Η ΔΙΔΑΣΚΑΛΙΑ ΤΩΝ ΑΛΓΟΡΙΘΜΙΚΩΝ ΔΟΜΩΝ ΣΤΑ ΠΛΑΙΣΙΑ ΤΟΥ ΜΑΘΗΜΑΤΟΣ «ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ»

Ξυνόγαλος Στέλιος
Διδάσκοντας Π.Δ. 407/80, Πανεπιστήμιο Μακεδονίας
Καθηγητής Πληροφορικής, Β/θμια Εκπ/ση Ξάνθης
stelios@uom.gr

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία γίνεται μια σύνοψη των δυσκολιών και παρανοήσεων που έχουν καταγραφεί στη διεθνή βιβλιογραφία σχετικά με την εκμάθηση των αλγοριθμικών δομών που διδάσκονται στα πλαίσια του μαθήματος Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον του Ενιαίου Λυκείου. Ο στόχος της σύνοψης αυτής είναι η στήριξη του διδάσκοντα στη σχεδίαση φύλλων ελέγχου και δραστηριοτήτων που αποσκοπούν στην αποτελεσματικότερη διδασκαλία του μαθήματος. Στην εργασία παρουσιάζονται ενδεικτικά κάποιες προτάσεις για φύλλα ελέγχου/εργασίας και δραστηριότητες που αξιοποιούν τα συμπεράσματα που παρουσιάζονται στη σύνοψη.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ***ανάπτυξη εφαρμογών, αλγοριθμικές δομές, παρανοήσεις, μαθησιακές δραστηριότητες, εκπαιδευτικό λογισμικό***

ΕΙΣΑΓΩΓΗ

Το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (ΑΕΠΠ) της Γ' τάξης του κύκλου Πληροφορικής και Υπηρεσιών της Τεχνολογικής κατεύθυνσης του Ενιαίου Λυκείου έχει ως γενικό σκοπό οι μαθητές: «να καλλιεργήσουν αναλυτική σκέψη και συνθετική ικανότητα, να αναπτύξουν τη δημιουργικότητα και τη φαντασία στο σχεδιασμό, να καλλιεργήσουν και να εθιστούν στην αυστηρότητα και σαφήνεια της έκφρασης και της διατύπωσης, να αναπτύξουν ικανότητες μεθοδολογικού χαρακτήρα, να αποκτήσουν δεξιότητες αλγοριθμικής προσέγγισης και να καταστούν ικανοί να υλοποιούν τις λύσεις απλών προβλημάτων με χρήση βασικών προγραμματιστικών γνώσεων» (Βακάλη κ.α., 1999, σελ. 14 – βιβλίο καθηγητή). Η διδασκαλία του μαθήματος τα τελευταία χρόνια έχει δείξει ότι οι μαθητές αντιμετωπίζουν ποικίλες δυσκολίες, με αποτέλεσμα ο γενικός σκοπός του μαθήματος να μην είναι εύκολο να επιτευχθεί (Αθανασόπουλος & Οικονόμου, 2004). Το πρόβλημα πιστεύουμε ότι έγκειται, κατά κύριο λόγο, στο γεγονός ότι:

- η διδασκαλία και εκμάθηση του προγραμματισμού είναι εν γένει δύσκολη
- οι δύο διδακτικές ώρες την εβδομάδα δεν επαρκούν για την αποτελεσματική διδασκαλία της ύλης του μαθήματος.

Στην παρούσα εργασία γίνεται μια προσπάθεια στήριξης της διδασκαλίας του μαθήματος μέσα από μια σειρά προτεινόμενων δραστηριοτήτων, οι οποίες αξιοποιούν τη διδακτική γνώση που συσσωρεύτηκε τις τελευταίες δεκαετίες. Συγκεκριμένα, παρέχεται μια σύνοψη των δυσκολιών και παρανοήσεων που έχουν καταγραφεί στη διεθνή βιβλιογραφία σχετικά με τη διδασκαλία των αλγοριθμικών δομών. Παρόλο που οι σχετικές μελέτες πραγματοποιήθηκαν στα πλαίσια διδασκαλίας του προγραμματισμού χρησιμοποιώντας την Basic και την Pascal, τα συμπεράσματα που καταγράφηκαν πιστεύουμε ότι μπορούν να αξιοποιηθούν για την αποτελεσματικότερη διδασκαλία του μαθήματος ΑΕΠΠ, αφού η γλώσσα προγραμματισμού ΓΛΩΣΣΑ που

χρησιμοποιείται σε αυτό αποτελεί εξελληνισμένο υποσύνολο των παραπάνω γλωσσών. Για την αποτελεσματικότερη αξιοποίηση - από τον διδάσκοντα - των συμπερασμάτων της έρευνας που έχει διεξαχθεί και έχοντας την πεποίθηση ότι αυτά δεν αλλοιώνονται, η παρουσίασή τους γίνεται χρησιμοποιώντας τη ΓΛΩΣΣΑ. Στη συνέχεια παρουσιάζονται κάποιες ιδέες/προτάσεις που έχουν ως στόχο να βοηθήσουν τον διδάσκοντα να αξιοποιήσει τα συμπεράσματα αυτά προκειμένου να σχεδιάσει *φύλλα ελέγχου*, *φύλλα εργασίας* και *δραστηριότητες* που θα βοηθήσουν τους μαθητές να αντιμετωπίσουν τις συνήθεις δυσκολίες και παρανοήσεις που αφορούν στις βασικές αλγοριθμικές δομές.

ΔΟΜΗ ΑΚΟΛΟΥΘΙΑΣ: ΜΕΤΑΒΛΗΤΕΣ ΚΑΙ ΕΝΤΟΛΕΣ ΕΙΣΟΔΟΥ

Πολλά από τα λάθη των σπουδαστών οφείλονται σε λανθασμένες αντιλήψεις σχετικά με τις μεταβλητές (Pane & Myers, 1996). Η περιγραφή μιας μεταβλητής ως μιας διεύθυνσης ή ενός ονόματος δεν είναι αρκετή για την ανάλυση του τρόπου λειτουργίας της μεταβλητής. Η ομάδα της (Samurcay, 1989) διεξήγαγε μια εμπειρική μελέτη στο Laboratoire de Psychologie du Travail EPHE/CNRS στο Παρίσι, με σκοπό να διαπιστώσει τις ικανότητες διαχείρισης της έννοιας της μεταβλητής από μια ομάδα σπουδαστών που φοιτούσαν στο 5ο έτος της Β/θμιας Εκπ/σης στη Γαλλία, και είχαν διδαχθεί 15 ώρες προγραμματισμό σε Pascal. Η ομάδα της (Samurcay, 1989) διακρίνει:

- Τέσσερις τύπους χρήσης του *τελεστή απόδοσης τιμής* όταν αυτός χρησιμοποιείται με μεταβλητές:

  Ανάθεση μιας σταθερής τιμής: lesson ← 'Πληροφορική'
  Ανάθεση μιας τιμής ως αποτέλεσμα υπολογισμών: b ← x * 4
  Αντιγραφή (duplication): l ← m
  Συσσώρευση (accumulation): sum ← sum + number

  Με εξαίρεση την περίπτωση της συσσώρευσης, ο σπουδαστής μπορεί να χρησιμοποιήσει τις γνώσεις του από τα μαθηματικά που σχετίζονται με την έννοια της μεταβλητής και της ισότητας. Το μαθηματικό μοντέλο όμως δεν είναι κατάλληλο για την περίπτωση της συσσώρευσης, καθώς και για την περίπτωση που οι μεταβλητές εμπλέκονται σε βρόχους. Η μεταβλητή δεν είναι πλέον μια διεύθυνση με μια τιμή, είναι και πρέπει να αντιμετωπίζεται ως μια συνάρτηση της εκτέλεσης του προγράμματος ή μια αλληλουχία τιμών.
- Δύο είδη μεταβλητών βάσει της *λειτουργικής τους σημα*σίας (functional meaning) στο μοντέλο του σχεδίου δράσης του σπουδαστή: (1) *εξωτερικές μεταβλητές* (external variables), των οποίων οι τιμές ελέγχονται από τους χρήστες του προγράμματος, (2) *εσωτερικές μεταβλητές* (internal variables), δηλαδή μεταβλητές των οποίων οι τιμές ελέγχονται από τον προγραμματιστή. Όπως επισημαίνει και η ομάδα της Samurcay, ο χειρισμός των εσωτερικών μεταβλητών είναι νοητικά πιο δύσκολος γιατί απαιτεί τη συνεχή ανάπτυξη αναπαραστάσεων της τόσο ευμετάβλητης κατάστασης του υπολογιστικού συστήματος.

Τα βασικά συμπεράσματα της εμπειρικής μελέτης που διεξήχθη (Samurcay, 1989) ήταν τα εξής:

- *Οι σπουδαστές μπορούν να επεξεργαστούν με μεγαλύτερη ευκολία μεταβλητές για τις οποίες ήδη διαθέτουν ένα σύστημα αναπαράστασης και επεξεργασίας* από κάποια άλλη γνωστική περιοχή από εκείνη του προγραμματισμού. Για παράδειγμα, οι σπουδαστές διαθέτουν ένα

σχήμα για τη μεταβλητή μετρητή. Ωστόσο, ακόμα και αν δεν είναι διαθέσιμο το σχήμα αυτό μπορεί εύκολα να γίνει κτήμα των σπουδαστών μέσω της διδασκαλίας.

- *Η αρχικοποίηση των μεταβλητών είναι μια πολύπλοκη νοητική διαδικασία.* Επιπλέον, οι σπουδαστές κατανοούν ευκολότερα μια εντολή ***Διάβασε*** παρά μια εντολή απόδοσης τιμής. Η πολυπλοκότητα εντοπίζεται: (1) στη δυσκολία διατύπωσης μιας υπόθεσης σχετικά με την αρχική κατάσταση του συστήματος, η οποία αφορά νοητικές δραστηριότητες πιο γενικές από τον προγραμματισμό, (2) η αρχικοποίηση των μεταβλητών κατά την επίλυση προβλημάτων πραγματοποιείται ασυναίσθητα από το σπουδαστή, ωστόσο η ανάδειξη των διαδικασιών που απαιτούνται για την αρχικοποίηση αυτή δεν είναι εύκολη.
- Οι δυσκολίες που σχετίζονται με την ενημέρωση και τον έλεγχο των τιμών των μεταβλητών σε προβλήματα με βρόχους παρουσιάζουν την ίδια πολυπλοκότητα. Η (Samurcay, 1989) επισημαίνει ότι απαιτείται πιο εκτεταμένη έρευνα προκειμένου να διαπιστωθεί ο ρόλος των μεταβλητών στην ανάπτυξη των βρόχων. Ωστόσο, υπάρχουν πολλές ενδείξεις ότι οι δυσκολίες των σπουδαστών στην ενημέρωση συνδέονται στενά με την πολυπλοκότητα της περιοχής του προβλήματος.

Επιπλέον, ο (Δαγδιλέλης, 1996) αναφέρει τις εξής δυσκολίες:

- *Μεταβλητές με «πεπλεγμένο» υπολογισμό τιμών.* Όταν οι μεταβολές μιας μεταβλητής - έστω Ν - εξαρτώνται από τις μεταβολές μια άλλης μεταβλητής – έστω Μ -, η οποία εξαρτάται από κάποιες άλλες μεταβλητές – έστω Χ και Υ – τότε ο σπουδαστής δυσκολεύεται να προσδιορίσει τόσο την τιμή της μεταβλητής Ν όσο και το ρόλο της.

```
X ← X – 3
  ...
Λ ← Y + 2
  ...
M ← Y + X
  ...
N ← N + M
```

- *Δυσκολία κατανόησης του γεγονότος ότι οι πράξεις επιτρέπονται – κατά κανόνα – μεταξύ μεταβλητών του ίδιου ή συγγενούς τύπου & του γεγονότος ότι υπάρχουν επιτρεπτές και μη πράξεις για κάθε κατηγορία μεταβλητών.*

Επίσης, οι (Sleeman et al., 1988; Putman et al., 1989) σε δύο μελέτες που διεξήγαγαν με μαθητές Γυμνασίου, χρησιμοποιώντας τις γλώσσες προγραμματισμού Basic και Pascal, κατέληξαν στα εξής συμπεράσματα:

- Αρκετοί σπουδαστές δυσκολεύονται να κατανοήσουν πως γίνεται η *απόδοση τιμής σε μια μεταβλητή με μια εντολή* ***Διάβασε***. Συγκεκριμένα, παρατηρήθηκαν οι παρακάτω κατηγορίες λαθών:
  - *Πολλές τιμές διαβάζονται σε μια μεταβλητή* (Multiple-value read): Αρκετοί σπουδαστές πιστεύουν ότι με μια εντολή ***Διάβασε*** μπορούν να διαβαστούν περισσότερες από μία τιμές σε μια μεταβλητή. Αυτή είναι η σημαντικότερη λανθασμένη αντίληψη που σχετίζεται με τις μεταβλητές και παρουσιάζεται με διάφορες μορφές. Εκτός από εκείνη που αναφέρθηκε, μερικοί σπουδαστές θεωρούν ότι οι τιμές συλλέγονται σε μια μεταβλητή μετά από κάθε διάβασμα ή ανάθεση.
  - *Σημασιολογικά περιορισμένη απόδοση τιμής* (Semantically Constrained Reads): μια εντολή ***Διάβασε*** στην οποία το όνομα της μεταβλητής έχει κάποια σημασία στη φυσική γλώσσα, έχει ως αποτέλεσμα την επιλογή μιας τιμής που βασίζεται στη σημασία του ονόματος.
  - *Η σειρά δήλωσης των μεταβλητών καθορίζει τη σειρά διαβάσματος τιμών.*
- *Συγχέονται δύο μεταβλητές.* Για παράδειγμα στην αλληλουχία

  Διάβασε P

  Q ← Q + 1 η δεύτερη εντολή εκτελείται σαν να ήταν Q ← P + 1

- *Αντιστροφή της εντολής απόδοσης τιμής*. Η εντολή Α ← Β θεωρείται ότι ανταλλάσσει τις τιμές των δύο μεταβλητών.
- *Χρησιμοποιείται κατ' επανάληψη η αρχική τιμή της μεταβλητής* και όχι η τελευταία τιμή της.
- Οι τιμές των μεταβλητών *εκτυπώνονται όταν η μεταβλητή συναντάται στο αριστερό μέρος μιας έκφρασης*.
- **Η τιμή της μεταβλητής που βρίσκεται στο αριστερό μέρος *εκτυπώνεται όποτε αλλάζει η τιμή της*.**

ΔΟΜΕΣ ΕΠΙΛΟΓΗΣ

Η πιο συνηθισμένη κατηγορία λανθασμένων αντιλήψεων στους αρχάριους προγραμματιστές έχει την πηγή της στην καθημερινή ζωή (Δαγδιλέλης, 1996). Πολλές φορές οι σπουδαστές μεταφέρουν τη διατύπωση της λύσης ενός προβλήματος, η οποία είναι εκφρασμένη σε φυσική γλώσσα, ή ακόμα και καθημερινές συνομιλίες μεταξύ ανθρώπων σε μια γλώσσα προγραμματισμού. Στη συνέχεια παρατίθεται ένα χαρακτηριστικό παράδειγμα (Δαγδιλέλης, 1996):

Διατύπωση σε φυσική γλώσσα	**Μεταφορά στη «Γλώσσα»**
	Διάβασε x
Κάτω από 30 είναι ΝΕΟΣ,	Αν x<30 τότε Γράψε 'ΝΕΟΣ'
Κάτω από 50 είναι ΜΕΣΟΚΟΠΟΣ,	Αν x<50 τότε Γράψε 'ΜΕΣΟΚΟΠΟΣ'
αλλιώς είναι ΓΕΡΟΣ	Γράψε 'ΓΕΡΟΣ'

Όπως επισημαίνει και ο (du Boulay, 1989) οι αρχάριοι προγραμματιστές δυσκολεύονται να κατανοήσουν ότι ένα πρόγραμμα γίνεται κατανοητό από το σύστημα βάσει πολύ αυστηρών κανόνων και εκπλήσσονται από το βαθμό της λεπτομέρειας που απαιτεί ο προγραμματισμός, ιδιαίτερα λόγω των *ανθρωπομορφικών χαρακτηριστικών* που προσδίδουν *στο σύστημα.*

Άλλες δυσκολίες που έχουν καταγραφεί είναι οι παρακάτω:

- Οι θετικές εναλλακτικές επιλογές σε μια δομή επιλογής παρουσιάζουν λιγότερες δυσκολίες από τις αρνητικές (Rogalski & Samurcay, 1990). Επιπλέον, η ύπαρξη σύνθετων εκφράσεων - boolean συναρτήσεων, συνδυασμοί προτάσεων με λογικά ΚΑΙ, ΟΧΙ, 'Η – καθιστά ακόμα πιο δύσκολη την κατανόηση μιας εντολής επιλογής (Δαγδιλέλης, 1996).
- Ο βαθμός δυσκολίας αυξάνεται όσο αυξάνεται το βάθος εμφώλευσης των εντολών. Στο (Δαγδιλέλης, 1996) μάλιστα επισημαίνεται ότι πολλοί ερευνητές, όπως οι Barron D. W. και Green T.R.G., συνιστούν την κατάργηση των εμφωλευμένων εντολών επιλογής γιατί προκαλούν σύγχυση στους σπουδαστές. Βέβαια, άλλοι ερευνητές δεν συμφωνούν με την κατάργηση των εμφωλευμένων εντολών επιλογής, αφού θεωρούν ότι σε πολύ μικρό βαθμό και σε ορισμένες μόνο περιπτώσεις η παράθεση των δομών επιλογής είναι πιο κατανοητή από την εμφώλευσή τους (Δαγδιλέλης, 1996).
- Η σωστή οργάνωση στο χώρο του πηγαίου κώδικα, δηλαδή η στοίχιση και η χρήση εσοχών, διευκολύνει την εκμάθηση και χρήση των δομών επιλογής.
- Οι σπουδαστές με καλύτερο υπόβαθρο στα μαθηματικά μαθαίνουν ευκολότερα νέες δομές, όποιες και αν είναι αυτές.
- Στις μελέτες των (Putman et al., 1989; Sleeman et al., 1988) καταγράφηκαν οι παρακάτω παρανοήσεις σχετικά με τη συμπεριφορά ενός προγράμματος ή τη ροή ελέγχου μετά από την εκτέλεση μιας εντολής επιλογής:
  - Όταν η συνθήκη είναι ψευδής, ο έλεγχος πηγαίνει στην αρχή του προγράμματος.

- Όταν μια εντολή ***Αν*** έχει ως αποτέλεσμα τη διακλάδωση σε μια εντολή ***Γράψε***, τότε τόσο η τιμή της μεταβλητής της εντολής ***Γράψε*** όσο και της έκφρασης της συνθήκης εκτυπώνεται.
- Τόσο το τμήμα ***τότε*** όσο και το τμήμα ***αλλιώς*** μιας εντολής ***Αν*** εκτελείται.
- Το τμήμα ***τότε*** της εντολής εκτελείται ανεξάρτητα από το αν αληθεύει η συνθήκη.
- Η εντολή: ***Αν*** <συνθήκη> ***τότε*** <εντολή1> <εντολή2> ερμηνεύεται ως ***Αν*** <συνθήκη> ***τότε*** <εντολή1> ***αλλιώς*** <εντολή2>

Τέλος, διαπιστώθηκε ότι οι σπουδαστές πιστεύουν ότι όταν η συνθήκη μιας εντολής ***Αν*** είναι ψευδής η εκτέλεση του προγράμματος τερματίζεται (Putman et al., 1989), αν δεν υπάρχει τμήμα ***αλλιώς*** (Sleeman et al., 1988).

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΔΟΜΕΣ

Η δημιουργία ενός επαναληπτικού σχεδίου περιλαμβάνει την αναγνώριση των βασικών ενεργειών οι οποίες πρέπει να επαναλαμβάνονται και της συνθήκης που καθορίζει τον τερματισμό ή τη συνέχιση της επανάληψης. Συνεπώς, τρεις διαδικασίες εμπλέκονται στη δημιουργία ενός βρόχου (Rogalski & Samurcay, 1990):

1. *Αρχικοποίηση*: καθορισμός της αρχικής κατάστασης των μεταβλητών.
2. *Ενημέρωση*: δημιουργία και εκφορά των *αναλλοίωτων σχέσεων που διέπουν μια επαναληπτική δομή* (loop invariant).
3. *Έλεγχος*: προσδιορισμός της συνθήκης τερματισμού και της θέσης της στο βρόχο.

Τα σημαντικότερα και πιο συνηθισμένα προβλήματα που αντιμετωπίζουν οι σπουδαστές, όπως προκύπτει από τη σχετική βιβλιογραφία, είναι:

- *Αδυναμία γενίκευσης*. Οι σπουδαστές παρουσιάζουν την τάση να χρησιμοποιούν στα προγράμματά τους μια λίστα επαναλαμβανόμενων εντολών αντί να χρησιμοποιούν ένα βρόχο (Hoc, 1989).
- *Ανεπαρκή νοητά μοντέλα*. Όπως προκύπτει από την έρευνα των (Kessler & Anderson, 1989) οι σπουδαστές διαθέτουν ανεπαρκή μοντέλα για τις επαναληπτικές δομές.
- *Οι αναλλοίωτες σχέσεις που διέπουν μια επαναληπτική δομή* (Loop invariant). Οι σπουδαστές δυσκολεύονται στον καθορισμό του τμήματος ενημέρωσης ενός βρόχου (loop invariant) που αποτελεί σημαντικό συστατικό του. Οι σπουδαστές βασίζουν τα μοντέλα των βρόχων στην αναπαράσταση μιας αλληλουχίας ενεργειών (δυναμικό μοντέλο) και όχι στην αναπαράσταση των αναλλοίωτων σχέσεων μεταξύ των μεταβλητών (στατικό μοντέλο). Επίσης, χρησιμοποιούν διαφορετικά ονόματα σε κάθε βήμα της επανάληψης για τη σηματοδότηση μιας μεταβλητής με συγκεκριμένη λειτουργία (functional variable) και δεν έχουν την ικανότητα να προσδιορίσουν αυθόρμητα μια συνθήκη εξόδου. (Rogalski & Samurcay, 1990).
- *Επιλογή κατάλληλης επαναληπτικής δομής*. Η επιλογή της καταλληλότερης επαναληπτικής δομής για ένα συγκεκριμένο πρόβλημα είναι αρκετά δύσκολη. Ακόμη και δευτεροετείς – τριτοετείς φοιτητές συναντούν δυσκολίες στην έκφραση σύνθετων επαναληπτικών δομών (Δαγδιλέλης, 1996).
- *Έξοδος από ένα βρόχο*. Οι βρόχοι διακρίνονται σε 4 κατηγορίες, ανάλογα με το σημείο στο οποίο ελέγχεται η συνθήκη εξόδου: (1) βρόχοι στους οποίους η συνθήκη ελέγχεται πριν να εκτελεστεί το σώμα του βρόχου (*top-exit loops)*, όπως για παράδειγμα στη δομή ***Όσο…επανάλαβε***, (2) βρόχοι στους οποίους η συνθήκη ελέγχεται μετά από την εκτέλεση του σώματος του βρόχου (*bottom-exit loops)*, όπως για παράδειγμα στη δομή ***Αρχή_επανάληψης…μέχρις_ότου***, (3) η έξοδος γίνεται από ένα σημείο ελέγχου στο

μέσο του βρόχου (*middle-exit loops)*, (4) η έξοδος μπορεί να πραγματοποιηθεί από οπουδήποτε μέσα στο βρόχο εφόσον η συνθήκη αποτυγχάνει (*daemon-exit loops*). Στο (Pane & Myers, 1996) οι Wu και Anderson αναφέρουν ότι οι σπουδαστές συνήθως χρησιμοποιούν τη στρατηγική bottom-exit, στη συγκεκριμένη περίπτωση δηλαδή τη δομή ***Αρχή_επανάληψης…μέχρις_ότου***. Ο ισχυρισμός αυτός ενισχύεται και από τους Rogalski και Samurcay που επισημαίνουν ότι οι σπουδαστές δυσκολεύονται περισσότερο με σχέδια που βασίζονται στη στρατηγική 'έλεγχος τιμής μεταβλητής/επεξεργασία μεταβλητής' παρά στη στρατηγική 'επεξεργασία/έλεγχος' (Rogalski & Samurcay, 1990). Ο μεγαλύτερος βαθμός δυσκολίας που παρουσιάζει η χρήση της στρατηγικής top-exit, δηλαδή η δομή ***Όσο…επανάλαβε***, σε σχέση με την bottom-exit, δηλαδή με τη δομή ***Αρχή_επανάληψης…μέχρις_ότου***, από τους αρχάριους προγραμματιστές (Δαγδιλέλης, 1996) ίσως να οφείλεται στη δυσκολία αναπαράστασης και εκφοράς μιας συνθήκης για ένα αντικείμενο το οποίο ακόμα δεν έχουν επεξεργαστεί (Rogalski & Samurcay, 1990).

Τα συχνότερα <u>λάθη και λανθασμένες αντιλήψεις</u> των μαθητών που σχετίζονται με βρόχους, όπως προκύπτει από τις δύο μελέτες των Putman, Sleeman, Baxter και Kuspa, αλλά και μελέτες άλλων ερευνητών, είναι:

- Οι εντολές **Γράψε** αμέσως μετά από ένα βρόχο συμπεριλαμβάνονται σ' αυτόν (Sleeman et al., 1988; Putman et al., 1989).
- Ελεγχόμενη από τα δεδομένα-εισόδου επανάληψη (Data-driven looping): το πλήθος των δεδομένων καθορίζει πόσες φορές εκτελείται το σώμα του βρόχου ή αλλιώς οι επαναλήψεις συνεχίζονται όσο υπάρχουν δεδομένα να διαβαστούν (Sleeman et al., 1988; Putman et al., 1989).
- *Προβλήματα εμβέλειας* (scope problems): αρκετά λάθη οφείλονται σε παρανοήσεις σχετικά με την αρχή και το τέλος των βρόχων, καθώς και τις εντολές που επαναλαμβάνονται σε αυτούς (Sleeman et al., 1988): (i) *Μόνο η τελευταία εντολή ενός βρόχου εκτελείται κατ' επανάληψη*, ενώ οι υπόλοιπες μία μόνο φορά. Τα λάθος αυτό που ήταν αρκετά συχνό παρατηρήθηκε μόνο σε βρόχους στους οποίους η τελευταία εντολή ήταν η ***Γράψε***, (ii) *Η στοίχιση καθορίζει την εμβέλεια ενός βρόχου*, (iii) *Μετά την εκτέλεση ενός βρόχου, ο έλεγχος πηγαίνει στην πρώτη εντολή του προγράμματος*. Αν και τα παραπάνω λάθη ήταν σπάνια, το 1/3 των σπουδαστών που πέρασαν από συνέντευξη είχαν δυσκολίες που σχετίζονταν με την εμβέλεια των βρόχων.
- Ειδικά για τους βρόχους ***Για*** έχουν καταγραφεί οι εξής δυσκολίες/παρανοήσεις:
  - Η μεταβλητή ελέγχου (control variable) δεν έχει τιμή μέσα στο βρόχο (Sleeman et al., 1988).
  - Μερικοί σπουδαστές δεν κατανοούν ότι η μεταβλητή ελέγχου είναι μια μεταβλητή που αυξάνεται κατά ένα βήμα σε κάθε επανάληψη (du Boulay, 1989; Sleeman et al., 1988) και επομένως έχουν τη δυνατότητα να αλλάζουν την τιμή της στο σώμα του βρόχου (Putman et al., 1989).
  - Οι παραπάνω παρανοήσεις σχετικά με τους βρόχους **Για** ενδεχομένως να οφείλονται στο γεγονός ότι τόσο η μεταβολή της μεταβλητής ελέγχου όσο και η πραγματοποίηση του ελέγχου είναι έμμεσες (Δαγδιλέλης, 1996).
  - *Το πεδίο τιμών της μεταβλητής ελέγχου αποτελεί περιορισμό για τις τιμές που μπορούν να διαβαστούν στις μεταβλητές στο σώμα του βρόχου* με μια εντολή ***Διάβασε*** (Sleeman et al., 1988; Putman et al., 1989).

Επίσης, πολλοί σπουδαστές πιστεύουν ότι ένας βρόχος ***Όσο…επανάλαβε*** *τερματίζεται τη στιγμή που η συνθήκη ελέγχου παύει να αληθεύει* (daemon-exit) και όχι αφού η συνθήκη ελεγχθεί κατά την αμέσως επόμενη επανάληψη (du Boulay, 1989; Sleeman et al., 1988).

Τέλος, όσον αφορά στις δομές **Όσο…επανάλαβε** και **Αρχή_πανάληψης… μέχρις_ότου** έχουν καταγραφεί δυσκολίες σε προγράμματα στα οποία διαβάζεται μια αλληλουχία τιμών που τερματίζεται με μια τιμή-σημαία (π.χ. 99999) και γίνεται κάποια επεξεργασία των τιμών αυτών, για παράδειγμα υπολογίζεται η μέση τιμή των τιμών εκτός της τιμής-σημαίας (99999). Οι δυσκολίες εντοπίζονται στην αλληλεπίδραση του διαβάσματος των δεδομένων και του βρόχου. Για παράδειγμα, το πρώτο δεδομένο διαβάζεται εκτός του βρόχου και δεν υπολογίζεται στο σύνολο ή η τιμή 99999 συμπεριλαμβάνεται στο σύνολο, ενώ επισημαίνεται ότι η τιμή σημαία δεν συμμετέχει στην επεξεργασία.

ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΤΗΝ ΑΝΤΙΜΕΤΩΠΙΣΗ ΤΩΝ ΔΥΣΚΟΛΙΩΝ

Προκειμένου να εντοπίσουμε και να αντιμετωπίσουμε τις δυσκολίες και τις παρανοήσεις των μαθητών μπορούμε να ετοιμάσουμε *φύλλα ελέγχου* και να σχεδιάσουμε *δραστηριότητες βασισμένες στην υπάρχουσα διδακτική γνώση και σε εκπαιδευτικό λογισμικό*. Το πρόβλημα έγκειται στο γεγονός ότι ο περιορισμένος αριθμός των διδακτικών ωρών σε συνδυασμό με την έκταση της ύλης του μαθήματος καθιστά, τις περισσότερες φορές, απαγορευτική τη χρήση εκπαιδευτικού λογισμικού από τους μαθητές. Ωστόσο, ο διδάσκοντας μπορεί να ετοιμάσει δραστηριότητες για εκπαιδευτικό λογισμικό και να τις παρουσιάσει στην τάξη. Στη συνέχεια, παρουσιάζονται ενδεικτικά κάποιες ιδέες για τη σχεδίαση φύλλων ελέγχου/εργασίας και δραστηριοτήτων.

Δραστηριότητα 1η *- εντοπισμός και αντιμετώπιση δυσκολιών/ παρανοήσεων για τη δομή ακολουθίας.* Για τον εντοπισμό των δυσκολιών και των παρανοήσεων των μαθητών όσον αφορά στη δομή ακολουθίας μπορεί να τους δοθεί ένα φύλλο ελέγχου με ειδικά σχεδιασμένες ερωτήσεις σύντομης απάντησης. Στη συνέχεια παρουσιάζονται ενδεικτικά κάποιες ερωτήσεις και πιθανές λανθασμένες απαντήσεις που υποδηλώνουν την ύπαρξη παρανοήσεων που παρουσιάστηκαν σε προηγούμενες ενότητες.

Ερώτηση	**Απάντηση/ Παρανόηση**
1. Ποιες θα είναι οι τιμές των μεταβλητών Largest και Smallest μετά από την εκτέλεση της εντολής ***Διάβασε*** Largest, Smallest αν δοθούν ως είσοδος οι τιμές 5 και 10;	Largest = 10, Smallest = 5 *σημασιολογικά περιορισμένη απόδοση τιμής*
2. Ποιες θα είναι οι τιμές των μεταβλητών Άρτιος και Περιττός μετά από την εκτέλεση της εντολής ***Διάβασε*** Άρτιος, Περιττός αν δοθούν ως είσοδος οι τιμές 5, 2, 10 και 7;	Άρτιος = 2 & 10 Περιττός = 5 & 7 *πολλές τιμές διαβάζονται σε μια μεταβλητή - σημασιολογικά περιορισμένη απόδοση τιμής*

3. Ποιες θα είναι οι τιμές των μεταβλητών Α, Β, C μετά από την εκτέλεση του παρακάτω προγράμματος αν δοθούν ως είσοδος οι τιμές 15, 20 και 30; ***Πρόγραμμα Παράδειγμα*** ***Μεταβλητές*** Ακέραιες: Α, Β, C ***Αρχή*** ***Διάβασε*** C, B, A ***Τέλος_Προγράμματος***	Α = 15 Β = 20 C = 30 *η σειρά δήλωσης καθορίζει τη σειρά διαβάσματος*
4. Ποια θα είναι η τιμή της μεταβλητής Q μετά από την εκτέλεση των παρακάτω εντολών Q ← 1 ***Διάβασε*** Ρ Q ← Q + 1 αν δοθεί ως είσοδος η τιμή 10;	Q = 11 *σύγχυση δύο μεταβλητών:* η δεύτερη εντολή εκτελείται σαν να ήταν Q ← Ρ + 1

Μετά από τη συμπλήρωση του φύλλου ελέγχου από τους μαθητές ακολουθεί συζήτηση στην τάξη. Συγκεκριμένα, ο διδάσκοντας ομαδοποιεί και καταγράφει στον πίνακα τις απαντήσεις των μαθητών. Σε περίπτωση που κάποιος ή κάποιοι μαθητές έχουν δώσει λανθασμένη απάντηση ακολουθεί διάλογος προκειμένου να εντοπιστεί η αιτία του λάθους. Αν και τα λάθη των μαθητών μπορεί να είναι τυχαία, τις περισσότερες φορές οφείλονται σε ελλιπή γνώση και λανθασμένες αντιλήψεις/παρανοήσεις. Μέσα από το διάλογο γίνεται προσπάθεια εξάλειψης των παρανοήσεων και οικοδόμησης σωστής και πλήρους γνώσης. Για την επίτευξη του στόχου αυτού ο διδάσκοντας μπορεί να έχει ετοιμάσει απλά παραδείγματα και να τα εκτελέσει με τη βοήθεια λογισμικού στην τάξη. Για το σκοπό αυτό μπορεί να χρησιμοποιηθεί το λογισμικό της ***Γλωσσομάθειας*** (http://glossomatheia.studies.gr) που διατίθεται ελεύθερα. Αφού πρώτα ζητηθεί από τους μαθητές να προσδιορίσουν το αποτέλεσμα του προγράμματος που τους παρουσιάζεται κάθε φορά, ο διδάσκοντας εκτελεί το πρόγραμμα και ακολουθεί συζήτηση. Οι μαθητές σημειώνουν στο φύλλο ελέγχου (που έχουν συμπληρώσει) τα λάθη τους, ενώ ο διδάσκοντας τους υπαγορεύει και γράφουν σύντομα και περιεκτικά σχόλια για τις ερωτήσεις του φύλλου ελέγχου που τους δυσκόλεψαν. Τα φύλλα ελέγχου τα κρατάνε οι μαθητές και παρακινούνται να τα μελετήσουν κατά τη διάρκεια της μελέτης του μαθήματος στο σπίτι. Αν στο τέλος της δραστηριότητας μείνει χρόνος οι μαθητές μπορούν να πειραματιστούν με έτοιμα προγράμματα χρησιμοποιώντας τη *Γλωσσομάθεια*, μαθαίνοντας από τα λάθη τους.

Δραστηριότητα 2η – εξάλειψη της αντίληψης ύπαρξης *ανθρωπομορφικών χαρακτηριστικών στο υπολογιστικό σύστημα.* Όπως αναφέρθηκε στην ενότητα *Δομές Επιλογής* πολλές φορές οι σπουδαστές αποδίδουν ανθρωπομορφικά χαρακτηριστικά στο υπολογιστικό σύστημα και μεταφέρουν τη διατύπωση της λύσης ενός προβλήματος, η οποία είναι εκφρασμένη σε φυσική γλώσσα, ή ακόμα και καθημερινές συνομιλίες μεταξύ ανθρώπων σε μια γλώσσα προγραμματισμού. Για να διαπιστώσει ο διδάσκοντας αν η παρανόηση αυτή υπάρχει στους μαθητές μπορεί να τους δώσει ένα φύλλο ελέγχου/εργασίας με την εξής άσκηση:

«Να γράψετε αλγόριθμο που θα δέχεται την ηλικία ενός ατόμου και θα εμφανίζει το μήνυμα ΝΕΟΣ, ΜΕΣΟΚΟΠΟΣ ή ΓΕΡΟΣ αν θεωρήσουμε ότι ένα άτομο:

Κάτω από 30 είναι ΝΕΟΣ,

Κάτω από 50 είναι ΜΕΣΟΚΟΠΟΣ,

αλλιώς είναι ΓΕΡΟΣ.

Στη συνέχεια να εκτελέσετε τον αλγόριθμο στο χαρτί και να προσδιορίσετε το αποτέλεσμά του αν δοθεί ως είσοδος η τιμή 25, 32 και 60.»

Αφού δοθεί στους μαθητές ο απαραίτητος χρόνος για να συμπληρώσουν το φύλλο ελέγχου, τους ζητείται να παρουσιάσουν τη λύση που προτείνουν. Οι πιο πιθανές λύσεις που ενδέχεται να προτείνουν οι μαθητές είναι οι 3 πρώτες από τις λύσεις που παρουσιάζονται παρακάτω, από τις οποίες οι 2 πρώτες είναι προφανώς λανθασμένες. Για την εξοικονόμηση χρόνου ο διδάσκοντας μπορεί να έχει ετοιμάσει φωτοτυπίες με τις λύσεις αυτές, τις οποίες και θα μοιράσει στους μαθητές μετά από τη συμπλήρωση του αντίστοιχου φύλλου ελέγχου. Επίσης, ο διδάσκοντας μπορεί να έχει γράψει τα αντίστοιχα προγράμματα στη ΓΛΩΣΣΑ για να τα εκτελέσει με το λογισμικό Γλωσσομάθεια στην τάξη και να εξακριβώσουν οι μαθητές ποιο είναι πράγματι το αποτέλεσμα του κάθε προγράμματος. Βαρύτητα δίνεται σε εκείνα τα προγράμματα που έχουν προτείνει και οι ίδιοι οι μαθητές. Η 4η λύση παρουσιάζεται στους μαθητές με σκοπό να επισημανθεί ο τρόπος με τον οποίο μια δομή πολλαπλής επιλογής μπορεί να μετατραπεί σε μια αλληλουχία απλών δομών επιλογής που θα έχει το ίδιο αποτέλεσμα. Σε κάθε περίπτωση γίνεται συζήτηση και οι μαθητές μετά από υπόδειξη του διδάσκοντα κρατούν, στο φύλλο ελέγχου και στη φωτοτυπία που τους έχει δοθεί, σημειώσεις που θα τους βοηθήσουν στη μελέτη στο σπίτι.

Λύση 1η

```
Διάβασε x
Αν x<30 τότε
   Γράψε 'ΝΕΟΣ'
Τέλος_αν
Αν x<50 τότε
   Γράψε 'ΜΕΣΟΚΟΠΟΣ'
Τέλος_αν
Γράψε 'ΓΕΡΟΣ'
```

Λύση 2η

```
Διάβασε x
Αν x<30 τότε
   Γράψε 'ΝΕΟΣ'
Τέλος_αν
Αν x<50 τότε
   Γράψε 'ΜΕΣΟΚΟΠΟΣ'
Αλλιώς
   Γράψε 'ΓΕΡΟΣ'
Τέλος_αν
```

Λύση 3η

```
Διάβασε x
Αν x<30 τότε
   Γράψε 'ΝΕΟΣ'
Αλλιώς_Αν x<50 τότε
   Γράψε 'ΜΕΣΟΚΟΠΟΣ'
Αλλιώς
   Γράψε 'ΓΕΡΟΣ'
Τέλος_αν
```

Λύση 4η

```
Διάβασε x
Αν x<30 τότε
   Γράψε 'ΝΕΟΣ'
Τέλος_αν
Αν όχι (x<30) και x<50 τότε
   Γράψε 'ΜΕΣΟΚΟΠΟΣ'
Τέλος_αν
Αν όχι (x<30) και όχι (x<50) τότε
   Γράψε 'ΓΕΡΟΣ'
Τέλος_αν
```

Εναλλακτικά, μπορεί να δοθεί στους μαθητές εξαρχής ο προσδιορισμός του προβλήματος και οι 4 λύσεις και να τους ζητηθεί: (1) να προσδιορίσουν ποιες είναι σωστές και ποιες όχι, (2) να εντοπίσουν τα λάθη, (3) να απλοποιήσουν - όπου είναι εφικτό - τις συνθήκες των δομών επιλογής, (4) να προσδιορίσουν την «καλύτερη» λύση αιτιολογώντας την απάντησή τους.

Δραστηριότητα 3η *– εξοικείωση με προβλήματα στα οποία διαβάζεται μια αλληλουχία τιμών που τερματίζεται με μια τιμή-σημαία* (π.χ. 99999) και γίνεται κάποια επεξεργασία των τιμών αυτών. Τα προβλήματα αυτής της κατηγορίας μπορούν να αντιμετωπιστούν χρησιμοποιώντας τις επαναληπτικές δομές ***Όσο…επανάλαβε*** και ***Αρχή_επανάληψης…Μέχρις_ότου***. Βέβαια, παρόλο που μπορούμε να αναπτύξουμε σωστό πρόγραμμα χρησιμοποιώντας και τις 2 δομές, είναι προτιμότερο να χρησιμοποιούμε τη δομή

- ***Όσο…επανάλαβε*** όταν δεν θέλουμε να επεξεργαστούμε την τιμή σημαία
- ***Αρχή_επανάληψης…Μέχρις_ότου*** όταν θέλουμε να την επεξεργαστούμε

προκειμένου να αποφύγουμε συνήθη λάθη, όπως για παράδειγμα να διαβάσουμε την πρώτη τιμή εκτός του βρόχου και να μην την επεξεργαστούμε ή να επεξεργαστούμε την τιμή σημαία παρόλο που το πρόβλημα επισημαίνει ότι αυτή η τιμή δεν συμμετέχει στην επεξεργασία. Για την εξοικείωση των μαθητών με προβλήματα αυτού του τύπου τους δίνεται ένα φύλλο εργασίας με δύο απλά προβλήματα, από τα οποία στο ένα απαιτείται η επεξεργασία της τιμής σημαίας και στο άλλο όχι. Για παράδειγμα, μπορεί να τους δοθεί το εξής πρόβλημα:

Α. Να γράψετε αλγόριθμο που θα διαβάζει αριθμούς και θα υπολογίζει το άθροισμά τους. Η είσοδος των δεδομένων θα σταματάει όταν δοθεί η τιμή -999, η οποία δεν θα συμμετέχει στο άθροισμα. Ο αλγόριθμος θα εμφανίζει το άθροισμα των αριθμών.

Β. Να κατασκευάσεις πίνακα τιμών προκειμένου να προσδιορίσεις ποιο θα είναι το άθροισμα αν δοθούν ως είσοδος οι αριθμοί 10, 20, 15, 5, 60, -999.

Γ. Μπορείς να γράψεις ισοδύναμο αλγόριθμο χρησιμοποιώντας κάποια άλλη επαναληπτική δομή από εκείνη που χρησιμοποίησες; Αν ναι, ποια δομή μπορείς να χρησιμοποιήσεις;»

Αρχικά, ζητείται από τους μαθητές να απαντήσουν στα 3 ερωτήματα. Στη συνέχεια καταγράφονται οι λύσεις που προτείνουν οι μαθητές και τα αποτελέσματα της εκτέλεσης όπως έχουν υπολογιστεί στο Β ερώτημα και ακολουθεί συζήτηση. Για την καλύτερη εκμετάλλευση του διαθέσιμου χρόνου, ο διδάσκοντας μπορεί να έχει ετοιμάσει διάφορες πιθανές λύσεις του προβλήματος ώστε να τις παρουσιάσει στους μαθητές και να σχολιαστούν στην τάξη από τους ίδιους. Οι λύσεις που θα παρουσιαστούν μπορεί να έχουν τη μορφή προγραμμάτων, τα οποία θα εκτελεστούν με τη Γλωσσομάθεια προκειμένου οι μαθητές να διαπιστώσουν άμεσα ποιο είναι το αποτέλεσμά τους για την είσοδο που προσδιορίζεται στο ερώτημα Β. Βέβαια, το πρόβλημα και η είσοδος επιλέχθηκαν έτσι ώστε ο μαθητής να μπορεί εύκολα να ελέγξει αν ο αλγόριθμός του – τον οποίο εκτελεί αρχικά στο χαρτί – δίνει σωστό αποτέλεσμα. Στις λύσεις που θα παρουσιάσει ο διδάσκοντας και τις οποίες θα έχουν προτείνει και οι ίδιοι οι μαθητές θα πρέπει να περιλαμβάνονται και οι λανθασμένες λύσεις. Παρακάτω παρουσιάζονται 3 πιθανές λύσεις, από τις οποίες η 3η είναι λανθασμένη, αλλά είναι πολύ πιθανό να προταθεί από ένα σημαντικό αριθμό μαθητών. Ακόμα και αν η λύση αυτή δεν προταθεί από κάποιον από τους μαθητές μπορεί να παρουσιαστεί στην τάξη και να ζητηθεί να προσδιορίσουν οι μαθητές αν είναι σωστή ή όχι.

Λύση 1η
```
Sum ← 0
Διάβασε x
Όσο x ≠ -999 επανάλαβε
   Sum ← Sum + x
   Διάβασε x
Τέλος_επανάληψης
Γράψε Sum
```

Λύση 2η
```
Sum ← 0
Διάβασε x
Αν x ≠ -999 τότε
   Αρχή_επανάληψης
      Sum ← Sum + x
      Διάβασε x
   Μέχρις_ότου χ = -999
Τέλος_αν
Γράψε Sum
```

Λύση 3η
```
Sum ← 0
Διάβασε x
Αρχή_επανάληψης
   Sum ← Sum + x
   Διάβασε x
Μέχρις_ότου χ = -999
Γράψε Sum
```

Μετά από την παρουσίαση των λύσεων και τη συζήτηση που θα πραγματοποιηθεί στην τάξη θα έχει δοθεί στην ουσία η απάντηση και στο ερώτημα Γ. Ο διδάσκοντας λοιπόν παρουσιάζει

ταυτόχρονα στους μαθητές τις 2 πρώτες από τις λύσεις που φαίνονται παραπάνω και ακολουθεί συζήτηση για τον τρόπο με τον οποίο μπορούμε να μετατρέψουμε μια δομή ***Όσο…επανάλαβε*** σε ***Αρχή_επανάληψης….Μέχρις_ότου*** και αντίστροφα. Οι μαθητές κρατάνε σημειώσεις στο φύλλο εργασίας, το οποίο και παροτρύνονται να μελετήσουν στο σπίτι. Αν ο χρόνος δεν επαρκεί για την επίλυση του 2ου προβλήματος, τότε αυτό ανατίθεται στους μαθητές ως εργασία για το σπίτι.

ΕΠΙΛΟΓΟΣ

Η εκμάθηση των αλγοριθμικών δομών στα πλαίσια του μαθήματος ΑΕΠΠ είναι γνωστό ότι παρουσιάζει αρκετές δυσκολίες. Σημαντική στήριξη στην αντιμετώπιση των δυσκολιών μπορεί αναμφισβήτητα να παρέχει η χρήση στην τάξη εκπαιδευτικού λογισμικού, η οποία όμως καθίσταται δύσκολη, αν όχι ανέφικτη, λόγω του περιορισμένου αριθμού των διδακτικών ωρών. Ο διδάσκοντας λοιπόν καλείται να διαμορφώσει κατάλληλες διδακτικές καταστάσεις που θα βοηθήσουν τους μαθητές να αντιμετωπίσουν τις δυσκολίες τους στο διαθέσιμο χρόνο. Σημαντική βοήθεια πιστεύουμε ότι παρέχει η σχεδίαση δραστηριοτήτων που βρίσκονται στο επίκεντρο των δυσκολιών που αντιμετωπίζουν οι μαθητές. Στην παρούσα εργασία έγινε προσπάθεια σύνοψης των αποτελεσμάτων της εκτεταμένης έρευνας που έχει διεξαχθεί σχετικά με τη διδασκαλία και εκμάθηση των βασικών αλγοριθμικών δομών, έχοντας ως στόχο την αξιοποίησή τους από το διδάσκοντα για τη σχεδίαση τέτοιων δραστηριοτήτων. Επίσης, παρουσιάστηκαν ενδεικτικά κάποιες δραστηριότητες που αξιοποιούν τα αποτελέσματα της σύνοψης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Du Boulay (1989), Some Difficulties Of Learning To Program, In E. Soloway, J. Sprohrer (Eds.) Studying The Novice Programmer, 283-300, Lawrence Erlbaum Associates
2. Hoc (1989), Do We Really Have Conditionals In Our Brains?, In E. Soloway, J. Sprohrer (Eds.) Studying The Novice Programmer, 179-190, Lawrence Erlbaum Associates
3. Kessler & Anderson (1989), Learning Flow of Control: Recursive and Iterative Procedures, In E. Soloway, J. Sprohrer (Eds.) Studying The Novice Programmer, 229-260, Lawrence Erlbaum Associates
4. Pane & Myers (1996), Usability Issues in the Design of Novice Programming Systems, Technical Report CMU-CS-96-132, Carnegie Mellon University
5. Putman, R., Sleeman, D., Baxter, J. & Kuspa, L. (1989), A Summary Of Misconceptions Of High-School BASIC Programmers. In E. Soloway, J. Sprohrer (Eds.) Studying The Novice Programmer, 301-314, Lawrence Erlbaum Associates
6. Rogalski & Samurcay (1990), Acquisition of Programming Knowledge and Skills. In J. Hoc, T. Green, R. Samurcay & D. Gilmore (Eds.), Psychology of Programming, 157-174, Academic Press
7. Samurcay (1989), The Concept of Variable in Programming: its Meaning and Use in Problem-Solving by Novice Programmers, In E. Soloway, J. Sprohrer (Eds.) Studying The Novice Programmer, 161-178, Lawrence Erlbaum Associates
8. Sleeman, D., Putman, R., Baxter, J. & Kuspa, L. (1988), An Introductory Pascal Class: A Case Study Of Students' Errors, In R. Mayer (Ed.), Teaching and Learning Computer Programming , 237-258, Lawrence Erlbaum Associates
9. Αθανασόπουλος, Δ. & Οικονόμου, Γ. (2004), Συμπεράσματα από τις επιδόσεις των μαθητών στις Πανελλαδικές εξετάσεις του μαθήματος της ανάπτυξης Εφαρμογών, 2η Πανελλήνια Διημερίδα «Διδακτική της Πληροφορικής», 315-318, Βόλος.
10. Βακάλη, Α., Γιανννόπουλος, Η., Ιωαννίδης, Ν. Κοίλιας, Χ., Μάλαμας, Κ., Μανωλόπουλος, Ι. & Πολίτης, Π. (1999), Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (Βιβλίο/ τετράδιο μαθητή & Βιβλίο καθηγητή), ΟΕΔΒ
11. Δαγδιλέλης, Β. (1996), Διδακτική της πληροφορικής. Η διδασκαλία του προγραμματισμού: αντιλήψεις των σπουδαστών για την κατασκευή κι επικύρωση προγραμμάτων και διδακτικές καταστάσεις για τη διαμόρφωσή τους. Διδακτορική διατριβή, Τμήμα Εφ. Πληροφορικής Πανεπιστήμιο Μακεδονίας