

Οι δυσκολίες μαθητών δευτεροβάθμιας εκπαίδευσης στην εφαρμογή της δομής ελέγχου για την ανάπτυξη αλγορίθμων. Μία μελέτη περίπτωσης

Α. Τζιμογιάννης, Β. Γεωργίου

1. Εισαγωγή

Η διδασκαλία του Προγραμματισμού Η/Υ αποτελεί μια δύσκολη και ταυτόχρονα ελκυστική εργασία, ιδιαίτερα όταν αφορά σε άτομα με περιορισμένη εμπειρία στους υπολογιστές, όπως είναι οι μαθητές. Στη χώρα μας η διδασκαλία του Προγραμματισμού Η/Υ στη δευτεροβάθμια εκπαίδευση εισάγεται το 1985 σε Τεχνικά Επαγγελματικά Λύκεια (ΤΕΛ) και Ενιαία Πολυκλαδικά Λύκεια (ΕΠΛ) με τη μορφή, κυρίως, της εκμάθησης γλωσσών προγραμματισμού (Basic, Pascal και Cobol). Το 1992 καθιερώνεται η διδασκαλία της Logo στο Γυμνάσιο στα πλαίσια του μαθήματος Πληροφορική-Τεχνολογία. Από το σχολικό έτος 1998-99 επεκτείνεται η διδασκαλία των αρχών του Προγραμματισμού στο νέο Ενιαίο Λύκειο μέσα από το νεοεισαγόμενο μάθημα της Πληροφορικής.

Στα Προγράμματα Σπουδών Πληροφορικής έχουν εφαρμοστεί γενικά δύο διαφορετικές μεθοδολογίες διδασκαλίας του Προγραμματισμού Η/Υ.

α) Προσέγγιση της εκμάθησης γλωσσών προγραμματισμού

Η μέθοδος αυτή εστιάζεται στα χαρακτηριστικά της χρησιμοποιούμενης γλώσσας προγραμματισμού και αναπτύσσει ένα πλαίσιο διδασκαλίας γύρω από αυτή. Η έμφαση δίνεται στη δομή της γλώσσας, στο λεξιλόγιο και τους συντακτικούς κανόνες της. Ο κύριος στόχος είναι η εκμάθηση της γλώσσας και η διδασκαλία είναι προσανατολισμένη σ' αυτό. Η προσέγγιση αυτή ακολουθήθηκε και στη χώρα μας στα πλαίσια του Προγράμματος Σπουδών των ΤΕΛ και ΕΠΛ. Η εμπειρία από τη διδασκαλία του προγραμματισμού τόσο σε ΤΕΛ όσο και σε ΙΕΚ έχει αποδείξει ότι το εκπαιδευτικό αποτέλεσμα είναι συχνά περιορισμένο. Οι περισσότεροι μαθητές περιορίζονται στην αποστήθιση εντολών ή διαδικασιών και δεν εξασκούνται στην ανάπτυξη μεθόδων επίλυσης ισοδύναμων προβλημάτων. Για το λόγο αυτό συναντούν μεγάλες δυσκολίες στην επίλυση παρόμοιων προβλημάτων χρησιμοποιώντας διαφορετική γλώσσα προγραμματισμού.

β) Προσέγγιση της επίλυσης προβλημάτων

Στην προσέγγιση αυτή η χρησιμοποιούμενη γλώσσα προγραμματισμού τίθεται σε δεύτερη μοίρα, και η έμφαση δίνεται στο πρόβλημα και τα λογικά βήματα που είναι απαραίτητα για την επίλυση του. Ο διδάσκων θέτει ή ο μαθητής επιλέγει το προς επίλυση πρόβλημα. Η έμφαση δίνεται στην ανάπτυξη της κατάλληλης λύσης (αλγορίθμου) και στη

συνέχεια γίνεται η δόμηση του προγράμματος. Κατά την προσέγγιση αυτή διδάσκονται και αναλύονται μόνο τα στοιχεία της γλώσσας που είναι απαραίτητα για την επίλυση του προβλήματος. Ο υπολογιστής είναι το εργαλείο και η γλώσσα το μέσο που επιτρέπει στο μαθητή να χρησιμοποιήσει αποδοτικά το εργαλείο.

Στη χώρα μας, η Διδακτική της Πληροφορικής και ειδικότερα του Προγραμματισμού Η/Υ βρίσκεται στη φάση της οργάνωσης. Η εκπαιδευτική έρευνα σχετικά με τη διδασκαλία του μαθήματος είναι αποσπασματική και υποτυπώδης. Για το λόγο αυτό θα πρέπει να αξιοποιηθεί, σε πρώτο επίπεδο, η εμπειρία και η μεθοδολογία της Διδακτικής των συναφών επιστημών όπως είναι τα Μαθηματικά και οι Φυσικές Επιστήμες. Η καταγραφή των αντιλήψεων και των δυσκολιών που συναντούν οι μαθητές στην ανάπτυξη προγραμμάτων έχει ιδιαίτερη παιδαγωγική αξία, καθώς μπορεί να αξιοποιηθεί τόσο για τη βελτίωση της διδακτικής πράξης όσο και για τη διαμόρφωση νέων αναλυτικών προγραμμάτων και βιβλίων.

Η εργασία αυτή αποτελεί μία μελέτη περίπτωσης (case study) με στόχο την καταγραφή, ταξινόμηση και διερεύνηση των αντιλήψεων και των δυσκολιών που συναντούν μαθητές Τεχνικών Επαγγελματικών Λυκείων κλάδου Πληροφορικής στην εφαρμογή της δομής ελέγχου για την ανάπτυξη απλών αλγορίθμων. Εντάσσεται σε ένα ευρύτερο πρόγραμμα που έχει ως στόχο την οργάνωση ενός πλαισίου διδασκαλίας του προγραμματισμού Η/Υ στη δευτεροβάθμια εκπαίδευση βασισμένου στη διεπιστημονική προσέγγιση του αντικείμενου, με έμφαση στην ανάπτυξη αλγορίθμων και μεθοδολογιών επίλυσης προβλημάτων [1,2]. Τα αποτελέσματα που παρουσιάζονται αποτελούν τα πρώτα ερευνητικά δεδομένα στην Ελληνική βιβλιογραφία σχετικά με τις δυσκολίες μαθητών της δευτεροβάθμιας εκπαίδευσης στην κατανόηση και λειτουργική εφαρμογή της δομής ελέγχου.

2. Γιατί είναι απαραίτητη η διδασκαλία τον Προγραμματισμού Η/Υ;

Η κύρια διαφορά της διδασκαλίας του Προγραμματισμού Η/Υ με τα υπόλοιπα μαθήματα του Προγράμματος Σπουδών βρίσκεται στο περιεχόμενο και στην αξία των παρεχόμενων γνώσεων και δεξιοτήτων. Συχνά διατυπώνονται απόψεις που θέτουν υπό αμφισβήτηση την παιδαγωγική αξία της διδασκαλίας του Προγραμματισμού Η/Υ, με βάση το γεγονός ότι τα σύγχρονα πακέτα εφαρμογών υποστηρίζουν εύκολα και σε ελάχιστο χρόνο την παραγωγή ποιοτικών και αξιόπιστων αποτελεσμάτων. Από τη στιγμή που η αξία και η διαχρονικότητα των παρεχόμενων γνώσεων είναι αμφισβητήσιμη, καθίσταται αναγκαία η αναζήτηση δεξιοτήτων που αποκτούν οι μαθητές μέσα από τη διδασκαλία συγκεκριμένων αντικειμένων. Η σημασία του Προγραμματισμού Η/Υ ως γνωστική δραστηριότητα, η συνεισφορά του στην ανάπτυξη δομημένης σκέψης και η επίδραση του στην καλλιέργεια πνευματικών δεξιοτήτων υψηλού επιπέδου έχει τεθεί για πρώτη φορά από τον Papert [3] και έχει αποδειχθεί από διάφορες έρευνες [4,5,6].

Για τους παραπάνω λόγους θα πρέπει η διδακτική προσέγγιση του Προγραμματισμού Η/Υ στη δευτεροβάθμια εκπαίδευση να βασίζεται στη διδασκαλία μεθοδολογιών επίλυσης προβλημάτων και όχι στην εκμάθηση γλωσσών προγραμματισμού, ώστε να υποστηρίξει με επιτυχία διαχρονικές δεξιότητες αντίστοιχες με άλλα μαθήματα

όπως τα Μαθηματικά ή η Φυσική. Το να είναι κανείς δεινός προγραμματιστής δεν σημαίνει αυτόματα ότι είναι και επιτυχημένος δάσκαλος του Προγραμματισμού Η/Υ. Η διδακτική προσέγγιση έχει ιδιαίτερα χαρακτηριστικά και δυσκολίες που δε συναντώνται σε άλλα γνωστικά αντικείμενα:

- 1.0 σχεδιασμός και η υλοποίηση αλγορίθμων αποτελεί μία σύνθετη και περίπλοκη νοητική δραστηριότητα, η οποία απαιτεί την καλλιέργεια δεξιοτήτων υψηλού επιπέδου και την ύπαρξη γνώσεων από διάφορα αντικείμενα.
2. Ο κώδικας επικοινωνίας μαθητή-υπολογιστή είναι απόλυτος και αυστηρά καθορισμένος, με βάση συγκεκριμένους συντακτικούς κανόνες. Η αυστηρότητα αυτή είναι εγγενής στον προγραμματισμό και δεν έχει την απλή έννοια που συναντάμε στις άλλες θετικές επιστήμες.
3. Βασικός παράγοντας στη διδασκαλία γλώσσών προγραμματισμού είναι η ενεργός συμμετοχή του μαθητή, καθώς ο προγραμματισμός είναι σύμφυτος με διαδικασίες επικοινωνίας-ανάδρασης μεταξύ μαθητή και υπολογιστή. Ο διδάσκων δεν είναι ο παραδοσιακός φορέας γνώσης, αλλά καθίσταται καθοδηγητής των μαθησιακών δραστηριοτήτων των μαθητών.

3. Μέθοδος

Στόχος της εργασίας μας είναι η καταγραφή και ταξινόμηση των δυσκολιών μαθητών Τεχνικών Επαγγελματικών Λυκείων κλάδου Πληροφορικής στην εφαρμογή της δομής ελέγχου για την επίλυση απλών προβλημάτων. Η επιλογή του αντικειμένου μελέτης βασίστηκε στο γεγονός ότι η δομή ελέγχου αποτελεί τη πιο βασική δομή προγραμματισμού που περιλαμβάνεται σε όλες τις γλώσσες. Παράλληλα, η διδακτική μας εμπειρία και τα αποτελέσματα προηγούμενων ερευνών στα ΤΕΛ έχουν δείξει ότι, οι μαθητές έχουν σοβαρές αδυναμίες στη λειτουργική χρήση των βασικών δομών του προγραμματισμού για την ανάπτυξη προγραμμάτων [1,2].

Η υπόθεση της έρευνας είναι ότι *"πολλοί μαθητές των ΤΕΛ κλάδου Πληροφορικής έχουν δυσκολίες να κατανοήσουν και να εφαρμόσουν επιτυχώς τη δομή ελέγχου σε προγράμματα για την επίλυση απλών προβλημάτων"*.

Για την επιβεβαίωση ή απόρριψη της παραπάνω υπόθεσης σχεδιάσαμε τρία εργασίες που δόθηκαν στους μαθητές για απάντηση με τη μορφή γραπτού ερωτηματολογίου. Βασική μας επιδίωξη ήταν να αποφύγουμε προβλήματα σχολικού τύπου, έτσι ώστε να καταγράψουμε τις πραγματικές αντιλήψεις και δυσκολίες των μαθητών του δείγματος. Η κωδικοποίηση των προγραμμάτων έγινε κυρίως σε Pascal, ενώ μικρός αριθμός μαθητών χρησιμοποίησε σε κάποιες περιπτώσεις λογικό διάγραμμα ή γλώσσα Cobol.

Η έρευνα διενεργήθηκε σε χρονική απόσταση τουλάχιστο τριών μηνών από την περίοδο του διδάχθηκε το αντικείμενο στην τάξη. Το δείγμα αποτελείται από 67 μαθητές της Γ' Τάξης Κλάδου Πληροφορικής του 1ου ΤΕΛ Χαλκίδας και του 2ου ΤΕΛ Ιωαννίνων. Σημειώνεται ότι οι μαθητές του δείγματος είχαν διδαχθεί τη Basic στη Β' Τάξη, ενώ παράλληλα διδάσκονταν τις γλώσσες Pascal και Cobol.

4. Αποτελέσματα

Η αξιολόγηση και ταξινόμηση των απαντήσεων των μαθητών έγινε με βάση τα λογικά λάθη, ενώ αγνοήθηκαν τα συντακτικά λάθη. Οι προσεγγίσεις των μαθητών στα έργα της έρευνας κατατάσσονται σε τέσσερις κατηγορίες. Επαρκείς θεωρούνται οι λύσεις που χρησιμοποιούν τη σωστή συνθήκη και δίνουν ολοκληρωμένη τη διαδικασία επίλυσης του έργου. Στη δεύτερη κατηγορία κατατάσσονται οι λύσεις που χειρίζονται σωστά τις απαραίτητες μεταβλητές του κατάλληλου τύπου αλλά έχουν λογικά λάθη ή ελλείψεις στη σύνταξη της λογικής συνθήκης. Στην τρίτη κατηγορία ταξινομούνται οι απαντήσεις που χαρακτηρίζονται από λανθασμένη χρήση των μεταβλητών (π.χ. χρήση μεταβλητών που δεν είναι απαραίτητες, λανθασμένου τύπου, έλλειψη των κατάλληλων μεταβλητών που επιλύουν το έργο). Θα πρέπει να σημειωθεί ότι, η πλειονότητα των απαντήσεων της κατηγορίας αυτής παρουσιάζει επίσης λογικά λάθη στη σύνταξη της λογικής συνθήκης. Τέλος, θεωρούμε ανεπαρκείς τις προσεγγίσεις που δεν δίνουν καμία απάντηση ή περιορίζονται μόνο στη μηχανική παράθεση τυποποιημένων εντολών και δηλώσεων βασισμένων στην αποστήθιση.

Στη συνέχεια παρουσιάζονται αναλυτικά τα αποτελέσματα της έρευνας για τα διάφορα έργα.

Έργο 1

Να γραφεί πρόγραμμα που θα διαβάζει τη βαθμολογία ενός μαθητή και, όταν αυτή είναι κάτω από τη βάση (10), θα τυπώνει το μήνυμα "ΑΠΟΤΥΧΩΝ".

Οι απαντήσεις που δόθηκαν στο 1ο έργο δίνονται στον Πίνακα 1. Το έργο αυτό μπορεί να ενταχθεί στην κατηγορία προβλημάτων που αντιμετωπίζονται συχνά στα εισαγωγικά μαθήματα του προγραμματισμού. Παρόλα αυτά 4 στους 10 μαθητές δεν μπορούν να δώσουν επαρκή λύση. Το 16.4 % κάνει λανθασμένη χρήση της μεταβλητής αριθμητικού τύπου (integer) που περιγράφει τη βαθμολογία.

Σημειώνεται ότι, σε 6 περιπτώσεις δηλώνεται (χωρίς να χρησιμοποιείται σωστά) μεταβλητή τύπου array, που όπως διαπιστώθηκε αποτέλεσε αντικείμενο πρόσφατης διδασκαλίας σε συγκεκριμένη τάξη. Τέλος, οι συχνότερες αδυναμίες στη σύνταξη λογικών συνθηκών που καταγράφηκαν είναι της μορφής:

1. IF bathmos NOT>10 THEN writeln ('αποτυχών');
 2. IF bathmos <=10THEN writeln ('αποτυχών');
- όπου η μεταβλητή bathmos θεωρείται τύπου integer.

Πίνακας 1. Κατηγορίες απαντήσεων στο έργο 1

α/α	ΔΙΑΔΙΚΑΣΙΑ	ΣΥΧΝΟΤΗΤΑ N=67	ΠΟΣΟΣΤΟ %
1	Επαρκής λύση	40	59.7
2	Λανθασμένη σύνταξη λογικής συνθήκης	13	19.4

3	Λανθασμένη χρήση μεταβλητών	11	16.4
4	Χωρίς απάντηση	3	4.5

Έργο 2

Να γραφεί πρόγραμμα που θα διαβάζει τη θερμοκρασία δωματίου και θα εμφανίζει το κατάλληλο μήνυμα ενεργοποίησης του συστήματος κλιματισμού:

ΘΕΡΜΑΝΣΗ, για θερμοκρασία μικρότερη από 20°C

ΨΥΞΗ, για θερμοκρασία μεγαλύτερη από 25 °C.

Στον Πίνακα 2 ταξινομούνται οι απαντήσεις των μαθητών στο 2^ο έργο. Λιγότεροι από 4 στους 10 μαθητές δίνουν επαρκή λύση. Το 41.8 % των μαθητών του δείγματος έχει σημαντικές δυσκολίες στη σύνταξη της σχετικής λογικής συνθήκης. Οι χαρακτηριστικότερες απαντήσεις που καταγράφηκαν είναι οι εξής:

1. IF therm<20 THEN writeln ('ΘΕΡΜΑΝΣΗ')
ELSE writeln ('ΨΥΞΗ');
2. IF therm NOT<25 THEN writeln ('ΨΥΞΗ');
IF therm NOT >20 THEN writeln ('ΘΕΡΜΑΝΣΗ');
3. IF therm<=20 THEN writeln ('ΘΕΡΜΑΝΣΗ');
IF therm>=25 THEN writeln ('ΨΥΞΗ');

όπου η μεταβλητή therm είναι τύπου integer.

Το 11.9 % κάνει λανθασμένη χρήση της μεταβλητής που περιγράφει τη θερμοκρασία ή δηλώνει χωρίς να χρησιμοποιεί στο πρόγραμμα περισσότερες μεταβλητές. Επαναλαμβάνεται και στο έργο αυτό η χρήση μεταβλητής τύπου array από 4 μαθητές.

Πίνακας 2. Κατηγορίες απαντήσεων στο έργο 2

α/α	ΔΙΑΔΙΚΑΣΙΑ	ΣΥΧΝΟΤΗΤΑ N=67	ΠΟΣΟΣΤΟ%
1	Επαρκής λύση	26	38.8
2	Λανθασμένη σύνταξη λογικής συνθήκης	28	41.8
3	Λανθασμένη χρήση μεταβλητών	8	11.9
4	Χωρίς απάντηση	5	7.5

Έργο 3

Να γραφεί πρόγραμμα που θα διαβάζει την κατάσταση των φωτεινού σηματοδότη (*ΚΟΚΚΙΝΟ* ή *ΠΡΑΣΙΝΟ*) και θα εμφανίζει το αντίστοιχο μήνυμα για τον οδηγό ενός αυτοκίνητου (*ΣΤΑΜΑΤΑ* ή *ΦΥΓΕ*).

Στον Πίνακα 3 ταξινομούνται οι απαντήσεις των μαθητών στο 3ο έργο, στο οποίο σημειώνονται δυσκολίες, σε βαθμό μεγαλύτερο από ότι αναμενόταν. Είναι χαρακτηριστικό ότι μόλις το 14.9% δίνει επαρκή απάντηση. Οι δυσκολίες που

καταγράφονται αφορούν τόσο στη χρήση της κατάλληλης μεταβλητής όσο και στη σύνταξη της λογικής συνθήκης. Οι περισσότεροι μαθητές χρησιμοποιούν για την περιγραφή της κατάστασης του φωτεινού σηματοδότη μεταβλητή τύπου integer, real, ακόμη και array. Οι πιο χαρακτηριστικές δυσκολίες που καταγράφηκαν στη σύνταξη της σχετικής λογικής συνθήκης είναι οι εξής:

1. IF not kokkino THEN \writeln ('ΦΥΓΕ')
ELSE writeln ('ΣΤΑΜΑΤΑ') όπου
η μεταβλητή kokkino είναι string ή integer.
2. IF state='κόκκινο' THEN writeln ('ΣΤΑΜΑΤΑ')
ELSE writeln ('ΦΥΓΕ');
όπου η μεταβλητή state είναι integer ή real.
3. IF state=K THEN writeln ('ΣΤΑΜΑΤΑ')
ELSE IF state=P THEN writeln ('ΦΥΓΕ');

όπου οι μεταβλητές K και P είναι string χωρίς περιεχόμενο, integer ή real. Επίσης, η μεταβλητή state είναι στις περισσότερες περιπτώσεις integer ή real.

Πίνακας 3. Κατηγορίες απαντήσεων στο έργο 3

α/α	ΔΙΑΔΙΚΑΣΙΑ	ΣΥΧΝΟΤΗΤΑ N=67	ΠΟΣΟΣΤΟ%
1	Επαρκής λύση	10	14.9
2	Λανθασμένη σύνταξη λογικής συνθήκης	25	37.3
3	Λανθασμένη χρήση μεταβλητών	25	37.3
4	Χωρίς απάντηση	7	10.5

Από τα αποτελέσματα της έρευνας προκύπτει ότι, η πλειονότητα των μαθητών του δείγματος έχει σημαντικές δυσκολίες στην κατανόηση και στη λειτουργική εφαρμογή της δομής ελέγχου για τη σύνταξη απλών αλγορίθμων. Κατά μέσο όρο, ένας στους τρεις (33%) έχει σημαντικές δυσκολίες στη διατύπωση λογικών συνθηκών, ενώ περίπου ένας στους τέσσερις (22%) δεν μπορεί να χρησιμοποιήσει τις κατάλληλες μεταβλητές για την επίλυση απλών προβλημάτων. Επίσης καταγράφεται ιδιαίτερα μεγάλη δυσκολία στη χρήση μεταβλητών τύπου string για τη σύνταξη λογικών συνθηκών, γεγονός που οφείλεται ενδεχομένως στην περιορισμένη εφαρμογή τους σε παραδείγματα σχολικού τύπου. Τέλος, διαπιστώνεται ότι πολλοί μαθητές αποστηθίζουν και μεταφέρουν μηχανιστικά στα προγράμματα τους εντολές ή ομάδες εντολών, οι οποίες συνήθως δεν έχουν καμία σχέση με το πρόβλημα που καλούνται να επιλύσουν.

Με δεδομένο ότι οι μαθητές του δείγματος έχουν διδαχθεί τη Basic στη Β' Τάξη (8 ώρες εβδομαδιαία) και στη Γ' Τάξη διδάσκονταν παράλληλα τις γλώσσες Pascal (5

ώρες εβδομαδιαία) και Cobol (7 ώρες εβδομαδιαία), μπορούμε να αξιολογήσουμε το γνωστικό αποτέλεσμα ως πενιχρό. Οι καταγραφείσες δυσκολίες δηλώνουν χαμηλό υπόβαθρο και περιορισμένες ικανότητες επίλυσης προβλημάτων. Θεωρούμε ότι το βασικό εκπαιδευτικό ζήτημα ενός σύγχρονου Προγράμματος Σπουδών είναι η καλλιέργεια δεξιοτήτων επίλυσης προβλημάτων. Η επιτυχής εφαρμογή των προγραμματιστικών εργαλείων από τους μαθητές προϋποθέτει την ύπαρξη τέτοιου είδους δεξιοτήτων, έτσι ώστε να γίνει η ανάπτυξη και εξέλιξη τους αξιοποιώντας τα δομικά χαρακτηριστικά του Προγραμματισμού Η/Υ.

5. Συμπεράσματα-Διδακτική πρόταση

Τα αποτελέσματα της εργασίας επιβεβαιώνουν την ερευνητική μας υπόθεση, καθώς διαπιστώνεται ότι οι μαθητές του δείγματος έχουν σοβαρές αδυναμίες στην κατανόηση και στην εφαρμογή βασικών προγραμματιστικών εργαλείων για τη σύνταξη αλγορίθμων και την επίλυση απλών προβλημάτων. Θεωρούμε ως βασική αιτία το γεγονός ότι η διδασκαλία του Προγραμματισμού Η/Υ στα ΤΕΛ είναι προσανατολισμένη κυρίως στην εκμάθηση γλωσσών προγραμματισμού και όχι στη μεθοδολογία επίλυσης προβλημάτων. Θα πρέπει να σημειωθεί τέλος το χαμηλό υπόβαθρο γνώσεων από άλλες γνωστικές περιοχές, το οποίο διαπιστώνεται σε πολλούς μαθητές των ΤΕΛ.

Για το λόγο αυτό η διδασκαλία του Προγραμματισμού Η/Υ στη δευτεροβάθμια εκπαίδευση θα πρέπει να κατευθύνει τους μαθητές, ώστε να καλλιεργούν δεξιότητες υψηλού επιπέδου (δομημένη, αναλυτική και συνθετική σκέψη, δημιουργικότητα, φαντασία κ.λ.π.) μέσα από την κατανόηση και λειτουργική εφαρμογή των βασικών προγραμματιστικών εργαλείων, όπως:

- της έννοιας της μεταβλητής
- της εντολής εκχώρησης
- της λογικής και των μεθοδολογιών εφαρμογής των δομών ελέγχου και επανάληψης
- των βασικών τύπων και δομών δεδομένων
- της έννοιας του λογικού διαγράμματος.

Ο Polya [7], στο κλασικό του βιβλίο *"Πώς να το λύσω"*, έχει οριοθετήσει τέσσερα βήματα για την επίλυση προβλημάτων:

1. Κατανόηση του προβλήματος
- 2.Επινοήση του σχεδίου επίλυσης
- 3.Εκτέλεση του σχεδίου επίλυσης
4. Έλεγχος αποτελεσμάτων

Η παρούσα εργασία, αξιοποιώντας τα αποτελέσματα της έρευνας και με βάση τη μεθοδολογία Polya, προτείνει την οργάνωση της διδασκαλίας του Προγραμματισμού Η/Υ στη δευτεροβάθμια εκπαίδευση στη βάση της διεπιστημονικής προσέγγισης και της καλλιέργειας δεξιοτήτων διαχρονικής αξίας. Το ζητούμενο στον προγραμματισμό δεν πρέπει να είναι απλά η παραγωγή αποτελεσμάτων αλλά η ανάπτυξη μεθόδων για την επίλυση ισοδύναμων προβλημάτων. Στο βαθμό που η παραπάνω αναγκαιότητα δεν καλύπτεται

ικανοποιητικά από τα άλλα αντικείμενα του Προγράμματος Σπουδών. είναι απαραίτητο, ανεξάρτητα από τη γλώσσα που θα χρησιμοποιηθεί, να διδάξουμε στους μαθητές μεθοδολογίες επίλυσης προβλημάτων οργανώνοντας εκπαιδευτικά τις παρακάτω φάσεις:

α) κατανόηση του προβλήματος

Η κατανόηση του προβλήματος αποτελεί βασική προϋπόθεση για την επιτυχή λύση του και συνίσταται στο να προσδιοριστούν όλες οι παράμετροι που είναι απαραίτητες για την επίλυση του, δηλαδή

- τα δεδομένα
- τα ζητούμενα
- οι περιορισμοί και οι συνθήκες.

Με βάση τα παραπάνω θα καθοριστούν οι κατάλληλες μεταβλητές και οι τύποι τους.

β) ανάπτυξη της μεθόδου επίλυσης (αλγορίθμου)

Η ανάπτυξη των διαδοχικών βημάτων της μεθόδου επίλυσης σε μορφή ψευδοκώδικα ή λογικού διαγράμματος αποτελεί βασικό βήμα για την λύση του προβλήματος. Η διδακτική εμπειρία δείχνει ότι η φάση αυτή είναι πολύ σημαντική. Θεωρούμε ότι, θα πρέπει να δοθεί μεγάλη βαρύτητα σ' αυτή, ειδικά στα εισαγωγικά μαθήματα.

γ) κωδικοποίηση του αλγορίθμου

Μετά την ανάπτυξη του αλγορίθμου ακολουθεί η κωδικοποίηση του και η σύνταξη του προγράμματος. Οι μαθητές πρέπει να εξασκούνται στην ακριβή μεταφορά του αλγορίθμου στη γλώσσα προγραμματισμού και στην τήρηση των συντακτικών κανόνων και του λεξιλογίου.

δ) ανίχνευση και διόρθωση λαθών

Η ανίχνευση και διόρθωση των λαθών του προγράμματος αποτελεί βασική φάση για την ολοκλήρωση της λύσης του προβλήματος. Τα λάθη που μπορεί να υπάρξουν σε ένα πρόγραμμα είναι τριών τύπων:

- συντακτικά λάθη (syntax errors)
- λάθη εκτέλεσης (run-time errors)
- λογικά λάθη (logical errors).

Οι μαθητές θα πρέπει να διδαχθούν πώς να αξιοποιούν σωστά και πώς να μαθαίνουν μέσα από τα λάθη τους. Η ανίχνευση και διόρθωση των σφαλμάτων (debugging) από τους ίδιους είναι μια διαδικασία ιδιαίτερης παιδαγωγικής αξίας και απαιτεί την καλλιέργεια δεξιοτήτων υψηλού επιπέδου. Βασίζεται στην κριτική ανάγνωση του κώδικα, στην κατανόηση των βασικών δομών και λογικών βημάτων, στην εμπέδωση των τύπων δεδομένων και στη γνώση των συντακτικών κανόνων της χρησιμοποιούμενης γλώσσας προγραμματισμού.

ε) έλεγχος προγράμματος

Τα λογικά λάθη σε ένα πρόγραμμα αποτελούν το μεγαλύτερο πρόβλημα, καθώς είναι δύσκολα ανιχνεύσιμα. Για τον εντοπισμό της αιτίας τους απαιτείται ο έλεγχος της ορθότητας του αλγόριθμου και της κωδικοποίησης του. Οι μαθητές θα πρέπει να εξασκούνται σε τεχνικές αποδοτικού ελέγχου. Βασική αρχή είναι να γίνεται κατάλληλη επιλογή των δεδομένων εισόδου, ώστε να ανιχνεύονται όλες οι περιπτώσεις και ιδιαίτερα οι οριακές.

στ) τεκμηρίωση προγράμματος

Η τεκμηρίωση είναι απαραίτητο βήμα στον κύκλο ανάπτυξης ενός προγράμματος. Περιλαμβάνει την περιγραφή του προβλήματος, την παράθεση σχολίων και παρατηρήσεων και τη σχεδίαση της τελικής λύσης σε ψευδοκώδικα. Αποτελεί βασική αρχή που τηρείται από τους έμπειρους προγραμματιστές και συμβάλει καθοριστικά στη συντήρηση του προγράμματος. Οι μαθητές θα πρέπει να εξοικειωθούν με τη φάση αυτή και να κατανοήσουν ότι τα διάφορα προβλήματα δεν είναι στατικά. Η επίλυση τους θα πρέπει να σχεδιάζεται έτσι, ώστε να μπορούν εύκολα να γίνουν αλλαγές, τροποποιήσεις, προσθήκες ή βελτιώσεις, σύμφωνα με τα νέα δεδομένα της εφαρμογής του προγράμματος.

Ο πρόσφατος σχεδιασμός του Παιδαγωγικού Ινστιτούτου για τα Προγράμματα Σπουδών Πληροφορικής του Ενιαίου Λυκείου βασίζεται σε μια αντίστοιχη προσέγγιση [8,9]. Δίνει δε την ευκαιρία να διαπιστωθεί στη διδακτική πράξη η αξία του Προγραμματισμού Η/Υ ως μεθοδολογία επίλυσης προβλημάτων. Απομένει να διαπιστωθεί και να καταγραφεί από την εκπαιδευτική έρευνα η αποτελεσματικότητα της εφαρμογής της. Στους άμεσους στόχους μας είναι η επέκταση της ερευνάς μας σε μαθητές του Ενιαίου¹ Λυκείου.

Τμήμα της εργασίας αυτής εκπονήθηκε στα πλαίσια του αρ. Σ.0205/111/1998-Α επιδοτούμενου Προγράμματος Κινητικότητας τον Ε.Π.Ε.Α.Ε.Κ με θέμα "Δημιουργία Βάσης Ασκήσεων Γνωστής Δυσκολίας".

6. Αναφορές

1. Γεωργίου Β. και Τζιμογιάννης Α.. 1998, *Η Διδασκαλία τον Προγραμματισμού Η/Υ στη Δευτεροβάθμια Εκπαίδευση ως Διαδικασία Ανάπτυξης Πνευματικών Δεξιοτήτων. Μια Πρόταση Βασισμένη στη Δημιουργία Βάσης Ασκήσεων Γνωστής Δυσκολίας*, στο Α. Τζιμογιάννης (επιμ.), Πρακτικά εργασιών 1^{ης} Πανεπιστημιακής Ημερίδας "Πληροφορική και Εκπαίδευση", Σύλλογος Καθηγητών Πληροφορικής Ηπείρου, 125-133.
2. Τζιμογιάννης Α. και Γεωργίου Β.. 1998, *Η Αναγκαιότητα της Διδασκαλίας τον Προγραμματισμού Η/Υ στη Δευτεροβάθμια Εκπαίδευση ως Μεθοδολογία Επίλυσης Προβλημάτων. Το παράδειγμα των πινάκων*, Πρακτικά Διημερίδας Πληροφορικής "Η Πληροφορική στη Δευτεροβάθμια Εκπαίδευση", ΕΠΥ, 28-34.
3. Papert S., 1980, *Νοητικές Θύελλες: Παιδιά, ηλεκτρονικοί υπολογιστές και δυναμικές ιδέες*, Εκδόσεις Οδυσσέας (Ελληνική μετάφραση 1991).
4. Kagan D. M., 1989, *Research on Computer Programming as a Cognitive Activity: implications for the study of classroom teaching*, Journal of Education for Teaching, Vol. 15, No. 3, 177-189.
5. 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, Vol.26, No. 4,489-496.
6. Charlton J. P. and Birkett P. E., 1994, *Specificity Versus Nonspecificity of Cognitive Skills in Elementary Computer Programming*. Journal of Research on Computing in Education, Vol.26, No. 3, 391-402.
7. Polya G., 1944, *Πώς να το λύσω*, Εκδόσεις Σπηλιώτη (Ελληνική μετάφραση)
8. ΥΠΕΠΘ, Παιδαγωγικό Ινστιτούτο, 1997, *Ενιαίο Πλαίσιο Προγράμματος Σπουδών Πληροφορικής*.
9. Παπαδόπουλος Γ., 1998. *Η Πληροφορική στο Σχολείο: Ο Σχεδιασμός του Παιδαγωγικού Ινστιτούτου*, στο Α. Τζιμογιάννης (επιμ.). Πρακτικά εργασιών 1 ης Πανεπιστημιακής Ημερίδας "Πληροφορική και Εκπαίδευση", Σύλλογος Καθηγητών Πληροφορικής Ηπείρου. 9-22.

