

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2003)

2ο Συνέδριο Σύρου στις ΤΠΕ



ΣΕΝΑΡΙΑ ΔΙΔΑΣΚΑΛΙΑΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΤΗ ΔΕΥΤΕΡΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ

Στέλιος Ξυνόγαλος

Βιβλιογραφική αναφορά:

Ξυνόγαλος Σ. (2025). ΣΕΝΑΡΙΑ ΔΙΔΑΣΚΑΛΙΑΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΤΗ ΔΕΥΤΕΡΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ . *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, 1, 783–795. ανακτήθηκε από <https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/7471>

ΣΕΝΑΡΙΑ ΔΙΔΑΣΚΑΛΙΑΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΤΗ ΔΕΥΤΕΡΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ

Ξυνόγαλος Στέλιος
Διδάκτορας Εκπαιδευτικής Τεχνολογίας
stelios@uom.gr

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία παρουσιάζονται σενάρια διδασκαλίας του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση και συγκεκριμένα στο Γυμνάσιο και το Ενιαίο Λύκειο. Τα σενάρια που παρουσιάζονται στοχεύουν στην αξιοποίηση των δυνατοτήτων που παρέχουν οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση σε συνδυασμό με την πολύτιμη διδακτική γνώση που έχει συσσωρευτεί από την έρευνα που διεξήχθη τις τελευταίες δεκαετίες για τη διδασκαλία του προγραμματισμού. Στόχος μας είναι η παρουσίαση και ο σχολιασμός προτάσεων που μπορούν να προσαρμοστούν κατάλληλα και να χρησιμοποιηθούν από τους εκπαιδευτικούς για την αποτελεσματικότερη διδασκαλία του προγραμματισμού.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: σενάρια διδασκαλίας του προγραμματισμού, εκπαιδευτικό προγραμματιστικό περιβάλλον, σύστημα δυναμικής προσομοίωσης εκτέλεσης προγράμματος, προγραμματιστικός μικρόκοσμος, εκπαιδευτικό λογισμικό

ΕΙΣΑΓΩΓΗ

Η διδασκαλία του προγραμματισμού είναι γνωστό ότι παρουσιάζει αρκετές δυσκολίες. Το γεγονός αυτό είχε ως αποτέλεσμα τη διεξαγωγή εκτεταμένης έρευνας, η οποία κατέδειξε σημαντικούς παράγοντες που καθιστούν τη διδασκαλία του προγραμματισμού δύσκολη. Παράλληλα αναπτύχθηκε εκπαιδευτικό λογισμικό για τη στήριξη της διδακτικής πράξης. Ωστόσο, ακόμα και σήμερα οι αρχές του προγραμματισμού εξακολουθούν να διδάσκονται με την ίδια μέθοδο, χρησιμοποιώντας δηλαδή μια συμβατική γλώσσα προγραμματισμού και ένα επαγγελματικό περιβάλλον προγραμματισμού. Επιπλέον, τα προβλήματα που καλούνται να λύσουν οι μαθητές δεν προκαλούν το ενδιαφέρον τους, μιας και αφορούν την επεξεργασία αριθμών και συμβόλων.

Στην παρούσα εργασία παρουσιάζονται σενάρια διδασκαλίας του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση, τα οποία βασίζονται στην υπάρχουσα διδακτική γνώση - τη γνώση δηλαδή που έχει συσσωρευτεί από την εκτεταμένη έρευνα που διεξήχθη τις τελευταίες δεκαετίες για τη διδασκαλία του προγραμματισμού και τις εμπειρίες των διδασκόντων - και τις δυνατότητες που παρέχουν οι ΤΠΕ. Αρχικά γίνεται μια σύντομη αναφορά στις αρχές του προγραμματισμού που διδάσκονται στη Δευτεροβάθμια Εκπαίδευση και στους διδακτικούς στόχους των αντίστοιχων διδακτικών ενοτήτων, έτσι όπως παρουσιάζονται στο *Ενιαίο Πλαίσιο Προγράμματος Σπουδών Πληροφορικής*. Στη συνέχεια παρέχεται μια *σύνοψη των δυσκολιών* που έχουν καταγραφεί στη διεθνή βιβλιογραφία σχετικά με την εκμάθηση των σχετικών εννοιών. Η σύνοψη αυτή έχει ως

στόχο να βοηθήσει τον εκπαιδευτικό στην προετοιμασία κατάλληλων διδακτικών καταστάσεων (επιλογή κατάλληλων εκπαιδευτικών εργαλείων, παρουσίαση κατάλληλων παραδειγμάτων, προετοιμασία ασκήσεων, προετοιμασία φύλλων ελέγχου κ.τ.λ.), προκειμένου να αντιμετωπιστούν οι δυσκολίες των μαθητών. Τέλος, παρουσιάζονται κάποια σενάρια διδασκαλίας του προγραμματισμού τόσο στο Γυμνάσιο όσο και στο Ενιαίο Λύκειο. Τα σενάρια αυτά, όπως ήδη αναφέραμε, επιδιώκουν να αξιοποιήσουν τη διδακτική γνώση και τις εμπειρίες που έχουν συσσωρευτεί από τη διδασκαλία του προγραμματισμού τις τελευταίες δεκαετίες σε συνδυασμό με τις δυνατότητες που παρέχουν οι ΤΠΕ.

Η ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΤΗ ΔΕΥΤΕΡΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ

Η διδασκαλία του προγραμματισμού στη Β/θμια Εκπαίδευση πραγματοποιείται υποχρεωτικά μόνο στα πλαίσια του μαθήματος της *Πληροφορικής στη Γ' Γυμνασίου*. Στο Ενιαίο Λύκειο η διδασκαλία του προγραμματισμού πραγματοποιείται στα πλαίσια των επιλεγόμενων μαθημάτων *Εφαρμογές Πληροφορικής* και *Εφαρμογές Υπολογιστών*. Στον Πίνακα 1 παρουσιάζονται οι διδακτικοί στόχοι και το περιεχόμενο των μαθημάτων αυτών όσον αφορά στον προγραμματισμό.

	Γενικός διδακτικός στόχος	«Άξονας περιεχομένου» & Διδακτέα ύλη
Πληροφορική Γ' Γυμνασίου	Οι μαθητές πρέπει να είναι σε θέση να επιλύουν απλά προβλήματα σε προγραμματιστικό περιβάλλον.	<p>«Ελέγχο – Προγραμματίζω τον Υπολογιστή»</p> <ul style="list-style-type: none"> ✓ Η έννοια του αλγορίθμου ✓ Ο κύκλος ανάπτυξης ενός προγράμματος ✓ Το περιβάλλον μιας γλώσσας προγραμματισμού ✓ Βασικές δομές μιας συμβολικής γλώσσας <p>Διδακτικές ώρες: 10</p> <p>«Διερευνώ – Δημιουργώ – Ανακαλύπτω»</p> <p>Στην πλαίσια του άξονα αυτού (12 διδακτικές ώρες) περιλαμβάνεται η ανάπτυξη συνθετικών εργασιών με λογισμικό εφαρμογών γενικής χρήσης, λογισμικό δικτύων, λογισμικό ανάπτυξης πολυμέσων, εκπαιδευτικό λογισμικό και προγραμματιστικά εργαλεία.</p>

<p>Εφαρμογές Πληροφορικής (Μάθημα επιλογής Α' τάξης Ενιαίου Λυκείου)</p>	<p>Οι μαθητές που θα έχουν παρακολουθήσει με επιτυχία αυτά τα μαθήματα, πρέπει να μπορούν: ✓ να περιγράφουν τα χαρακτηριστικά και τις δυνατότητες των σύγχρονων προγραμματιστικών εργαλείων</p>	<p>«Ο κόσμος της Πληροφορικής» Στην πλαίσια του άξονα αυτού (20 διδακτικές ώρες) περιλαμβάνεται και η ενότητα: <i>Προγραμματιστικά περιβάλλοντα</i> «Διερευνώ – Δημιουργώ - Ανακαλύπτω» Συνθετικές εργασίες με λογισμικό εφαρμογών γενικής χρήσης, εκπαιδευτικό λογισμικό και <i>προγραμματιστικά περιβάλλοντα</i>. Διδακτικές ώρες: 27</p>
<p>Εφαρμογές Υπολογιστών (Μάθημα επιλογής Β' ή Γ' τάξης Ενιαίου Λυκείου)</p>	<p>✓ να επιλύουν απλά προβλήματα με χρήση προγραμματιστικών εργαλείων</p>	<p>«Διερευνώ – Δημιουργώ – Ανακαλύπτω» Συνθετικές εργασίες με λογισμικό εφαρμογών γενικής χρήσης, λογισμικό ανάπτυξης πολυμέσων, λογισμικό δικτύων, <i>εκπαιδευτικό λογισμικό</i> και <i>προγραμματιστικά περιβάλλοντα</i>. Διδακτικές ώρες: 30</p>

Πίνακας 1: Η ύλη και οι στόχοι της διδασκαλίας του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση.

Επίσης, στον Κύκλο Πληροφορικής και Υπηρεσιών της Τεχνολογικής Κατεύθυνσης του Ενιαίου Λυκείου διδάσκεται το πανελλαδικώς εξεταζόμενο μάθημα *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*.

ΔΥΣΚΟΛΙΕΣ ΕΚΜΑΘΗΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η εκμάθηση του προγραμματισμού είναι αναμφισβήτητα δύσκολη. Οι δυσκολίες εκμάθησης του προγραμματισμού, σύμφωνα με τον (Du Boulay, 1989), κατηγοριοποιούνται ως εξής:

Προσανατολισμός (orientation): τι είναι ο προγραμματισμός, τι είδους προβλήματα μπορούν να λυθούν, ποιες είναι οι θετικές επιδράσεις της εκμάθησης του προγραμματισμού;

Νοητή μηχανή (notional machine): δυσκολίες που αφορούν στην κατανόηση των γενικών ιδιοτήτων της μηχανής που μαθαίνει να ελέγχει ο σπουδαστής και της σχέσης της με τη φυσική μηχανή. Οι σημαντικότερες δυσκολίες, τα λάθη και οι λανθασμένες αντιλήψεις που σχετίζονται με τη νοητή μηχανή (Du Boulay, 1989) είναι οι εξής:

- Η εκτέλεση ενός προγράμματος είναι ένα είδος μηχανισμού και ο σπουδαστής χρειάζεται πολύ χρόνο για να εντοπίσει τη σχέση ανάμεσα σε ένα πρόγραμμα και στο μηχανισμό που περιγράφει.

- Ο υπολογιστής μαζί με το προγραμματιστικό περιβάλλον συνιστούν επίσης ένα μηχανισμό, ο οποίος όμως χρησιμοποιείται για τη δημιουργία άλλων μηχανισμών, δηλαδή των προγραμμάτων. Η θεώρηση του υπολογιστικού συστήματος ως μια *μηχανή* πρέπει να αναπτυχθεί σε ένα σωστό επίπεδο λεπτομέρειας. Ο σπουδαστής θα πρέπει να είναι σε θέση να αντιλαμβάνεται τι συμβαίνει στο εσωτερικό του υπολογιστή, γιατί διαφορετικά ενδέχεται να αναπτύξει τη δική του θεωρία, η οποία κατά πάσα πιθανότητα θα είναι ανεπαρκής, αν όχι λανθασμένη.

- Οι σπουδαστές δυσκολεύονται να κατανοήσουν τι παρουσιάζεται κάθε φορά στην οθόνη. Οι πληροφορίες στην οθόνη του τερματικού μπορεί να είναι καταγραφή προηγούμενων αλληλεπιδράσεων χρήστη-υπολογιστή, αποτέλεσμα εκτέλεσης συγκεκριμένων εντολών ή ένα παράθυρο στο εσωτερικό της μηχανής.

- Αρκετές δυσκολίες και λάθη σχετίζονται με τη *διαχείριση της νοητής μηχανής*: (i) Εκμάθηση της γλώσσας διαχείρισης (ανάκληση, επεξεργασία, αποθήκευση) και έκδοσης (χρήση editor) των προγραμμάτων, (ii) Το γεγονός ότι ένα πρόγραμμα γίνεται κατανοητό από το σύστημα βάσει πολύ αυστηρών κανόνων προκαλεί δυσκολίες στους αρχάριους προγραμματιστές. Πολλές φορές μάλιστα εκπλήσσονται από το βαθμό της λεπτομέρειας που απαιτεί ο προγραμματισμός, ιδιαίτερα λόγω των *ανθρωπομορφικών χαρακτηριστικών που προσδίδουν στο σύστημα*.

- Η *χρήση αγγλικών λέξεων* στις γλώσσες προγραμματισμού: (i) προσδίδει στο σύστημα, φαινομενικά τουλάχιστον, μεγαλύτερη νοημοσύνη, (ii) έχει ως αποτέλεσμα τη λανθασμένη ερμηνεία τους από τον αρχάριο προγραμματιστή, όταν οι λέξεις αυτές έχουν αρκετές σημασίες (iii) δημιουργεί προβλήματα όταν αυτές έχουν διαφορετικό νόημα στη γλώσσα προγραμματισμού απ' ότι στην Αγγλική γλώσσα.

- Πολλές σημαντικές *εσωτερικές αλλαγές* της κατάστασης της μηχανής δεν παρουσιάζονται στο χρήστη.

Οι κανόνες της γλώσσας προγραμματισμού (notation): η σύνταξη και η σημασιολογία ως επέκταση των ιδιοτήτων και της συμπεριφοράς της νοητής μηχανής.

Εκμάθηση των καθιερωμένων δομών (structures): οι δυσκολίες που αντιμετωπίζουν οι αρχάριοι προγραμματιστές κατά την εκμάθηση της σημασιολογίας των δομών μιας γλώσσας προγραμματισμού είναι πολλές και οφείλονται κατά κύριο λόγο στα παρακάτω γενικά προβλήματα:

- *Πρόβλημα φυσικής γλώσσας (natural-language problem)*: πολλές δομές των γλωσσών προγραμματισμού έχουν ονόματα που σχετίζονται με λέξεις της φυσικής γλώσσας δημιουργώντας έτσι προβλήματα κατανόησης των δομών ή/και παρανοήσεις.

- *Πρόβλημα ανθρώπινης μετάφρασης (human interpreter problem)*: οι αρχάριοι προγραμματιστές πολλές φορές υποθέτουν ότι ο υπολογιστής θα μεταφράσει τις δομές με τον ίδιο τρόπο που τις μεταφράζουν και οι ίδιοι.

- *Πρόβλημα μη συνοχής (inconsistency problem)*: αρκετές φορές οι αρχάριοι προγραμματιστές αφού καταλάβουν πως «δουλεύει» μια δομή σε μια συγκεκριμένη κατάσταση, υποθέτουν ότι η δομή θα «δουλέψει» με τον ίδιο τρόπο και σε μια άλλη παρόμοια κατάσταση.

Επίλυση προβλήματος (pragmatics): απόκτηση της ικανότητας καθορισμού, ανάπτυξης, ελέγχου και αποσφαλμάτωσης ενός προγράμματος με τα διαθέσιμα εργαλεία. Τα σημαντικότερα προβλήματα είναι τα εξής:

- *Αδυναμία εντοπισμού των στοιχείων κλειδιών για την επίλυση ενός προβλήματος*.

- *Προβλήματα συνδυασμού των επιμέρους βημάτων ενός αλγόριθμου* (plan composition problems) (Sprohrer & Soloway, 1986).

- *Η εκφορά της νοητικής αναπαράστασης ενός αλγορίθμου σε μια μορφή αποδεκτή από τον υπολογιστή είναι μια διαδικασία επίπονη και διανοητικά πολύπλοκη*. Ο μαθητής πρέπει να συνδυάσει αρκετές εντολές προκειμένου να μεταφράσει κάθε βήμα του αλγορίθμου του σε μια κατανοητή από τον υπολογιστή μορφή. Επίσης,

αρκετές φορές κατά την προσαρμογή του αλγορίθμου εντοπίζονται αρκετές ασυμβατότητες μεταξύ των βημάτων του και των δυνατοτήτων του υπολογιστή.

Μια μέθοδος που χρησιμοποιείται συχνά για τη στήριξη των σπουδαστών στην αντιμετώπιση των παραπάνω δυσκολιών είναι η σχεδίαση και ανάπτυξη *εκπαιδευτικών προγραμματιστικών περιβαλλόντων* (educational programming environments). Στην ενότητα που ακολουθεί περιγράφονται σύντομα κάποια από τα περιβάλλοντα αυτά. Συγκεκριμένα, περιγράφονται δύο περιβάλλοντα που ανήκουν στην κατηγορία των *συστημάτων δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων* (program animators), τα οποία έχουν ως βασικό στόχο την αντιμετώπιση των δυσκολιών που σχετίζονται με τη νοητή μηχανή, την κατανόηση των σημασιολογικών κανόνων της γλώσσας προγραμματισμού και των καθιερωμένων δομών, καθώς επίσης και την αποσφαλμάτωση των προγραμμάτων. Επίσης, περιγράφονται δύο περιβάλλοντα που ανήκουν στην κατηγορία των *προγραμματιστικών μικρόκοσμων* (programming microworlds), οι οποίοι ενσωματώνουν μια γλώσσα προγραμματισμού με *απλή σύνταξη και σημασιολογία*, μια νοητή μηχανή που βασίζεται σε ένα *υπαρκτό μοντέλο* που είναι ήδη γνωστό στο σπουδαστή και την τεχνολογία της δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων. Στόχος τους είναι η αντιμετώπιση του συνόλου των προαναφερθέντων δυσκολιών.

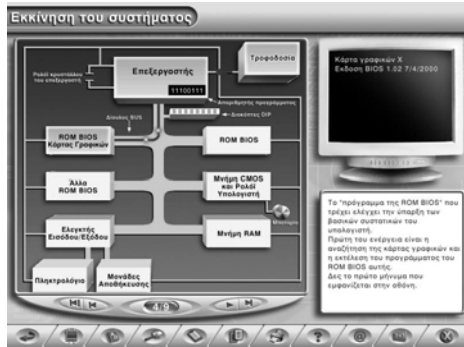
ΣΕΝΑΡΙΑ ΔΙΔΑΣΚΑΛΙΑΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

ΚΑΤΑΝΟΗΣΗ ΤΩΝ ΒΑΣΙΚΩΝ ΑΡΧΩΝ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΥΠΟΛΟΓΙΣΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Πριν να διδαχθεί ο μαθητής τις βασικές αρχές του προγραμματισμού θα πρέπει να κατανοήσει τις βασικές αρχές λειτουργίας του υπολογιστικού συστήματος, όπως άλλωστε προβλέπει και το Πρόγραμμα Σπουδών. Για το σκοπό αυτό μπορεί να χρησιμοποιηθεί το εκπαιδευτικό λογισμικό ΔΕΛΥΣ (Dagdilelis et al., 2003; <http://macedonia.uom.gr/~delys>), το οποίο συνοδεύεται από πληθώρα δραστηριοτήτων. Χρησιμοποιώντας το λογισμικό αυτό, το οποίο βασίζεται στην τεχνολογία των πολυμέσων και της οπτικοποίησης, ο μαθητής μπορεί μέσω διερεύνησης και πειραματισμού να κατανοήσει ευκολότερα τι συμβαίνει στο εσωτερικό του υπολογιστή. Από την κεντρική οθόνη ο μαθητής μπορεί να επιλέξει μία από τις πέντε ενότητες του λογισμικού:

Εκκίνηση του Συστήματος. Στην ενότητα αυτή ο μαθητής παρακολουθεί βήμα προς βήμα τις λειτουργίες που εκτελούνται σε ένα υπολογιστικό σύστημα κατά την έναρξή του. Συγκεκριμένα, παρουσιάζονται με την τεχνολογία των πολυμέσων τα βασικά τμήματα του υπολογιστή που λαμβάνουν μέρος στη διαδικασία εκκίνησης, τα σχετικά μηνύματα που εμφανίζονται στην οθόνη του υπολογιστή και ταυτόχρονα δίνεται μια περιγραφή της όλης διαδικασίας (Εικόνα 1).

Διερεύνηση του Η/Υ. Στην ενότητα αυτή παρουσιάζονται οι διάφορες εσωτερικές και εξωτερικές μονάδες του υπολογιστή. Ο μαθητής μπορεί να δει εικόνες για μια συσκευή, να διαβάσει τη σχετική θεωρία, να μεταβεί στην αίθουσα των ασκήσεων, καθώς επίσης και να μεταβεί στο εικονικό εργαστήριο προκειμένου να συνδέσει τη συσκευή στη μητρική κάρτα (Εικόνα 2).



Εικόνα 1: Οθόνη από την ενότητα Εκκίνηση του συστήματος.



Εικόνα 2: Οθόνη από την ενότητα Διερεύνηση του H/Y.

Εικονικό Εργαστήριο. Στην ενότητα αυτή παρέχεται στο μαθητή η δυνατότητα να: (i) εξοικειωθεί με τη μετατροπή αριθμών από το δεκαδικό στο δυαδικό σύστημα και αντίστροφα, χρησιμοποιώντας μια εικονική ζυγαριά με δεκαδικά και ψηφιακά βαρίδια (Εικόνα 3), (ii) συναρμολογήσει ένα υπολογιστή και να δοκιμάσει τη λειτουργία του (Εικόνα 4), (iii) συνδέσει ένα περιφερειακό στον υπολογιστή.



Εικόνα 3: Η Εικονική ζυγαριά.



Εικόνα 4: Συναρμολόγηση του H/Y.

Επεξεργασία Δεδομένων. Στην ενότητα αυτή ο μαθητής έχει τη δυνατότητα να γνωρίσει τον τρόπο με τον οποίο: (i) πραγματοποιείται η αποθήκευση και ανάκτηση των δεδομένων στο και από το σκληρό δίσκο αντίστοιχα, (ii) μετατρέπεται ένας αριθμός από το δεκαδικό στο δυαδικό σύστημα, (iii) πραγματοποιείται η πρόσθεση αριθμών στον υπολογιστή, (iv) λειτουργούν οι λογικές πύλες.

Ο Προγραμματισμός. Το εκπαιδευτικό προγραμματιστικό περιβάλλον που ενσωματώνει το λογισμικό βασίζεται στη γλώσσα X, η οποία αποτελεί ένα υποσύνολο της Pascal. Στο περιβάλλον αυτό ο χρήστης έχει τη δυνατότητα να παρακολουθήσει όλα τα στάδια της εκτέλεσης ενός προγράμματος: τη μεταγλώττιση και την έξοδο του προγράμματος, καθώς επίσης και τον assembly κώδικα, τα περιεχόμενα των καταχωρητών και τις ενδιάμεσες τιμές των μεταβλητών εφόσον επιλεγεί η κλάση προχωρημένων χρηστών (Εικόνα 5).



Εικόνα 5: Το περιβάλλον του X-Compiler.

ΑΝΑΠΤΥΞΗ ΔΕΞΙΟΤΗΤΩΝ ΓΙΑ ΤΗΝ ΑΛΓΟΡΙΘΜΙΚΗ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ

Οι μαθητές πριν να ξεκινήσουν να αναπτύσσουν προγράμματα θα πρέπει να είναι σε θέση να αναλύσουν ένα πρόβλημα σε μια ακολουθία σαφώς ορισμένων και πεπερασμένων βημάτων. Ο εκπαιδευτικός μπορεί να ετοιμάσει φύλλα με τον ορισμό ενός προβλήματος, στα οποία οι μαθητές θα συμπληρώσουν τι δίνεται και τι ζητάτε από το πρόβλημα - ποια είναι τα στοιχεία κλειδιά του προβλήματος και θα αναπτύξουν ένα αλγόριθμο για την επίλυσή του. Τα φύλλα αυτά έχουν ως στόχο την καθοδήγηση των μαθητών στην ανάπτυξη αλγορίθμων και μπορούν να χρησιμοποιηθούν από τους ίδιους σε επόμενα μαθήματα για την κωδικοποίηση του προγράμματος.

Η ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Χρήση μιας συμβατικής γλώσσας προγραμματισμού. Η διδασκαλία του προγραμματισμού τόσο στο Γυμνάσιο όσο και στο Λύκειο πραγματοποιείται συνήθως χρησιμοποιώντας ως γλώσσα προγραμματισμού την *Pascal* και κάποιο εμπορικό μεταγλωττιστή για τη γλώσσα αυτή. Για τη διδασκαλία των αρχών του προγραμματισμού μπορεί να χρησιμοποιηθεί το εκπαιδευτικό προγραμματιστικό περιβάλλον *Dynalab* (Birch et al., 1995) (Εικόνα 6), το οποίο είναι ένα σύστημα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων. Στο περιβάλλον του Dynalab ο μαθητής μπορεί να αναπτύξει, να αποθηκεύσει, να ανακαλέσει, να επεξεργαστεί, να αποθηκεύσει και να εκτελέσει ένα πρόγραμμα σε Pascal. Το σημαντικό χαρακτηριστικό του περιβάλλοντος είναι η δυνατότητα της βήμα προς βήμα εκτέλεσης ενός προγράμματος - τόσο προς τα εμπρός όσο και προς τα πίσω - με ταυτόχρονη οπτικοποίηση των δεδομένων, εμφάνιση δηλαδή των μεταβολών των τιμών των μεταβλητών κατά την εκτέλεση ενός προγράμματος. Παρέχεται δηλαδή στο μαθητή ένα «παράθυρο» στο εσωτερικό της μηχανής που μαθαίνει να χειρίζεται βοηθώντας τον έτσι:

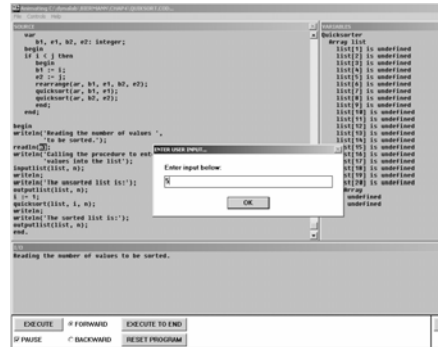
- να ελέγξει και να αποσφαλματώσει τα προγράμματά του
- να κατανοήσει τη ροή εκτέλεσης ενός προγράμματος
- να κατανοήσει το δυναμικό χαρακτήρα των μεταβλητών και να εξαλείψει τις σχετικές παρανοήσεις που δημιουργούνται - κατά κύριο λόγο - από τον παραλληλισμό της έννοιας της μεταβλητής στον προγραμματισμό με την έννοια της στα μαθηματικά
- να εξαλείψει παρανοήσεις που σχετίζονται με τη σημασιολογία των δομών επιλογής και επανάληψης

Το Dynalab μαζί με βιβλιοθήκες έτοιμων προγραμμάτων διατίθεται ελεύθερα στο διαδίκτυο (<http://www.cs.montana.edu/~dynamalab/download/download.html>).

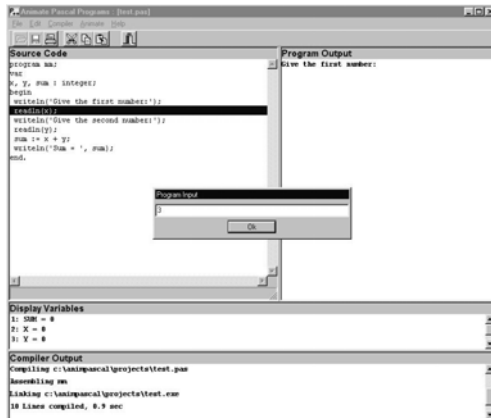
Εναλλακτικά, μπορεί να χρησιμοποιηθεί το εκπαιδευτικό προγραμματιστικό περιβάλλον *AnimPascal* (Σατρατζέμη κ.α., 2000), το οποίο αποτελεί και αυτό ένα σύστημα δυναμικής προσομοίωσης εκτέλεσης προγραμμάτων. Σε αντίθεση με το Dynalab, το AnimPascal (Εικόνα 7) δεν υποστηρίζει την προς τα πίσω εκτέλεση ενός προγράμματος, η οποία δίνει στο μαθητή τη δυνατότητα να διερευνήσει το τμήμα του προγράμματος που τον δυσκολεύει, αποφεύγοντας έτσι την ανάγκη επανεκκίνησης της εκτέλεσής του. Επίσης, δεν υποστηρίζει την οπτικοποίηση της εκτέλεσης των υποπρογραμμάτων. Ωστόσο, ο μεταγλωττιστής Free Pascal που χρησιμοποιείται δεν ενημερώνει απλά το μαθητή για τα σφάλματα (errors), αλλά τον προειδοποιεί και για πιθανές παραλείψεις (warnings, hints, notes) του προγράμματός του.

Επιπλέον, το AnimPascal ενσωματώνει τη *δυνατότητα καταγραφής των ενεργειών των μαθητών*, αποθηκεύει δηλαδή τον πηγαίο κώδικα και τα αποτελέσματα της μεταγλώττισης κάθε φορά που ο μαθητής μεταγλωττίζει το πρόγραμμά του. Οι εκδόσεις των προγραμμάτων παρουσιάζονται σε ένα διαφορετικό παράθυρο (Εικόνα 8) και παρέχουν τη δυνατότητα στο διδάσκοντα:

- να διερευνήσει τις δυσκολίες και τις παρανοήσεις των μαθητών, καθώς επίσης και τις στρατηγικές επίλυσης προβλημάτων που εφαρμόζουν
- να προσαρμόσει το μάθημα στις ανάγκες των μαθητών.



Εικόνα 6: Το περιβάλλον του Dynalab.



Εικόνα 7: Το βασικό παράθυρο του AnimPascal.



Εικόνα 8: Ιστορία των μεταγλωττίσεων.

Χρήση μιας εκπαιδευτικής γλώσσας προγραμματισμού. Εφόσον ο διδακτικός στόχος της διδασκαλίας του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση είναι η κατανόηση των βασικών αρχών του και όχι η εκμάθηση μιας γλώσσας προγραμματισμού, πιστεύουμε ότι είναι προτιμότερο να χρησιμοποιηθεί μια εκπαιδευτική γλώσσα προγραμματισμού. Ιδανικός για την επίτευξη αυτού του στόχου είναι ένας *προγραμματιστικός μικρόκοσμος* (programming microworld), ο οποίος:

- ενσωματώνει μια γλώσσα προγραμματισμού με περιορισμένο ρεπερτόριο εντολών, απλή σύνταξη και σημασιολογία, έτσι ώστε να αποφευχθεί το πρόβλημα της κατανόησης των σχετικών κανόνων της γλώσσας προγραμματισμού
- βασίζεται σε ένα υπαρκτό μοντέλο που είναι ήδη γνωστό στο μαθητή και όχι στη μηχανή του von Neumann, η οποία διέπεται από ένα σύνολο δυσνόητων (για τον αρχάριο προγραμματιστή) κανόνων. Ένας μικρόκοσμος ενσωματώνει συνήθως ένα πρωταγωνιστή, τον οποίο ο μαθητής κατευθύνει με τα προγράμματα που αναπτύσσει. Χαρακτηριστικό παράδειγμα είναι η **Logo**
- ενσωματώνει τη δυνατότητα της δυναμικής προσομοίωσης εκτέλεσης των προγραμμάτων, η οποία παρέχει τα παιδαγωγικά οφέλη που ήδη αναφέραμε
- επιτρέπει την επίλυση προβλημάτων που προκαλούν το ενδιαφέρον του σπουδαστή.

Ένας από τους πιο δημοφιλείς προγραμματιστικούς μικρόκοσμους είναι το **ρομπότ Karel** (Pattis et al., 1995) που χρησιμοποιείται για τη διδασκαλία του δομημένου προγραμματισμού. Ο Karel, ο πρωταγωνιστής του μικρόκοσμου, εκτελεί διάφορες αποστολές (προγράμματα) σε ένα κόσμο που αποτελείται από οριζόντιους δρόμους και κάθετες λεωφόρους, τμήματα τοίχου που τοποθετούνται μεταξύ των διασταυρώσεων δημιουργώντας εμπόδια (όπως λαβύρινθος, σκάλες, βουνά κ.τ.λ.) και beepers, μικροί πλαστικοί κώνοι που παράγουν ένα ήχο (μπιπ). Ο Karel έχει τη δυνατότητα να κινείται κατά ένα μπλοκ (move), να στρίβει αριστερά κατά 90° (turnLeft), να σηκώνει beepers από τη διασταύρωση που βρίσκεται (pickBeeper) και να αφήνει beepers από την τσάντα του στη διασταύρωση (putBeeper). Επίσης, ο Karel

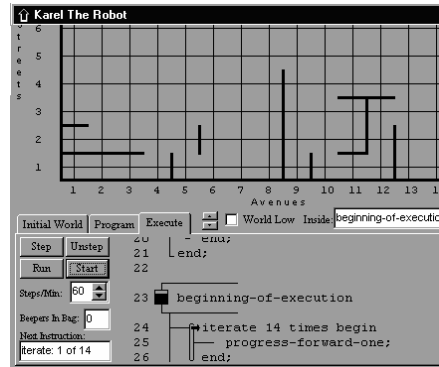
έχει τη δυνατότητα να ελέγχει την κατάσταση του κόσμου. Η γλώσσα προγραμματισμού περιλαμβάνει τις βασικές δομές επιλογής και επανάληψης, αλλά δεν υποστηρίζει μεταβλητές. Εξέλιξή του αποτελεί ο μικρόκοσμος **Karel++** (Bergin et al., 1997) για τη διδασκαλία του αντικειμενοστραφούς προγραμματισμού. Η βασική κλάση ρομπότ έχει τις δυνατότητες που έχει και το ρομπότ Karel. Αν οι δυνατότητες αυτές δεν επαρκούν για την εκτέλεση μιας αποστολής ο μαθητής δηλώνει νέες κλάσεις με αυξημένες δυνατότητες.

Ένας προσομοιωτής του ρομπότ Karel που διατίθεται ελεύθερα στο διαδίκτυο (<http://ocweb.otterbein.edu/csc/cs115/web/Karel>) είναι ο **KarelTheRobot** (Εικόνα 9) που παρέχει τη δυνατότητα ανάπτυξης, αποθήκευσης, ανάκλησης και βηματικής εκτέλεσης - τόσο προς τα εμπρός όσο και προς τα πίσω - των προγραμμάτων.

Ένας προσομοιωτής που βασίζεται στον Karel++ είναι ο διδακτικός μικρόκοσμος **objectKarel** (Ξυνόγαλος, 2002; Xinogalos & Satratzemi 2002). Ο μικρόκοσμος αυτός ενσωματώνει μια σειρά μαθημάτων στα ελληνικά (Εικόνα 10) και δραστηριοτήτων για την εξοικείωση του μαθητή με τις βασικές έννοιες του αντικειμενοστραφούς προγραμματισμού και τις βασικές δομές. Για παράδειγμα, ο μαθητής πατώντας απλά κάποια κουμπιά μπορεί να παρακολουθήσει πως ανταποκρίνεται ένα ρομπότ στα τέσσερα μηνύματα της βασικής κλάσης ρομπότ, καθώς επίσης και τη σύνταξη της εντολής στη γλώσσα προγραμματισμού (Εικόνα 11).

Επιπλέον, ο διδακτικός μικρόκοσμος:

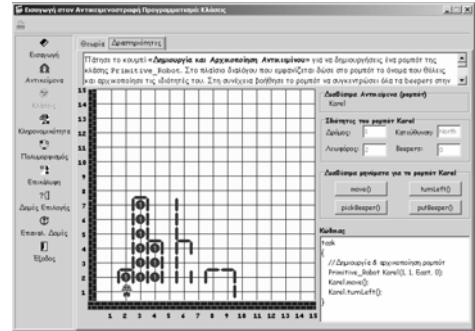
- ενσωματώνει έναν *εκδότη δομής*. Ο μαθητής αναπτύσσει τα προγράμματά του μέσω ενός μενού και πλαισίων διαλόγου (Εικόνα 12). Έτσι, η διδασκαλία επικεντρώνεται στις έννοιες και όχι στις συντακτικές λεπτομέρειες της γλώσσας προγραμματισμού
- ενσωματώνει τη δυνατότητα της *βήμα προς βήμα εκτέλεσης των προγραμμάτων*
- ενσωματώνει τη δυνατότητα της *επεξηγηματικής οπτικοποίησης*, την εμφάνιση δηλαδή επεξηγήσεων της τρέχουσας εντολής στα ελληνικά. Οι επεξηγήσεις αυτές εμφανίζονται στο κάτω μέρος του βασικού παραθύρου του προγραμματιστικού περιβάλλοντος (Εικόνα 12). Η δυνατότητα αυτή βοηθάει το μαθητή στην πληρέστερη κατανόηση της σημασίας των διαφόρων δομών και της ροής εκτέλεσης ενός προγράμματος



Εικόνα 9: Το περιβάλλον του KarelTheRobot.

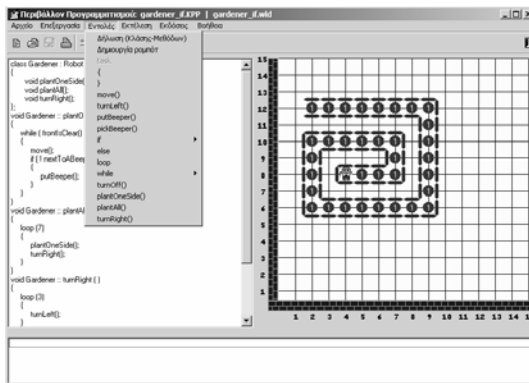


Εικόνα 10: Το περιβάλλον των Μαθημάτων.



Εικόνα 11: Το περιβάλλον των Δραστηριοτήτων.

- αναφέρει εξελληνισμένα μηνύματα λάθους που χρησιμοποιούν απλή γλώσσα και χαρακτηρίζονται από σαφήνεια και πληρότητα
- ενσωματώνει σύστημα βοήθειας, το οποίο έχει τη μορφή ενός υπερκειμένου.
- ενσωματώνει τη δυνατότητα της καταγραφής των ενεργειών των μαθητών. Η ιστορία των μεταγλωττίσεων παρουσιάζεται σε ένα διαφορετικό παράθυρο με τη μορφή ενός δένδρου δύο επιπέδων (Εικόνα 13)



Εικόνα 12: Το περιβάλλον Προγραμματισμού.



Εικόνα 13: Η ιστορία των μεταγλωττίσεων.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η διδασκαλία του προγραμματισμού, ακόμα και στις μέρες μας, παρουσιάζει αρκετές δυσκολίες. Ιδιαίτερα στη Δευτεροβάθμια Εκπαίδευση όπου υπάρχουν αρκετοί περιορισμοί - με σημαντικότερους την ανεπάρκεια χρόνου, τη μικρή αναλογία υπολογιστών/μαθητών και το γεγονός ότι εξακολουθεί να χρησιμοποιείται η κλασική μέθοδος διδασκαλίας - η διδασκαλία του προγραμματισμού πραγματοποιείται υποτυπωδώς. Ωστόσο, πιστεύουμε ότι η αξιοποίηση της διδακτικής γνώσης που έχει συσσωρευτεί από την έρευνα που διεξήχθη τις τελευταίες δεκαετίες σε συνδυασμό με τις δυνατότητες που παρέχουν οι ΤΠΕ μπορεί να δώσει λύση σε πολλά από τα προβλήματα που παρουσιάζει η διδασκαλία του προγραμματισμού.

Καταρχήν, ο εκπαιδευτικός πρέπει να έχει γνώση των αποτελεσμάτων της έρευνας σχετικά με τις δυσκολίες που αντιμετωπίζουν οι μαθητές κατά την εισαγωγή τους στον προγραμματισμό προκειμένου να διαμορφώσει αποτελεσματικές διδακτικές καταστάσεις. Ο εκπαιδευτικός μπορεί να προετοιμάσει κατάλληλα παραδείγματα και να κάνει χρήση του εκπαιδευτικού λογισμικού που περιγράψαμε για να τα παρουσιάσει στους μαθητές με τη βοήθεια ενός video projector. Επίσης, μπορεί πολύ εύκολα να ελέγξει το βαθμό κατανόησης των διδασκόμενων εννοιών διακόπτοντας τη βήμα προς βήμα εκτέλεση ενός προγράμματος και ζητώντας από τους μαθητές να προσδιορίσουν - για παράδειγμα - το αποτέλεσμα εκτέλεσης της τρέχουσας εντολής ή την εντολή που θα εκτελεστεί στη συνέχεια.

Όσον αφορά στα εκπαιδευτικά εργαλεία που θα χρησιμοποιηθούν πιστεύουμε ότι η διδασκαλία του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση θα ήταν ιδανικό να ξεκινάει - στο Γυμνάσιο - με ένα προγραμματιστικό μικρόκοσμο. Με τη χρήση του objectKarel ή του KarelTheRobot η διδασκαλία μπορεί να επικεντρωθεί στις έννοιες και στην ανάπτυξη δεξιοτήτων για την αλγοριθμική επίλυση προβλημάτων. Ειδικά το λογισμικό objectKarel, το οποίο ενσωματώνει μαθήματα - δραστηριότητες στα ελληνικά και έναν εκδότη δομής, πιστεύουμε ότι μπορεί να συντελέσει ουσιαστικά στην επίτευξη αυτού του στόχου. Βέβαια, στους παραπάνω μικρόκοσμους δεν υποστηρίζονται μεταβλητές. Πιστεύουμε όμως ότι η έννοια της μεταβλητής, η οποία είναι γνωστό ότι προκαλεί πολλές δυσκολίες στους μαθητές, είναι προτιμότερο να μην διδάσκεται εξ' αρχής.

Στο Ενιαίο Λύκειο και εφόσον οι μαθητές έχουν κατανοήσει τις βασικές αρχές του προγραμματισμού θα μπορούσε να χρησιμοποιηθεί το Dynalab ή το AnimPascal προκειμένου να διδαχθούν οι μαθητές την έννοια της μεταβλητής και παράλληλα να πραγματοποιηθεί μια εισαγωγή σε μια συμβατική γλώσσα προγραμματισμού όπως η Pascal.

Επιπλέον, η χρήση του objectKarel και του AnimPascal θα δώσει στον εκπαιδευτικό τη δυνατότητα να διερευνήσει τις δυσκολίες των μαθητών διεξοδικά και να προσαρμόσει το μάθημα στις ανάγκες τους. Να επισημάνουμε βέβαια ότι η μελέτη των διαδοχικών εκδόσεων των προγραμμάτων όλων των μαθητών είναι ιδιαίτερα χρονοβόρα. Μπορεί ωστόσο ο εκπαιδευτικός να μελετήσει επιλεκτικά κάποια από τα προγράμματα των μαθητών.

Τέλος, θα θέλαμε να αναφέρουμε κάποια στοιχεία που αφορούν στην αξιολόγηση των εκπαιδευτικών εργαλείων που παρουσιάσαμε:

- Το εκπαιδευτικό λογισμικό ΔΕΛΥΣ ελέγχθηκε και αξιολογήθηκε σε σχολεία της Δευτεροβάθμιας Εκπαίδευσης τόσο από καθηγητές κατά τη φάση της ανάπτυξης του όσο και από το Παιδαγωγικό Ινστιτούτο μετά την ολοκλήρωσή του (Dagdilelis et al., 2003). Οι καθηγητές που συμμετείχαν στην αξιολόγηση ανέφεραν ότι οι μαθητές έδειξαν ιδιαίτερο ενδιαφέρον για το λογισμικό, ενώ το Παιδαγωγικό Ινστιτούτο έκρινε ότι το λογισμικό παρέχει σημαντική βοήθεια στη διδασκαλία των μαθημάτων πληροφορικής του Προγράμματος Σπουδών.

- Το προγραμματιστικό περιβάλλον Dynalab, το οποίο χρησιμοποιείται ευρέως στο εξωτερικό για τη διδασκαλία του προγραμματισμού, παρουσιάστηκε στους συμμετέχοντες του έργου «E42: Επιμόρφωση Επιμορφωτών στα Νέα Δικτυακά και Υπολογιστικά Συστήματα», στα πλαίσια του ΕΠΕΑΕΚ. Οι συμμετέχοντες έκριναν ότι

το Dynalab μπορεί να στηρίξει το μαθητή στην εκμάθηση του προγραμματισμού και εξέφρασαν την πρόθεση να το χρησιμοποιήσουν στις επιμορφώσεις των εκπαιδευτικών.

- Τα περιβάλλοντα που βασίζονται στο ρομπότ Karel χρησιμοποιούνται ευρέως στο εξωτερικό για τη διδασκαλία του προγραμματισμού σε όλες τις βαθμίδες της εκπαίδευσης, ακόμα και στην Πρωτοβάθμια. Επιπλέον, θα θέλαμε να αναφέρουμε ότι το προγραμματιστικό περιβάλλον objectKarel και η σειρά μαθημάτων που ενσωματώνει χρησιμοποιήθηκαν για τη διδασκαλία του αντικειμενοστραφούς προγραμματισμού σε φοιτητές του τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας (Ξυνόγαλος, 2002). Όλοι οι φοιτητές αξιολόγησαν θετικά το λογισμικό, ενώ από την ανάλυση των δεδομένων που συγκεντρώθηκαν προέκυψε ότι το λογισμικό στήριξε ουσιαστικά τόσο το διδάσκοντα όσο και τους φοιτητές στη διδασκαλία και εκμάθηση του προγραμματισμού αντίστοιχα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Bergin, J., Stehlik, M., Roberts, & J. Pattis (1997), *Karel++: A Gentle Introduction to the Art of Object-Oriented Programming, 2nd edn. New York, Wiley*
2. Birch, M., Boroni, C., Goosey, F., Patton, S., Poole, D., Pratt, C. & Ross, R. (1995), DYNALAB: A Dynamic Computer Science Laboratory Infrastructure Featuring Program Animation, *ACM, SIGSCE Bulletin, Vol. 27, No. 1*, 29-33
3. Dagdilelis, V., Evangelidis, G., Satratzemi, M., Efoopoulos, V. & Zagouras, C. (2003), DELYS: a novel microworld-based educational software for teaching computer science subjects, *Computers & Education 40 (2003)*, 307-325, Elsevier Publishers
4. Du Boulay, B. (1989), Some Difficulties Of Learning To Program. In Soloway, E. & Sprohrer, J. (Eds.), *Studying The Novice Programmer*, 283-300, Lawrence Erlbaum Associates
5. Pattis, R. E., Roberts, J. & Stehlik, M. (1995), *Karel The Robot: A Gentle Introduction to the Art of Programming, 2nd edn. New York, John Wiley & Sons*
6. Spohrer, J. C. & Soloway, E. (1986), Novice Mistakes: Are the Folk Wisdoms Correct? *Communications of the ACM, Vol. 29, No. 7*, 624-632
7. Xinogalos, S. & Satratzemi, M. (2002), An Integrated Programming Environment for Teaching the Object-Oriented Programming Paradigm, In Shafazand, H. A & A Min Tjoa (Eds.), *“Lecture Notes in Computer Science” (LNCS), 2510, EurAsia-ICT 2002*, 544-551, Springer Verlag
8. Ξυνόγαλος, Σ. (2002), Εκπαιδευτική Τεχνολογία: Ένας Διδακτικός Μικρόκοσμος για την Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό, *Διδακτορική διατριβή, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας*
9. Παιδαγωγικό Ινστιτούτο (1998) Η Πληροφορική στο σχολείο
10. Σατρατζέμη, Μ., Χατζηαθανασίου, Κ. & Δαγδiléλης, Β. (2000), AnimPascal: Ένα Εκπαιδευτικό Περιβάλλον για τη Στήριξη Εισαγωγικών Μαθημάτων Προγραμματισμού, *Πρακτικά 2^{ου} Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή – οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση*, 125-135, Πάτρα, Οκτώβριος 2000