

Η διδασκαλία του Προγραμματισμού και της αλγοριθμικής επίλυσης προβλημάτων στο Ενιαίο Λύκειο

Αθανάσιος Τζιμογιάννης

Σκοπός

Σκοπός αυτής της θεματικής ενότητας είναι η διαπραγμάτευση του πλαισίου διδασκαλίας του προγραμματισμού και της αλγοριθμικής επίλυσης προβλημάτων στο Ενιαίο Λύκειο. Αναλύονται οι διδακτικοί στόχοι των Προγραμμάτων Σπουδών Πληροφορικής και οι βασικές παρανοήσεις και γνωστικές δυσκολίες των μαθητών στον προγραμματισμό, οι οποίες συνιστούν σημαντικά διδακτικά εμπόδια στην τάξη. Παρουσιάζονται παιδαγωγικές και διδακτικές στρατηγικές με στόχο το σχεδιασμό μαθησιακών δραστηριοτήτων που συμβάλλουν στην ενεργοποίηση των μαθητών, στην εποικοδόμηση των βασικών προγραμματιστικών εννοιών και στην καλλιέργεια δεξιοτήτων για την αλγοριθμική επίλυση προβλημάτων. Τέλος, γίνεται αναλυτική αναφορά στη διδασκαλία της έννοιας της προγραμματιστικής μεταβλητής και της εντολής εκχώρησης, καθώς και των ρόλων μεταβλητών στο πρόγραμμα.

Προσδοκώμενα αποτελέσματα

Στο τέλος αυτής της θεματικής ενότητας οι επιμορφούμενοι εκπαιδευτικοί πληροφορικής θα πρέπει

- να κατανοήσουν του γενικούς σκοπούς και τους ειδικούς στόχους της διδασκαλίας του προγραμματισμού και της αλγοριθμικής επίλυσης προβλημάτων, στα πλαίσια των μαθημάτων Πληροφορικής το Ενιαίου Λυκείου
- να γνωρίσουν και να εντοπίσουν τις βασικές παρανοήσεις και γνωστικές δυσκολίες που αντιμετωπίζουν οι μαθητές
- να κατανοήσουν τα βασικά χαρακτηριστικά των διαφόρων διδακτικών προσεγγίσεων και παιδαγωγικών στρατηγικών για τη διδασκαλία του προγραμματισμού
- να είναι σε θέση να σχεδιάζουν μαθησιακές δραστηριότητες που θα στοχεύουν στην εποικοδόμηση επαρκών αναπαραστάσεων για τις προγραμματιστικές έννοιες και στην προετοιμασία των μαθητών για την επίλυση αυθεντικών προβλημάτων σε προγραμματιστικά περιβάλλοντα.

Έννοιες-Κλειδιά

- Ενιαίο Πλαίσιο Προγραμμάτων Σπουδών Πληροφορικής
- Διδακτική του προγραμματισμού
- Παρανοήσεις και εννοιολογικές δυσκολίες των μαθητών
- Μοντέλα ανάπτυξης προγραμματιστικών δεξιοτήτων
- Διερευνητικές και εποικοδομηστικές προσεγγίσεις
- Σχεδιασμός μαθησιακών δραστηριοτήτων

Η Πληροφορική ως αντικείμενο γενικής παιδείας στο Ενιαίο Λύκειο

Η εισαγωγή του μαθήματος της Πληροφορικής στο Ενιαίο Λύκειο έγινε το 1998, στα πλαίσια του **Ενιαίου Πλαισίου Προγραμμάτων Σπουδών Πληροφορικής (ΕΠΠΣΠ)**. Η προσέγγιση που ακολουθήθηκε (ΥΠΕΠΘ, 1998) αντιμετωπίζει την Πληροφορική ως αυτόνομο γνωστικό αντικείμενο, το οποίο έχει ιδιαίτερη αξία για την ολοκληρωμένη προετοιμασία των μαθητών, σε ότι αφορά στην πορεία τους μετά το Λύκειο, και παρέχει επαρκείς γνώσεις και δεξιότητες στην Πληροφορική.

Η φιλοσοφία που διατρέχει το ΕΠΠΣΠ δεν θεωρεί τη διδασκαλία της Πληροφορικής ως μέσο κατάρτισης των μαθητών στο χειρισμό των υπολογιστών και στη χρήση λογισμικών γενικού σκοπού. Η Πληροφορική αποτελεί **αντικείμενο γενικής παιδείας** που απευθύνεται σε όλους τους μαθητές και στοχεύει

- στην κατανόηση **βασικών εννοιών** και στην απόκτηση **διαχρονικών γνώσεων** στην Πληροφορική
- στην ανάπτυξη **διαχρονικών δεξιοτήτων** χρήσης του υπολογιστή ως εργαλείο έρευνας και μάθησης
- στην καλλιέργεια **δεξιοτήτων μεθοδολογικού χαρακτήρα** (έκφραση και διερεύνηση ιδεών, δημιουργικότητα, διαθεματική προσέγγιση της γνώσης, πειραματισμός και ανίχνευση λαθών)
- στην καλλιέργεια **δεξιοτήτων υψηλού επιπέδου** (κριτική και αναλυτική σκέψη, συνθετική ικανότητα)
- στην απόκτηση **ευρύτερης παιδείας και κουλτούρας** γύρω από την Πληροφορική.
Το Πρόγραμμα Σπουδών του Ενιαίου Λυκείου περιλαμβάνει τα εξής μαθήματα επιλογής:
- **στην Α' Τάξη**
 - «Εφαρμογές Πληροφορικής»
- **στη Β' ή/και στη Γ' Τάξη:**
 - «Εφαρμογές Υπολογιστών»
 - «Τεχνολογία Υπολογιστών»
 - «Πολυμέσα-Δίκτυα»
 - «Εφαρμογές Λογισμικού».

Τέλος, στη Γ' Τάξη Τεχνολογικής Κατεύθυνσης διδάσκεται ως βασικό το μάθημα «*Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*», το οποίο εξετάζεται πανελλαδικά στις εισαγωγικές εξετάσεις για τα ΑΕΙ και ΤΕΙ.

Ο γενικός σκοπός των μαθημάτων Πληροφορικής στο Λύκειο αφορά στην κατανόηση των **βασικών αρχών** της επιστήμης της Πληροφορικής και του προγραμματισμού, καθώς και στην εξοικείωση των μαθητών με νέες **μεθοδολογίες επίλυσης προβλημάτων** που αξιοποιούν τα σύγχρονα εργαλεία των ΤΠΕ. Στην κατεύθυνση αυτή, ένα σύγχρονο Πρόγραμμα Σπουδών της Πληροφορικής για τη δευτεροβάθμια εκπαίδευση, καθώς και οι ακολουθούμενες προσεγγίσεις διδασκαλίας, καλούνται να εξειδικεύσουν στην πράξη τον όρο 'αντικείμενο γενικής παιδείας'. Το προτεινόμενο διδακτικό πλαίσιο προσδιορίζεται από τέσσερις αλληλοεξαρτώμενες συνιστώσες (Τζιμογιάννης 2005):

Η Πληροφορική ως επιστήμη: Διαπραγμάτευση των αρχών που θεμελιώνουν την Πληροφορική ως επιστήμη στο ευρύτερο πλαίσιο των θετικών επιστημών. Ο άξονας αυτός θα πρέπει να στοχεύει στην οικοδόμηση **διαχρονικών γνώσεων** που αφορούν σε έννοιες, αντικείμενα και διαδικασίες (π.χ. υλικό, λογισμικό, λογισμικό συστήματος, αρχείο, αναπαράσταση της πληροφορίας, επεξεργασία δεδομένων, αλγόριθμος, μοντελοποίηση, προγραμματιστικές δομές κ.λπ.).

Η Πληροφορική ως μεθοδολογία επίλυσης προβλημάτων: Τα σύγχρονα εργαλεία παρέχουν νέες μεθοδολογίες επίλυσης προβλημάτων (π.χ. υπολογιστικά φύλλα, βάσεις δεδομένων, προγραμματιστικά περιβάλλοντα κ.λπ.) που απαιτούν την ανάπτυξη νέου τύπου

δεξιοτήτων από τους μαθητές. Αυτό μπορεί να επιτευχθεί μέσω της εμπλοκής των μαθητών σε **αυθεντικές δραστηριότητες** που θα έχουν ως σκοπό την καλλιέργεια δεξιοτήτων μεθοδολογικού χαρακτήρα (επεξεργασία δεδομένων, σχεδιασμός και υλοποίηση αλγορίθμων, μοντελοποίηση λύσεων, προγραμματισμός υπολογιστών, δημιουργικότητα) και δεξιοτήτων υψηλού επιπέδου (κριτική και αναλυτική σκέψη, συνθετική ικανότητα, διερευνητική-συνεργατική μάθηση).

Η Πληροφορική ως εργαλείο: Υπάρχει ακόμη αναγκαιότητα επέκτασης και εμπάθυνσης των γνώσεων και των τεχνικών δεξιοτήτων που, ενδεχόμενα, κατέχουν οι μαθητές της δευτεροβάθμιας εκπαίδευσης μέσα από το οικογενειακό ή το κοινωνικό τους περιβάλλον. Επιπλέον, το δημόσιο σχολείο θα πρέπει να καλύψει τις ενδεχόμενες ανισότητες και να προσφέρει ευκαιρίες για εξοικείωση με τους υπολογιστές στα παιδιά που δεν τις έχουν για λόγους κοινωνικούς, οικονομικούς ή άλλους. Βασικός στόχος, για όλους τους μαθητές, πρέπει να είναι η καλλιέργεια **διαχρονικών τεχνικών δεξιοτήτων** χειρισμού των σύγχρονων υπολογιστικών περιβαλλόντων (λειτουργικό σύστημα, λογισμικά γενικής χρήσης, αναζήτηση και αξιολόγηση πληροφοριών κ.λπ.).

Η Πληροφορική ως κοινωνικό φαινόμενο: Κριτική επισκόπηση και αξιολόγηση των σύγχρονων εφαρμογών της Πληροφορικής με αναφορές στον κοινωνικό, εργασιακό, εκπαιδευτικό, επιστημονικό και πολιτιστικό τομέα. Απόκτηση ευρύτερης **παιδείας και κουλτούρας** γύρω από την Πληροφορική. Όπως οι μαθητές της δευτεροβάθμιας εκπαίδευσης μελετούν τις φυσικές επιστήμες, ώστε να κατανοήσουν το φυσικό κόσμο και τις δυνατότητες παρέμβασής τους σ' αυτόν, έτσι θα πρέπει να μελετήσουν την επιστήμη και τις τεχνολογίες των υπολογιστών ώστε να εμπεδώσουν το νέο κοινωνικό, οικονομικό και πολιτισμικό περιβάλλον που διαμορφώνεται στην Κοινωνία της Πληροφορίας.

Η Πληροφορική ως αντικείμενο γενικής παιδείας στο Λύκειο στοχεύει σε τρεις γενικούς εκπαιδευτικούς σκοπούς που αφορούν σε:

Γνώσεις: Κατανόηση βασικών εννοιών και απόκτηση διαχρονικών γνώσεων στην Πληροφορική.

Δεξιότητες: Διαχρονικές δεξιότητες επίλυσης προβλημάτων με χρήση περιβαλλόντων προγραμματισμού και λογισμικών γενικού σκοπού

Ικανότητες: Χρήση λειτουργικών περιβαλλόντων, επεξεργασία κειμένου, υπολογιστικά φύλλα, βάσεις δεδομένων, παρουσιάσεις, πλοήγηση στο Διαδίκτυο, επικοινωνία κ.λπ.

Το πλαίσιο γνώσεων και δεξιοτήτων στον προγραμματισμό

Η διδασκαλία του προγραμματισμού και της αλγοριθμικής στο Ενιαίο Λύκειο αποτελεί βασική συνιστώσα του Προγράμματος Σπουδών και εντάσσεται στο πλαίσιο που αναλύθηκε στην προηγούμενη ενότητα. Σύμφωνα με το Πρόγραμμα Σπουδών (ΥΠΕΠΘ 1998), οι ειδικοί σκοποί της διδασκαλίας του προγραμματισμού στο Ενιαίο Λύκειο είναι οι μαθητές

- να αναπτύξουν αναλυτική και συνθετική σκέψη
- να αποκτήσουν ικανότητες μεθοδολογικού χαρακτήρα
- να μπορούν να επιλύουν απλά προβλήματα σε προγραμματιστικό περιβάλλον
- να αναπτύξουν δεξιότητες αλγοριθμικής προσέγγισης (ανάλυση προβλήματος, σχεδίαση αλγορίθμου, δομημένη σκέψη, αυστηρότητα έκφρασης)
- να αναπτύξουν τη δημιουργικότητα και τη φαντασία τους.

Στο μάθημα «Εφαρμογές Πληροφορικής» της Α' Λυκείου προβλέπεται ικανός αριθμός διδακτικών ωρών (20 συνολικά) για την εμπλοκή των μαθητών με την αλγοριθμική και τον προγραμματισμό. Στα πλαίσια της διδασκαλίας του μαθήματος «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον», στη Γ' Τάξη Τεχνολογικής Κατεύθυνσης, παρέχονται

ευκαιρίες για εξοικείωση με κλασσικούς αλγορίθμους, εξάσκηση των μαθητών στην επίλυση προβλημάτων με χρήση προγραμματιστικών εργαλείων και εμπάθυνση στην αλγοριθμική σκέψη.

Η γνωστική αξία της διδασκαλίας του προγραμματισμού σε μαθητές της δευτεροβάθμιας εκπαίδευσης είναι ευρύτερα αποδεκτή, ήδη από τις αρχές τις δεκαετίας του '80. Για παράδειγμα, κατά την επίλυση προβλημάτων σε προγραμματιστικό περιβάλλον, χρησιμοποιούνται θεμελιώδεις έννοιες (όπως μεταβλητή, δομή επιλογής, δομές επανάληψης, διαδικασία κ.λπ.), οι οποίες είναι δύσκολο να οικοδομηθούν από τους μαθητές με τα παραδοσιακά διδακτικά αντικείμενα.

Ο προγραμματισμός ως γνωστική δραστηριότητα περιλαμβάνει την πρόσκτηση και την εφαρμογή τριών αλληλοεξαρτώμενων μορφών γνώσης (Bayman & Mayer 1988, Pair 1990, Τζιμογιάννης 2003, Κόμης 2005):

Συντακτική γνώση: Είναι η γνώση των ειδικών χαρακτηριστικών μιας γλώσσας προγραμματισμού και των κανόνων χρήσης της.

Εννοιολογική γνώση: Η εννοιολογική γνώση αφορά στην πλήρη κατανόηση των προγραμματιστικών δομών και αρχών. Διακρίνεται στη **σημασιολογική (semantic knowledge)** και στη **σχηματική γνώση (schematic knowledge)**. Η σημασιολογική γνώση βασίζεται σε ολοκληρωμένα εννοιολογικά μοντέλα για το σύστημα του υπολογιστή, το τι συμβαίνει στο εσωτερικό του κατά την εκτέλεση των εντολών του προγράμματος, την έννοια της μεταβλητής, της δομής επιλογής και των δομών επανάληψης κ.λπ. Η σχηματική γνώση συνίσταται στο ρεπερτόριο ρουτινών και αλγορίθμων που διαθέτει ο μαθητής. Σε αντίθεση με τους μαθητές, οι έμπειροι προγραμματιστές έχουν ένα μεγάλο, δομημένο σύνολο ρουτινών, τις οποίες εύκολα μπορούν να ανακαλέσουν και να εφαρμόσουν σε νέες καταστάσεις.

Στρατηγική γνώση (μεταγνώση): Είναι η ικανότητα εφαρμογής των συντακτικών και εννοιολογικών γνώσεων για την επίλυση αυθεντικών προβλημάτων στον προγραμματισμό. Βασίζεται στην ανάπτυξη δεξιοτήτων υψηλού επιπέδου για το σχεδιασμό προγραμμάτων (ανάλυση-σύνθεση, διατύπωση συνθηκών και αιτιακών συσχετισμών κ.λπ.) και στην ικανότητα μεταφοράς δεξιοτήτων επίλυσης προβλημάτων από/σε άλλα γνωστικά αντικείμενα.

Με βάση τα πορίσματα της γνωστικής ψυχολογίας διακρίνονται δύο τύποι γνώσεων (Anderson 1983):

Δηλωτική γνώση (declarative knowledge): Ορίζεται ως η γνώση που αφορά γεγονότα, έννοιες και αρχές: **γνωρίζω κάτι**. Η βάση της γνώσης αυτής οργανώνεται σε εννοιολογικές δομές που λέγονται **σχήματα**. Σε ότι αφορά στον προγραμματισμό, οι δηλωτικές γνώσεις αφορούν στο συντακτικό της γλώσσας και στις βασικές αρχές του προγραμματισμού (π.χ. αναγνώριση του ρόλου μιας υπολογιστικής δομής ή ενός τμήματος αλγορίθμου).

Διαδικαστική γνώση (procedural knowledge): Αναφέρεται στην αποτελεσματική χρήση και εφαρμογή των δηλωτικών γνώσεων για την επίλυση προβλημάτων: **γνωρίζω πώς και γιατί**. Η διαδικαστική γνώση στον προγραμματισμό σχετίζεται με

- την ικανότητα εφαρμογής των σχετικών γνώσεων (συντακτικών και εννοιολογικών) για την ανάπτυξη ολοκληρωμένων προγραμμάτων
- τη μετατροπή των δηλωτικών γνώσεων σε προγραμματιστικές ρουτίνες και διαδικασίες
- τη δυνατότητα αυτοματοποιημένης ανάκλησης και εφαρμογής των διαδικασιών.

Είναι προφανές ότι, στη γνωστική περιοχή του προγραμματισμού και ειδικότερα στο πλαίσιο της επίλυσης προβλημάτων, οι δύο αυτοί τύποι γνώσεων δύσκολα μπορούν να διακριθούν και να αντιμετωπιστούν ξεχωριστά. Στον Πίνακα 1 δίνεται το πλαίσιο γνώσεων στον προγραμματισμό και οι μεταξύ τους συσχετίσεις.

Θα πρέπει να σημειωθεί ότι, σύμφωνα με το Πρόγραμμα Σπουδών της Πληροφορικής (ΥΠΕΠΘ 1998), η έμφαση δεν δίνεται στις συντακτικές γνώσεις και στα ιδιαίτερα χαρακτηριστικά του χρησιμοποιούμενου περιβάλλοντος προγραμματισμού αλλά στις

εννοιολογικές γνώσεις και στις δεξιότητες αλγοριθμικής επίλυσης προβλημάτων. Αυτό αναδεικνύεται με επιτυχία στο σχεδιασμό του μαθήματος «*Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*», όπου προτείνεται η χρήση ενός ιδεατού περιβάλλοντος ψευδογλώσσας (ΓΛΩΣΣΑ), με στόχο να απλοποιηθεί το πλαίσιο συντακτικών δυσκολιών, να περιοριστούν οι εννοιολογικές δυσκολίες και να διευκολυνθούν οι μαθητές ώστε να εστιάσουν στα κομβικά χαρακτηριστικά της αλγοριθμικής επίλυσης προβλημάτων.

Πίνακας 1. Πλαίσιο γνώσεων στον προγραμματισμό

	Δηλωτική γνώση	Διαδικαστική γνώση
Συντακτική γνώση	<p>Γνώση των ιδιαίτερων συντακτικών χαρακτηριστικών της γλώσσας, όπως:</p> <ul style="list-style-type: none"> • Γνώση της σημασίας του ερωτηματικού στην Pascal, ως δήλωση του τέλους εντολής • Γνώση του τρόπου δήλωσης σταθερών, μεταβλητών και τύπων δεδομένων, πινάκων, υποπρογραμμάτων • Γνώση χειρισμού λογικών δεδομένων και σύνταξης συνθηκών 	<p>Δεξιότητες εφαρμογής κανόνων σύνταξης για την ανάπτυξη προγραμμάτων, όπως:</p> <ul style="list-style-type: none"> • Δεξιότητα ορθής διατύπωσης και ελέγχου της σύνταξης εμφωλευμένων δομών επιλογής • Δεξιότητα ορθής διατύπωσης και ελέγχου της σύνταξης επαναληπτικών δομών • Δεξιότητα αναζήτησης μιας τιμής που έχει καταχωριστεί σε έναν πίνακα δεδομένων
Εννοιολογική γνώση	<p>Κατανόηση της σημασίας των δράσεων που υλοποιούνται στο εσωτερικό του υπολογιστή κατά την εκτέλεση του προγράμματος</p> <ul style="list-style-type: none"> • Κατανόηση της έννοιας της μεταβλητής • Κατανόηση της λειτουργίας της δομής επιλογής • Κατανόηση της λειτουργίας των επαναληπτικών δομών • Κατανόηση του τρόπου με τον οποίο υλοποιείται η λειτουργία μιας διαδικασίας ή μιας συνάρτησης, καθώς και η επικοινωνία της με το κύριο πρόγραμμα 	<p>Ικανότητα σχεδιασμού αλγορίθμων για την επίλυση προβλημάτων όπως</p> <ul style="list-style-type: none"> • Δεξιότητα ερμηνείας της λειτουργίας ενός τμήματος προγράμματος ή ψευδοκώδικα • Δεξιότητα μετατροπής ενός διαγράμματος ροής σε πρόγραμμα ή ψευδοκώδικα • Δεξιότητα σχεδιασμού μιας διαδικασίας υπολογισμού (π.χ. μέσος όρος των στοιχείων ενός πίνακα) • Δεξιότητα σχεδιασμού υποπρογραμμάτων για την υλοποίηση απλών επεξεργασιών
Στρατηγική γνώση		<ul style="list-style-type: none"> • Ικανότητα σχεδιασμού, κωδικοποίησης και ελέγχου ενός προγράμματος με στόχο την επίλυση αυθεντικών προβλημάτων • ικανότητα μεταφοράς δεξιοτήτων επίλυσης προβλημάτων σε νέα γνωστικά πλαίσια

Γνωστικές δυσκολίες των μαθητών στον προγραμματισμό

Η διδασκαλία του προγραμματισμού σε μαθητές της δευτεροβάθμιας εκπαίδευσης συνιστά ένα εξαιρετικά ενδιαφέρον έργο που έχει ιδιαίτερα χαρακτηριστικά και διαφορές σε σχέση με τα άλλα γνωστικά αντικείμενα του Προγράμματος Σπουδών. Οι μαθητές της δευτεροβάθμιας εκπαίδευσης, όταν έρχονται για πρώτη φορά σε επαφή με τη χρήση προγραμματιστικών εργαλείων για την επίλυση προβλημάτων, αντιμετωπίζουν πολλές δυσκολίες (Bonar & Soloway 1985, Ennis 1994). Παρότι οι προϋπάρχουσες γνώσεις και εμπειρίες από συναφείς γνωστικές περιοχές, όπως είναι τα μαθηματικά και η φυσική, παίζουν σημαντικό ρόλο στην ανάπτυξη αλγοριθμικής σκέψης, στις περισσότερες περιπτώσεις όμως αυτό δεν είναι αρκετό.

Η εκπαιδευτική έρευνα έχει δώσει σημαντικές πληροφορίες σχετικά με τις εναλλακτικές αντιλήψεις μαθητών (αλλά και φοιτητών) για τις υπολογιστικές δομές και τις εννοιολογικές δυσκολίες που αντιμετωπίζουν κατά την εφαρμογή των βασικών υπολογιστικών δομών για την επίλυση προβλημάτων σε περιβάλλον προγραμματισμού (Samurçay 1989, Soloway & Sprohler 1989, Green 1990). Παρόμοια αποτελέσματα έχουν αναφερθεί στη χώρα μας σχετικά με τις παρανοήσεις μαθητών της δευτεροβάθμιας εκπαίδευσης στον προγραμματισμό, όπως είναι η έννοια της μεταβλητής (Τζιμογιάννης & Κόμης 2000) και του πίνακα (Τζιμογιάννης & Γεωργίου 1998), και τις δυσκολίες στην εφαρμογή της δομής επιλογής για την επίλυση απλών προβλημάτων (Τζιμογιάννης & Κόμης 1999). Πολλοί μαθητές δεν μπορούν να γράψουν ολοκληρωμένα και λογικά ορθά προγράμματα ακόμη και μετά από πολλά μαθήματα στον προγραμματισμό.

Με βάση τα αποτελέσματα της εκπαιδευτικής έρευνας και της διδακτικής εμπειρίας έχουμε κωδικοποιήσει ένα σύστημα γνωστικών δυσκολιών που αντιμετωπίζουν οι μαθητές της δευτεροβάθμιας εκπαίδευσης στον προγραμματισμό, το οποίο θεωρούμε ότι θα πρέπει να γνωρίζει ο καθηγητής της Πληροφορικής:

1. Η διδασκαλία του προγραμματισμού περιλαμβάνει πολύ περισσότερα πράγματα από τη διδασκαλία του συντακτικού και της σημασιολογίας μιας γλώσσας προγραμματισμού. Οι μαθητές πρέπει να κατανοήσουν πώς **χρησιμοποιούνται και πώς λειτουργούν τα δομικά εργαλεία** της γλώσσας για την επίλυση ενός προβλήματος. Για παράδειγμα, ένα πρόγραμμα απαιτεί το χειρισμό πολλών αφηρημένων οντοτήτων που έχουν μικρή σχέση με τα στοιχεία της καθημερινής εμπειρίας του μαθητή (π.χ. λογικά δεδομένα, εμφωλευμένες δομές επιλογής, βρόχος, μετρητής, αρχικοποίηση μεταβλητής, δείκτης πίνακα, αναδρομή κ.α.).
2. Υπάρχουν εγγενή χαρακτηριστικά και δυσκολίες στο πεδίο του προγραμματισμού, καθώς είναι, συνήθως, απαραίτητο να σκεφτόμαστε σχετικά με δεδομένα και αλγορίθμους με τρόπους **πολύ διαφορετικούς από αυτούς που σκεφτόμαστε σε άλλες γνωστικές περιοχές** (π.χ. μαθηματικά ή φυσική) και, πολύ περισσότερο, στην καθημερινή μας ζωή. Έτσι, οι μαθητές συναντούν δυσκολίες να εκφράσουν λύσεις που δεν αποτελούν φυσική συνέπεια της μεταφοράς γνώσεων από άλλες γνωστικές περιοχές σε περιβάλλοντα προγραμματισμού. Για παράδειγμα, ας συγκρίνουμε τον τρόπο υπολογισμού του αθροίσματος μιας σειράς αριθμών στην Pascal ή στη ΓΛΩΣΣΑ, σε σχέση με τα μαθηματικά ή ακόμη και με τη συνάρτηση sum που χρησιμοποιεί το Excel.
3. Οι μαθητές δεν έχουν τη δυνατότητα διαπραγμάτευσης του συντακτικού και της σημασιολογίας των εντολών μιας γλώσσας προγραμματισμού. Για να μπορέσουν να χρησιμοποιήσουν αποτελεσματικά ένα προγραμματιστικό περιβάλλον για την επίλυση προβλημάτων απαιτείται η προσαρμογή τους στην **αυστηρότητα σύνταξης και δόμησης** του προγράμματος, η οποία είναι εγγενής στον προγραμματισμό, και αυτό δεν επιτυγχάνεται εύκολα.
4. Η κατανόηση της δομής του προγράμματος και της **βηματικής φύσης** (step by step) του αλγορίθμου συνιστούν σημαντικό εμπόδιο στην ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων σε προγραμματιστικό περιβάλλον. Πρόσθετες εννοιολογικές δυσκολίες

αποτελούν οι ανεπαρκείς ή ελλιπείς αναπαραστάσεις για το ρόλο των μεταβλητών ως στοιχεία αποθήκευσης δεδομένων, για την έννοια του πίνακα, για τον έλεγχο της ροής του προγράμματος (δομές επιλογής, δομές επανάληψης, αναδρομή κ.λπ.), το ρόλο και τη χρήση υποπρογραμμάτων κ.α.

5. Ένα πρόγραμμα που εκτελείται στον υπολογιστή αποτελεί ένα **μηχανισμό, ο οποίος δεν είναι εύκολα προσιτός στους μαθητές**. Οι δεξιότητες που είναι απαραίτητες, ώστε οι μαθητές να εμπεδώσουν το πρόγραμμα ως μια ενιαία οντότητα, να κατανοήσουν τα κύρια μέρη του και τις σχέσεις μεταξύ τους, απαιτούν χρόνο και αναπτύσσονται σταδιακά. Η έλλειψη επαρκών αναπαραστάσεων για τη ροή των δεδομένων, το ρόλο και τη λειτουργία των βασικών μονάδων του υπολογιστή φαίνεται να παρεμβαίνει καθοριστικά στο μοντέλο της μηχανής που οικοδομούν οι μαθητές (Τζιμογιάννης & Κόμης 2004). Η διδακτική εμπειρία δείχνει ότι, σε πολλές περιπτώσεις, η μηχανή αντιμετωπίζεται από τους μαθητές ως μαύρο κουτί (black box). Τα μοντέλα και οι αντιλήψεις των μαθητών εισάγουν πρόσθετα γνωστικά εμπόδια στο να κατανοήσουν το επίπεδο λεπτομέρειας, το οποίο απαιτείται για να περιγραφεί μια υπολογιστική διαδικασία σε γλώσσα προγραμματισμού.
6. Επιπρόσθετα, υπάρχουν ιδιαίτερα χαρακτηριστικά στη διδασκαλία του προγραμματισμού, τα οποία σχετίζονται με το ρόλο της μηχανής. Ο υπολογιστής και το περιβάλλον προγραμματισμού έχουν διπλό ρόλο: αποτελούν ένα **μηχανισμό** ο οποίος μπορεί να χρησιμοποιηθεί για την **ανάπτυξη και εκτέλεση άλλων μηχανισμών** (προγραμμάτων). Ο Du Boulay (1989) έχει εισάγει τον όρο **ενοσιολογική μηχανή (notional machine)** για να περιγράψει το ρόλο του υπολογιστή στον προγραμματισμό και τις αντιλήψεις που έχουν οι αρχάριοι γι' αυτόν. Πρόκειται για έναν ιδεατό ενοσιολογικό υπολογιστή (σε επίπεδο γλώσσας και όχι υλικού), του οποίου τα χαρακτηριστικά καθορίζονται από τα ιδιαίτερα στοιχεία και τις κατασκευές της χρησιμοποιούμενης γλώσσας προγραμματισμού. Η ενοσιολογική μηχανή θα πρέπει να έχει δύο βασικά χαρακτηριστικά ώστε να είναι αποτελεσματική στην ανάπτυξη ενοσιολογικών μοντέλων από τους μαθητές: α) να είναι ενοσιολογικά απλή και β) να παρέχει δυνατότητες στους μαθητές να αναπτύξουν επαρκείς αναπαραστάσεις για τις βασικές λειτουργίες της (Du Boulay et al., 1999).
7. Η μελέτη των έμπειρων προγραμματιστών δείχνει ότι οι προγραμματιστικές γνώσεις είναι οργανωμένες σε **σύνθετες ενοσιολογικές δομές** και δεν επικεντρώνονται στο συντακτικό της χρησιμοποιούμενης γλώσσας. Οι δομές αυτές περιλαμβάνουν ένα **ευρύ ρεπερτόριο αλγορίθμων** μαζί με τις σχετικές πληροφορίες και τα ενοσιολογικά εργαλεία. Αντίθετα, οι νέοι προγραμματιστές δεν έχουν αναπτύξει υποδείγματα προγραμματισμού που να επιτρέπουν ένα καινούργιο πρόβλημα να προσαρμοστεί σε μια λύση που είναι γνωστή ή έχει διδαχθεί προηγουμένα. Συνήθως οργανώνουν φτωχά τις προγραμματιστικές τους γνώσεις, σε ένα πλαίσιο προσανατολισμένο στο συντακτικό της γλώσσας, με αποτέλεσμα να μη μπορούν εύκολα να τις εφαρμόσουν για την επίλυση προβλημάτων που δεν είναι εκ των προτέρων γνωστά. Οι περισσότεροι μαθητές έχουν αδυναμίες να λειτουργήσουν αφαιρετικά σε ένα πρόγραμμα, να συνθέσουν νέους αλγορίθμους και να προσαρμόσουν εντολές ή υπολογιστικές διαδικασίες, αξιοποιώντας αποτελεσματικά τις προηγούμενες γνώσεις τους. Οι δεξιότητες που είναι απαραίτητες, ώστε οι μαθητές να εμπεδώσουν το πρόγραμμα ως μια ενιαία οντότητα, να κατανοήσουν τα κύρια μέρη του και τις σχέσεις μεταξύ τους απαιτούν χρόνο και αναπτύσσονται σταδιακά.
8. Τα συνήθη προγραμματιστικά περιβάλλοντα και οι γλώσσες προγραμματισμού που χρησιμοποιούνται για εκπαιδευτικούς σκοπούς έχουν σχεδιαστεί για την ανάπτυξη εφαρμογών **και όχι για διδασκαλία**. Είναι, συνεπώς, προσαρμοσμένα στο πλαίσιο γνώσεων και δεξιοτήτων των έμπειρων προγραμματιστών, γεγονός που ενισχύει τις δυσκολίες και τα εμπόδια που συναντούν οι μαθητές και οι αρχάριοι στον προγραμματισμό.
9. Σε πολλές περιπτώσεις οι ίδιες οι γλώσσες προγραμματισμού ενσωματώνουν **ανθρωπομορφικά χαρακτηριστικά**, τα οποία εισάγουν πρόσθετες δυσκολίες και

παρανοήσεις στους μαθητές. Η γλώσσα που χρησιμοποιούν συχνά οι εκπαιδευτικοί ενισχύει τις αντιλήψεις αυτές (για παράδειγμα, «Έτσι που έγραψες την εντολή ο υπολογιστής θεωρεί ότι ...»). Τα πρώτα διδακτικά παραδείγματα αλγορίθμων βασίζονται σε μια σειρά οδηγιών (βημάτων) που εκτελεί ο άνθρωπος για την υλοποίηση εργασιών της καθημερινής ζωής (π.χ. πώς παρασκευάζεται ένα γλυκό, πώς υλοποιείται μια κατασκευή κ.λπ.). Η χρήση παραδειγμάτων του τύπου αυτού για την εισαγωγή των μαθητών στην αλγοριθμική επίλυση προβλημάτων οδηγεί, συνήθως, σε μια σειρά λανθασμένων αντιλήψεων, όπως ότι

- η γλώσσα σύνταξης των εντολών είναι ελεύθερη
- υπάρχουν εντολές που μπορούν να υποστηρίξουν με απλό τρόπο κάθε περίπλοκη υπολογιστική διαδικασία.

Από τις εμπειρικές στις παιδαγωγικές διδακτικές προσεγγίσεις

Για μια μεγάλη περίοδο, ιδιαίτερα μέχρι τα μέσα της δεκαετίας του '80, η διδασκαλία της Πληροφορικής ταυτιζόταν, ουσιαστικά, με τη διδασκαλία του προγραμματισμού, τόσο στην τριτοβάθμια όσο και στη δευτεροβάθμια εκπαίδευση. Το μοντέλο που ακολουθήθηκε, επηρεάστηκε από τις *τεχνοκεντρικές προσεγγίσεις*, που ήταν κυρίαρχες κατά την περίοδο αυτή, και είχε ως κύριο στόχο την εκμάθηση συγκεκριμένων γλωσσών προγραμματισμού. Το πλαίσιο διδασκαλίας που διαμορφώθηκε ήταν προσανατολισμένο στα χαρακτηριστικά της χρησιμοποιούμενης γλώσσας προγραμματισμού, με έμφαση στη δομή, στο λεξιλόγιο και τους συντακτικούς κανόνες της γλώσσας.

Οι κυρίαρχες διδακτικές προσεγγίσεις ακολουθούσαν ένα 'επικοινωνιακό' μοντέλο, στο οποίο ο δάσκαλος μεταδίδει τις κατάλληλες προγραμματιστικές γνώσεις στους μαθητές που καλούνται να τις εφαρμόζουν, στη συνέχεια, στο περιβάλλον μιας γλώσσας. Το μοντέλο αυτό εμπνέεται, σε μεγάλο βαθμό, από τη **συμπεριφοριστική προσέγγιση** για τη διδασκαλία και τη μάθηση. Στον Πίνακα 2 δίνονται τα βασικά χαρακτηριστικά του τεχνοκεντρικού-συμπεριφοριστικού μοντέλου διδασκαλίας του προγραμματισμού και οι ρόλοι διδάσκοντα και μαθητή.

Πίνακας 2. Συμπεριφοριστικό μοντέλο διδασκαλίας του προγραμματισμού

Διδάσκων	Μαθητής
Πηγή πληροφοριών και γνώσεων στον προγραμματισμό	Παθητικός δέκτης πληροφοριών, εντολών, προγραμμάτων
Μεταφορέας γνώσεων, αναλύει και εξηγεί εντολές και προγράμματα	Απομνημονεύει συντακτικούς κανόνες, εντολές, προγράμματα
Δίνει έμφαση στην εκμάθηση των εντολών και του συντακτικού μιας γλώσσας	Συντάσσει προγράμματα αναπαράγοντας εντολές, μαθαίνει έτοιμους αλγορίθμους
Δίνει έμφαση στην προγραμματιστική αυστηρότητα και όχι στο σχεδιασμό αλγορίθμων	Εξασκείται σε τυποποιημένα προβλήματα και προγράμματα
Δίνει μεγάλη σημασία στην εξάσκηση και στην επίλυση τυπικών προβλημάτων	Η μάθηση είναι μοναχική διαδικασία
Ελέγχει και κατευθύνει τη μάθηση των μαθητών	

Η προσέγγιση αυτή ακολουθήθηκε και στη χώρα μας στα πλαίσια της διδασκαλίας του προγραμματισμού στα ενιαία πολυκλαδικά λύκεια και στα τεχνικά επαγγελματικά λύκεια (με τη διδασκαλία των γλωσσών προγραμματισμού Basic, Pascal και Cobol), καθώς και στο γυμνάσιο (με τη διδασκαλία της Logo και της Basic). Στον ελληνικό χώρο, η εμπειρία από τη διδασκαλία του προγραμματισμού στην τεχνική εκπαίδευση και κατάρτιση έχει αποδείξει ότι το εκπαιδευτικό αποτέλεσμα ήταν, συχνά, περιορισμένο. Οι περισσότεροι μαθητές επικεντρώνονταν στην αποστήθιση εντολών, τμημάτων κώδικα ή/και ολόκληρων

διαδικασιών και δεν εξασκούνταν στην ανάπτυξη δεξιοτήτων και μεθοδολογιών επίλυσης προβλημάτων. Από την άλλη μεριά, συναντούσαν μεγάλες δυσκολίες στην ανάπτυξη ‘γνωστών’ αλγορίθμων στο περιβάλλον μιας νέας γλώσσας προγραμματισμού (Τζιμογιάννης και Γεωργίου 1999). Χαρακτηριστικό παράδειγμα ήταν η μετάβαση από τη Basic στην Pascal για τους μαθητές των τεχνικών επαγγελματικών λυκείων.

Η ερευνητική μελέτη ανέδειξε από πολύ νωρίς τα αδιέξοδα που εμπειρείχε η εμπειρική-τεχνοκεντρική προσέγγιση (Dahl et al. 1972). Το ενδιαφέρον στη διδασκαλία του προγραμματισμού εστιάζεται, πλέον, στις μορφές συλλογισμού και στα νοητικά μοντέλα που χρησιμοποιούν οι αρχάριοι και οι έμπειροι προγραμματιστές, καθώς και στις μεθόδους εργασίας με στόχο τη σύλληψη αλγορίθμων και την ανάπτυξη προγραμμάτων.

Η ιδέα της **ανακαλυπτικής προσέγγισης** του προγραμματισμού (programming by discovery) επικεντρώνεται στο σχεδιασμό εκπαιδευτικών περιβαλλόντων που θα βοηθούν αποτελεσματικά τους αρχάριους προγραμματιστές (μαθητές ή φοιτητές) στη σύνταξη προγραμμάτων (Ramadhan 2000). Εξερευνώντας τα περιβάλλοντα αυτά, οι μαθητές γίνονται ενεργά υποκείμενα της μάθησης και υποστηρίζονται με στόχο την κατανόηση της δυναμικής συμπεριφοράς του προγράμματος, την ανίχνευση των σφαλμάτων του προγράμματος και των παρανοήσεών τους που σχετίζονται με αυτά. Σε τελική ανάλυση, οι μαθητές μαθαίνουν μέσα από διαδικασίες δοκιμής και άμεσης παρατήρησης του αποτελέσματος των προγραμμάτων τους στην οθόνη του υπολογιστή. Στον Πίνακα 3 δίνονται τα βασικά χαρακτηριστικά του ανακαλυπτικού μοντέλου και οι ρόλοι διδάσκοντα και μαθητή κατά τη μάθηση του προγραμματισμού.

Πίνακας 3. Ανακαλυπτικό μοντέλο μάθησης του προγραμματισμού

Διδάσκων	Μαθητής
Έχει βαθιά και πολύπλευρη γνώση του προγραμματισμού	Γίνεται ενεργό υποκείμενο της μάθησης αναπτύσσοντας προγράμματα
Είναι εξοικειωμένος με την επιστημονική και την πειραματική μεθοδολογία	Ανακαλύπτει έννοιες και συσχετίσεις μετά από διαδικασίες δοκιμής και ελέγχου
Είναι διευκολυντής των μαθησιακών δραστηριοτήτων στην τάξη ή στο εργαστήριο	Μαθαίνει να ελέγχει και να διορθώνει τα λάθη στα προγράμματα που αναπτύσσει
Δημιουργεί κλίμα εμπιστοσύνης και σιγουριάς στους μαθητές	Μαθαίνει να σχεδιάζει λύσεις, να διατυπώνει επιχειρήματα, ιδέες και συμπεράσματα
Παρέχει έτοιμα προγράμματα για εξάσκηση, πειραματισμό, ανίχνευση λαθών	Εξοικειώνεται με μεθοδολογίες και στρατηγικές επίλυσης προβλημάτων
Επιδιώκει την επίλυση ανοικτών προβλημάτων και προωθεί την ανακάλυψη των βασικών χαρακτηριστικών των αλγορίθμων από τους ίδιους τους μαθητές	

Το **εποικοδομιστικό μοντέλο** έχει ως βασική διδακτική παραδοχή ότι οι γνώσεις δεν μεταδίδονται αλλά ‘οικοδομούνται’ και αναδομούνται ατομικά από τον μαθητή, ο οποίος είναι υπεύθυνος για τη μάθησή του. Η διδασκαλία του προγραμματισμού στο πλαίσιο των αρχών του εποικοδομισμού αποτέλεσε αντικείμενο ενδιαφέροντος και μελέτης από τις αρχές της δεκαετίας του ’80 (Papert 1980). Σε αντίθεση με τη θεώρηση του συμπεριφορισμού, οι **εποικοδομιστές υποστηρίζουν ότι η μαθησιακή διαδικασία του προγραμματισμού δεν μπορεί να πραγματοποιηθεί ουσιαστικά εάν δεν λάβει υπόψη τον τρόπο, με τον οποίο οικοδομούν τις γνώσεις τους τα υποκείμενα που μαθαίνουν** (Ben-Ari 2001, Κόμης 2001).

Οι εποικοδομιστικές μαθησιακές διαδικασίες τοποθετούν το μαθητή στο κέντρο της μάθησης. Ο μαθητής συμμετέχει ενεργά στη μαθησιακή διαδικασία και οικοδομεί τις γνώσεις του με βάση μια διερεύνηση της πραγματικότητας, η οποία συμπεριλαμβάνει επιπλέον, τη γνώση μαζί με τις διάφορες μορφές μετάδοσής της (διδασκαλία, επικοινωνία, συνεργασία κ.λπ.). Η πρόσκτηση της νέας γνώσης γίνεται με μη γραμμικό τρόπο και βασίζεται πάνω σε

εξατομικευμένες οικοδομήσεις αλλά και σε συλλογικές καταστάσεις της τάξης, όπου εμφανίζονται γνωστικές συγκρούσεις, ικανές να προωθήσουν την οικοδόμηση των γνώσεων. Στον Πίνακα 4 δίνονται τα βασικά χαρακτηριστικά του εποικοδομιστικού μοντέλου μάθησης του προγραμματισμού και οι ρόλοι διδάσκοντα και μαθητή.

Πίνακας 4. Εποικοδομιστικό μοντέλο μάθησης του προγραμματισμού

Διδάσκων	Μαθητής
Παρέχει και διαμορφώνει ευκαιρίες για μάθηση, οικοδόμηση γνώσεων και ανάπτυξη δεξιοτήτων	Συμμετέχει ενεργά σε προγραμματιστικές δραστηριότητες μάθησης
Εκτιμά και λαμβάνει υπόψη τις προϋπάρχουσες γνώσεις των μαθητών στον προγραμματισμό	Διαμορφώνει δομές γνώσεων και οικοδομεί ένα ικανό ρεπερτόριο ρουτινών
Λαμβάνει υπόψη τις γνωστικές δυσκολίες των μαθητών στον προγραμματισμό	Οικοδομεί νέες γνώσεις και αναπτύσσει δεξιότητες
Συμμετέχει στη μάθηση	Συnergάζεται με συμμαθητές και διδάσκοντα
Είναι συντονιστής και καθοδηγητής των μαθητικών δραστηριοτήτων	Μαθαίνει πώς να μαθαίνει (αναλύει, συνθέτει, αξιολογεί)
	Αναπτύσσει μεταγνωστικές δεξιότητες

Τα τελευταία χρόνια, αναπτύσσεται έντονο ενδιαφέρον για το σχεδιασμό εποικοδομητικών μαθησιακών δραστηριοτήτων στον προγραμματισμό που βασίζονται στη **διερευνητική μάθηση (inquiry learning)** (Ramadhan 2000, Kolikant & Pollack 2004, Τσέλιος κ.α. 2006) και στη **συνεργατική μάθηση** (Γρηγοριάδου κ.α. 2004). Οι Williams & Kessler (2000) υποστηρίζουν ότι η **συνεργασία μεταξύ των μαθητών αποτελεί καθοριστικό παράγοντα στη μάθηση του προγραμματισμού**. Οι προσεγγίσεις αυτές δίνουν έμφαση στον παιδαγωγικό σχεδιασμό της διδασκαλίας του προγραμματισμού και στη μετατόπιση από το συντακτικό στην καλλιέργεια δεξιοτήτων επίλυσης προβλημάτων (αναλυτική-συνθετική σκέψη, αφαιρετική ικανότητα, μοντελοποίηση λύσεων). Στο πλαίσιο αυτό προτείνεται, επίσης, ο σχεδιασμός κατάλληλων διδακτικών στρατηγικών με χρήση εκπαιδευτικών περιβαλλόντων προσομοίωσης αλγορίθμων (Jehng et al. 1999, Τζιμογιάννης κ.α. 2006).

Το πλαίσιο εποικοδόμησης εννοιών του προγραμματισμού

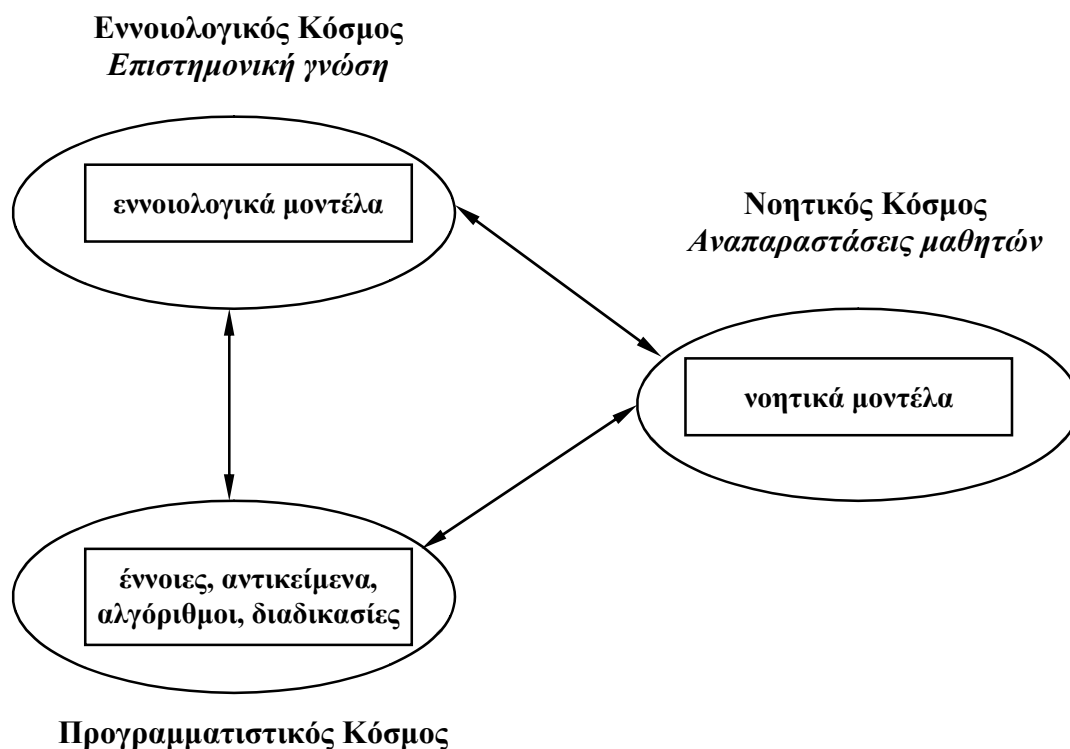
Η γνωστική σύγκρουση και η εννοιολογική αλλαγή για τις θεμελιώδεις έννοιες και τα δομικά στοιχεία του προγραμματισμού, καθώς και η ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων είναι διαδικασίες προοδευτικές και πραγματοποιούνται σε βάθος χρόνου. Οι **γνωστικές πηγές** που χρησιμοποιούν οι μαθητές για την ανάπτυξη νέων γνώσεων στον προγραμματισμό είναι

- η καθημερινή τους εμπειρία και το περιβάλλον που δραστηριοποιούνται και αναπτύσσονται
- οι νοητικές διεργασίες που αναπτύσσουν
- τα συστήματα αναπαράστασεων και νοητικών μοντέλων που χρησιμοποιούν.

Τα μοντέλα αποτελούν μια τεχνική αναπαράστασης της συμπεριφοράς στοιχείων, καταστάσεων και υπολογιστικών διαδικασιών του προγραμματισμού. Ως **μοντέλο** ορίζεται ένα **νοητικό σχήμα** που είναι **οικείο ή προσιτό στην προϋπάρχουσα γνώση** και βοηθάει τα άτομα να φτάσουν στην κατανόηση των νέων γνώσεων. Αποτελεί μια εσωτερική αναπαράσταση που παίζει το ρόλο του υποκατάστατου της πραγματικότητας.

Τα μοντέλα είναι, συνήθως, σύνθετα εννοιολογικά συστήματα που αναπτύσσονται από τους μαθητές δύσκολα και με αργούς ρυθμούς. Στο πλαίσιο οικοδόμησης νέων στον προγραμματισμό μπορούμε να διακρίνουμε τρεις διαφορετικούς κόσμους (Σχήμα 1):

- τον **προγραμματιστικό κόσμο (programming world)**, όπου ‘ζουν’ τα διάφορα υπολογιστικά αντικείμενα και διαδραματίζονται οι συσχετίσεις μεταξύ τους (μεταβλητές, υπολογιστικές δομές, δομές δεδομένων, υποπρογράμματα κ.λπ.).
- τον **εννοιολογικό κόσμο (conceptual world)** που περιλαμβάνει τα μοντέλα εννοιών, αντικειμένων και διαδικασιών του προγραμματιστικού κόσμου, τα οποία αναπτύσσουν οι έμπειροι προγραμματιστές.
- το **νοητικό κόσμο (mental world)** που αποτελείται από τις υποκειμενικές, ατομικές αντιλήψεις που δημιουργούνται στο μυαλό των μαθητών. Συνήθως, πρόκειται για ατελή μοντέλα που αντικατοπτρίζουν τις παρανοήσεις και τις εννοιολογικές δυσκολίες που αντιμετωπίζουν οι μαθητές σε μια συγκεκριμένη περιοχή ή σε ένα πρόβλημα.



Σχήμα 1. Το πλαίσιο οικοδόμησης εννοιών προγραμματισμού

Η διδασκαλία θα πρέπει να παρέχει στους μαθητές ευκαιρίες ώστε να μετασχηματίσουν τα δικά τους ατελή νοητικά μοντέλα σε επιστημονικά επαρκή μοντέλα που θα τους βοηθήσουν να κατανοήσουν τις βασικές έννοιες του προγραμματισμού και τις αρχές της αλγοριθμικής επίλυσης προβλημάτων.

Οι παρανοήσεις και τα εναλλακτικά νοητικά μοντέλα των μαθητών προκαλούν **σφάλματα (bugs)**, άμεσα παρατηρήσιμα στην οθόνη του υπολογιστή κατά την εκτέλεση των προγραμμάτων. Ο υπολογιστής παίζει σημαντικό ρόλο στη διδασκαλία του προγραμματισμού. Δεν αποτελεί απλά τη μηχανή που εκτελεί το πρόγραμμα. **Η μεγάλη παιδαγωγική του αξία οφείλεται στο ότι επιτρέπει τη σύνταξη, τον έλεγχο και τη διόρθωση του προγράμματος από τους ίδιους τους μαθητές οι οποίοι, ερχόμενοι σε άμεση επαφή με τη μηχανή, παίρνουν **ανάδραση** στις δικές τους αναπαραστάσεις για τις εμπλεκόμενες έννοιες και αντικείμενα του προγραμματισμού.** Με άλλα λόγια, ο υπολογιστής παρέχει στους μαθητές δυνατότητες εξάσκησης, πειραματισμού και ελέγχου των νοητικών μοντέλων για τα προγραμματιστικά αντικείμενα και τις λύσεις των υπό μελέτη προβλημάτων.

Σχεδιασμός μαθησιακών δραστηριοτήτων για τον προγραμματισμό

Οι σύγχρονες παιδαγωγικές προσεγγίσεις έχουν επηρεαστεί από το ευρύτερο θεωρητικό πλαίσιο του **εποικοδομισμού**. Με βάση την προβληματική που αναπτύχθηκε προηγουμένα, προτείνουμε ένα πλαίσιο παιδαγωγικού σχεδιασμού της διδασκαλίας του προγραμματισμού, το οποίο θα δίνει έμφαση στη μετατόπιση

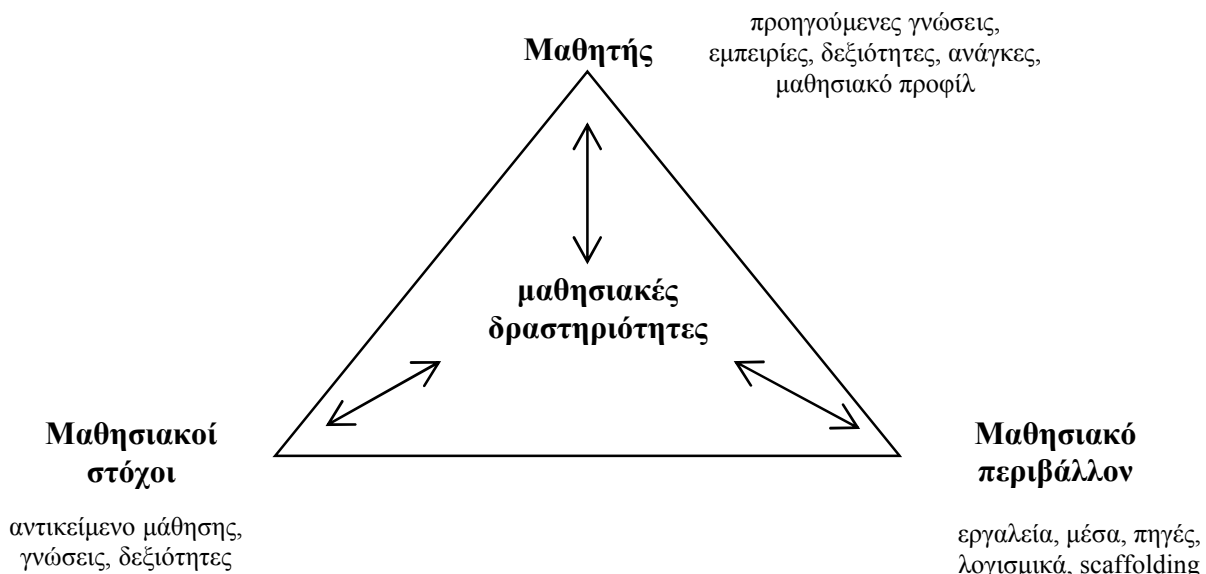
- από τη διδασκαλία στη ενεργητική μάθηση μέσω δραστηριοτήτων
- από το συντακτικό της χρησιμοποιούμενης γλώσσας στην καλλιέργεια δεξιοτήτων επίλυσης προβλημάτων (αναλυτική σκέψη, αφαιρετική ικανότητα, μοντελοποίηση λύσεων).

Το προτεινόμενο πλαίσιο οικοδόμησης νέων γνώσεων στον προγραμματισμό λαμβάνει υπόψη τις **προϋπάρχουσες γνώσεις** και εναλλακτικές **αντιλήψεις** των μαθητών με στόχο την πρόκληση καταστάσεων **γνωστικής σύγκρουσης**. Επιπλέον, αξιοποιεί τα **πλεονεκτήματα της διερευνητικής μάθησης** και της **καθοδηγούμενης ανακάλυψης** μέσα από το σχήμα:

- Παράθεση ερωτημάτων-υποθέσεων
- Σχεδιασμός λύσης - Πρόβλεψη
- Παρατήρηση
- Εναλλακτικές ερμηνείες
- Απόδειξη (χρησιμοποίηση τεχνικών ελέγχου)
- Επικοινωνία
- Ανάδραση-Συνεργασία.

Βασικός παράγοντας είναι ο **πλουραλισμός** και η **πολλαπλότητα** των χρησιμοποιούμενων μεθόδων, των μέσων και των περιβαλλόντων διδασκαλίας του προγραμματισμού, έτσι ώστε να προσφέρονται στους μαθητές δυνατότητες διαπραγμάτευσης των ιδεών τους και ευκαιρίες **συνεργατικής μάθησης**.

Το πλαίσιο σχεδιασμού των μαθησιακών δραστηριοτήτων για τον προγραμματισμό ορίζεται από τρεις αλληλοεξαρτώμενες συνιστώσες (Σχήμα 2):



Σχήμα 2. Το πλαίσιο σχεδιασμού μαθησιακών δραστηριοτήτων

Μαθητής: Γνωστικές ανάγκες, προϋπάρχουσες γνώσεις, εμπειρίες και δεξιότητες, μαθησιακό προφίλ, μαθησιακά ενδιαφέροντα.

Μαθησιακοί στόχοι: Αντικείμενο μάθησης, γνώσεις και δεξιότητες που προσδιορίζονται από το Πρόγραμμα Σπουδών.

Μαθησιακό περιβάλλον: Προγραμματιστικά εργαλεία, διδακτικό υλικό, μέσα, πηγές, εκπαιδευτικό λογισμικό και μαθησιακή υποστήριξη (scaffolding).

Ο κατάλληλος σχεδιασμός της μαθησιακής υποστήριξης αποτελεί τον πιο σημαντικό παράγοντα για μια αποτελεσματική μαθησιακή δραστηριότητα στον προγραμματισμό. Ο εκπαιδευτικός θα πρέπει να έχει υπόψη

- τις **πρωταρχικές αντιλήψεις και παρανοήσεις** των μαθητών για τον προγραμματισμό και τα πληροφορικά αντικείμενα
- την **έλλειψη επαρκών αναπαραστάσεων και νοητικών μοντέλων** που έχουν, εν γένει, οι μαθητές για τη λειτουργία του υπολογιστή και την εννοιολογική μηχανή, τα βασικά εργαλεία του προγραμματισμού, τις υπολογιστικές δομές και διαδικασίες
- τις δυσκολίες να αντιμετωπίσουν το πρόγραμμα (ή τον αλγόριθμο) ως μια ολότητα και να διακρίνουν τις επιμέρους ρουτίνες και το ρόλο τους, καθώς και τη σχέση κάθε εντολής με δράσεις που επάγονται στην εννοιολογική μηχανή
- τις **δυσκολίες** πρόσκτησης δεξιοτήτων και τεχνικών σχεδίασης αλγορίθμων και προγραμμάτων, τις οποίες χρησιμοποιούν οι έμπειροι προγραμματιστές
- **ιδανικά παραδείγματα** πρόκλησης γνωστικής σύγκρουσης και ενεργοποίησης διαδικασιών οικοδόμησης εννοιών του προγραμματισμού
- τα **αρνητικά πρότυπα** που δημιουργούνται από την επίλυση τυποποιημένων προβλημάτων στα μαθηματικά ή στη φυσική και τις δυσκολίες εξοικείωσης με τη διερευνητική μεθοδολογία (υπόθεση-έλεγχος-συμπέρασμα)
- την **αρνητική στάση** που, συνήθως, έχουν οι μαθητές για τον προγραμματισμό, η οποία οφείλεται σε μεγάλο βαθμό στις δυσκολίες που συναντούν στα εισαγωγικά μαθήματα και στην εξοικείωσή τους με άλλα ελκυστικότερα περιβάλλοντα, όπως είναι τα ηλεκτρονικά παιχνίδια και το Διαδίκτυο.

Η μαθησιακή υποστήριξη μπορεί να δοθεί είτε από τα χρησιμοποιούμενα μέσα (φύλλο εργασίας μαθητή, περιβάλλον της Αλγοριθμικής) είτε/και από τον ίδιο εκπαιδευτικό. Θα πρέπει να περιλαμβάνει τρία διακριτά επίπεδα:

- **Καθοδήγηση**
 - Περίγραμμα διδακτικών στόχων
 - Περιγραφή δραστηριότητας και εργασίας των μαθητών (παραδοτέο)
 - Ερμηνεία-εξήγηση δύσκολων εννοιών και αποριών
 - Οδηγίες χρήσης του συνοδευτικού εκπαιδευτικού υλικού
 - Οδηγίες εργασίας των μαθητών(π.χ. trial and error)
- **Διαμεσολάβηση**
 - Υποδείξεις, υπενθύμιση γνωστών
 - Υποδείξεις αυτορρύθμισης του μαθητή
 - Εξατομίκευση οδηγιών
 - Καθοδήγηση εργασίας, και συνεργασίας
 - Ενίσχυση, ενθάρρυνση
- **Υποχώρηση** της διαμεσολάβησης του διδάσκοντα, όταν οι μαθητές προχωρούν αυτόνομα και δεν έχουν ανάγκη γνωστικής υποστήριξης.

Φάσεις σχεδιασμού μαθησιακών δραστηριοτήτων

Προσδιορισμός των διδακτικών στόχων της δραστηριότητας (Με βάση τους γενικούς και ειδικούς διδακτικούς στόχους του Προγράμματος Σπουδών)

Εκτίμηση των γνωστικών δυσκολιών των μαθητών (Προϋπάρχουσες ιδέες, βασικές παρανοήσεις και εννοιολογικές δυσκολίες)

Επιλογή διδακτικής και παιδαγωγικής προσέγγισης

Προσδιορισμός του τύπου της δραστηριότητας (επίλυση προβλήματος, διερευνητική δραστηριότητα, εφαρμογή γνώσεων κ.λπ.)

Σχεδιασμός των βημάτων εργασίας των μαθητών

Σχεδιασμός της γνωστικής καθοδήγησης (scaffolding)

Παράθεση διαγνωστικών ερωτήσεων-αξιολόγηση μαθητή

Οι ιδέες των μαθητών για την έννοια της μεταβλητής

Παρότι οι μαθητές αντιμετωπίζουν την έννοια της μεταβλητής από τα εισαγωγικά μαθήματα του προγραμματισμού, η οικοδόμησή της φαίνεται ότι παρουσιάζει ιδιαίτερες δυσκολίες (Samurçay, 1989, Τζιμογιάννης & Κόμης, 2000, Τζιμογιάννης κ.α., 2005). Η οικοδόμηση αποτελεσματικών αναπαραστάσεων για την έννοια της προγραμματιστικής μεταβλητής και την εντολή εκχώρησης αποτελεί πρωταρχικό βήμα για την ανάπτυξη των βασικών προγραμματιστικών γνώσεων και δεξιοτήτων (Τζιμογιάννης & Γιούνης, 2003). Είναι ιδιαίτερα σημαντικό ο εκπαιδευτικός της Πληροφορικής να λαμβάνει υπόψη τις γνωστικές δυσκολίες, τις προϋπάρχουσες ιδέες και παρανοήσεις των μαθητών, οι οποίες είναι γνωστές από τη βιβλιογραφία ή/και τη διδακτική του εμπειρία.

Ο βασικός στόχος, από τα πρώτα μαθήματα, είναι οι μαθητές να κατανοήσουν και να διακρίνουν το λογικό από το φυσικό νόημα της μεταβλητής. Ο εκπαιδευτικός θα πρέπει να επιμείνει στην ανάδειξη της **λογικής μεταβλητής**, ως εργαλείο του αλγορίθμου, και της **φυσικής μεταβλητής** ως μέσο αποθήκευσης δεδομένων του προγράμματος.

Η έννοια της πληροφορικής μεταβλητής οικοδομείται, συνήθως, πάνω στην προϋπάρχουσα γνώση της από τα μαθηματικά, με αποτέλεσμα η κοινή αντίληψη των μαθητών για τη μεταβλητή περιορίζεται στη μαθηματική αναπαράσταση, ακόμη και μετά από πολλά μαθήματα στον προγραμματισμό. Από το πρώτο μάθημα θα πρέπει να τονιστεί στους μαθητές η βασική διαφορά, δηλαδή ότι η μεταβλητή στα μαθηματικά είναι **στατική**, αφού αναπαριστά μια λειτουργική σχέση. Αντίθετα η μεταβλητή στον προγραμματισμό έχει φυσικό και λογικό περιεχόμενο, ενώ κατά την εκτέλεση του προγράμματος η τιμή της μπορεί να **τροποποιηθεί δυναμικά** μέσω εντολών εκχώρησης ή εισόδου δεδομένων.

Οι οικείες από τα μαθηματικά διαδικασίες επίλυσης (αυτές που συνήθως γίνονται με το χέρι, όπως η χρήση πίνακα τιμών) άλλοτε παίζουν θετικό ρόλο, ενισχύοντας την οικοδόμηση της έννοιας της μεταβλητής, και άλλοτε συνιστούν ένα ισχυρό γνωστικό εμπόδιο. Πολλοί μαθητές δεν κατανοούν εύκολα ότι η εντολή εκχώρησης καταχωρεί δεδομένα πάνω στην προϋπάρχουσα τιμή μιας μεταβλητής, η οποία χάνεται. Είναι πιθανό οι αναπαραστάσεις που χρησιμοποιούνται στα εισαγωγικά μαθήματα προγραμματισμού (π.χ. η αναπαράσταση του κουτιού) να ενισχύουν την παρανόηση αυτή. Επίσης, αρκετοί μαθητές της δευτεροβάθμιας εκπαίδευσης θεωρούν ότι μια μεταβλητή διατηρεί περισσότερες από μία τιμές ή ότι η έχει τη δυνατότητα να 'θυμάται' την ιστορία των εκχωρήσεων που έχουν προηγηθεί.

Πολλοί μαθητές θεωρούν ότι κατά την εκτέλεση της εντολής $y \leftarrow x$ η μεταβλητή x μένει χωρίς περιεχόμενο, καθώς έχει εκχωρήσει την τιμή της στη μεταβλητή y . Οι μαθητές θα πρέπει να κατανοήσουν ότι η τιμή μιας μεταβλητής η τιμή της μεταβλητής x **διατηρείται αναλλοίωτη**, όταν εκχωρείται σε μια άλλη μεταβλητή y . Η τιμή μιας μεταβλητής x μπορεί να

αλλάξει μόνο μέσω της εντολής εκχώρησης ή με εισαγωγή δεδομένων από το πληκτρολόγιο (ΔΙΑΒΑΣΕ x).

Η **εντολή εκχώρησης** ενέχει μια μαθηματική υπόσταση, η οποία προέρχεται από τα ονόματα των εμπλεκόμενων μεταβλητών και το σύμβολο εκχώρησης ($=$, $:=$, ή \leftarrow), το οποίο συχνά συγγέεται με το σύμβολο της ισότητας στα μαθηματικά. Όμως, υπάρχει μια ασυμμετρία στον προγραμματισμό, η οποία δημιουργεί στους μαθητές απορίες, συγχύσεις και παρανοήσεις. Για παράδειγμα, στην εντολή $x \leftarrow x + 5$, **το x στις δύο πλευρές της εντολής εκχώρησης δεν αφορά στην ίδια οντότητα**. Στο αριστερό μέρος σχετίζεται με την περιοχή μνήμης (δηλαδή τη μεταβλητή αυτή καθαυτή) ενώ στο δεξιό μέρος αντιπροσωπεύει την τρέχουσα τιμή της μεταβλητής. Για τον ίδιο λόγο δεν είναι ισοδύναμες οι εντολές $x \leftarrow y$ και $y \leftarrow x$, πράγμα που φαίνεται μαθηματικά εύλογο.

Δραστηριότητα 1

Να μελετήσετε τη δραστηριότητα **B.4.1_L1** και να επικεντρωθείτε στην προτεινόμενη προσέγγιση και στον τρόπο που αντιμετωπίζει τις προϋπάρχουσες ιδέες και τις αναμενόμενες δυσκολίες των μαθητών. Να σχεδιάσετε ένα δικό σας διδακτικό σενάριο για την έννοια της μεταβλητής και της εντολής εκχώρησης.

Η διδασκαλία ρόλων μεταβλητών

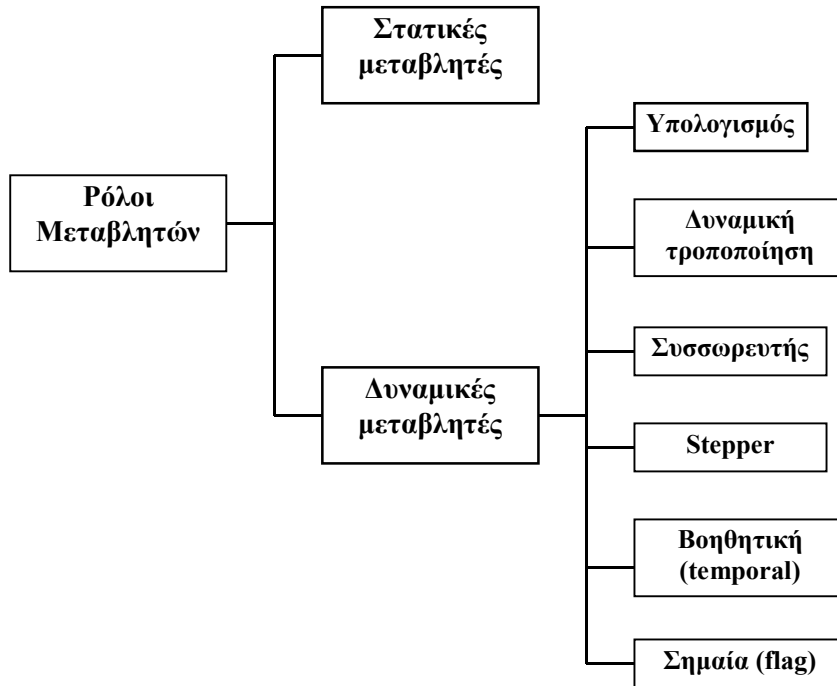
Ακολουθώντας μια εναλλακτική προσέγγιση, στο πλαίσιο της εποικοδόμησης αποτελεσματικών μοντέλων και αναπαραστάσεων για έννοιες του προγραμματισμού, έχει προταθεί η διδασκαλία των **ρόλων μεταβλητών (roles of variables)** στα εισαγωγικά μαθήματα (Sajaniemi 2005). Οι ρόλοι μεταβλητών βοηθούν στη δημιουργία λειτουργικών αναπαραστάσεων για τις μεταβλητές του προγράμματος και τις εντολές εκχώρησης, στις οποίες εμπλέκονται οι μεταβλητές αυτές. Επιπλέον, βοηθούν τους αρχάριους προγραμματιστές να διαπιστώσουν τα λάθη τους και να κατανοήσουν εκτεταμένα ή περίπλοκα προγράμματα. Οι βασικοί άξονες της προσέγγισης αυτής είναι οι εξής:

- Οι μεταβλητές δεν χρησιμοποιούνται με έναν τυχαίο, γενικό τρόπο αλλά με βάση συγκεκριμένα πρότυπα που σχετίζονται με διαφορετικούς ρόλους ή 'συμπεριφορές' που έχουν στο πρόγραμμα.
- Η τιμή μιας μεταβλητής δεν σχετίζεται με το ρόλο της στο πρόγραμμα.
- Η διδασκαλία και οι χρησιμοποιούμενες δραστηριότητες πρέπει να συμβάλλουν στη λειτουργική κατανόηση του ρόλου κάθε μεταβλητής στο πρόγραμμα. **Για το λόγο αυτό, προτείνεται ο σχεδιασμός διδακτικών καταστάσεων που θα στοχεύουν στην κατανόηση του ρόλου κάθε μεταβλητής στο πρόγραμμα και όχι στην εκμάθηση προγραμμάτων.**

Με βάση την παραπάνω προσέγγιση έχουμε ομαδοποιήσει ένα σύνολο ρόλων μεταβλητών που μπορούν να καλύψουν με επάρκεια τις γνωστικές απαιτήσεις του μαθήματος ΑΕΠΠ της Γ' Λυκείου (Σχήμα 3). Διακρίνουμε δύο μεγάλες κατηγορίες μεταβλητών με βάση το ρόλο που υποστηρίζουν στο πρόγραμμα:

α) στατικές μεταβλητές, που διαχειρίζονται δεδομένα τα οποία δεν τροποποιούνται κατά την εκτέλεση του προγράμματος. Ο ρόλος τους στο πρόγραμμα είναι, συνήθως, οργανωτικός ή επικουρικός.

β) δυναμικές μεταβλητές, που διαχειρίζονται δεδομένα τα οποία αλλάζουν δυναμικά κατά την εκτέλεση του προγράμματος. Ο ρόλος τους στο πρόγραμμα είναι πολύ σημαντικός και καθορίζεται από τη λογική του αλγορίθμου. Μπορούμε να διακρίνουμε έξι τύπους δυναμικών μεταβλητών, με βάση το ρόλο τους στο πρόγραμμα:



Σχήμα 3. Οι βασικοί ρόλοι μεταβλητών

Υπολογισμός (calculation): Χρησιμοποιείται για τον υπολογισμό της τιμής μιας μεταβλητής, η οποία προκύπτει μέσω εντολών εκχώρησης που περιλαμβάνουν άλλες μεταβλητές.

Δυναμική τροποποίηση: Αφορά σε μια μεταβλητή που αλλάζει συνεχώς τιμές, κατά μήκος του προγράμματος, και διατηρεί κάθε φορά την πιο πρόσφατη τιμή (most-recent holder).

Συσσωρευτής (gatherer): Με τον όρο αυτό περιγράφονται μεταβλητές οι οποίες χρησιμοποιούνται για τη συσσώρευση των τιμών δεδομένων. Για παράδειγμα, μεταβλητές τύπου αθροιστή ή πολλαπλασιαστή σε μια επαναληπτική δομή.

Βηματικός μετρητής (stepper): Αφορά σε μεταβλητές που δημιουργούνται και αλλάζουν τιμή αυτόματα, κατά ένα δεδομένο βήμα (π.χ. η μεταβλητή ελέγχου σε μια επαναληπτική δομή τύπου FOR).

Βοηθητική μεταβλητή (temporal): Χρησιμοποιείται για την προσωρινή καταχώρηση τιμών ώστε να γίνουν διάφοροι υπολογισμοί (π.χ. αντιμετάθεση τιμών δύο μεταβλητών, αλγόριθμος ταξινόμησης πίνακα κ.λ.π.).

Σημαία (one-way flag): Είναι συνήθως μεταβλητές λογικού τύπου που σηματοδοτούν την ύπαρξη ή όχι ενός γεγονότος κατά την εκτέλεση του προγράμματος (π.χ. ύπαρξη ή όχι μιας συγκεκριμένης τιμής σε έναν μονοδιάστατο πίνακα).

Δραστηριότητα 2

Να σχεδιάσετε ένα διδακτικό σενάριο (σχέδιο μαθήματος και φύλλο εργασίας μαθητή), το οποίο θα στοχεύει στην εξοικείωση των μαθητών με ρόλους μεταβλητών. Προτείνεται να χρησιμοποιηθεί σε συνδυασμό με επαναληπτικές δομές.

Σύνοψη

Στην παρούσα θεματική ενότητα έγινε μια συνολική διαπραγμάτευση του πλαισίου διδασκαλίας του προγραμματισμού και της αλγοριθμικής επίλυσης προβλημάτων στο Ενιαίο Λύκειο. Αναλύθηκαν οι διδακτικοί στόχοι των Προγραμμάτων Σπουδών, οι γνώσεις και οι δεξιότητες που πρέπει να αποκτήσουν οι μαθητές στον προγραμματισμό. Έγινε μια παρουσίαση των βασικών παρανοήσεων και γνωστικών δυσκολιών που αντιμετωπίζουν οι μαθητές και συνιστούν σημαντικά εμπόδια στην οικοδόμηση επαρκών μοντέλων για τα προγραμματιστικά αντικείμενα. Παρουσιάστηκε αναλυτικά το παιδαγωγικό πλαίσιο διδασκαλίας του προγραμματισμού και τεκμηριώθηκε η αναγκαιότητα μετάβασης από τις εμπειρικές διδακτικές προσεγγίσεις σε παιδαγωγικά σχεδιασμένες μαθησιακές δραστηριότητες που στοχεύουν στην ενεργοποίηση των μαθητών με στόχο την εποικοδόμηση επαρκών αναπαραστάσεων για τις προγραμματιστικές έννοιες και δομές και στην επίλυση αυθεντικών προβλημάτων σε προγραμματιστικά περιβάλλοντα. Η προτεινόμενη διδακτική προσέγγιση συνοδεύεται από κατάλληλα φύλλα εργασίας που βασίζονται, κυρίως, στο ολοκληρωμένο περιβάλλον της Αλγοριθμικής.

Βιβλιογραφία

- Anderson J. R. (1983), *The architecture of cognition*, Cambridge, MA: Harvard University Press
- Bayman P. & Mayer R. (1988), Using conceptual models to teach BASIC computer programming, *Journal of Educational Psychology*, 80(3), 291-298
- Ben-Ari M. (2001), Constructivism in computer science education, *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45-73
- Bonar J., & Soloway E. (1985), Preprogramming knowledge: a major source of misconceptions in novice programmers, *Human-Computer Interaction*, 1, 133-161
- Dahl, O. J., Dijkstra, E. W. and Hoare, C. A. R. (1972), *Structured Programming*, Academic Press
- Du Boulay B. (1989), Some difficulties of learning to program, in E. Soloway & J. C. Spohrer (Eds.), *Studying the novice programmer*, 283-299, Hillsdale, NJ: Lawrence Erlbaum
- Ennis D. L. (1994), Computing, problem-solving instruction and programming instruction to increase the problem-solving ability of high school students, *Journal of Research on Computing in Education*, 26(4) 489-496
- Green T. R. G. (1990), (Ed.), *Psychology of Programming*, London: Academic Press
- Jehng J. C. J., Tung S. H. S. & Chang C. T. (1999), A visualisation approach to learning the concept of recursion, *Journal of Computer Assisted Learning*, 15, 279-290
- Kolikant Y. B.-D. & Pollack S. (2004), Establishing computer science professional norms among high-school students, *Computer Science Education*, 14(1), 21-35
- Pair C. (1990), Programming, programming languages and programming methods, in T. R. G. Green (Ed.), *Psychology of Programming*, 9-19, London: Academic Press
- Papert S. (1980), *Νοητικές Θύελλες: Παιδιά, ηλεκτρονικοί υπολογιστές και δυναμικές ιδέες*, Εκδόσεις Οδυσσέας (Ελληνική μετάφραση 1991)
- Ramadhan H. A. (2000), Programming by discovery, *Journal of Computer Assisted Learning*, 16, 83-93
- Sajaniemi, J. (2005). Roles of variables and learning to program. Στο Α. Τζιμογιάννης (επιμ.), *Πρακτικά 3^{ου} Πανελληνίου Συνεδρίου 'Διδακτική της Πληροφορικής'*, Κόρινθος.
- Samurçay R. (1989), The concept of variable in programming: Its meaning and use in problem-solving by novice programmers, in E. Soloway & J. C. Spohrer (Eds.), *Studying the Novice Programmer*, 161-178, Hillsdale, NJ: Lawrence Erlbaum
- Soloway E. & Spohrer J. C. (1989), (Eds.), *Studying the Novice Programmer*, Hillsdale, NJ: Lawrence Erlbaum
- Williams L. & Kessler R. (2000), All I rally need to know about pair programming I learned in kindergarten, *Communications of ACM*, 43(5), 108-144.

- Γρηγοριάδου Μ., Γόγουλου Α., Γουλή Ε. & Σαμαράκου (2004), Σχεδιάζοντας «Διερευνητικές + Συνεργατικές» δραστηριότητες σε εισαγωγικά μαθήματα προγραμματισμού, στο Π. Πολίτης (επιμ.), *Πρακτικά 2^{ης} Διημερίδας με Διεθνή Συμμετοχή 'Διδακτική της Πληροφορικής'*, 86-96, Βόλος
- Εφόπουλος Β., Ευαγγελίδης Γ., Δαγδιλέλης Β. και Κλεφτοδήμος Α. (2005), Οι δυσκολίες των αρχάριων προγραμματιστών, στο Α. Τζιμογιάννης (επιμ.), *Πρακτικά 3^{ου} Πανελληνίου Συνεδρίου 'Διδακτική της Πληροφορικής'*, 51-60, Κόρινθος.
- Κόμης Β. (2001), Μελέτη βασικών εννοιών του προγραμματισμού στο πλαίσιο μιας οικοδομηστικής διδακτικής προσέγγισης, *ΘΕΜΑΤΑ στην Εκπαίδευση*, 2(2-3), 243-270
- Κόμης Β. (2005), *Εισαγωγή στη Διδακτική της Πληροφορικής*, Αθήνα: Κλειδάριθμος
- Κόμης Β. και Τζιμογιάννης Α. (2006), Ο Προγραμματισμός ως μαθησιακή δραστηριότητα: από τις εμπειρικές προσεγγίσεις στη γνώση παιδαγωγικού περιεχομένου, *Θέματα στην Εκπαίδευση*, 7(3), 229-255
- Τζιμογιάννης Α. (2000), Η διδασκαλία του Προγραμματισμού Η/Υ στη Δευτεροβάθμια Εκπαίδευση. Δυσκολίες και αντιλήψεις των μαθητών για την έννοια της μεταβλητής, *Η Βάση*, 2, 35-42, Ιωάννινα
- Τζιμογιάννης Α. (2002), Διδακτική Πληροφορικής, Προγράμματα Σπουδών και διδακτικές πρακτικές στο Ενιαίο Λύκειο, *Πρακτικά 3^{ου} Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή 'Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση'*, Τόμος Α', 229-238, Ρόδος
- Τζιμογιάννης Α. (2003), Η διδασκαλία του Προγραμματισμού στο Ενιαίο Λύκειο: προς ένα ολοκληρωμένο πλαίσιο με στόχο την ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων, *Πρακτικά 2^{ου} Πανελληνίου Συνεδρίου των Εκπαιδευτικών για τις ΤΠΕ: «Αξιοποίηση των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών στη Διδακτική Πράξη»*, Τόμος Α', 706-720, Σύρος
- Τζιμογιάννης Α. (2005), Προς ένα παιδαγωγικό πλαίσιο διδασκαλίας του προγραμματισμού στη δευτεροβάθμια εκπαίδευση, στο Α. Τζιμογιάννης (επιμ.), *Πρακτικά 3^{ου} Πανελληνίου Συνεδρίου 'Διδακτική της Πληροφορικής'*, 99-111, Κόρινθος
- Τζιμογιάννης, Α. και Γεωργίου, Β. (1998), Η αναγκαιότητα της διδασκαλίας του προγραμματισμού Η/Υ στη δευτεροβάθμια εκπαίδευση ως μεθοδολογία επίλυσης προβλημάτων. Το παράδειγμα των πινάκων, *Πρακτικά Διημερίδας Πληροφορικής "Η Πληροφορική στη Δευτεροβάθμια Εκπαίδευση"*, 28-34, Αθήνα: ΕΠΥ
- Τζιμογιάννης, Α. και Γεωργίου, Β. (1999), Οι δυσκολίες μαθητών της Δευτεροβάθμιας Εκπαίδευσης στην εφαρμογή της δομής ελέγχου για την ανάπτυξη απλών αλγορίθμων, *Πρακτικά Πανελληνίου Συνεδρίου "Πληροφορική και Εκπαίδευση"*, 183-192, Ιωάννινα
- Τζιμογιάννης Α. & Γιούνης Α. (2003), *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*, Τόμος Α', Αθήνα: Εκδόσεις Σαββάλα
- Τζιμογιάννης Α. και Κόμης Β. (2000), Η έννοια της μεταβλητής στον Προγραμματισμό: δυσκολίες και παρανοήσεις μαθητών του Ενιαίου Λυκείου, *Πρακτικά 2^{ου} Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή 'Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση'*, 103-114, Πάτρα.
- Τζιμογιάννης Α. και Κόμης Β. (2004), Μελέτη των αναπαραστάσεων μαθητών του Ενιαίου Λυκείου για τη ροή δεδομένων και το ρόλο των βασικών μονάδων του υπολογιστή, *Πρακτικά 2^{ης} Διημερίδας 'Διδακτική της Πληροφορικής'*, 73-85, Βόλος
- Τζιμογιάννης Α., Πολίτης Π. και Κόμης Β. (2005), Μελέτη των αναπαραστάσεων τελειόφοιτων μαθητών Ενιαίου Λυκείου για την έννοια της μεταβλητής, στο Α. Τζιμογιάννης (επιμ.), *Πρακτικά 3^{ου} Πανελληνίου Συνεδρίου 'Διδακτική της Πληροφορικής'*, 61-70, Κόρινθος.
- Τζιμογιάννης Α., Τσιωτάκης Π. και Sajaniemi J. (2006), Μελετώντας το ρόλο των προσομοιώσεων αλγορίθμων στη διδασκαλία του προγραμματισμού στο Ενιαίο Λύκειο, στο Ε. Σταυρίδου & Χ. Σολομωνίδου (επιμ.), *Πρακτικά Πανελληνίου Συνεδρίου 'Ψηφιακό Εκπαιδευτικό Υλικό: Ζητήματα δημιουργίας, διδακτικής αξιοποίησης και αξιολόγησης'*, 99-108, Βόλος
- Τσέλιος, Ν., Γεωργόπουλος, Α., Πολίτης, Π., Πύρζα, Φ., Φανίκος, Α., Κουμπούρη, Δ. και Κόμης, Β. (2006), Μελέτη της χρήσης πολλαπλών αναπαραστάσεων προγράμματος σε εκπαιδευτικό περιβάλλον προγραμματισμού, *ΘΕΜΑΤΑ στην Εκπαίδευση*, 7(3), 221-249
- ΥΠΕΠΘ (1998), *Η Πληροφορική στο σχολείο*, Αθήνα: Παιδαγωγικό Ινστιτούτο