τους, καταρχήν σε επίπεδο αλγοριθμικής προσέγγισης και εν συνεχεία σε επίπεδο προγραμματιστικού περιβάλλοντος.”

Η σπειροειδής προσέγγιση είχε σαν στόχο με την επανάληψη, το μάθημα να επιτύχει τόσο σε θεωρητικό όσο και σε προγραμματιστικό επίπεδο. Το πρόγραμμα σπουδών όμως περιόρισε τις ώρες διδασκαλίας από

τρεις που ήταν αρχικά σε δύο. Ο λιγότερος διδακτικός χρόνος σε συνδυασμό με την μεγάλη ύλη, είχε σαν αποτέλεσμα οι καθηγητές να μην έχουν τον απαραίτητο χρόνο ώστε να παρουσιάσουν την ύλη του μαθήματος με πληρότητα, ούτε να ακολουθήσουν τη μέθοδο που προτάθηκε. Οι μαθητές από την άλλη έχουν την δυνατότητα να επιλέξουν οποιαδήποτε μορφή αναπαράστασης αλγορίθμων για την επίλυση των θεμάτων τους, με επικρατέστερες τις “ψευδογλώσσα” και “ΓΛΩΣΣΑ”.

Η “ελευθερία” επιλογής της μορφής κωδικοποίησης δημιουργεί και θετικά και αρνητικά χαρακτηριστικά στον τρόπο διεξαγωγής του μαθήματος. Από τη μία πλευρά έρχεται σε αντίθεση με το αυστηρό λεξιλόγιο και συντακτικό μιας γλώσσας προγραμματισμού και δίνει τη δυνατότητα στους μαθητές να επιλέξουν τον τρόπο κωδικοποίησης που επιθυμούν ή έχουν κατανοήσει καλύτερα. Αντίθετα, κατά την αξιολόγηση των γραπτών, δημιουργούνται ασάφειες και παρανοήσεις από αυτή την ελευθερία σε θέματα παρουσίασης και ερμηνείας αλγοριθμικών δομών. Χαρακτηριστικό παράδειγμα αποτελούν τα **Υποπρογράμματα** που διδάσκονται στο 10^ο Κεφάλαιο και παρουσιάζονται μόνο σε προγραμματιστικό περιβάλλον. Αν υπάρξει σχετικό θέμα ανάπτυξης υποπρογράμματος, οι μαθητές καλούνται να χρησιμοποιήσουν τη ΓΛΩΣΣΑ για την κωδικοποίηση του θέματος. Με τον τρόπο αυτό καταρρίπτεται η δυνατότητα επιλογής των μαθητών να διατυπώνουν τις λύσεις των ασκήσεων σε οποιαδήποτε μορφή αναπαράστασης.

Τέλος, οι χρονικοί περιορισμοί αποθαρρύνουν τους εκπαιδευτικούς να επιχειρήσουν την εργαστηριακή υποστήριξη του μαθήματος, όπως προτείνεται στο πρόγραμμα σπουδών, με αποτέλεσμα η ΓΛΩΣΣΑ να

διδάσκεται στον πίνακα και να εξετάζεται γραπτώς, πρακτική που δεν θυμίζει προγραμματισμό.

2.4.2 Προβλήματα που αντιμετωπίζουν οι μαθητές

Η διδασκαλία του προγραμματισμού, αποτελεί μια ενδιαφέρουσα και χρήσιμη διαδικασία, που έχει σαν στόχο την ανάπτυξη μεθοδολογικού χαρακτήρα και της δομημένης σκέψης. Με το μάθημα ΑΕΠΠ οι μαθητές της δευτεροβάθμιας εκπαίδευσης έρχονται για πρώτη φορά σε επαφή με την επιστήμη αυτή. Ωστόσο, η “νέα” αυτή επιστήμη, δημιουργεί και ορισμένες δυσκολίες.

Μια σημαντική δυσκολία μπορεί να χαρακτηριστεί το υπόβαθρο των μαθητών. Ένα μεγάλο ποσοστό στο παρελθόν δεν είχε καμία επαφή με τον προγραμματισμό και οι γνώσεις τους περιορίζονται στη χρήση διαδικτύου, παιχνιδιών και προγραμμάτων όπως ζωγραφική ή Word. Το γεγονός αυτό έχει σαν αποτέλεσμα να δυσκολεύονται να κατανοήσουν τη διαφορά ανάμεσα σε χρήστη και προγραμματιστή, ειδικά όταν έχουν να αντιμετωπίσουν μια δραστηριότητα μέσα στην τάξη, π.χ. συγγραφή ενός προγράμματος. Επίσης, οι μαθητές αγνοούν βασικές αρχές λειτουργίας του Η/Υ και κατ’ επέκταση δεν μπορούν να αντιληφθούν τον τρόπο επικοινωνίας μεταξύ μηχανής-προγραμματιστή. Ιδανικά, θα έπρεπε σε κάποια προηγούμενα μαθήματα, να διδάσκεται η ιστορία της προσπάθειας επικοινωνίας ανθρώπου-Η/Υ, η εξέλιξη των γλωσσών προγραμματισμού και να καταλήγουμε στη διδασκαλία του προγραμματισμού.

Παρατηρούνται έτσι προβλήματα στους μαθητές όπως:

- Αδυνατούν να κατανοήσουν την έννοια του προβλήματος και να την αντιστοιχίσουν με παραδείγματα της καθημερινότητάς τους

- Σε πολλές περιπτώσεις, είναι περισσότερο κατανοητή μια προσέγγιση του αλγορίθμου στη φυσική γλώσσα, όμως δύσκολη η μεταφορά της σε κάποια γλώσσα προγραμματισμού, έστω και αν αυτή είναι ψευδοκώδικας

Επιπλέον, με τη σπειροειδή προσέγγιση, προσεγγίζονται επαναληπτικά οι ίδιες έννοιες, που όμως φέρνουν σε σύγχυση τόσο τους μαθητές όσο και τους εκπαιδευτικούς. Για παράδειγμα, κάποιες λέξεις-κλειδιά της **ΓΛΩΣΣΑΣ** δυσκολεύουν τους μαθητές, ενώ κατανοούν το νόημά τους σε φυσική γλώσσα. Π.χ. τότε χρησιμοποιούμε τις λέξεις **Δεδομένα** και **Αποτελέσματα**, ή τότε τις λέξεις **Εμφάνισε** και τότε **Γράψε**.

Τέλος, η έλλειψη ή η μη χρήση προγραμματιστικού λογισμικού, έχει σαν συνέπεια να δυσκολεύονται οι μαθητές σε βασικές έννοιες όχι μόνο του συγκεκριμένου μαθήματος, αλλά και γενικότερα του προγραμματισμού. Αδυνατούν να κατανοήσουν την έννοια της προγραμματιστικής μεταβλητής, την οποία συχνά συγκρίνουν με τη μεταβλητή στα μαθηματικά. Κάποια διδακτικά προβλήματα που παρατηρούνται είναι:

- Ανάθεση τιμής
- Ο τύπος της μεταβλητής
- Η αλλαγή της τιμής της (μετά από κάποια πράξη)
- Η σύγχυση της εκχώρησης με την ισότητα

Οι δομές ελέγχου και επανάληψης, αποτελούν επίσης πρόβλημα το οποίο οφείλεται κυρίως στην μη διδασκαλία της Λογικής. Για παράδειγμα, ένα πρόβλημα που εκφράζεται σε φυσική γλώσσα και περιέχει διάζευξη (το

λογικό ή) , όπως “*κάνε αυτό ή το άλλο*” , δημιουργεί πρόβλημα όταν χρησιμοποιείται στην εκφώνηση κάποιου προβλήματος.

Παρ’ όλες τις δυσκολίες όμως, το μάθημα κρίνεται πολύ σημαντικό, καθώς αποτελεί μια πρώτη εμπειρία με την επιστήμη του προγραμματισμού και όλα τα θετικά της, διαχωρίζει τις έννοιες χρήστης υπολογιστή-προγραμματιστής και αποτελεί προπομπό για τους μαθητές που θα επιλέξουν για τις σπουδές τους τις θετικές επιστήμες και ιδιαίτερα της πληροφορικής.

Ενότητα 3: Διδακτικές προσεγγίσεις και εκπαιδευτικά λογισμικά

3.1 Η έννοια της μάθησης

Ως μάθηση, μπορεί να θεωρηθεί ένα σύνθετο εσωτερικό βιολογικό και πνευματικό φαινόμενο που έχει μελετηθεί από διάφορους κλάδους της επιστήμης όπως ψυχολογία, παιδαγωγική, φυσιολογία, ιατρική κ.α. Οι διαδικασίες της μάθησης είναι τόσο ποικιλόμορφες και διαφορετικές, ώστε η ένταξή τους σε μία και μοναδική κατηγορία δεν μπορεί να είναι βάσιμη και πλήρης.

Διάφοροι ορισμοί της μάθησης που έχουν δοθεί κατά καιρούς είναι:

- Δημιουργία υποκατάστατων ανακλαστικών (Pavlov)
- Δοκιμή και πλάνη (Thorndike)
- Επανάληψη μιας ανάδρασης μετά από θετική ενίσχυση (Skinner)
- Μίμηση προτύπου (Bandura)
- Επεξεργασία πληροφοριών (Neisser, Seymour, Gagne)

Αντιπροσωπευτικός μπορεί να θεωρηθεί ο ορισμός που προτάθηκε από τον Kimble (1980): *“Μάθηση είναι μια σχετικά σταθερή αλλαγή σε μία δυνατότητα της συμπεριφοράς, η οποία συμβαίνει ως αποτέλεσμα ενισχυμένης πρακτικής”*.

Η μάθηση έχει προσωπικό και ατομικό χαρακτήρα. Κάθε άτομο διακρίνεται από ένα μοναδικό τρόπο να μαθαίνει, είτε πρόκειται για γνώσεις που προέρχονται από ένα σχολικό μάθημα, είτε κάποια άλλη δραστηριότητα. Η μάθηση δεν ολοκληρώνεται μόνο μέσα από ένα

εκπαιδευτικό σύστημα, αλλά αποτελεί μια συνεχή διεργασία που συντελείται δια βίου και χρησιμοποιεί διάφορους τρόπους και μέσα.

Τα στάδια μάθησης μπορούν να θεωρηθούν οι επιμέρους διαδικασίες που εκτελούνται κατά την πραγμάτωση της μάθησης και συνοψίζονται στα εξής :

- Διαδικασία στροφής της προσοχής
- Διατήρηση στη βραχυπρόθεσμη μνήμη
- Κωδικοποίηση
- Συγκέντρωση και διαφύλαξη
- Ανάκτηση
- Γεννήτρια αντιδράσεων
- Εκτέλεση
- Επανατροφοδότηση
- Διαδικασίες εκτελεστικού ελέγχου

Ιδιαίτερο ενδιαφέρον παρατηρείται στην μελέτη και έρευνα της **τυπικής μάθησης** (formal learning), της **μη τυπικής μάθησης** (non formal learning) και της **άτυπης μάθησης** (informal learning).

Τυπική μάθηση, ορίζεται το ιεραρχημένο, δομημένο και οργανωμένο χρονικά σε βαθμίδες εκπαιδευτικό σύστημα, από την πρωτοβάθμια εκπαίδευση ως το πανεπιστήμιο, που περιλαμβάνει τόσο τις γενικές ακαδημαϊκές σπουδές όσο και τα εξειδικευμένα προγράμματα και θεσμούς ολοκληρωμένης επαγγελματικής και τεχνικής εκπαίδευσης (Jeffs & Smith 1999). Οι βαθμίδες εκπαίδευσης στην Ελλάδα είναι : Νηπιαγωγείο, Δημοτικό, Γυμνάσιο, Λύκειο, ΤΕΕ, ΑΕΙ, ΤΕΙ, ΙΕΚ.

Η τυπική μάθηση αναπαριστά ουσιαστικά την “παραδοσιακή” μέθοδο μετάδοσης γνώσης, όπου κύριος καθοδηγητής θεωρείται ο δάσκαλος. Αυτός διδάσκει στην αίθουσα και οι μαθητές γίνονται αποδέκτες της γνώσης. Η γνώση θεωρείται ως ετοιμοπαράδοτο αγαθό το οποίο μεταβιβάζεται από το δάσκαλο στο μαθητή. Η διαδικασία μάθησης αποτελεί αντικείμενο ενός προγράμματος σπουδών, ενώ το παιδαγωγικό ενδιαφέρον επικεντρώνεται στη διδακτέα ύλη και την εφαρμογή του προγράμματος σπουδών.

Μη τυπική μάθηση, ορίζεται οποιαδήποτε εκπαιδευτική δραστηριότητα εκτός του τυπικού εκπαιδευτικού συστήματος, που απευθύνεται σε συγκεκριμένους εκπαιδευόμενους και έχει συγκεκριμένους εκπαιδευτικούς στόχους (Jeffs & Smith 1990). Διάφορα εκπαιδευτικά προγράμματα, κυρίως για εκπαιδευτικούς όλων των βαθμίδων αποτελούν μέρος της τυπικής μάθησης.

Παράδειγμα τυπικής μάθησης αποτελεί η Περιβαλλοντική Εργασία, η διαδικασία η οποία στοχεύει στην ανάπτυξη, σε διάφορους τομείς της κοινωνίας, περιβαλλοντικών εννοιών, δεξιοτήτων, στάσεων και περιβαλλοντικού ήθους, η οποία πραγματοποιείται από διάφορα ιδρύματα όπως σχολεία και πανεπιστήμια (Φραγκούλης & Καραγιάννης 1997).

Άτυπη μάθηση, θεωρείται σύμφωνα με τους Jeffs και Smith η διαδικασία με την οποία κάθε άτομο, σε όλη τη διάρκεια της ζωής του μαθαίνει να αποκτά αξίες, ικανότητες, δεξιότητες και γνώσεις από την καθημερινή του εμπειρία και τις επιδράσεις που δέχεται από το περιβάλλον του (εργασία, οικογένεια, ασχολίες, βιβλία, μέσα μαζικής ενημέρωσης κ.α.). Είναι η μάθηση που προκύπτει από δραστηριότητες της καθημερινής

ζωής οι οποίες σχετίζονται με την εργασία, την οικογένεια ή τον ελεύθερο χρόνο και δεν είναι διαρθρωμένη από άποψη μαθησιακών στόχων, χρόνου μάθησης ή διδακτικής υποστήριξης, γι' αυτό και δεν οδηγεί τυπικά σε επίσημη πιστοποίηση.

Ο μαθητής, μαθαίνει κυρίως έξω από την τάξη, χωρίς την καθοδήγηση του δασκάλου του, ακολουθώντας δικά του “εσωτερικά” κίνητρα, ενώ μπορεί να συνυπάρξει σε αυτού του είδους μάθηση με άτομα εκτός επίσημου προγράμματος σπουδών και να αναπτύσσει πολύτιμες δεξιότητες.

Η άτυπη μάθηση μπορεί να θεωρηθεί ως “ελαστική” ενώ προσαρμόζεται πιο εύκολα και γρήγορα σε νέες ανάγκες σε σχέση με τα επίσημα προγράμματα σπουδών. Η γνώση που αποκτάται από την άτυπη μάθηση αποτελεί αντικείμενο έρευνας, για το λόγο αυτό είναι επιθυμητό να μπορούν να συνδυάζονται τα πλεονεκτήματα της με αυτά της αναγνωρισμένης “τυπικής” γνώσης.

Παραδείγματα άτυπης μάθησης μπορούν να θεωρηθούν:

- Η μάθηση μιας γλώσσας
- Πολιτιστικές αναφορές
- Μάθηση από τον εργασιακό περίγυρο (συμμαθητές, συνάδελφοι)
- Χειρισμός νέων τεχνολογιών (H/Y, κινητό τηλέφωνο)

3.2 Θεωρίες μάθησης

Ο τρόπος που μαθαίνουμε, έχει αποτελέσει και συνεχίζει να αποτελεί αντικείμενο μελέτης και έρευνας. Παρατηρείται μία προσπάθεια κατανόησης του τρόπου με τον οποίο η ψυχολογία, καθώς και το σύνολο των γνωστικών επιστημών, αντιλαμβάνονται τις διαδικασίες μάθησης όταν αυτές λαμβάνουν χώρα σε σχολικές συνθήκες.

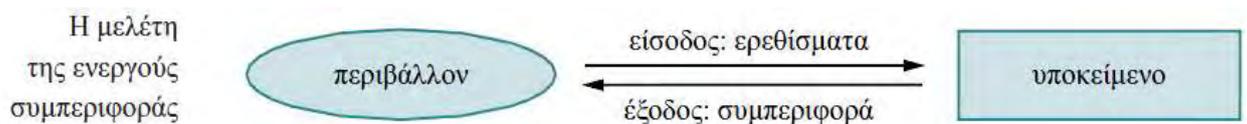
Μπορούμε να διακρίνουμε τρία μεγάλα ψυχολογικά ρεύματα που επηρέασαν και συνεχίζουν να επηρεάζουν τη διαδικασία της σχολικής μάθησης και να περιγράψουν τις αντίστοιχες διδακτικές παρεμβάσεις :

- **Συμπεριφορισμός** (behaviorism) , (αναμετάδοση της πληροφορίας και τροποποίηση της συμπεριφοράς)
- **Εποικοδομισμός** (constructivism) , (οικοδόμηση νέων γνώσεων στις ήδη υπάρχουσες)
- **Θεωρία της δραστηριότητας** (activity theory) , (οι γνωστικές διεργασίες δεν θεωρούνται αυτόνομες ,αλλά συστατικά ενός οργανωμένου κοινωνικοπολιτισμικού περιβάλλοντος)

Για το **συμπεριφορισμό**, η μάθηση συνίσταται στην **τροποποίηση της ανθρώπινης συμπεριφοράς**. Πρόδρομος της σχολής αυτής υπήρξε ο **I. Pavlov** (με πειράματα σε ζώα σχετικά με την εξαρτημένη συμπεριφορά) και βασικοί εκπρόσωποί της οι **J.B. Watson**, **E.L. Thorndike** και **B.F. Skinner**.

Ο συμπεριφορισμός δεν αποδέχεται τη δυνατότητα πρόσβασης στις νοητικές καταστάσεις των υποκειμένων. Το μόνο που πρέπει να γίνει είναι η περιγραφή της συμπεριφοράς και όχι η εξήγηση. Αποδέχεται επίσης ότι

υπάρχουν γενικοί νόμοι που διέπουν την ανθρώπινη συμπεριφορά : οι νόμοι αυτοί μπορούν να ανακαλυφθούν εάν συσχετίσουμε τα φυσικά χαρακτηριστικά των ερεθισμάτων που δέχεται το υποκείμενο με τα φυσικά χαρακτηριστικά της συμπεριφοράς του. Πρόκειται για το μοντέλο **Stimuli – Response** (Ερέθισμα - Απόκριση).



Εικόνα 1. Μοντέλο Stimuli-Response

Με άλλα λόγια, ο συμπεριφορισμός δεν ενδιαφέρεται για την εσωτερική (νοητική) λειτουργία των υποκειμένων που μαθαίνουν, αφού θεωρεί τον νου ως μαύρο κουτί χωρίς δυνατότητα πρόσβασης, αλλά εστιάζει την προσοχή του στην ανάλυση των χαρακτηριστικών εισόδου-εξόδου της ανθρώπινης συμπεριφοράς.

Παράδειγμα διδακτικού μοντέλου που αναπαριστά μια συστηματική και δομημένη προσέγγιση για το σχεδιασμό διδακτικών διαδικασιών και παρεμβάσεων και βασίζεται άμεσα στις συμπεριφοριστικές προσεγγίσεις, είναι το μοντέλο του *Διδακτικού Σχεδιασμού*. Τα στάδια ανάλυσης του διδακτικού σχεδιασμού είναι σύμφωνα με τον **Boyle** τα εξής :

- **Αξιολόγηση αναγκών** : προσδιορίζει κάθε δραστηριότητα του μαθητή και κάθε τμήμα της γνώσης που πρέπει να προσκτηθεί από αυτόν

- **Επιλογή διδακτικών μεθόδων και υλικού** : στηρίζονται σε μετρήσιμα μεγέθη συμπεριφοράς
- **Αξιολόγηση του μαθητή** : βασίζεται κυρίως σε μία σειρά από τεστ, τα οποία επιτρέπουν να αποφανθούμε για την επίτευξη των διδακτικών στόχων

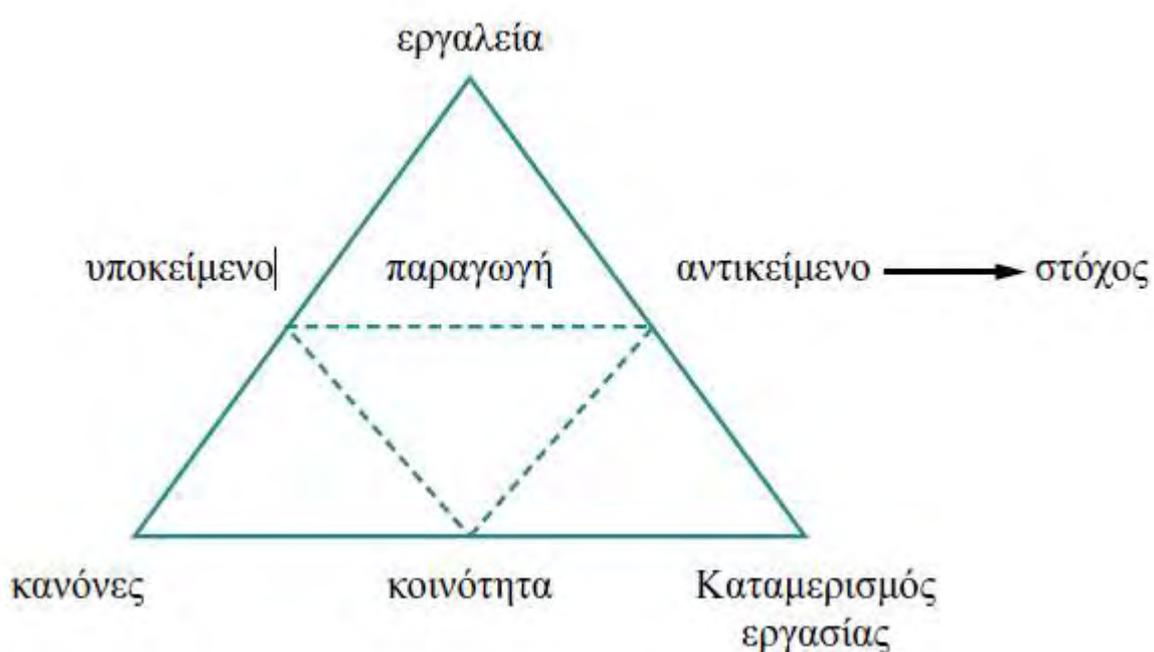
Αντίθετα με τη θεώρηση του συμπεριφορισμού, μελέτες και έρευνες υποστηρίζουν ότι η μαθησιακή διαδικασία δεν μπορεί να πραγματοποιηθεί ουσιαστικά εάν δεν πάρει υπόψη της τον τρόπο με τον οποίο οικοδομούν τις γνώσεις τους τα υποκείμενα που μαθαίνουν. Για το λόγο αυτό, οι προσεγγίσεις που υποστηρίζουν ότι οι γνώσεις δεν μεταδίδονται αλλά “οικοδομούνται” με προσωπικό τρόπο, ονομάζονται **εποικοδομιστικές**.

Η εμφάνιση της εποικοδομιστικής θεωρίας χρωστάει πολλά στην εξέλιξη της ψυχολογίας, με τις εργασίες του **Jean Piaget** και της επιστημολογίας, οι οποίες επηρέασαν σε μεγάλο βαθμό την παιδαγωγική σκέψη και τον προσανατολισμό της εκπαιδευτικής έρευνας (Σολομωνίδου 1999). Σύμφωνα με τη θεωρία αυτή, η νόηση είναι μια λειτουργία κατασκευής νοημάτων βασισμένη πάνω σε όλη την εμπειρία του ατόμου. Η δόμηση της γνώσης είναι μια λειτουργία που βασίζεται στις προϋπάρχουσες εμπειρίες, τις νοητικές κατασκευές, τις πεποιθήσεις, τις “θεωρίες” που ο καθένας χρησιμοποιεί, προκειμένου να ερμηνεύσει αντικείμενα ή γεγονότα και τις οποίες δεν υποτιμά ο δάσκαλος κατά τις διδακτικές του επιδιώξεις (Ράπτης, Ράπτη 2001).

Οι βασικές παραδοχές της εποικοδομιστικής θεωρίας έχουν διαμορφωθεί με βάση ένα σημαντικό αριθμό ερευνητικών δεδομένων :

- Οι μαθητές δεν θεωρούνται πλέον παθητικοί δέκτες, αλλά τελικοί υπεύθυνοι της δικής τους μάθησης. Σε κάθε μαθησιακή διαδικασία φέρνουν τις δικές τους προηγούμενες αντιλήψεις και απόψεις.
- Η μάθηση θεωρείται ότι εμπλέκει το μαθητή με ενεργό τρόπο στη μαθησιακή διαδικασία. Η μάθηση προϋποθέτει την οικοδόμηση νοήματος και συμβαίνει συχνά μέσα από προσωπική διαπραγμάτευση.
- Η γνώση οικοδομείται με προσωπικό και κοινωνικό τρόπο.
- Οι διδάσκοντες φέρνουν στις μαθησιακές καταστάσεις τις δικές τους ιδέες και αντιλήψεις. Φέρνουν όχι μόνο τη γνώση που έχουν για το αντικείμενο, αλλά και τις απόψεις τους για τη διδασκαλία και τη μάθηση και όλα αυτά επηρεάζουν τον τρόπο αλληλεπίδρασης με τα παιδιά μέσα στην τάξη.
- Η διδασκαλία δεν είναι η μετάδοση της γνώσης, αλλά προϋποθέτει την οργάνωση των καταστάσεων μέσα στην τάξη και το σχεδιασμό των δραστηριοτήτων με τρόπο να προωθούν την οικοδόμηση της επιστημονικής γνώσης.
- Το αναλυτικό πρόγραμμα δεν είναι αυτό το οποίο θα πρέπει να μάθει κανείς, αλλά αποτελεί ένα πρόγραμμα από μαθησιακές δραστηριότητες, υλικά, πηγές, μέσα από τα οποία οι μαθητές οικοδομούν τη γνώση.

Τέλος, η **Θεωρία της Δραστηριότητας**, έχει τις ρίζες της στην ψυχολογική σχολή των **Vygotsky, Leontiev, Luria**. Βασική αρχή της θεωρίας αυτής είναι ότι η ανθρώπινη δράση διαμεσολαβείται από πολιτισμικά σύμβολα (cultural signs) : λέξεις και εργαλεία τα οποία επιδρούν στη δραστηριότητα του ατόμου και συνεπώς στις νοητικές του διεργασίες.



Εικόνα 2. Η θεωρία της δραστηριότητας

Η θεωρία της δραστηριότητας αποτελεί ένα πλαίσιο για τη μελέτη διαφορετικών μορφών ανθρώπινων πράξεων ως αναπτυξιακών διαδικασιών, διασυνδεδεμένα με το κοινωνικό και ατομικό επίπεδο. Μελετά τα άτομα στο φυσικό τους περιβάλλον λαμβάνοντας υπόψη τους πολιτιστικούς παράγοντες και τις αναπτυξιακές πτυχές της πνευματικής ανθρώπινης ζωής. Η βασική μονάδα ανάλυσης είναι η δραστηριότητα.

Τέλος, έχει σημαντικές εφαρμογές στις έρευνες που αφορούν την Επικοινωνία Ανθρώπου-Μηχανής και ειδικότερα στο σχεδιασμό συνεργατικών μαθησιακών περιβαλλόντων με υπολογιστή, τα οποία ευνοούν τις συνεργατικές δραστηριότητες και συνακόλουθα τη **συνεργατική μάθηση** (collaborative learning).

3.3 Η γλώσσα Logo και Logo like εκπαιδευτικά περιβάλλοντα

Η γλώσσα προγραμματισμού **Logo** θεωρείται μια γλώσσα υψηλού επιπέδου και είναι απόγονος της **Lisp**. Δημιουργήθηκε στα μέσα της δεκαετίας του 1960 από τους **S. Papert**, **W. Feurzeig** και **D. Bobrow** και είναι μια γλώσσα που επιτρέπει το χειρισμό και την οπτική απόδοση συμβόλων, ώστε να είναι εύκολη και αλληλεπιδραστική η εργασία των παιδιών με αυτή.

Το όνομά της προέρχεται από την Ελληνική λέξη *λόγος*, που συνδέεται με τις έννοιες *συλλογισμός*, *γλώσσα* και *υπολογισμός*. Χρησιμοποιήθηκε ως πληροφορική βάση για να αναπτύξει ένα καινοτόμο εκπαιδευτικό σχέδιο, το οποίο φιλοδοξούσε να απαντήσει στο ερώτημα “*πώς να αλλάξουμε την εκπαίδευση χρησιμοποιώντας την τεχνολογία*”.

Η Logo, στηρίζεται στις αρχές του δομημένου προγραμματισμού και υπήρξε η πρώτη γλώσσα προγραμματισμού που προσέφερε στους χρήστες της ένα περιβάλλον εργασίας με χρήση γραφικών. Ιδιαίτερη έμφαση κατά το σχεδιασμό της, δόθηκε στην απλότητα, στη δυναμική δημιουργία και στον απλό χειρισμό εικόνων, ήχων, λέξεων και αριθμών.

Η απλότητά της, την έκανε να ταυτιστεί με τη χρήση υπολογιστών σε παιδιά μικρής ηλικίας, διότι τόσο το περιβάλλον όσο και η σύνταξή της, παρέχουν τη δυνατότητα σε παιδιά κάθε ηλικίας να επικοινωνήσουν εύκολα με τον υπολογιστή. Η διδασκαλία της θεωρήθηκε ως ένα εργαλείο αυτόνομης μάθησης που αφορά στην ανάπτυξη γνωστικών δεδεξιότητων υψηλού επιπέδου.

Χαρακτηριστικό της γλώσσας είναι οι δυνατότητες των γραφικών και ο διαλογικός τρόπος προγραμματισμού, ο οποίος επιτυγχάνεται με ένα τριγωνικό σύμβολο στην οθόνη, που ονομάζεται **χελώνα**. Δίνεται η ευκαιρία στο χρήστη να κατασκευάσει και δικές του εντολές (για παράδειγμα μικρά προγράμματα, **διαδικασίες**), οι οποίες μπορούν να χρησιμοποιηθούν και αυτές από τον κατασκευαστή. Σύμφωνα με τον Papert, η χρήση της Logo *“δίνει στα παιδιά ένα αντικείμενο με το οποίο μπορούν να σκέπτονται, ένα εργαλείο που τα βοηθάει να μαθαίνουν ανακαλύπτοντας”*.

Η Logo σαν προγραμματιστικό περιβάλλον, όπως και άλλες γλώσσες, μπορεί να αντιμετωπίσει πληθώρα προβλημάτων. Το κύριο γνώρισμα που την κάνει κατάλληλη για εκπαιδευτική χρήση είναι ότι σχεδιάστηκε εξ' αρχής ως γνωστικό εργαλείο από τον Papert, σαν αποτέλεσμα ερευνών στις δυσκολίες κατανόησης μαθηματικών εννοιών από τα παιδιά. Η επιτυχία και η ισχύς της βασίζεται σε δύο τομείς : 1^ον ότι το περιβάλλον εργασίας της είναι σχετικά απλό, ικανό να βοηθήσει τον άπειρο χρήστη να αντιληφθεί και να αναπτύξει σε σχετικά σύντομο χρονικό διάστημα τις διαδικασίες που απαιτούνται για την επίλυση ενός προβλήματος στον υπολογιστή και 2^ον ότι βοηθά τον προγραμματιστή να συγκεντρωθεί στο πρόβλημά του, χωρίς να τον απασχολούν οι ιδιαιτερότητες και οι περιορισμοί του εργαλείου που χρησιμοποιεί.

Ορισμένα χαρακτηριστικά που την διαφοροποιούν σε σχέση με άλλες γλώσσες προγραμματισμού είναι:

- Ευκολία εκμάθησης
- Εντολές εύκολες στην απομνημόνευση
- Διευκρινιστικά μηνύματα λάθους
- Προωθεί καλές προγραμματιστικές τεχνικές
- Ενθαρρύνει την ανάπτυξη μικρών προγραμμάτων (διαδικασίες)
- Παρέχει επαναχρησιμοποιήσιμο κώδικα
- Αποτελεί ευχάριστη αρχή για απόκτηση τεχνολογικής κουλτούρας

Αξίζει τέλος να αναφερθεί πως η Logo δημιουργεί την έννοια του *μικρόκοσμου* , μιας απλοποιημένης έννοιας του πραγματικού κόσμου ή αλλιώς μιας οντότητας που αποκόπτει πληροφορία και αφήνει λίγες εντολές για ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων. Είναι ο χώρος που μαθαίνει ο μαθητής, που συνεργάζεται και δημιουργεί. Εξερευνεί εναλλακτικές λύσεις, δοκιμάζει υποθέσεις και ανακαλύπτει δεδομένα για τον κόσμο. Μαθαίνει μέσα από τη δράση.

Η Logo, αν και υπάρχει από το 1960, αποτέλεσε και συνεχίζει να αποτελεί πρόδρομο για τη δημιουργία Logo like περιβαλλόντων. Παραδείγματα αποτελούν τα :

- **MicroWorlds Pro**, το οποίο παρέχεται μέσω του ΥΠΕΠΘ σε μαθητές δημοτικού και γυμνασίου
- **Lego Mindstorms**, εκπαιδευτικό λογισμικό που αναπτύχθηκε για να δίνει “ζωή” σε απλές κατασκευές από τουβλάκια, τα οποία μέσω προγραμματισμού γίνονται ρομπότ

- **Scratch**, εκπαιδευτικό λογισμικό που στην ουσία πρόκειται για σύνθεση παρά στην συγγραφή κώδικα

Ενότητα 4: Το εκπαιδευτικό λογισμικό Scratch

4.1 Ιστορικά στοιχεία και περιγραφή

Το **Scratch** είναι μια διερμηνευόμενη δυναμική οπτική γλώσσα προγραμματισμού, βασισμένη και υλοποιημένη σε **Squeak**. Όντας δυναμική, επιτρέπει σε αλλαγές του κώδικα ακόμη και κατά τη διάρκεια της εκτέλεσης των προγραμμάτων. Έχει ως στόχο τη διδασκαλία εννοιών προγραμματισμού σε παιδιά και εφήβους και να τους επιτρέψει να δημιουργήσουν παιχνίδια, βίντεο και μουσική. Μπορεί να μεταφορτωθεί δωρεάν και χρησιμοποιείται σε μια ευρεία ποικιλία δράσεων εντός και εκτός του σχολείου ανά τον κόσμο.

Έχει αναπτυχθεί από μια μικρή ομάδα ερευνητών στο **Lifelong Kindergarten Group**, στο **MIT Media Lab**, με επικεφαλή τον **Mitchel Resnick**. Το όνομα Scratch προέρχεται από την τεχνική του scratching στα πικάπ, και αναφέρεται τόσο στη γλώσσα όσο και στην υλοποίηση της. Η ομοιότητα προς το scratching στη μουσική είναι η εύκολη επαναχρησιμοποίηση κομματιών: στο Scratch όλα τα αλληλεπιδραστικά αντικείμενα, γραφικά και ήχοι μπορούν εύκολα να εισαχθούν σε ένα νέο πρόγραμμα και να συνδυαστούν με νέους τρόπους. Έτσι οι αρχάριοι μπορούν να λάβουν γρήγορα αποτελέσματα και αποκτούν κίνητρο να προσπαθήσουν περαιτέρω.

Χρησιμοποιείται παγκοσμίως σε διάφορα σχολεία και εκπαιδευτικούς οργανισμούς. Ο δικτυακός τόπος του Scratch έχει αναπτύξει μια κοινότητα από αρχόμενους προγραμματιστές, μαθητές, δασκάλους και ερασιτέχνες, που αλληλοπαρακινούνται να αναπτύξουν την δημιουργικότητά τους και

τις προγραμματιστικές δεξιότητές τους. Ένα τμήμα στον ιστότοπο του Scratch είναι αφιερωμένο σε συζητήσεις μεταξύ εκπαιδευτικών.

Το σλόγκαν του Scratch είναι *Imagine – Program – Share* (Φαντάσου - Φτιάξε - Μοιράσου). Η έμφαση στο μοίρασμα είναι σημαντικό μέρος της παιδαγωγικής για το Scratch: τα προγράμματα δεν θεωρούνται μαύρα κουτιά, αλλά αντικείμενα που μπορούν να αναμιχθούν για τη δημιουργία νέων έργων. Ακολουθεί δηλαδή την αρχή του *εποικοδομισμού* (**constructionism, Seymour Papert**), όπου οι μαθητές δεν αποκτούν τη γνώση, αλλά την οικοδομούν και μαθαίνουν να συνεργάζονται.



Εικόνα 3. Scratch Logo

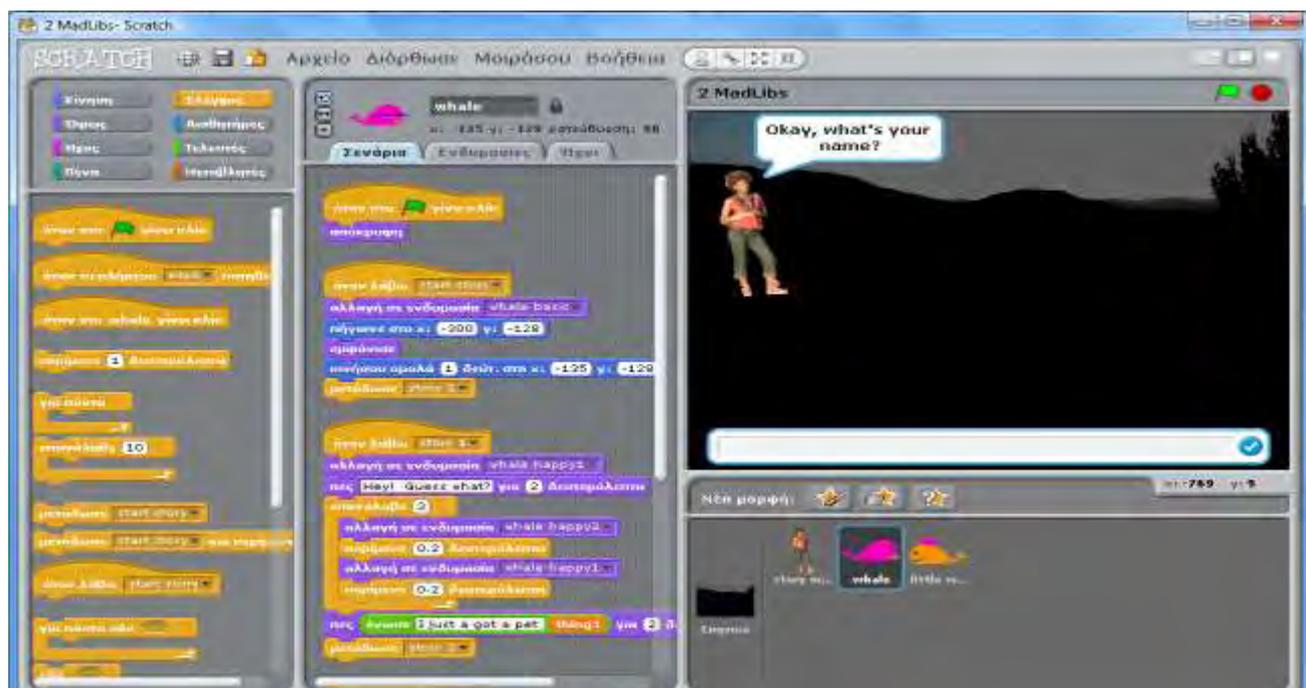
Η δημοτικότητα του Scratch στην εκπαίδευση οφείλεται στην ευκολία με την οποία μπορούν να δημιουργηθούν προγράμματα: οι εντολές

και οι δομές δεδομένων είναι απλές, γραμμένες στην καθομιλουμένη και η δομή του προγράμματος μπορεί να σχεδιαστεί σαν ένα πάζλ, με αποσπώμενα κομμάτια κώδικα που μπορούν να μετακινηθούν και προσαρμοστούν μαζί.

Τέλος, όσον αφορά το περιβάλλον ανάπτυξης του Scratch, μπορεί να μεταφορτωθεί δωρεάν από τον δικτυακό τόπο <http://scratch.mit.edu/> και είναι συμβατό με λειτουργικά *Windows*, *Mac OS X* και διάφορες εκδόσεις του *Linux*. Τα προγράμματα μπορούν να φορτωθούν αυτόματα από το περιβάλλον ανάπτυξης σε προσωπικές σελίδες στον ιστότοπο του Scratch, όπου άλλα μέλη της κοινότητας του Scratch μπορούν να τα μεταφορτώσουν (συμπεριλαμβανομένου του πλήρους πηγαίου κώδικα) για μάθηση ή ανάμιξη σε νέα έργα. Ο ιστότοπος, που είναι υλοποιημένος πάνω στην πλατφόρμα ScratchR, παρέχει της δυνατότητα στα μέλη της κοινότητας να σχολιάσουν έργα, πέραν της παροχής γενικών χώρων συζητήσεων και χώρων επίδειξης έργων. Προγράμματα που έχουν αναπτυχθεί σε Scratch μπορούν να εκτελούνται είτε στο περιβάλλον ανάπτυξης ή μέσω μιας μικροεφαρμογής Java γνωστής ως Scratch Player. Ο Scratch Player επιτρέπει σε προγράμματα Scratch να εκτελεστούν από σχεδόν οποιαδήποτε εφαρμογή εμφάνισης ιστοσελίδων.

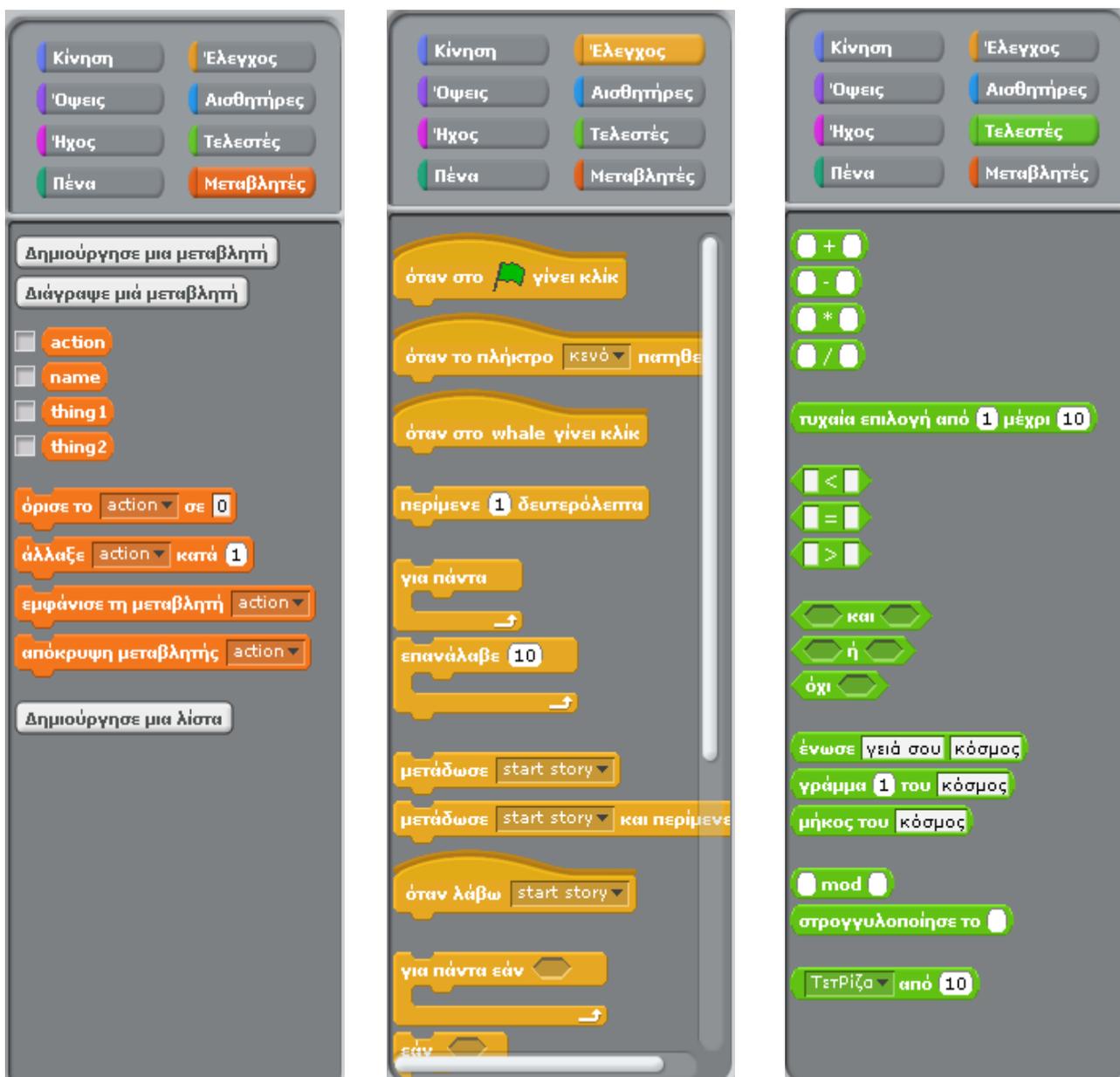
4.2 Δομή του προγράμματος

Ένα πρόγραμμα σε Scratch θυμίζει αρκετά “πάζλ” από κομμάτια κώδικα, τα οποία μπορούν να μετακινηθούν μαζί. Παράδειγμα προγράμματος είναι το ακόλουθο :



Εικόνα 4. Η δομή του προγράμματος Scratch

Όπως παρατηρούμε, το περιβάλλον αποτελείται από τέσσερα τμήματα. Στην αριστερή πλευρά της οθόνης βρίσκεται η **Περιοχή Εντολών**, όπου απεικονίζονται οι βασικές δομές και εντολές. Παραδείγματα αυτών είναι μεταβλητές, δομές ελέγχου, τελεστές, εντολές ήχου – εικόνας – κίνησης – ζωγραφικής , εντολές έναρξης προγράμματος.



Εικόνα 5. Αναπαράσταση μεταβλητών, δομών ελέγχου και τελεστών

Στη συνέχεια υπάρχει η **Περιοχή Σεναρίων**, στην οποία μπορούμε να “σύρουμε” (“*drag and drop*”) εντολές και να δημιουργήσουμε σενάρια, δηλαδή προγράμματα. Πρόκειται ουσιαστικά για τον χώρο όπου γίνεται η σύνθεση του κώδικα με χρήση έτοιμων εντολών. Τα σενάρια αποτελούνται από αντικείμενα (*sprites*) και καθορίζουν τη συμπεριφορά τους.



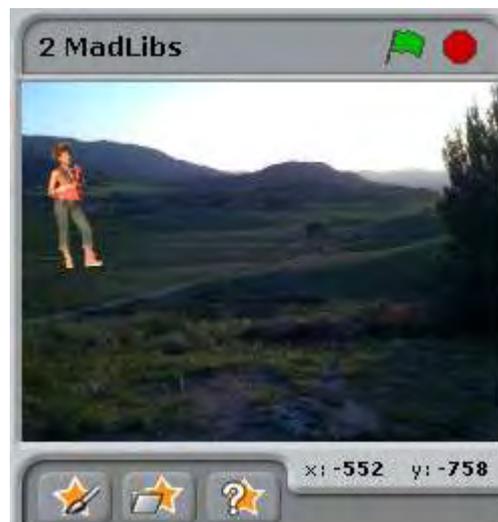
Εικόνα 6. Περιοχή Σεναρίων με τμήματα κώδικα

Στο δεξί κάτω μέρος της οθόνης βρίσκεται η *Λίστα των Μορφών*, όπου παρουσιάζονται οι μικρογραφίες με τα ονόματα των μορφών. Ως μορφή θεωρείται η απεικόνιση κάθε τμήματος κώδικα.



Εικόνα 7. Λίστα Μορφών για κάθε τμήμα κώδικα

Τέλος, πάνω από τη λίστα των μορφών, βρίσκεται η **Σκηνή**, ο χώρος που αλληλεπιδρούν τα σενάρια. Στη σκηνή, τα προγράμματα του χρήστη κατά κάποιο τρόπο “ζωντανεύουν”.



Εικόνα 8. Σκηνή ενός σεναρίου

Για τη δημιουργία ενός νέου προγράμματος, ο χρήστης θα πρέπει να επιλέξει από τη γραμμή επιλογών **Αρχείο Διόρθωσε Μοιράσου Βοήθεια** :

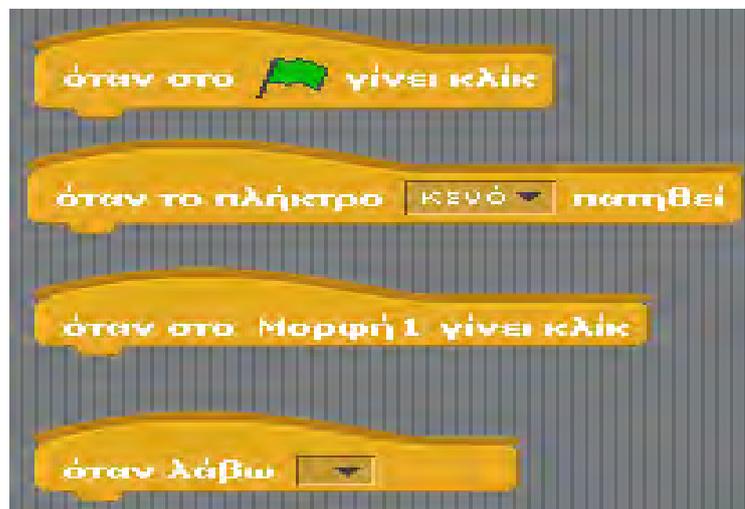
- **Αρχείο**
- **Νέο**

και είναι σε θέση να δημιουργήσει το δικό του σενάριο. Κάθε καινούριο σχέδιο εργασίας διαθέτει ως προκαθορισμένο αντικείμενο τη γάτα. Για τη δημιουργία νέων αντικειμένων, δίνεται η δυνατότητα :

- Να ζωγραφίσει το δικό του αντικείμενο
- Να εισάγει ένα αποθηκευμένο αντικείμενο
- Να επιλέξει ένα τυχαίο αντικείμενο

Στη συνέχεια, επιλέγει από την περιοχή εντολών το σύνολο αυτών που θα χρησιμοποιήσει στο πρόγραμμά του και να το αποθηκεύσει. Τέλος υπάρχει η δυνατότητα διόρθωσης του προγράμματος, η επιλογή βοήθειας καθώς και η απευθείας μετάβαση στην κοινότητα του Scratch, όπου κάποιος μπορεί να μοιραστεί τη δημιουργία του (επιλογή **Μοιράσου**) με τα υπόλοιπα μέλη της κοινότητας.

Η εκκίνηση κάθε προγράμματος Scratch γίνεται με ένα από τα παρακάτω block :



Εικόνα 9. Block εντολών εκκίνησης

4.3 Παιδαγωγικά οφέλη του περιβάλλοντος Scratch

Το Scratch χρησιμοποιείται παγκοσμίως σε διάφορα σχολεία και εκπαιδευτικούς οργανισμούς. Έχει κατασκευαστεί με τέτοιο τρόπο ώστε να είναι εύκολη η εφαρμογή του *Κύκλου Μάθησης*, που σύμφωνα με τον Resnick προάγει την δημιουργική σκέψη.



Εικόνα 10. Κύκλος Μάθησης

Σύμφωνα με τον κύκλο μάθησης, ο μαθητής *φαντάζεται* μια εργασία (imagine), ξεκινά να *δημιουργεί* την σύνθεσή της (create), *πειραματίζεται* με αυτή (play) και τη *μοιράζεται* με άλλους (share). Από την ανατροφοδότηση που λαμβάνει μπορεί να *αναστοχαστεί* (reflect) και να σκεφτεί κάτι νέο ώστε να συνεχιστεί ο κύκλος (Resnick, 2007).

Το περιβάλλον χρησιμοποιείται σαν μια πρώτη επαφή των αρχαρίων μαθητών/σπουδαστών με τον προγραμματισμό, την εκμάθηση προχωρημένων εννοιών, την εισαγωγή τους σε επαγγελματικές γλώσσες

προγραμματισμού, όπως π.χ. η Java και τη διδασκαλία εργαστηριακών μαθημάτων στην Πρωτοβάθμια και Δευτεροβάθμια Εκπαίδευση.

Κάποια από τα πλεονεκτήματα του είναι :

- Ευκολία στη χρήση από άτομα χωρίς κάποια προγραμματιστική εμπειρία
- Περιβάλλον γραμμένο σε πολλές γλώσσες (και στα Ελληνικά)
- Αναπαραστάσεις από την καθημερινή ζωή
- Δυνατότητα στο χρήστη να συνθέσει μια προσωπική δημιουργία (personalization) (προσθήκη δικών τους φωτογραφιών, μουσικών κομματιών κ.α.)
- Δυνατότητα διαμοιρασμού των εργασιών (share)

Παιδαγωγικά Οφέλη

Ανάπτυξη της κριτικής σκέψης : κατά τον προγραμματισμό οι μαθητές έρχονται σε επαφή με ευφυή συστήματα, γεγονός αρκετά σημαντικό στον σημερινό διαρκώς μεταβαλλόμενο κόσμο.

Δημιουργικότητα : οι μαθητές, δεν λύνουν απλά ένα καθορισμένο πρόβλημα αλλά μαθαίνουν να αντιμετωπίζουν οποιοδήποτε πρόβλημα προκύψει.

Συνεργατική μάθηση : οι μαθητές μπορούν να μοιραστούν τις δημιουργίες τους με άλλους, να επικοινωνήσουν, να λύσουν τυχών απορίες, να μάθουν από άλλες δημιουργίες.

Υπευθυνότητα : οι μαθητές, γνωρίζοντας πως οι δημιουργίες τους μπορεί να απευθυνθούν και σε άλλους, αποκτούν υπεύθυνο τρόπο σκέψης και υλοποίησης.

Διερεύνηση και επίλυση προβλημάτων : οι μαθητές κατά την αντιμετώπιση ενός προβλήματος, καλούνται να αναγνωρίσουν το πρόβλημα (σύλληψη του αλγορίθμου), να το διαχωρίσουν σε απλούστερα τμήματα και να το υλοποιήσουν με τη χρήση των εργαλείων που τους προσφέρει το περιβάλλον του Scratch.

Ενότητα 5 : Σχέδιο Μαθήματος

Στην παρούσα ενότητα, μελετάται και αναλύεται η διδασκαλία της πληροφορικής και ειδικότερα του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση. Πιο Συγκεκριμένα μελετάται το μάθημα της Γ΄ τάξης Λυκείου Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, η σημασία του σαν εισαγωγικό μάθημα σε εφαρμογές προγραμματισμού, τα οφέλη που αποκομίζουν οι μαθητές, καθώς και παρανοήσεις ή δυσκολίες που παρατηρούνται κατά την εκμάθηση βασικών εννοιών.

Γίνεται χρήση και αξιοποίηση εκπαιδευτικού λογισμικού, που έχει σαν σκοπό όχι να αλλοιώσει τον μέχρι τώρα τρόπο διδασκαλίας του μαθήματος, αλλά να παρέχει μία χρήσιμη εργαστηριακή υποστήριξη του.

Τέλος, παρουσιάζεται το διδακτικό πλάνο το οποίο έχει σαν σκοπό να αναδείξει τα προβλήματα που παρατηρούνται σε βασικές έννοιες και δομές προγραμματισμού χωρίς τη χρήση λογισμικού, και μέσα από προτάσεις, παραδείγματα και ασκήσεις να επιδιώξει να βελτιώσει όσο είναι δυνατό αυτές τις αδυναμίες.

5.1 ΑΕΕΠ: Η σημασία του, Παρανοήσεις, Δυσκολίες

Στην 2^η Ενότητα έγινε πλήρης περιγραφή του μαθήματος, ανάλυση της δομής του, αναφέρθηκαν οι στόχοι καθώς και ο τρόπος διδασκαλία του.

Το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, διδάσκεται στην Γ΄ τάξη Λυκείου, είναι Πανελλαδικώς εξεταζόμενο και έχει σαν σκοπό οι μαθητές να αναπτύξουν αναλυτική και συνθετική σκέψη, να αποκτήσουν ικανότητες μεθοδολογικού χαρακτήρα και να μπορούν να επιλύουν προβλήματα. Η ύπαρξη μαθήματος πληροφορικής που σχετίζεται με την αλγοριθμική σχεδίαση και την επίλυση προβλημάτων, έχει σαν πρωτεύοντα ρόλο τη γνωστική ανάπτυξη των μαθητών.

Χαρακτηρίζεται ως εργαστηριακό μάθημα και διδάσκεται σε δύο διδακτικές ώρες την εβδομάδα. Κατά τη διεξαγωγή του μαθήματος, προτείνεται η διδασκαλία σε αίθουσα για το πρώτο (θεωρητικό) μέρος, αλλά και σε εργαστήριο για τις ενότητες που αναφέρονται στο δεύτερο μέρος όπου γίνεται η ανάπτυξη των προγραμμάτων. Ακολουθείται η χρήση της “ελεύθερης” ψευδογλώσσας και της ιδεατής γλώσσας προγραμματισμού ΓΛΩΣΣΑ. Η τελευταία, ακολουθεί τις αρχές των σύγχρονων γλωσσών προγραμματισμού υψηλού επιπέδου και έχουν αναπτυχθεί διερμηνευτές και μεταγλωττιστές που την υποστηρίζουν.

Σύμφωνα με τους συγγραφείς του προτεινόμενου συγγράμματος στόχος δεν είναι η διδαχή και η εκμάθηση κάποιου συγκεκριμένου προγραμματιστικού περιβάλλοντος, ούτε η καλλιέργεια προγραμματιστικών δεξιοτήτων από τη μεριά των μαθητών, αλλά η δόμηση σκέψης και η διδασκαλία τεχνικών επίλυσης προβλημάτων.

Το μάθημα εκτός από τις γνώσεις και τις ικανότητες που προσφέρει στους μαθητές, κρίνεται ιδιαίτερα σημαντικό και για τους νεοεισερχόμενους φοιτητές και σπουδαστές σε ΑΕΙ και ΤΕΙ αντίστοιχα, τόσο σε τμήματα που σχετίζονται με το αντικείμενο της Πληροφορικής και Η/Υ όσο και σε άλλα τμήματα με διαφορετικό γνωστικό αντικείμενο, που χρησιμοποιούν τον προγραμματισμό σαν βοηθητικό εργαλείο. Αποτελεί μάθημα υποδομής και δόμησης της σκέψης και έχει αποδειχθεί ότι σε πολλά τμήματα απαιτείται η γνώση της αλγοριθμικής σχεδίασης.

“Η διαφοροποίηση μεταξύ των μαθητών που εισέρχονται στην τριτοβάθμια εκπαίδευση με γνώσεις αλγοριθμικής από τους υπόλοιπους, είναι εμφανής και σχολιάζεται ποικιλοτρόπως”. (Κοίλιας Χ. ,2004)

Επιπλέον, η συνεχής ζήτηση σε τμήματα Πληροφορικής, έχει σαν αποτέλεσμα να καθίσταται αναγκαία η γνώση της αλγοριθμικής. Επίσης, σε πολλές σχολές όπως και της Πληροφορικής, τα προγράμματα σπουδών θεωρούν αναγκαίο να διδάσκεται ήδη από το πρώτο εξάμηνο ο προγραμματισμός σαν βασικό μάθημα και μάλιστα σε διάφορες γλώσσες προγραμματισμού.

Ωστόσο, παρά τη σημασία και τη σπουδαιότητα του μαθήματος παρατηρούνται και αρκετές δυσκολίες.

Καταρχήν οι δύο διδακτικές ώρες που προτείνονται για τη διδασκαλία του μαθήματος, θεωρούνται λίγες τόσο για την κάλυψη της διδακτέας ύλης όσο και την εκμάθηση του περιεχομένου του μαθήματος. Σαν μάθημα κατεύθυνσης που εξετάζεται Πανελλαδικά, είναι επιτακτική η ανάγκη για επιπλέον ώρες διδασκαλίας.

Το διδακτικό πακέτο προσδιορίζει τις αλγοριθμικές δομές με ένα είδος “ελευθερίας” στην κωδικοποίηση, σε αντίθεση με το αυστηρό λεξιλόγιο μιας γλώσσας προγραμματισμού. Σαν αποτέλεσμα, μια αλγοριθμική ενέργεια εκφράζεται με περισσότερους από έναν τρόπους. Αυτή η “ελευθερία” παρέχει πολλά πλεονεκτήματα στην αλγοριθμική σχεδίαση, δημιουργεί ωστόσο και προβλήματα. Για παράδειγμα, οι μαθητές μπορούν να εκφράσουν έννοιες, δομές ή ακόμη και προβλήματα με το δικό τους τρόπο και σύνταξη, το οποίο δεν είναι πάντα σωστό. Ερμηνεύουν βασικές έννοιες και δομές με λάθος τρόπο με αποτέλεσμα να δημιουργούνται ασάφειες και παρανοήσεις.

Τέλος, το μάθημα αν και εργαστηριακής φύσης, δεν είναι σύνηθες να χρησιμοποιείται το εργαστήριο σαν κύριος χώρος διδασκαλίας. Δεν διδάσκεται κάποια “εμπορική” γλώσσα προγραμματισμού όπως Basic, Pascal, C, διότι μια τέτοια ενέργεια θα ήταν αντίθετη με τους στόχους του μαθήματος αλλά και θα επιβάρυνε τους καθηγητές με τη διδασκαλία μιας γλώσσας προγραμματισμού, όπως και τους μαθητές με την εκμάθησή της. Επιπλέον, σύμφωνα με έρευνες που έχουν πραγματοποιηθεί, παρατηρείται ότι το μεγαλύτερο ποσοστό των καθηγητών δεν χρησιμοποιεί τα εργαστήρια, αλλά επιμένει στις παραδοσιακές μεθόδους διδασκαλίας στην τάξη, κάνοντας χρήση του πίνακα και χρησιμοποιώντας το χαρτί για την εκτέλεση προγραμμάτων. Ωστόσο, είναι θετικό το γεγονός ότι ένα μικρό έστω ποσοστό καθηγητών, κάνει χρήση του εργαστηρίου χρησιμοποιώντας το πρόγραμμα Γλωσσομάθεια, Διερμηνευτή Γλώσσας ή κάποια άλλη γλώσσα προγραμματισμού.

5.2 Επιλογή Εκπαιδευτικού Λογισμικού

Σε κάθε μάθημα αλγοριθμικής, κρίνεται απαραίτητη η ύπαρξη εργαστηριακού μέρους. Στο εργαστήριο ο μαθητής μπορεί να καταλάβει ότι οι εντολές εκτελούνται σειριακά, να εμπεδώσει βασικές δομές όπως είναι η επιλογή και η επανάληψη, χωρίς να χρειάζεται να αποκτήσει ιδιαίτερες προγραμματιστικές δεξιότητες. Ο εκπαιδευτικός οφείλει παράλληλα με την αλγοριθμική σχεδίαση, να μοιράζεται το χρόνο μεταξύ διδασκαλίας με εποπτικά μέσα, όπου θα τεκμηριώνονται, θα “τρέχουν” και θα ελέγχονται οι αλγόριθμοι που έχουν σχεδιαστεί στο χαρτί.

Ο πλέον διαδεδομένος τρόπος για τη διδασκαλία προγραμματισμού στηρίζεται στη χρήση μιας επαγγελματικής γλώσσας και ενός κατάλληλου περιβάλλοντος για την υλοποίηση προγραμμάτων. Αυτού του είδους διδασκαλία, έρχεται αφενός σε αντίθεση με τους στόχους και τον τρόπο διεξαγωγής του μαθήματος ΑΕΠΠ, χαρακτηρίζεται δε και από διάφορα προβλήματα, καθώς τέτοιου τύπου γλώσσες περιλαμβάνουν πολλές εντολές και απαιτούν αρκετές συντακτικές λεπτομέρειες. Αποτέλεσμα των παραπάνω είναι ότι οι μαθητές αναγκάζονται να αφομοιώνουν έναν μεγάλο όγκο πληροφοριών και να ασχολούνται περισσότερο με τεχνικές λεπτομέρειες μιας γλώσσας, παρά με τη χρήση βασικών αρχών προγραμματισμού.

Τα παραπάνω προβλήματα μπορούν να αποφευχθούν καθώς τα τελευταία χρόνια έχουν αναπτυχθεί εκπαιδευτικά λογισμικά για τη διδασκαλία του προγραμματισμού, όπως :

- LegoMindStroms
- Scratch
- MicroWorldsPro
- Γλωσσομάθεια

Για τις ανάγκες του διδακτικού πλάνου, θα γίνει χρήση του εκπαιδευτικού λογισμικού Scratch. Πρόκειται για ένα προγραμματιστικό περιβάλλον που αναπτύχθηκε στο ερευνητικό εργαστήριο του MIT, με σκοπό να ενθαρρύνει τους αρχάριους προγραμματιστές να δημιουργήσουν εύκολα πολυμεσικές εφαρμογές.

Χρησιμοποιώντας το, οι μαθητές εύκολα αποφεύγουν συντακτικά λάθη κατά τη δημιουργία του προγράμματος, ενώ οι εκπαιδευτικοί δίνουν μεγαλύτερη βαρύτητα στην εκμάθηση της λογικής και της δομής του προγράμματος. Διαθέτει γραφική γλώσσα, χάρη στην οποία καθιστά τον προγραμματισμό οικείο σε παιδιά και εφήβους. Επικεντρώνεται στις θεμελιώδεις προγραμματιστικές έννοιες και παρέχει ένα σύνολο βασικών τεχνικών και δομών όπως μεταβλητές, δομή επιλογής, δομή επανάληψης. Τέλος, η ύπαρξη της κοινότητας, ένας χώρος δηλαδή όπου τα έργα μπορούν να μοιράζονται, να αξιολογούνται και να βελτιώνονται, το καθιστούν πιο αποτελεσματικό και σε πλεονεκτικότερη θέση σε σχέση με άλλα εκπαιδευτικά λογισμικά.

Από την εμφάνισή του κιόλας έχει κάνει αισθητή τη συνεισφορά του στην εκπαίδευση, καθώς χρησιμοποιείται για:

- Να εισάγει τους εκπαιδευόμενους σε έννοιες προγραμματισμού
- Να χρησιμοποιηθεί για τη διδασκαλία εργαστηριακών μαθημάτων
- Να βοηθήσει στην ανάπτυξη χρήσης προγραμματιστικών εργαλείων
- Να βοηθήσει τους μαθητές να μεταβούν πιο ομαλά σε πιο σύνθετες γλώσσες προγραμματισμού

Συνοπτικά, σαν πλεονεκτήματα του Scratch μπορούν να θεωρηθούν τα εξής :

- Ευκολία στη χρήση από άτομα με καθόλου ή πολύ μικρή προγραμματιστική εμπειρία
- Υποστήριξη των κυριότερων προγραμματιστικών δομών
- Υποστήριξη εργαλείων αποσφαλμάτωσης (debugging tools)
- Υποστήριξη περιβάλλοντος στα Ελληνικά
- Δυνατότητα διαμοιρασμού (share) των projects των χρηστών, σχολιασμού (comment on it) ή ψηφοφορίας (vote for it) άλλων projects και επικοινωνίας (forum) με άλλους χρήστες μέσω της ιστοσελίδας του Scratch

5.3 Διδακτικό Πλάνο

Στην παρούσα υποενότητα αναπτύσσεται ένα σχέδιο μαθήματος με τη χρήση του λογισμικού Scratch.

Στόχος του σχεδίου δεν είναι να αλλοιώσει τη φύση του μαθήματος και να το επιφορτίσει με επιπλέον όγκο πληροφοριών, αλλά να εντοπίσει τα “αδύνατα” σημεία των μαθητών και με τη συμβολή του λογισμικού να τα βελτιώσει ή και αν είναι εφικτό να τα εξαλείψει.

Χώρος δραστηριότητας θεωρείται το σχολικό εργαστήριο και είναι απαραίτητη η εγκατάσταση του λογισμικού στους υπολογιστές, το οποίο όπως έχει αναφερθεί είναι δωρεάν με σχετικά μικρές απαιτήσεις συστήματος και μνήμης.

Προτείνεται η κοινή χρήση ενός Η/Υ για κάθε δύο μαθητές, καθώς είναι σημαντικό και τα δύο μέλη – συνεργάτες να συμμετέχουν ενεργά στη διαδικασία ανάπτυξης ενός προγράμματος και να είναι εξίσου υπεύθυνα για την επίτευξη του στόχου. Άλλωστε, με τη συνεργασία οι δύο μαθητές αποκτούν ρόλους “οδηγού” και “παρατηρητή” , οι οποίοι εναλλάσσονται και τους βοηθούν να ανταλλάσσουν ιδέες, να προτείνουν εναλλακτικές λύσεις και να επιλύουν προβλήματα που ενδεχομένως σαν μονάδες δεν θα μπορούσαν.

Η πρόταση του σχεδίου είναι να χρησιμοποιηθούν έξι (1) διδακτικές ώρες, με την πρώτη να αποτελεί εισαγωγικό μάθημα για την παρουσίαση και αρχική ενασχόληση με το λογισμικό. Θα καλυφθούν οι ενότητες Μεταβλητή, Δομή Επιλογής, Δομή Επανάληψης, Υποπρογράμματα, Αναδρομή σε μαθήματα μιας διδακτικής ώρας. Είναι στην ευχέρεια να του καθηγητή να επιλέξει το χρόνο που θα διεξαχθούν τα παραπάνω μαθήματα,

ωστόσο προτείνεται να γίνονται στο σημείο όπου έχει καλυφθεί η αντίστοιχη θεωρητική ενότητα από το βιβλίο.

Μάθημα 1^ο : Η Έννοια της Μεταβλητής

Μία μεταβλητή παριστάνει μια ποσότητα που η τιμή της μπορεί να μεταβάλλεται.

Ορισμός : Μεταβλητή ορίζουμε μία ποσότητα που μπορεί να λαμβάνει διαφορετικές τιμές κατά τη διάρκεια εκτέλεσης ενός προγράμματος. Τα χαρακτηριστικά μιας μεταβλητής είναι :

- Το όνομά της
- Ο τύπος της (ένα σύνολο από ιδιότητες)
- Η τιμή της
- Μια αναφορά (η θέση μνήμης που την προσδιορίζει)

Διδακτικός Σκοπός : Σκοπός του σχεδίου μαθήματος είναι να κατανοήσουν οι μαθητές την έννοια της μεταβλητής. Στο τέλος του σχεδίου μαθήματος οι μαθητές θα πρέπει να είναι σε θέση :

- Να διακρίνουν τις μεταβλητές από τις σταθερές
- Να αναγνωρίζουν τους διάφορους τύπους μεταβλητών
- Να γνωρίζουν τι είναι εκχώρηση τιμής σε μεταβλητή
- Να χειρίζονται ειδικές μεταβλητές
- Να δημιουργούν μεταβλητές στο Scratch

Η επιλογή κατάλληλου ονόματος για μια μεταβλητή, διευκολύνει στην κατανόηση της έννοιας ως ονομασία μιας θέσης μνήμης. Όταν χρησιμοποιούμε ένα όνομα διευκολύνεται η αναγνωσιμότητα ενός προγράμματος. Για παράδειγμα :

Μεταβλητή που υπολογίζει ένα άθροισμα

$$S = S + X$$

ή καλύτερα...

$$\text{Sum} = \text{Sum} + X$$

Μεταβλητή που αποθηκεύεται ένα όνομα

$N = \text{"George"}$

ή καλύτερα...

$\text{Name} = \text{"George"}$

Το αποτέλεσμα ενός αγώνα μπάσκετ, το ύψος ενός ανθρώπου, η θερμοκρασία είναι παραδείγματα καταστάσεων που μεταβάλλονται. Για να μπορούμε να χειριζόμαστε αυτές τις μεταβαλλόμενες τιμές, πρέπει με κάποιο τρόπο να τις κρατάμε κάπου, ώστε να τις έχουμε κάθε στιγμή στη διάθεσή μας. Για παράδειγμα χρησιμοποιούμε τη μεταβλητή Score, στην οποία κάθε στιγμή αποθηκεύουμε το αποτέλεσμα ενός αγώνα για να βρούμε το νικητή.

Από την άλλη, οι σταθερές είναι ποσότητες που η τιμή τους είναι κάθε στιγμή η ίδια, π.χ. $\pi = 3,14\dots$

Μεταβλητή : η τιμή της *αλλάζει*

Σταθερά : η τιμή της είναι *πάντα η ίδια*

Ένα σφάλμα στο οποίο υποπίπτουν οι μαθητές είναι πως πιστεύουν ότι μια μεταβλητή συγκρατεί στη μνήμη της όλες τις τιμές που της δίνουν.

Δεν κατανοούν ότι η κάθε νέα τιμή αντικαθιστά την παλιά, η οποία τελικά χάνεται. Το πρόβλημα αυτό οφείλεται στο γεγονός ότι οι μαθητές δεν γνωρίζουν πως μοιάζει η μνήμη ενός υπολογιστή και ποια η θέση των μεταβλητών μέσα σε αυτή.

Ας υποθέσουμε ότι η μνήμη ενός Η/Υ είναι μία τράπεζα που περιέχει θυρίδες (θέσεις μνήμης). Κάθε θυρίδα (μεταβλητή) μπορεί να είναι άδεια ή να περιέχει ένα χρηματικό ποσό. Μπορούμε να προσθέσουμε ή να αφαιρέσουμε χρήματα στην θυρίδα, οπότε το ποσό που περιέχει αλλάζει (άρα αλλάζει και η τιμή της μεταβλητής).

Μία μεταβλητή χαρακτηρίζεται και από έναν τύπο, χαρακτηριστικό το οποίο δεν αλλάζει από τη στιγμή που ορίζεται αυτή. Ο τύπος μιας μεταβλητής καθορίζει τι είδους δεδομένα μπορεί να έχει αυτή, π.χ. ακέραιους αριθμούς, πραγματικούς αριθμούς, χαρακτήρες, λογικές τιμές, συμβολοσειρές αλλά και τις δυνατές πράξεις που μπορούν να πραγματοποιηθούν από τις αντίστοιχες μεταβλητές. Όταν ορίζεται μια μεταβλητή πρέπει να οριστεί και ο τύπος της.

Για να δώσουμε τιμή σε μία μεταβλητή κατά τη διάρκεια εκτέλεσης ενός προγράμματος χρησιμοποιούμε την εντολή εκχώρησης. Ωστόσο η εντολή αυτή συνιστά βασικό διδακτικό εμπόδιο στον προγραμματισμό. Οι μαθητές, οι οποίοι διδάσκονται για πρώτη φορά τις μεταβλητές στα μαθηματικά, συγχέουν πολλές φορές την εντολή εκχώρησης με το σύμβολο της ισότητας “ = ”. Θα πρέπει να γίνει κατανοητό πως η εκχώρηση τιμής και η ισότητα είναι δυο διαφορετικές έννοιες, άσχετα με το συμβολισμό της κάθε γλώσσας προγραμματισμού. Σε χρήση ψευδοκώδικα, είναι

προτιμότερο να χρησιμοποιούνται πιο ουδέτεροι συμβολισμοί, όπως για παράδειγμα “ \leftarrow ” (Du Boulay, 1986).

```
Ανάθεση τιμής σε μεταβλητή  
X = 5  
ή  
X := 5  
ή  
X  $\leftarrow$  5
```

Σαν ειδικές μεταβλητές μπορούν να θεωρηθούν ο *άθροιστής* ($\text{Sum} \leftarrow \text{Sum} + N$) και ο *μετρητής* ($\text{Counter} \leftarrow \text{Counter} + 1$). Πρόκειται για μεταβλητές στο εσωτερικό ενός προγράμματος που επιτελούν συγκεκριμένο υπολογιστικό ρόλο, ανεξάρτητα από το συνολικό πρόβλημα. Στην περίπτωση αυτή παρουσιάζονται δύο διδακτικά προβλήματα. Το πρώτο αφορά τη μη αρχικοποίηση τιμών στις μεταβλητές αυτές. Η αρχικοποίηση απαιτεί τη διατύπωση της υπόθεσης μιας αρχικής κατάστασης του συστήματος. Για παράδειγμα σε ένα πρόβλημα που ζητείται να βρεθεί το άθροισμα από 1 έως 10, θα πρέπει αρχικά η μεταβλητή άθροισμα να πάρει την τιμή 0 ($\text{Sum} = 0$). Το δεύτερο έγκειται στη σύγχυση της χρήσης μίας μεταβλητής δύο φορές μέσα στην ίδια έκφραση.

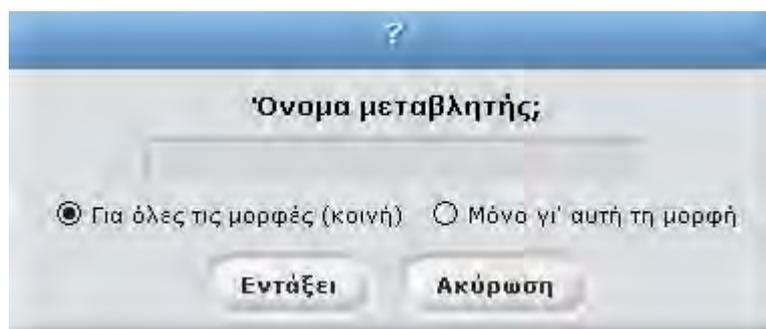
```
Το άθροισμα σε δύο εκδοχές  
Sum  $\leftarrow$  Sum + N  
ή  
Temp  $\leftarrow$  Sum + N  
Sum  $\leftarrow$  Temp
```

Μεταβλητές στο Scratch

Για τη δημιουργία μιας μεταβλητής στο Scratch, επιλέγουμε από την παλέτα **Μεταβλητές** την εντολή **Δημιούργησε μία μεταβλητή**.



Στην περίπτωση αυτή το πρόγραμμα ζητάει το όνομα της μεταβλητής.



Επιλέγεις αν η μεταβλητή θα είναι ορατή από όλα τα αντικείμενα (Για όλες τις μορφές) ή από το επιλεγμένο αντικείμενο (Μόνο γι' αυτή τη μορφή).

Δίνοντας το όνομα **score** , παρατηρούμε ένα νέο σύνολο εντολών που παρουσιάζεται :



- Όνομα μεταβλητής
- Αρχικοποίηση της μεταβλητής στην τιμή 0
- Μεταβολή της τιμής της μεταβλητής κατά 1
- Εμφάνιση της μεταβλητής
- Απόκρυψη της μεταβλητής

Μάθημα 2^ο : Η δομή Επιλογής

Η δομή επιλογής αποτελεί μια από τις βασικές λογικές δομές του προγραμματισμού και βρίσκεται σε όλα τα προγραμματιστικά περιβάλλοντα.

Διδακτικός Σκοπός : Σκοπός του σχεδίου μαθήματος είναι να κατανοήσουν οι μαθητές την δομή επιλογής. Στο τέλος του σχεδίου μαθήματος οι μαθητές θα πρέπει να είναι σε θέση :

- Να μπορούν να ελέγξουν μια συνθήκη
- Να γνωρίζουν τις τιμές μιας συνθήκης
- Να γνωρίζουν τους τύπους της δομής επιλογής
- Να γνωρίζουν και να χειρίζονται το περιεχόμενο μιας συνθήκης
- Να δημιουργούν προγράμματα που περιέχουν δομές επιλογής (στο χαρτί και στο Scratch)

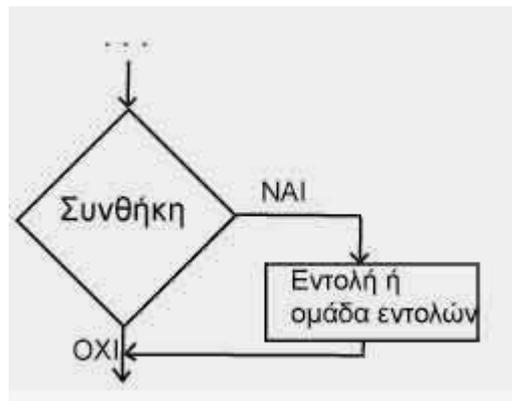
Η λειτουργία της δομής επιλογής είναι ο έλεγχος μιας συνθήκης, δηλαδή μιας λογικής έκφρασης η οποία παίρνει τις εξής τιμές :

- **Αληθής** (True)
- **Ψευδής** (False)

Μία συνθήκη περιέχει τελεστές (αριθμητικούς, συγκριτικούς, λογικούς) ή μια έκφραση. Όταν η συνθήκη είναι Αληθής, εκτελείται μια εντολή (ή ένα σύνολο εντολών) , ενώ όταν είναι Ψευδής εκτελείται μια **άλλη** εντολή (ή ένα **άλλο** σύνολο εντολών). Μπορούμε να διακρίνουμε τη δομή επιλογής σε :

- Απλή
- Σύνθετη (διπλή)
- Εμφωλευμένες δομές επιλογής (απλή & σύνθετη)
- Πολλαπλή

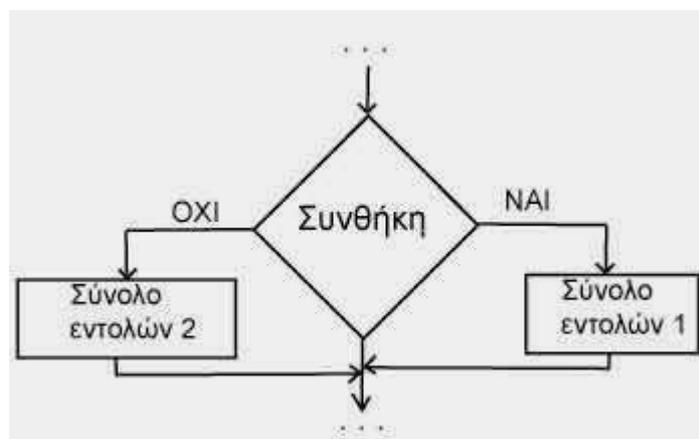
Απλή δομή επιλογής



Αν η συνθήκη είναι **αληθής**, εκτελούνται οι εντολές που βρίσκονται **μέσα στη δομή**.

Αν η συνθήκη είναι **ψευδής**, εκτελούνται οι εντολές που βρίσκονται **αμέσως μετά το τέλος της δομής**.

Σύνθετη δομή επιλογής

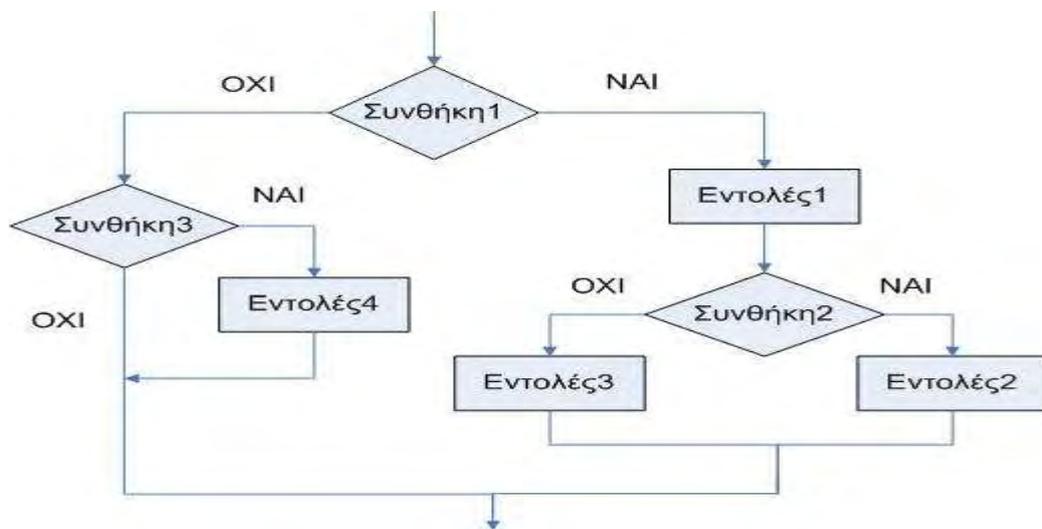


Στη δομή αυτή ελέγχονται δύο περιπτώσεις :

Αν η συνθήκη είναι **αληθής**, εκτελούνται οι εντολές που βρίσκονται **κάτω από τη λέξη τότε**.

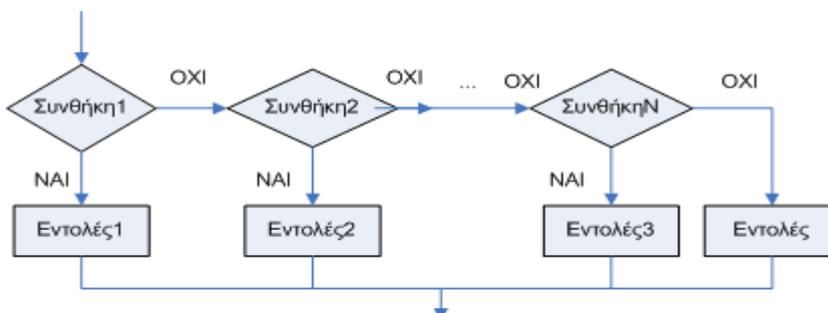
Αν η συνθήκη είναι **ψευδής**, εκτελούνται οι εντολές που βρίσκονται **κάτω από τη λέξη αλλιώς**.

Εμφωλευμένες δομές επιλογής



Οι εμφωλευμένες δομές επιλογής αποτελούν έναν **συνδυασμό** της απλής και της σύνθετης δομής επιλογής.

Πολλαπλή δομή επιλογής



Στην πολλαπλή δομή επιλογής ελέγχονται περισσότερες από δύο περιπτώσεις, ανεξάρτητες μεταξύ τους. Αν π.χ. η Συνθήκη1 είναι ψευδής θα ελέγξουμε τη Συνθήκη2, που αν και αυτή με τη σειρά της είναι ψευδής θα ελέγξουμε τη Συνθήκη3 κ.ο.κ. Αν για κάποια Συνθήκη ισχύει ότι είναι αληθής, θα εκτελεσθεί το σύνολο εντολών που την ακολουθεί (Συνθήκη1 → Εντολές1) και εκεί τελειώνει οριστικά η δομή.

Συνηθισμένα λάθη και δυσκολίες

Η δομή επιλογής προκαλεί αρκετές δυσκολίες στους μαθητές, τόσο κατά τη διδασκαλία όσο και ως προς την κατανόησή της. Οι μαθητές έχουν σαν παράδειγμα την σειριακή (ακολουθιακή) εκτέλεση ενός προγράμματος, όπου υπάρχει μια αλληλουχία εντολών. Αυτή η αλληλουχία διακόπτεται όταν υπάρχει μια δομή ελέγχου καθώς ακολουθείται μόνο μια από τις δυνατές καταστάσεις. Κάποια λάθη που παρατηρούνται είναι :

- Αν η συνθήκη είναι ψευδής, τότε πηγαίνουμε στην αρχή του προγράμματος
- Αν η δομή περιέχει και το “Τότε” και το “Αλλιώς” , θα εκτελεστούν και οι δύο περιπτώσεις

Ένα ακόμα σημείο δυσκολίας αποτελεί και η φύση μιας συνθήκης, όταν για παράδειγμα γίνεται χρήση Boolean συναρτήσεων ή συνδιασμοί προτάσεων με τους λογικούς τελεστές Ή , ΚΑΙ , ΟΧΙ (or, and, not).

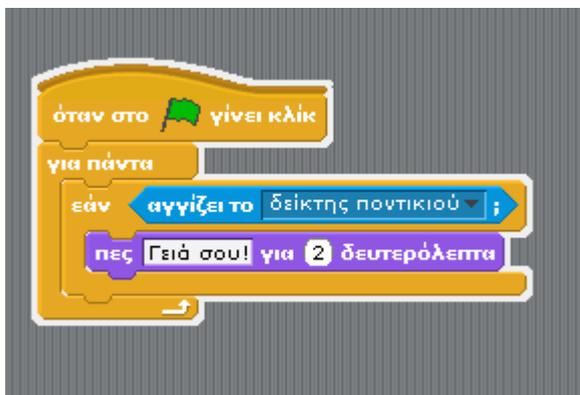
Τέλος, οι εμφωλευμένες δομές επιλογής, αποτελούν σημείο παρανοήσεων με αποτέλεσμα να αυξάνεται ο βαθμός δυσκολίας ανάλογα με το πόσες δομές εμπεριέχονται η μια μέσα στην άλλη.

Η δομή επιλογής στο Scratch

Η εντολή Αν στο Scratch ονομάζεται “Εάν” . Από την παλέτα Έλεγχος μπορούμε να επιλέξουμε εάν, για απλή δομή ή εάν – αλλιώς για σύνθετη δομή. Οι εμφωλευμένες και η πολλαπλή δομή προκύπτουν από το συνδυασμό τους.



1^ο Παράδειγμα : Θέλουμε χρησιμοποιώντας εντολές του Scratch να συμβαίνει το εξής : “*Αν ο δείκτης από ποντίκι αγγίζει τη Γάτα, τότε αυτή να λέει ‘Γειά σου!’*”.



Στην περίπτωση αυτή, η συνθήκη είναι η **αγγίζει το δείκτης ποντικιού ;** και όταν είναι αληθής εκτελείται η εντολή **πες Γειά σου! για 2 δευτερόλεπτα**. Αν η συνθήκη είναι ψευδής δεν συμβαίνει τίποτα.

2^ο Παράδειγμα : Στο προηγούμενο παράδειγμα θέλουμε τις εξής αλλαγές :
“*Αν ο δείκτης του ποντικιού αγγίζει τη Γάτα τότε αυτή να λέει ‘Με αγγίζεις!’*”
αλλιώς να λέει ‘Δεν με αγγίζεις!’”.



Στην περίπτωση αυτή, ενώ η συνθήκη παραμένει η ίδια, έχουμε δύο πιθανά αποτελέσματα.

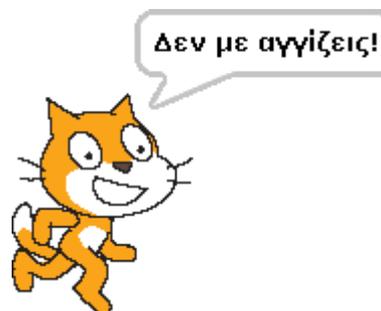
Όταν η συνθήκη είναι αληθής εκτελείται η εντολή

1^η Εντολή →



Όταν η συνθήκη είναι ψευδής εκτελείται η εντολή

2^η Εντολή →

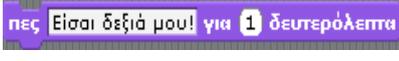


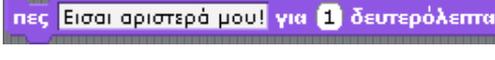
3^ο Παράδειγμα : Στο παράδειγμα αυτό θα δούμε πως λειτουργούν οι εμφωλευμένες δομές επιλογής, αλλά και οι συνθήκες που περιέχουν τελεστές.

“Δημιουργήστε πρόγραμμα που θα ελέγχει τη θέση που έχει ο δείκτης του ποντικιού σε σχέση με τη Γάτα. Συγκεκριμένα τη συντεταγμένη x .”



Το παραπάνω παράδειγμα ίσως φαντάζει πιο δύσκολο. Μία επεξήγηση : Θέλουμε ο δείκτης x να είναι είτε μεγαλύτερος, είτε μικρότερος είτε ίσος με την τιμή 0.

1^η περίπτωση : $x > 0$, άρα η αντίστοιχη συνθήκη είναι . Αν η συνθήκη είναι αληθής θα εκτελεστεί η εντολή . Αν είναι ψευδής τότε ή το $x < 0$ ή το $x = 0$. Έστω ότι ισχύει το πρώτο.

2^η περίπτωση : $x < 0$, άρα η αντίστοιχη συνθήκη είναι . Αν η συνθήκη είναι αληθής θα εκτελεστεί η εντολή . Αν είναι ψευδής τότε επαγωγικά το $x = 0$, καθώς οι άλλες δύο περιπτώσεις έχουν εξεταστεί.

3^η περίπτωση : $x = 0$, όμως σε αυτή την περίπτωση δεν χρειαζόμαστε συνθήκη καθώς βρισκόμαστε στο τμήμα *αλλιώς*, που είναι και η τελευταία επιλογή `πες Είσαι ακριβώς επάνω μου! για 1 δευτερόλεπτα`

Άρα κατά την εκτέλεση των παραπάνω εντολών θα έχουμε:



Μάθημα 3^ο : Η δομή επανάληψης

Η δομή επανάληψης χρησιμοποιείται για την **επαναληπτική εκτέλεση** μίας εντολής ή πολλών εντολών (ακολουθία εντολών), ενός αλγορίθμου ή προγράμματος. Το τμήμα των εντολών που εκτελείται επαναληπτικά ονομάζεται **βρόχος**. Ο βρόχος μπορεί να εκτελείται από καμία έως και άπειρες φορές. Στην περίπτωση που εκτελείται άπειρες φορές, ονομάζεται **ατέρμων βρόχος**, δηλαδή δεν τερματίζει ποτέ. Το πλήθος των επαναλήψεων καθορίζεται από τη **συνθήκη της δομής**.

Διδακτικός Σκοπός : Σκοπός του σχεδίου μαθήματος είναι να κατανοήσουν οι μαθητές την έννοια της επανάληψης. Συγκεκριμένα :

- Να γνωρίζουν τι είναι βρόχος
- Πόσες φορές θα εκτελεστεί ένα βρόχος
- Τι είναι ατέρμων βρόχος
- Τις μορφές/τύπους της δομής επανάληψης
- Να χειρίζονται επαναληπτικές δομές στο Scratch

Υπάρχουν τρεις μορφές αναπαράστασης για τη δομή επανάληψης, οι οποίοι σύμφωνα με το σχολικό εγχειρίδιο είναι οι εξής :

1. Όσο ... Επανάλαβε
2. Επανάλαβε ... Μέχρις _Ότου
3. Για ... Από ... Μέχρι ... Με βήμα

Σε μια σύγχρονη γλώσσα προγραμματισμού αντίστοιχα έχουμε :

1. **While**
2. **Do ... While**
3. **For**

Στην πρώτη περίπτωση ο βρόχος επαναλαμβάνεται όσο η συνθήκη ελέγχου είναι **Αληθής**. Όταν γίνει **Ψευδής** σταματά η επανάληψη και το πρόγραμμα πηγαίνει στην **επόμενη εντολή** (εκτός βρόχου).

Στη δεύτερη περίπτωση ο βρόχος θα εκτελεστεί σίγουρα μία φορά και θα εξακολουθεί να εκτελείται μέχρι να ικανοποιηθεί η συνθήκη, να γίνει δηλαδή **Αληθής** (όσο η συνθήκη είναι **Ψευδής** επαναλαμβάνεται ο βρόχος). Όταν συμβεί αυτό σταματά η επανάληψη και το πρόγραμμα πηγαίνει στην **επόμενη εντολή** (εκτός βρόχου).

Στην τρίτη περίπτωση δίνεται μια αρχική τιμή, μία συνθήκη και το βήμα (ένας μετρητής δηλαδή). Η αρχική τιμή χρησιμοποιείται πριν την έναρξη της επαναληπτικής διαδικασίας. Όσο η συνθήκη είναι Αληθής εκτελείται το τμήμα εντολών που ακολουθεί, αυξάνεται το βήμα (μπορεί να είναι κατά 1) και η διαδικασία επαναλαμβάνεται. Όταν η συνθήκη γίνει **Ψευδής** σταματά η επανάληψη και το πρόγραμμα πηγαίνει στην **επόμενη εντολή** (εκτός βρόχου).

Συνηθισμένα λάθη και δυσκολίες

Παρ' όλη τη σπουδαιότητα της δομής επανάληψης, παρατηρούνται αρκετά διδακτικά προβλήματα.

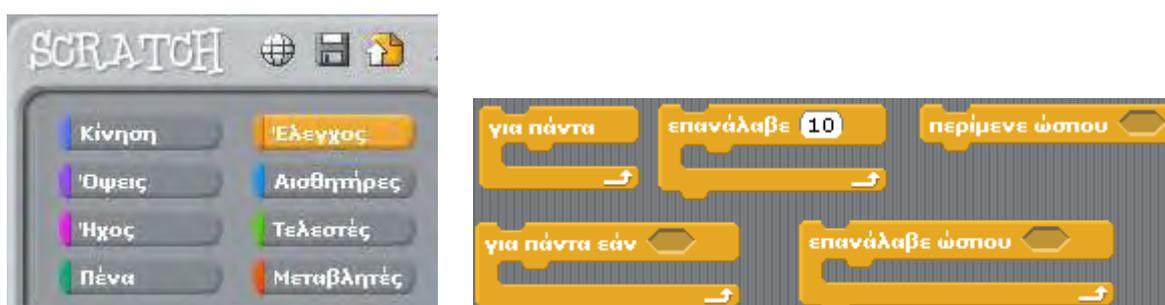
Κύριο πρόβλημα αποτελεί η συνθήκη ελέγχου. Πολλές φορές οι μαθητές δεν είναι σε θέση να διακρίνουν πότε ένας βρόχος θα εκτελεστεί, πότε δεν θα εκτελεστεί καθόλου και πότε θα τερματίσει. Η φύση της συνθήκης ελέγχου (αν για παράδειγμα πρόκειται για μια σύνθετη λογική έκφραση), όπως και στη δομή επιλογής είναι ένα επιπλέον πρόβλημα.

Κατά την αρχικοποίηση ενός βρόχου, πολλοί μαθητές δυσκολεύονται σε περιπτώσεις όπου οι μεταβλητές παίζουν το ρόλο του αθροιστή ή του μετρητή.

Τέλος, η αναπροσαρμογή της τιμής μιας μεταβλητής εντός βρόχου, αποτελεί περίπτωση όπου πολλοί (αρχάριοι κυρίως) την παραλείπουν, καθώς δεν μπορούν να την προσδιορίσουν επαρκώς.

Η δομή επιλογής στο Scratch

Από την παλέτα Έλεγχος έχουμε πέντε δυνατές επιλογές για τη δημιουργία δομής επανάληψης : **για πάντα**, **επανάλαβε**, **για πάντα εάν**, **περίμενε ώσπου**, **επανάλαβε ώσπου**.



1^ο Παράδειγμα : Στο παράδειγμα αυτό θα δείξουμε πως μπορούμε να δημιουργήσουμε επαναληπτικές δομές.

Εντολή για πάντα



Στο παράδειγμα αυτό παρατηρούμε ότι η συνθήκη θα ισχύει **για πάντα**, επομένως οι εντολές στο εσωτερικό της επανάληψης (βρόχος) θα εκτελούνται πάντα.

Εντολή επανέλαβε

Τώρα θέλουμε οι ίδιες εντολές να εκτελούνται όχι για πάντα όπως πριν, αλλά για **συγκεκριμένο** αριθμό φορών.



Εντολή για πάντα εάν

Η εντολή αυτή είναι η αντίστοιχη **Όσο ... Επανέλαβε**. Επομένως, όσο η συνθήκη που δίνεται είναι Αληθής θα εκτελούνται οι εντολές στο εσωτερικό της επανάληψης.

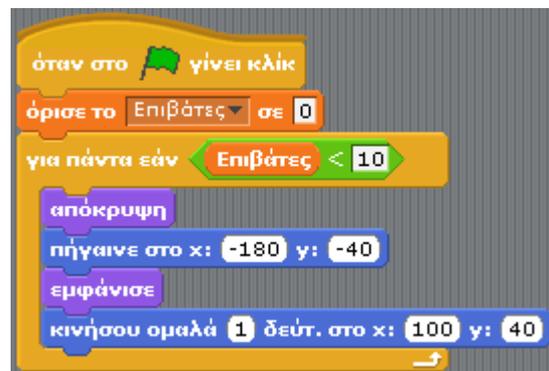


Όσο η συνθήκη είναι αληθής, η μορφή θα κινηθεί μέχρι το προκαθορισμένο σημείο. Όταν γίνει ψευδής, ο βρόχος τερματίζει και θα εκτελεστούν οι εντολές που ακολουθούν μετά από αυτόν. Εφόσον στο παράδειγμα δεν έχουμε κάποια εντολή μετά το τέλος του βρόχου, η μορφή θα σταματήσει.

2^ο Παράδειγμα : Στο παράδειγμα αυτό θα επιχειρήσουμε να επιλύσουμε ένα πιο σύνθετο πρόβλημα και να το αντιστοιχίσουμε στον τρόπο επίλυσης του βιβλίου.

“Εστω ένα ιστιοφόρο το οποίο έχει 10 κενές θέσεις. Θέλουμε να επιβιβαστούν σε αυτό μόνο 10 επιβάτες.”

Ο κώδικας για το παραπάνω παράδειγμα είναι ο εξής :



Αρχικοποίηση βρόχου : Επιβάτες ← 0 

Συνθήκη ελέγχου : Επιβάτες < 10 

Εντολή που εκτελείται : Επιβίβαση 



Στο παραπάνω παράδειγμα παρατηρούμε ότι το πρόγραμμα ποτέ δεν σταματά. Θα έπρεπε να είχε σταματήσει όταν μέσα στο ιστιοφόρο είχαν επιβαστεί συνολικά 10 επιβάτες. Επίσης η τιμή επιβάτες έχει πάντα την τιμή 0 .

Που είναι το λάθος ;

Το πρόβλημα εκτελείται ξανά και ξανά, δηλαδή έχει δημιουργηθεί, όπως αναφέρθηκε και στην αρχή του μαθήματος, **Ατέρμων Βρόχος**.

Επομένως το πρόγραμμα θα πρέπει να διορθωθεί. Παρατηρούμε ότι δεν μετράμε κάθε φορά πόσοι επιβάτες έχουν επιβαστεί στο πλοίο. Άρα χρειάζεται να μετράμε κάθε φορά το πλήθος των επιβατών. Αυτό μπορεί να συμβεί αν κάθε φορά που επιβιβάζεται ένας καινούριος επιβάτης, η μεταβλητή επιβάτες να αλλάζει κατά 1.

επιβάτες ← επιβάτες + 1 .

Με τις διορθώσεις ο κώδικας γίνεται :



και οι αντίστοιχες εντολές θα είναι:

Αρχικοποίηση βρόχου : $\text{Επιβάτες} \leftarrow 0$



Συνθήκη ελέγχου : $\text{Επιβάτες} < 10$



Εντολή που εκτελείται : Επιβίβαση



Μετρητής : $\text{Επιβάτες} \leftarrow \text{Επιβάτες} + 1$



Τώρα το πρόγραμμα τερματίζει και ο συνολικός αριθμός επιβατών θα είναι $\text{Επιβάτες} = 10$.



Μάθημα 4^ο : Τμηματικός Προγραμματισμός

Τμηματικός προγραμματισμός ονομάζεται η τεχνική σχεδίασης και ανάπτυξης των προγραμμάτων ως ένα σύνολο από απλούστερα τμήματα προγραμμάτων.

Διδακτικός Σκοπός : Σκοπός του σχεδίου μαθήματος είναι να κατανοήσει ο μαθητής την έννοια και το λόγο ύπαρξης τμηματικού προγραμματισμού.
Συγκεκριμένα :

- Να αναλύει ένα σύνθετο πρόγραμμα σε απλούστερα υποπρογράμματα
- Να διακρίνει τις διαδικασίες από τις συναρτήσεις
- Να κατανοήσει την έννοια της αναδρομής
- Να συντάσσει αναδρομικά υποπρογράμματα και να συγκρίνει αναδρομικές και επαναληπτικές διαδικασίες

Όταν έχουμε να ασχοληθούμε με απλά προβλήματα, μπορούμε να τα επιλύσουμε χρησιμοποιώντας ένα απλό πρόγραμμα. Στην πράξη, τα προβλήματα είναι αρκετά πιο σύνθετα, οπότε επιλύοντάς τα όλα μαζί, δημιουργείται ένα πρόγραμμα αρκετά μεγάλο σε όγκο και δύσκολο σε κατανόηση. Πρόσθετα υπάρχει και ο κίνδυνος να δημιουργηθούν λάθη, τα οποία πολλές φορές είναι δύσκολο να εντοπιστούν.

Η λύση είναι να διαιρέσουμε πρόβλημα σε μικρότερα και απλούστερα υποπροβλήματα, τα οποία θα έχουν ευκολότερη σχεδίαση και στον αλγόριθμο και στη λύση.

Η τεχνική του τμηματικού προγραμματισμού αποτελεί ένα από τα βασικότερα συστατικά του δομημένου προγραμματισμού, ο οποίο εξασφαλίζει σε μεγάλο βαθμό την επιτυχή και εύκολη δημιουργία σωστών προγραμμάτων.

Υπάρχουν δύο ειδών υποπρογραμμάτων, οι διαδικασίες και οι συναρτήσεις. Οι διαδικασίες μπορούν να εκτελέσουν οποιαδήποτε λειτουργία από αυτές που μπορεί να εκτελέσει ένα πρόγραμμα. Αντίθετα μια συνάρτηση μπορεί να υπολογίσει μόνο μια τιμή, αριθμητική, λογική ή χαρακτήρα και μόνο αυτή επιστρέφει στο πρόγραμμα που την κάλεσε.

Κάποια από τα βασικά πλεονεκτήματα του τμηματικού προγραμματισμού είναι ότι:

- Διευκολύνει την ανάπτυξη ενός αλγορίθμου και του αντίστοιχου προγράμματος
- Διευκολύνει την κατανόηση και διόρθωση του προγράμματος
- Απαιτεί λιγότερο χρόνο στην προσπάθεια στη συγγραφή ενός προγράμματος
- Επεκτείνει τη δυνατότητα μιας γλώσσας προγραμματισμού, καθώς τμήματα κώδικα μπορούν να χρησιμοποιηθούν σε διάφορα προβλήματα

Υποπρογράμματα στο Scratch

Για να δημιουργήσουμε υποπρογράμματα στο Scratch, από την παλέτα Έλεγχος μπορούμε να επιλέξουμε :

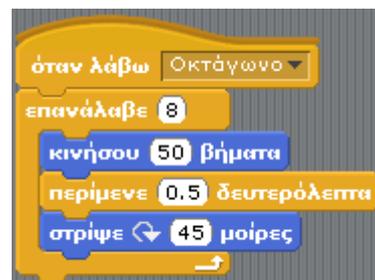
-  για τη δημιουργία ενός υποπρογράμματος και τη δήλωση του ονόματός του
-  για την κλήση ενός υποπρογράμματος
-  για την κλήση ενός υποπρογράμματος

Παράδειγμα : “Θέλουμε να δημιουργήσουμε στο Scratch πολλά οκτάγωνα στη σειρά”.

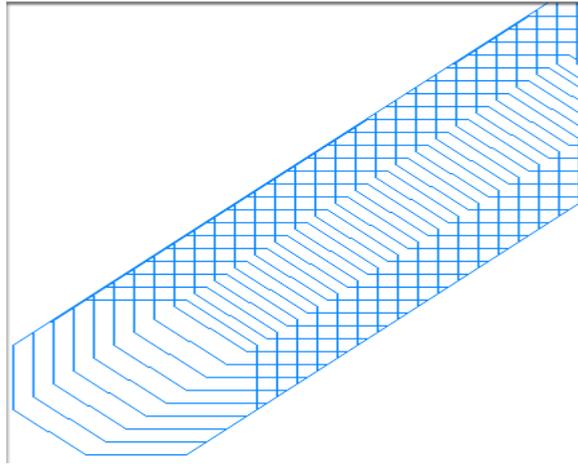
Θα δημιουργήσουμε ένα υποπρόγραμμα με όνομα *Οκτάγωνο*, το οποίο ζωγραφίζει ένα οκτάγωνο και θα το καλούμε από το πρόγραμμα όσες φορές θέλουμε ανάλογα με το πόσα οκτάγωνα θα δημιουργήσουμε.



Κύριο πρόγραμμα



Υποπρόγραμμα Οκτάγωνο



Αποτέλεσμα

Η έννοια της αναδρομής

Ένα υποπρόγραμμα καλείται από το κύριο πρόγραμμα ή κάποιο άλλο υποπρόγραμμα. Υπάρχει όμως και η δυνατότητα ένα υποπρόγραμμα να καλεί τον εαυτό του. Η δυνατότητα αυτή καλείται **αναδρομή**.

Η αναδρομή υποστηρίζεται από όλες τις γλώσσες προγραμματισμού και σαν έννοια μοιάζει με την δομή επανάληψης, είναι όμως πιο περιληπτική σαν έννοια και πιο “κομψή”, διότι μειώνει κατά πολύ το μέγεθος ενός κώδικα. Για το λόγο αυτό χρησιμοποιείται από έμπειρους προγραμματιστές.

Παρ’ όλα αυτά δημιουργεί αρκετά διδακτικά προβλήματα. Οι μαθητές δυσκολεύονται να κατανοήσουν τη ροή εκτέλεσης ενός προβλήματος με αναδρομή και για το λόγο αυτό προτιμούν να χρησιμοποιήσουν έναν αντίστοιχο αλγόριθμο με επαναληπτική δομή. Επίσης, μια λάθος εκτέλεση προγράμματος με αναδρομή, εκτός ότι καταναλώνει αρκετούς πόρους, μπορεί να οδηγήσει και σε ξαφνικό τερματισμό της εκτέλεσης του προγράμματος.

Οι κύριες δυσκολίες που παρατηρούνται είναι οι εξής :

1. Οι μαθητές δυσκολεύονται να διακρίνουν το ρόλο της εντολής τερματισμού της αναδρομικής διαδικασίας. Συγκεκριμένα δεν μπορούν να προβλέψουν σωστά ποια θα είναι η συνθήκη τερματισμού και σε ποιο σημείο του κώδικα θα βρίσκεται αυτή.
2. Γενικά δεν χρησιμοποιούν με σωστή σειρά τις εντολές που περιέχονται στα δύο σκέλη μιας αναδρομικής διαδικασίας.

Με ένα παράδειγμα στο Scratch θα προσπαθήσουμε να διακρίνουμε τα σημεία όπου παρουσιάζονται δυσκολίες και να τα βελτιώσουμε.

Παράδειγμα : “Δοκιμάστε το προηγούμενο παράδειγμα με τα οκτάγωνα με αναδρομικό τρόπο. Συγκεκριμένα να σταματά η διαδικασία να εκτελείται όταν σχηματιστούν 20 οκτάγωνα”.

Αρχικά πρέπει να δημιουργήσουμε ένα υποπρόγραμμα που θα σχεδιάζει ένα οκτάγωνο και το ονομάζουμε **Οκτάγωνο**. Αυτό θα είναι περίπου όπως στο προηγούμενο παράδειγμα. Θα τροποποιήσουμε αυτό το υποπρόγραμμα έτσι ώστε να δημιουργεί 20 οκτάγωνα τώρα. Επομένως θα χρειαστούμε ένα μετρητή για να μετράμε το πλήθος των οκταγώνων. Ονομάζουμε αυτό το μετρητή **Counter** (αρχικά θα είναι 0).

Δημιουργούμε αρχικά το πρώτο οκτάγωνο. Άρα **Counter** ← **Counter+1** .

Η διαδικασία αυτή θα πρέπει να επαναλαμβάνεται μέχρι να δημιουργηθούν 20 οκτάγωνα. Άρα το υποπρόγραμμα **Οκτάγωνο** θα καλέσει τον εαυτό του για όσο ισχύει μια συνθήκη. Η συνθήκη αυτή θα είναι **Counter** < 20.



Μετρητής



Συνθήκη



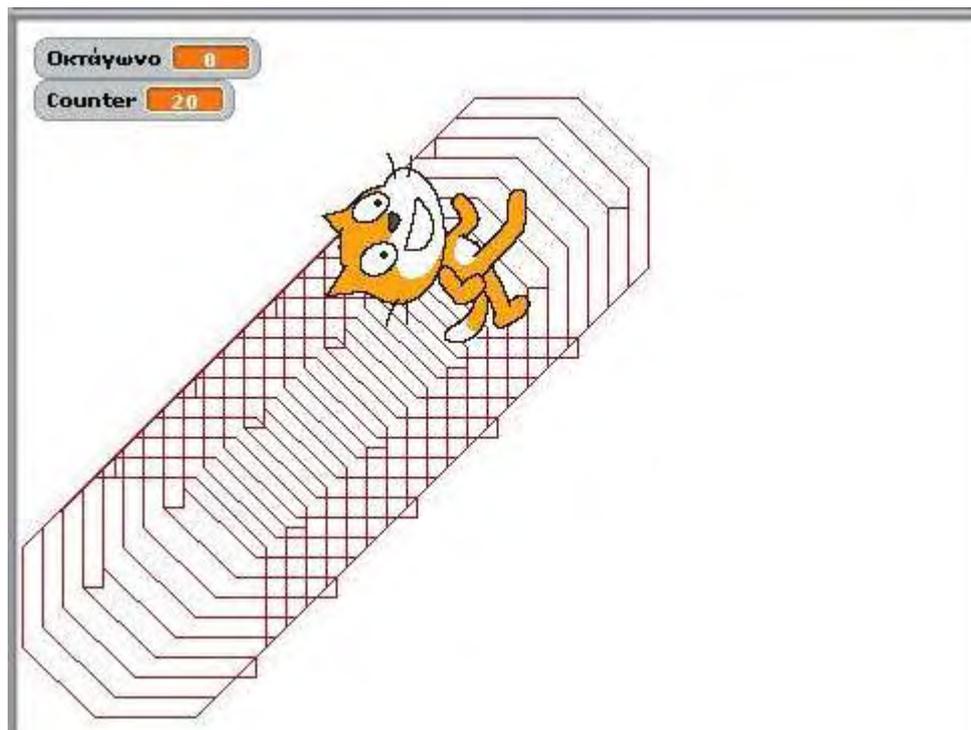
Αναδρομική κλήση →



Όταν θέλουμε να καλέσουμε από το πρόγραμμα την *Οκτάγωνο* :



Τελικό αποτέλεσμα:



Αξιολόγηση του Σχεδίου Μαθημάτων

Η αξιολόγηση θεωρείται ουσιαστικό μέρος της διδασκαλίας και της μάθησης, καθώς αποτελεί ένα μέσο για να προσδιορίσει την επιτυχία των στόχων. Κύριος σκοπός της θα πρέπει να είναι η βοήθεια που μπορεί να προσφέρει στον εκπαιδευτικό να κατανοήσει τις αδυναμίες των μαθητών αλλά και να συμβάλει στη διαρκή βελτίωση των μαθητών σε γνώσεις και δεξιότητες.

Το μάθημα ΑΕΠΠ είναι Πανελλαδικώς εξεταζόμενο και αξιολογείται στο τέλος του σχολικού έτους, αλλά και στο ενδιάμεσο με διαγωνίσματα από τον εκάστοτε εκπαιδευτικό. Το γεγονός αυτό δημιουργεί ιδιαίτερο άγχος στους μαθητές αλλά και στον εκπαιδευτικό τους.

Το σχέδιο μαθήματος που προτείνεται στην παρούσα εργασία, δεν έχει σαν στόχο την “παραδοσιακή” εξέταση του μαθητή, αλλά να καλύψει

τυχών αδυναμίες που παρατηρούνται πάνω σε βασικές έννοιες του προγραμματισμού. Είναι στην ευχέρεια του εκπαιδευτικού, μέσα από παραδείγματα και ασκήσεις με τη χρήση του προγραμματιστικού περιβάλλοντος, να ανιχνεύσει αδυναμίες των μαθητών και να τις βελτιώσει.

Ενότητα 6: Συμπεράσματα

Το μέχρι σήμερα μοντέλο διδασκαλίας που καθιστά τον διδάσκοντα πομπό πληροφοριών και τον μαθητή δέκτη που απλά αποστηθίζει, οφείλει να εκλείψει. Οι εκπαιδευτικοί δεν θα πρέπει απλά να εφαρμόζουν το σχέδιο μαθήματος, με τη μορφή που αυτό προτείνεται στα σχολικά εγχειρίδια, αλλά να το βελτιώνουν και να το προσαρμόζουν στις δικές τους ιδιαίτερες ανάγκες. Ιδιαίτερα οι διδάσκοντες μαθημάτων πληροφορικής, είναι απαραίτητο να χρησιμοποιούν τις νέες τεχνολογίες και τα μέσα που αυτές παρέχουν. Οι ΤΠΕ θα πρέπει να γίνουν το μέσο ώστε να επιτευχθεί ο στόχος, δηλαδή η μόρφωση και η γνώση.

Οι μαθητές, ως ενεργά μέλη μιας σύγχρονης κοινωνίας, κοινωνίας γνώσης, είναι απαραίτητο να αναπτύξουν γνώσεις και δεξιότητες ανωτέρου επιπέδου. Μέσα από τη μαθησιακή διαδικασία θα πρέπει να βρίσκουν κίνητρα που ενεργοποιούν το ενδιαφέρον τους, να ανάγουν τα προβλήματα στην καθημερινότητά τους, να δουλεύουν και να εξοικειώνονται με καινούρια εργαλεία. Στόχος θα πρέπει να είναι η κατάκτηση χρήσιμης γνώσης.

Η τεχνολογική ευχέρεια θεωρείται μία από τις δεξιότητες ανωτέρου επιπέδου, καθώς με τη χρήση τεχνολογικών εργαλείων ο μαθητής όχι μόνο καταναλώνει περιεχόμενο, αλλά μπορεί και να το κατασκευάζει. Ένας τρόπος για να μπορεί να παράγει κανείς έργο στον υπολογιστή είναι και ο προγραμματισμός. Θεωρείται ένα ισχυρό μέσο για την ανάπτυξη δεξιοτήτων υψηλού επιπέδου, για τη διδασκαλία βασικών εννοιών που βρίσκουν εφαρμογή στα μαθηματικά και στη φυσική, αλλά και τη

μεταφορά δεξιοτήτων επίλυσης προβλημάτων σε άλλα γνωστικά αντικείμενα. Η μάθηση προγραμματισμού μπορεί να προκαλέσει αλλαγές στο γνωστικό σύστημα των μαθητών, όπως:

- Αυστηρότητα στη σκέψη
- Κατανόηση γενικών εννοιών
- Απόκτηση ευρετικών ικανοτήτων και μεθοδολογίας
- Τεχνικές αναζήτησης λαθών
- Οικοδόμηση μιας λύσης

Το μάθημα ΑΕΠΠ εισάγει για πρώτη φορά τους μαθητές σε έννοιες προγραμματισμού. Παρ' όλες τις δυσκολίες που παρατηρούνται, μπορεί να προσφέρει σημαντική γνώση στο μαθητή και να τον βοηθήσει να αναπτύξει αναλυτική και συνθετική σκέψη, ικανότητες μεθοδολογικού χαρακτήρα και να τον βοηθήσει να επιλύσει απλά προβλήματα σε προγραμματιστικό περιβάλλον.

Τέλος, το εκπαιδευτικό λογισμικό Scratch, έχει σαν σκοπό την ανάπτυξη βασικών ικανοτήτων όπως η δημιουργική σκέψη, η επικοινωνία, η επικοδομητική συνεργασία, η δυνατότητα σχεδιασμού και επίλυσης προβλημάτων και οι δεξιότητες δια βίου μάθησης. Πλεονεκτήματά του είναι το γραφικό, “οπτικοποιημένο” περιβάλλον του, που το καθιστά εύχρηστο και ευχάριστο, όπως ένα παιχνίδι. Η δυνατότητα να μπορεί κανείς να συγγράφει, να μεταγλωττίζει και να εκτελεί ταυτόχρονα τις δημιουργίες του, επιτρέπει να δει κανείς το αποτέλεσμα των εντολών ενός προγράμματος χωρίς τον κίνδυνο συντακτικών λαθών. Έτσι, αμέσως οι μαθητές επικεντρώνονται περισσότερο στη μάθηση της διαδικαστικής γνώσης και των προγραμματιστικών δομών, παρά στον τρόπο σύνταξης

τους. Έχουν το αίσθημα να πειραματιστούν και απαλλάσσονται από το άγχος μην τυχόν κάνουν λάθη, αφού από αυτά τελικά μαθαίνουν.

Εν κατακλείδι, στην ψηφιακή εποχή που ζούμε, η γνώση, η μελέτη και η απόκτηση νέων δεξιοτήτων, αποτελεί προτέρημα όχι μόνο για τις μαθησιακές δραστηριότητες, αλλά και γενικότερα. Ο προγραμματισμός και οι νέες τεχνολογίες μπορούν να συμβάλλουν στην ανάπτυξη αυτών των δεξιοτήτων.

Ενότητα 7: Βιβλιογραφία

- [1] Διδακτική της Πληροφορικής Ι, ΤΜΗΥΤΔ, Σημειώσεις μαθήματος 2011-2012
- [2] Διδακτική της Πληροφορικής ΙΙ, ΤΜΗΥΤΔ, Σημειώσεις μαθήματος 2011-2012
- [3] Εισαγωγή στη διδακτική της πληροφορικής, Βασίλης Ι. Κόμης
- [4] <http://scratch.mit.edu/>
- [5] ΑΕΠΠ Γ' Λυκείου Τεχνολογικής Κατεύθυνσης, Βακάλη Α., Γιαννόπουλος Η., Κοίλιας Χ., Μάλαμας Κ., Μανωλόπουλος Ι., Πολίτης Π.
- [6] Γνωστικές δυσκολίες μαθητών σχετικά με την έννοια της προγραμματιστικής μεταβλητής και προτεινόμενες παρεμβάσεις, Φεσάκης Γ., Δημητρακοπούλου Α.
- [7] Η διδασκαλία του προγραμματισμού στο Ενιαίο Λύκειο: Προς ένα ολοκληρωμένο πλαίσιο με στόχο την ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων, Τζιμογιάννης Α.
- [8] Μια διδακτική πρόταση για τη γλώσσα προγραμματισμού Scratch, Νικολός Δ., Κόμης Β.
- [9] Οι νέες τεχνολογίες στην εκπαίδευση, προβλήματα και προοπτικές, Μπαβέλης Α.
- [10] Μια στατιστική έρευνα των παραμέτρων διδασκαλίας του μαθήματος ΑΕΠΠ, Κοίλιας, Δουκάκης, Γιαννοπούλου, Ψαλτίδου
- [11] Τα προβλήματα της διδασκαλίας της πληροφορικής στη δευτεροβάθμια εκπαίδευση, Ματάνας, Παπαβασιλείου, Παπαμήτσιου
- [12] Προτάσεις για την αναμόρφωση του μαθήματος ΑΕΠΠ, Δουκάκης Α., Κοίλιας Χ.
- [13] Η σημασία του αλγορίθμου και τα πλεονεκτήματα της αλγοριθμικής επίλυσης στο μάθημα ΑΕΠΠ, Κοίλιας Χ., Δουκάκης Α., Ψαλτίδου
- [14] Διδακτικές προσεγγίσεις υποψηφίων καθηγητών πληροφορικής, Κορδάκη Μ.
- [15] Η διδασκαλία του προγραμματισμού Η/Υ στη δευτεροβάθμια εκπαίδευση ως διαδικασία ανάπτυξης πνευματικών δεξιοτήτων, Γεωργίου, Τζιμογιάννης Α.
- [16] Η έννοια της μεταβλητής στον προγραμματισμό: Δυσκολίες και παρανοήσεις των μαθητών Ενιαίου Λυκείου, Τζιμογιάννης Α., Κόμης Β.

- [17] Knuth D. (1968), The art of computer programming, Volume I, Fundamental Algorithms, Addison-Wesley Inc.
- [18] Papert S. (1991), Νοητικές Θύελλες: Παιδιά, ηλεκτρονικοί υπολογιστές και δυναμικές ιδέες, Αθήνα: Εκδόσεις Οδυσσέας (Ελληνική μετάφραση)
- [19] Pea R. (1986), Language-independent conceptual ‘bugs’ in novice programming, Journal of Educational Computing Research
- [20] Du Boulay B. (1989), Some difficulties of learning to program, in E. Soloway & J. C. Spohrer (Eds.), Studying the novice programmer
- [21] Pair C. (1990), Programming, programming languages and programming methods, in T. R. G. Green (Ed.), Psychology of Programming
- [22] Soloway E. & Spohrer J. C. (1989), (Eds.), Studying the Novice Programmer, Hillsdale, NJ: Lawrence Erlbaum
- [23] Papert, S. (1980). Mindstorms. Children, Computers and Powerful Ideas. New York: Basic Books.
- [24] Resnick, M. (2007). All I Really Need to Know (about Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarden. In B. Shneiderman (Ed.), SIGCHI Conference on Creativity and Cognition
- [25] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. & Kafai, Y. (2009). Scratch: Programming for All. Communications of the ACM
- [26] Bonar J., & Soloway E. (1985), Preprogramming Knowledge: a major source of misconceptions in novice programmers, Human-Computer Interaction
- [27] Clements D. H. (1987), Longitudinal study of the effects of Logo programming on cognitive abilities and achievement, Journal of Educational Computing Research