

Διδάσκοντας το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον¹

Π. Κ. Γιαννοπούλου
Εκπαιδευτικός ΠΕ19, Λεόντειο Λύκειο Πατησίων
nagia@math.ntua.gr
Σ. Γ. Δουκάκης
Εκπαιδευτικός ΠΕ19, Αμερικανικό Κολλέγιο Ελλάδος, Pierce College
sdoukakis@acg.edu
Χ. Κοΐλιας
Επίκουρος Καθηγητής ΤΕΙ Αθήνας, Τμήμα Πληροφορικής
ckoilias@teiath.gr
Α. Ψαλτίδου
Εκπαιδευτικός ΠΕ19, Αμερικανικό Κολλέγιο Ελλάδος, Pierce College
alex_psa@yahoo.com

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αλγόριθμος, Ψευδογλώσσα, Προγραμματισμός, Επίλυση προβλημάτων
Εκπαιδευτικό Λογισμικό, Διδακτική Πληροφορικής

ΠΕΡΙΛΗΨΗ

Στην εργασία αυτή παρουσιάζεται η σημαντικότητα της αλγοριθμικής επίλυσης προβλημάτων. Επεξηγείτε η σημασία του μαθήματος Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον για την ανάπτυξη της αλγοριθμικής σκέψης των μαθητών και παρουσιάζονται κάποιες παρανοήσεις και προβλήματα που εμφανίζονται κατά τη διδασκαλία του μαθήματος και την αξιολόγηση των μαθητών. Διατυπώνονται προτάσεις για τη βελτίωση του εκπαιδευτικού υλικού του μαθήματος τόσο για τη διδασκαλία στην τάξη, όσο και στο εργαστήριο.

1. Εισαγωγή

Η αλγοριθμική προσέγγιση ενός προβλήματος δίνει έμφαση στον τρόπο επίλυσης του προβλήματος, ανεξάρτητα από το χώρο από τον οποίο προέρχεται το πρόβλημα. Η ικανότητα περιγραφής της επίλυσης ενός προβλήματος με μια μορφή αναπαράστασης αλγορίθμου εφοδιάζει το μαθητή με έναν γενικό οδηγό για την αντιμετώπιση προβλημάτων. Η αλγοριθμική σχεδίαση επίλυσης προβλημάτων αναπτύσσει γνωστικές δεξιότητες υψηλού επιπέδου και αποτελεί ένα ισχυρό μέσο για τη διδασκαλία βασικών εννοιών που βρίσκουν εφαρμογή σε όλους τους τομείς δραστηριοτήτων [Papert, 1980]. Αφού, λοιπόν, ένας από τους σκοπούς της εκπαίδευσης είναι και η ανάπτυξη της σκέψης των μαθητών, ένα αντικείμενο διδασκαλίας στον τρόπο με τον οποίο οι μαθητές πρέπει να σκέπτονται αλγοριθμικά είναι ένα απαραίτητο και άκρως δημιουργικό εφόδιο.

2. Η σημασία και οι δυνατότητες της αλγοριθμικής σχεδίασης

Οι μαθητές καθημερινά, τόσο στο σχολείο όσο και σε άλλες τους δραστηριότητες καλούνται να επιλύσουν προβλήματα. Αυτό δεν είναι πάντα εύκολο. Πολλές φορές το πρόβλημα δεν είναι κατανοητό, αφού μπορεί να μην έχει διατυπωθεί σωστά από το δημιουργό του ή μπορεί να ερμηνεύεται λάθος από τη μεριά του επίδοξου λύτη. Άλλες φορές, η επίλυση του προβλήματος δεν είναι προφανής και υπάρχει και η περίπτωση οι μαθητές να έχουν καταλήξει

¹ Γιαννοπούλου Π., Δουκάκης Σ., Κοΐλιας Χ., Ψαλτίδου Α. (2003), Διδάσκοντας το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, 6η Δημερίδα για την Πληροφορική στη Δευτεροβάθμια Εκπαίδευση, ΕΠΥ, 31/10-01/11/2003, Αθήνα, σ. 105-118

στον τρόπο επίλυσης, αλλά να μην μπορούν να επικοινωνήσουν (δηλαδή να διατυπώσουν ή να αναπαράγουν) τον τρόπο επίλυσης που έχουν σκεφθεί.

Σε όλα τα σχολικά μαθήματα, από την έκθεση έως και τα μαθηματικά, οι μαθητές καλούνται να αναπτύξουν θέματα με σκοπό να επιλύσουν ένα πρόβλημα (συγγραφή έκθεσης, σχολιασμός κειμένου, επίλυση μιας εξίσωσης, τακτική άμυνας σε έναν αγώνα ποδοσφαίρου). Αφού, λοιπόν, ο μαθητής κατανοήσει το πρόβλημα και σκεφτεί αλγοριθμικά την επίλυσή του, καταγράφει τη λύση. Έτσι, στην έκθεση αφού κατανοήσει το θέμα δημιουργεί ένα σχεδιάγραμμα, το οποίο ακολουθεί για τη συγγραφή της. Στα μαθηματικά, για την επίλυση της δευτεροβάθμιας εξίσωσης, ο μαθητής συγκεντρώνει τα δεδομένα και με εφαρμογή των μαθηματικών κανόνων προσδιορίζει τις λύσεις της εξίσωσης. Στη σχολική ομάδα καλαθοσφαίρισης, ο γυμναστής-προπονητής δημιουργεί ένα σχεδιάγραμμα θέσεων των παικτών για τον τρόπο επίθεσης. Συνεπώς, οι μαθητές λύνουν προβλήματα χωρίς να έχουν ποτέ διδαχτεί τον αλγοριθμικό τρόπο σκέψης.

Η αλγοριθμική επίλυση ενός προβλήματος, μπορεί να καταγραφεί με τους ακόλουθους τρόπους:

1. **Ελεύθερο κείμενο**, το οποίο αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου.
2. **Φυσική γλώσσα με βήματα**.
3. **Διαγραμματικές τεχνικές** που συνιστούν ένα γραφικό τρόπο παρουσίασης του αλγορίθμου.
4. **Κωδικοποίηση** δηλαδή με ένα πρόγραμμα γραμμένο είτε με μία ψευδογλώσσα, είτε σε κάποιο προγραμματιστικό περιβάλλον που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο [Βακάλη et al., 1999].

Η περιγραφή της επίλυσης ενός προβλήματος με έναν από τους παραπάνω τρόπους αναπαράστασης εφοδιάζει το μαθητή με έναν γενικό οδηγό για την αντιμετώπιση των προβλημάτων που του θέτονται προς επίλυση.

Η εκπαιδευτική διαδικασία, λοιπόν, απαιτεί από τους εκπαιδευόμενους να μπορούν να επιλύουν καθημερινά ή επιστημονικά προβλήματα, γεγονός που δηλώνει ότι πρέπει να σκέπτονται αλγοριθμικά και στη συνέχεια να καταγράφουν τη λύση του προβλήματος χρησιμοποιώντας την αλγοριθμική προσέγγιση.

3. Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον:

Διδάσκοντας την αλγοριθμική σχεδίαση επίλυσης προβλημάτων

Το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (ΑΕΠΠ) διδάσκεται για τέταρτη συνεχή χρονιά. Το μάθημα υποστηρίζεται από διδακτικό πακέτο τριών βιβλίων, που συμπεριλαμβάνουν βιβλίο μαθητή, τετράδιο μαθητή και βιβλίο καθηγητή. Οι εκπαιδευτικοί, μέσα από το διδακτικό πακέτο, επιχειρούν να εξοικειώσουν τους μαθητές με την αλγοριθμική επίλυση προβλημάτων.

Σύμφωνα με τη συγγραφική ομάδα, το μάθημα δεν έχει στόχο τη διδασχία και την εκμάθηση κάποιου συγκεκριμένου προγραμματιστικού περιβάλλοντος, ούτε την καλλιέργεια προγραμματιστικών δεξιοτήτων από τη μεριά των μαθητών. Στόχος του είναι η δόμηση της σκέψης των μαθητών και η διδασκαλία τεχνικών επίλυσης προβλημάτων. Ο τρόπος προσέγγισης στις αλγοριθμικές δομές από το διδακτικό πακέτο, δίνει έμφαση στην "ελευθερία" της αλγοριθμικής κωδικοποίησης, τονίζοντας έτσι την αντίθεση με το αυστηρό συντακτικό και λεξιλόγιο μιας γλώσσας προγραμματισμού. Αυτή η "ελευθερία", που αποτελεί ένα από τα βασικότερα πλεονεκτήματα της αλγοριθμικής σχεδίασης, δημιουργεί και προβλήματα. Για παράδειγμα, η ίδια αλγοριθμική ενέργεια μπορεί να εκφραστεί με πολλούς τρόπους. Οι εκπαιδευτικοί όμως, κατά τη διδασκαλία αλλά και κατά την αξιολόγηση των γραπτών εξετάσεων, προσπαθούν να αφαιρέσουν πολλές από τις "ελευθερίες" με σκοπό την τυποποίηση της ανάπτυξης αλγορίθμων. Η τυποποίηση αυτή, "καταδυναστεύει" την "ελευθερία" έκφρασης του αλγορίθμου με αποτέλεσμα να δημιουργούνται ασάφειες και παρανοήσεις σε θέματα παρουσίασης, ερμηνείας και αξιολόγησης αλγοριθμικών δομών [Δουκάκης et al, 2003].

3.1. Προβλήματα κατά τη διδασκαλία του μαθήματος ΑΕΠΠ στην τάξη

Επιλέξαμε ένα χαρακτηριστικό παράδειγμα αλγορίθμου από το διδακτικό πακέτο, για να συζητήσουμε κάποια από αυτά τα προβλήματα που αντιμετωπίζουν εκπαιδευτικοί και μαθητές. Ο παρακάτω αλγόριθμος επιλύει αλγοριθμικά την αναζήτηση ενός στοιχείου σε μη ταξινομημένο πίνακα.

Αλγόριθμος Σειριακή_Αναζήτηση

Δεδομένα // N, πίνακας, ζητούμενο //

βρέθηκε ← ψευδής

θέση ← 0

i ← 1

Όσο (βρέθηκε = ψευδής) **και** ($i \leq N$) **επανάλαβε**

Αν πίνακας[i] = ζητούμενο **τότε**

 βρέθηκε ← αληθής

 θέση ← i

αλλιώς

 i ← i + 1

Τέλος_αν

Τέλος_επανάληψης

Αποτελέσματα // βρέθηκε, θέση //

Τέλος Σειριακή_Αναζήτηση

Η παραπάνω σειρά ενεργειών είναι αποτελεσματική για οποιοδήποτε τύπο δεδομένων. Ο ίδιος αλγόριθμος μπορεί να χρησιμοποιηθεί για οποιοδήποτε πλήθος στοιχείων, τα οποία μπορεί να είναι είτε αλφαριθμητικά σε περίπτωση αναζήτησης ενός ονόματος μέσα σε πίνακα που περιέχει ονόματα μαθητών, είτε αριθμητικά (ακέραια ή πραγματικά) σε περίπτωση αναζήτησης ενός συγκεκριμένου ποσού μέσα σε πίνακα που περιέχει τιμές προϊόντων. Στη φάση της αλγοριθμικής σχεδίασης για την επίλυση του προβλήματος αναζήτησης σε μη ταξινομημένο πίνακα, είναι περιττή η δήλωση του τύπου των μεταβλητών. Η αλγοριθμική σχεδίαση κωδικοποιεί τον τρόπο επίλυσης ανεξάρτητα από τον τύπο των μεταβλητών που χρησιμοποιεί, δίνοντας βαρύτητα στον τρόπο που επεξεργάζονται οι μεταβλητές και όχι στον τύπο τους.

Έτσι στο μάθημα ΑΕΠΠ, οι εκπαιδευτικοί οφείλουν να τονίζουν ότι οι δηλώσεις μεταβλητών δεν επιτελούν κάποιο σκοπό κατά την ανάπτυξη αλγορίθμου. Ο βασικός ρόλος του αλγορίθμου είναι να κωδικοποιήσει αφαιρετικά την επίλυση ενός προβλήματος και στη συνέχεια, αυτός ο αλγόριθμος να μεταφερθεί σε κάποια γλώσσα προγραμματισμού. Είναι γνωστό ότι στις περισσότερες γλώσσες προγραμματισμού, είναι υποχρεωτική η δήλωση τύπου των μεταβλητών που χρησιμοποιούνται. Έτσι, ο εκπαιδευτικός οφείλει παράλληλα με την αλγοριθμική σχεδίαση να διδάσκει στους μαθητές τους τύπους των μεταβλητών, αλλά και τις πράξεις που υποστηρίζει κάθε τύπος. Για παράδειγμα, όταν ο εκπαιδευτικός παρουσιάσει στους μαθητές τον παραπάνω αλγόριθμο, μετά από τις απαραίτητες επεξηγήσεις για τον τρόπο που αντιμετωπίζει το πρόβλημα της αναζήτησης, πρέπει να αναφερθεί στους τύπους των μεταβλητών, χωρίς όμως αυτό να καταγραφεί μέσα στον αλγόριθμο. Συγκεκριμένα, είναι χρήσιμο να δοθεί έμφαση στο γεγονός ότι το στοιχείο που αναζητείται πρέπει να είναι ίδιου τύπου με τον πίνακα, ότι το i, το N και η θέση πρέπει να είναι μεταβλητές ακέραιου τύπου και ότι η μεταβλητή βρέθηκε είναι λογικού τύπου και μπορεί να λάβει μόνο τις τιμές αληθής ή ψευδής. Καθιερώνοντας την ερώτηση "Τι τύπου μεταβλητές χρησιμοποιήσατε;", ο εκπαιδευτικός ελέγχει τακτικά τη γνώση των μαθητών, και οι μαθητές αντιλαμβάνονται τη σημασία του τύπου των μεταβλητών, εφόδιο απαραίτητο για τον προγραμματισμό. Παράλληλα ο εκπαιδευτικός, πρέπει να επεξηγήσει τη φυσική σημασία των

μεταβλητών, δηλαδή την αντιστοιχία τους με μια θέση μνήμης. Έτσι, ο μαθητής θα εξοικειωθεί με την οικονομία των μεταβλητών, συνθήκη απαραίτητη στον προγραμματισμό.

Μια άλλη παρανόηση μπορεί να προέλθει από τη χρήση των δηλωτικών εντολών **Δεδομένα** και **Αποτελέσματα**. Έτσι, στο παράδειγμα της σειριακής αναζήτησης εμφανίζονται οι δηλωτικές εντολές:

Δεδομένα // N, πίνακας, ζητούμενο // και **Αποτελέσματα** // βρέθηκε, θέση //

Ποιος είναι ο ρόλος αυτών των εντολών; Ο αλγόριθμος δεν είναι εκτελέσιμος από υπολογιστή. Ωστόσο, κάθε αλγόριθμος χρειάζεται στοιχεία εισόδου, τα οποία ονομάζονται δεδομένα και παράγει στοιχεία εξόδου, τα οποία ονομάζονται αποτελέσματα. Όταν σε ένα πρόβλημα δεν καθορίζεται ο τρόπος εισαγωγής των δεδομένων, τότε ο συγγραφέας δεν χρειάζεται να ασχολείται με αυτόν. Στο πρόβλημα της σειριακής αναζήτησης, ο αλγόριθμος δέχεται το πλήθος των στοιχείων ενός πίνακα, τις τιμές των στοιχείων του και την τιμή που αναζητείται. Έτσι, αν η εκφώνηση προσδιορίζει "Δίνεται μη ταξινομημένος μονοδιάστατος πίνακας, N διαφορετικών αριθμητικών στοιχείων. Να αναπτύξετε αλγόριθμο ο οποίος...", τότε πρέπει να χρησιμοποιηθεί η δηλωτική εντολή **Δεδομένα** για την εισαγωγή αυτών των δεδομένων. Με αυτό τον τρόπο καθορίζεται ότι ο αλγόριθμος λαμβάνει κάποιες τιμές ως στοιχεία εισόδου.

Ωστόσο, αν η εκφώνηση προσδιορίζει "Να αναπτύξετε αλγόριθμο ο οποίος θα διαβάζει τις τιμές 100 αριθμητικών στοιχείων ενός μονοδιάστατου πίνακα, ...", τότε πρέπει να χρησιμοποιηθεί μια επαναληπτική διαδικασία και η εντολή **Διάβασε** για την εισαγωγή των δεδομένων. Αν μάλιστα η εκφώνηση απαιτεί και έλεγχο εγκυρότητας των τιμών που θα εισαχθούν στον αλγόριθμο, τότε η εντολή **Διάβασε** πρέπει να εμπεριέχεται σε μια επαναληπτική δομή, έτσι ώστε να διαβάζονται μόνο οι επιθυμητές τιμές.

Αν η εκφώνηση του αλγορίθμου, υποδεικνύει τη χρήση της εντολής **Δεδομένα**, αλλά ο λύτης χρησιμοποιήσει την εντολή **Διάβασε**, τότε επιλύεται ένα διαφορετικό πρόβλημα, με διαφορετικές απαιτήσεις. Με αυτήν την επιλογή, ελλοχεύει ο κίνδυνος της τροποποίησης του τρόπου εισόδου δεδομένων που έχει επιλεγεί από το πρόβλημα, αλλά και της αλλοίωσης των στοιχείων που θεωρεί δεδομένα ο αλγόριθμος.

Τα αποτελέσματα του αλγορίθμου δεν παρουσιάζονται σε κάποια μονάδα εξόδου, αλλά επειδή η κατασκευή ενός αλγορίθμου αποσκοπεί στη μετέπειτα μεταφορά του σε κάποιο προγραμματιστικό περιβάλλον, συνηθίζουμε να θεωρούμε ότι τα αποτελέσματα θα παρουσιαστούν είτε στην οθόνη, είτε στον εκτυπωτή. Επιπλέον, τα αποτελέσματα, μπορεί να χρησιμοποιηθούν ως τιμές εισόδου σε ένα νέο αλγόριθμο. Έτσι, αν το πρόβλημα δεν προσδιορίζει σαφώς τον τρόπο με τον οποίο θα γίνει η έξοδος των αποτελεσμάτων, τότε χρησιμοποιείται η δηλωτική εντολή **Αποτελέσματα** για την έξοδό τους.

Έτσι, αν η εκφώνηση προσδιορίζει "Δίνεται μη ταξινομημένος μονοδιάστατος πίνακας A, N διαφορετικών αριθμητικών στοιχείων. Να αναπτύξετε αλγόριθμο ο οποίος θα προσπελαύνει τον πίνακα με τη μέθοδο της σειριακής αναζήτησης ψάχνοντας κάποιο στοιχείο. Η επαναληπτική δομή θα πρέπει να τερματίζει αν εντοπισθεί το ζητούμενο στοιχείο.", τότε πρέπει να χρησιμοποιηθεί η δηλωτική εντολή **Αποτελέσματα** για την παρουσίαση των αποτελεσμάτων, τα οποία βάσει του παραπάνω αλγορίθμου θα είναι η τιμή της μεταβλητής βρέθηκε και η θέση του στοιχείου μέσα στον πίνακα.

Ωστόσο, αν η εκφώνηση προσδιορίζει "Δίνεται μη ταξινομημένος μονοδιάστατος πίνακας A, N διαφορετικών αριθμητικών στοιχείων. Να αναπτύξετε αλγόριθμο ο οποίος θα προσπελαύνει τον πίνακα με τη μέθοδο της σειριακής αναζήτησης ψάχνοντας κάποιο στοιχείο. Η επαναληπτική δομή θα πρέπει να τερματίζει αν εντοπισθεί το ζητούμενο στοιχείο. Ο αλγόριθμος θα εμφανίζει το μήνυμα "Δεν βρέθηκε" σε περίπτωση που δεν υπάρχει, ενώ αν υπάρχει θα εμφανίζει το μήνυμα "Βρέθηκε" καθώς και τη θέση του ζητούμενου στοιχείου", τότε πρέπει να χρησιμοποιηθεί η εντολή **Εμφάνισε** για την παρουσίαση των αποτελεσμάτων στην οθόνη. Ισοδύναμα, αν στην εκφώνηση χρησιμοποιηθεί η λέξη "...τυπώνει ή εκτυπώνει το μήνυμα...", τότε πρέπει να χρησιμοποιηθεί η εντολή **Εκτύπωσε** ή **Γράψε** για την παρουσίαση των αποτελεσμάτων στον εκτυπωτή. Οι εντολές **Εμφάνισε**, **Εκτύπωσε** και **Γράψε** είναι ισοδύναμες, αφού

στην πραγματικότητα δηλώνουν την παρουσίαση των αποτελεσμάτων και όχι την ενεργοποίηση διαφορετικών μονάδων εξόδου.

Η χρήση των εντολών **Δεδομένα** και **Αποτελέσματα** βοηθάει και ως προς την κατανόηση των αλγορίθμων που καλούνται από άλλο αλγόριθμο. Πράγματι, αν ένας μαθητής έχει κατανοήσει ότι ένας αλγόριθμος δέχεται κάποια δεδομένα και επιστρέφει κάποια αποτελέσματα, τότε είναι εύκολο να κατανοήσει την έννοια των παραμέτρων σε άλλο αλγόριθμο. Ένας αλγόριθμος μπορεί να κληθεί από έναν άλλο αλγόριθμο με χρήση της εντολής **Κάλεσε**. Η επικοινωνία μεταξύ των δύο αλγορίθμων μπορεί να γίνει με τους επόμενους τρόπους:

α) Με μεταβίβαση τιμών

Στον τρόπο αυτό η εντολή **Κάλεσε** συνοδεύεται με το όνομα του καλούμενου αλγορίθμου ακολουθούμενο από λίστα μεταβλητών ή σταθερών μέσα σε παρενθέσεις. Οι τιμές των μεταβλητών ή σταθερών μεταβιβάζονται κατά την κλήση στις αντίστοιχες μεταβλητές της γραμμής **Δεδομένα** του καλούμενου αλγορίθμου. Όταν ο καλούμενος αλγόριθμος τερματίσει τη λειτουργία του, γίνεται επιστροφή στην αμέσως επόμενη εντολή της **Κάλεσε**. Κατά την επιστροφή μπορεί επίσης να μεταβιβάζονται τιμές της εντολής **Αποτελέσματα**. Για παράδειγμα:

Αλγόριθμος Καλών

Δεδομένα // πλήθος, πίνακας_τιμών, τιμή //

Κάλεσε Σειριακή_Αναζήτηση (πλήθος, πίνακας_τιμών, τιμή, ύπαρξη, θέση)

Γράψε ύπαρξη, θέση

Τέλος Καλών

Ο μαθητής έχει ήδη εκπονήσει τον αλγόριθμο Σειριακή_Αναζήτηση που αναφέρθηκε προηγούμενα και χρειάζεται να τον χρησιμοποιήσει σε ένα άλλο πρόβλημα. Η κλήση του αλγορίθμου Σειριακή_Αναζήτηση, από έναν άλλον αλγόριθμο γίνεται με τη χρήση της εντολής **Κάλεσε** Σειριακή_Αναζήτηση (πλήθος, πίνακας_τιμών, τιμή, ύπαρξη, θέση).

Η αντιστοιχία των μεταβλητών των δύο αλγορίθμων γίνεται με τη σειρά που αναφέρονται στις αντίστοιχες γραμμές, δηλαδή:

πλήθος \Rightarrow N, πίνακας_τιμών \Rightarrow πίνακας, τιμή \Rightarrow ζητούμενο, ύπαρξη \Rightarrow βρέθηκε, θέση \Rightarrow θέση. Συνόνομες μεταβλητές διαφορετικών αλγορίθμων δεν έχουν καμία σχέση μεταξύ τους.

Ο τρόπος αυτός προτιμάται όταν επιδιώκεται ανεξαρτησία των καλούμενων αλγορίθμων από τους καλούντες. Έτσι, επιτυγχάνεται η επαναχρησιμοποίηση των αλγορίθμων σε διάφορα προβλήματα. Οι μεταβλητές των δύο αλγορίθμων είναι ξένες μεταξύ τους, ακόμη και αν έχουν ίδια ονόματα (όπως στην περίπτωση της μεταβλητής θέση).

β) Με χρήση καθολικών μεταβλητών

Στην περίπτωση αυτή οι μεταβλητές των δύο αλγορίθμων είναι κοινές. Το γεγονός αυτό επισημαίνεται με τη χρήση της δηλωτικής εντολής **Καθολικές Μεταβλητές** στην αρχή του καλούντος αλγορίθμου.

Στον καλούμενο αλγόριθμο δεν θα υπάρχουν οι εντολές **Δεδομένα** και **Αποτελέσματα** μια και χρησιμοποιούνται ίδιες μεταβλητές με αυτές του καλούντος αλγορίθμου. Όλες οι εντολές του καλούμενου αλγορίθμου μπορούν να ενθεθούν στον καλούντα στη θέση της εντολής **Κάλεσε**.

Ο τρόπος αυτός γενικά προτιμάται, όταν υπάρχουν πολλές μεταβλητές σε ένα πρόβλημα ή όταν καλούνται πολλοί και διαφορετικοί αλγόριθμοι από έναν (κύριο) αλγόριθμο, οι οποίοι επεξεργάζονται τα ίδια δεδομένα.

γ) Μεικτός τρόπος

Και οι δύο προηγούμενοι τρόποι μπορούν να αναμειχθούν. Σε αυτήν την περίπτωση άλλες μεταβλητές θα είναι καθολικές και άλλες θα μεταβιβάζονται κατά την κλήση. Για παράδειγμα, στον αλγόριθμο της σειριακής αναζήτησης, οι μεταβλητές πίνακας και N γίνονται καθολικές. Τότε η γραμμή δεδομένων του αλγορίθμου Σειριακή_Αναζήτηση γίνεται **Δεδομένα** // ζητούμενο // και η κλήση αυτού γίνεται **Κάλεσε** Σειριακή_Αναζήτηση (τιμή, ύπαρξη, θέση).

Ο τρόπος αυτός προτιμάται όταν από έναν (κύριο) αλγόριθμο καλούνται πολλοί άλλοι αλγόριθμοι που επεξεργάζονται τα ίδια δεδομένα (π.χ. πίνακες, πεδία εγγραφής αρχείου κ.λπ.). Τα κοινά αυτά δεδομένα ορίζονται ως καθολικές μεταβλητές, ενώ τα ιδιαίτερα δεδομένα κάθε καλούμενου αλγόριθμου μεταβιβάζονται [Κοίλιας 2003].

Τέλος, σε αρκετές ενότητες του διδακτικού πακέτου, παρουσιάζονται οι αλγοριθμικές "εντολές", οι οποίες όταν "εκτελεστούν" παρουσιάζουν τα αποτελέσματα στην οθόνη. Αυτό δημιουργεί σύγχυση μεταξύ αλγορίθμου και προγράμματος. Πρέπει να γίνει σαφές πως ο αλγόριθμος δεν είναι εκτελέσιμος από τον υπολογιστή, αλλά κατασκευάζεται στο "χαρτί", ελέγχεται και εκτελείται στο "χαρτί". Ο αλγόριθμος είναι ένα σκαρίφημα επίλυσης ενός προβλήματος, και περιγράφει τον τρόπο επίλυσης. Αντίθετα, το πρόγραμμα, συντάσσεται σε ένα προγραμματιστικό περιβάλλον, μεταφράζεται ή μεταγλωττίζεται, λαμβάνει τα δεδομένα εισόδου από κάποια μονάδα εισόδου (πληκτρολόγιο κτλ) ή από άλλο υποπρόγραμμα, εκτελείται από τον επεξεργαστή του υπολογιστή και τέλος αποδίδει τα αποτελέσματά του σε κάποια μονάδα εξόδου (οθόνη, εκτυπωτής κτλ) ή σε άλλο υποπρόγραμμα. Η εκσφαλμάτωση, διορθώνει τυχόν λάθη στον κώδικα του προγράμματος και έτσι διασφαλίζεται η αυστηρότητα σύνταξης της γλώσσας προγραμματισμού. Ο αλγόριθμος, δεν έχει συντακτικά λάθη. Για παράδειγμα, για τον αλγόριθμο της σειριακής αναζήτησης, οι παρακάτω δομές υποδηλώνουν την ίδια ενέργεια, είναι ισοδύναμες και πρέπει να είναι αποδεκτές.

Όσο (βρέθηκε = ψευδής) **και** ($i \leq N$) **επανάλαβε**

ή

Όσο (βρέθηκε = ψευδής) **και** ($i \leq N$) **κάνε**

ή

Επανάλαβε όσο (βρέθηκε = ψευδής) **και** ($i \leq N$)

Το γεγονός ότι υπάρχει ελευθερία στην καταγραφή του αλγορίθμου, διευκολύνει σχετικά μαθήματα (σχολικά ή πανεπιστημιακά) που διδάσκονται στην τάξη και εξετάζονται στο "χαρτί". Ο αλγόριθμος ούτε μεταγλωττίζεται, ούτε εκτελείται από υπολογιστή, αλλά είναι μία παρουσίαση της λογικής της επίλυσης του προβλήματος. Πρέπει, λοιπόν, οι διδάσκοντες, οι εξεταστές και οι διδασκόμενοι, να εκμεταλλευτούν αυτό το πλεονέκτημα, ώστε να ενισχύουν τη διδασκαλία της λογικής της επίλυσης και να μην αναλώνονται στη διδασκαλία της συντακτικής αυστηρότητας.

Με δεδομένο την υλικοτεχνική υποδομή των σχολικών και πανεπιστημιακών μονάδων, είναι αδύνατο να πραγματοποιηθεί εξέταση μαθημάτων αλγοριθμικής σχεδίασης σε υπολογιστή. Έτσι το πρόσφορο μέσο είναι η γραπτή εξέταση. Στο "χαρτί" δεν έχει ιδιαίτερη αξία η αξιολόγηση του μαθητή σε συντακτικούς κανόνες. Το αποτέλεσμα του αλγορίθμου δεν επηρεάζεται από τη χρήση μονού ή διπλού εισαγωγικού, τη χρήση του συγκριτικού τελεστή \neq ή του $\langle \rangle$, τη δήλωση των μεταβλητών κτλ.

Η διαδικασία αλγοριθμικής επίλυσης, διευκολύνει τον επίδοξο επίλυτη, αφού του επιτρέπει να καταγράψει τη λύση του προβλήματος σε μία φυσική γλώσσα. Ένας αλγόριθμος γραμμένος στα ελληνικά θα έχει το ίδιο αποτέλεσμα με τον ίδιο αλγόριθμο γραμμένο σε οποιαδήποτε φυσική γλώσσα, αφού και οι δύο αλγόριθμοι παρουσιάζουν την ίδια σειρά ενεργειών.

Φαίνεται, λοιπόν, ότι μια αλγοριθμική επίλυση είναι ανεξάρτητη από τη φυσική γλώσσα που έχει χρησιμοποιήσει ο λύτης, γεγονός που κάνει την κατανόησή της ευκολότερη σε σχέση με την κωδικοποίηση σε γλώσσα προγραμματισμού. Πολλοί διαφωνούν με τη χρήση αλγορίθμου για την επίλυση ενός προβλήματος, αφού δεν είναι δυνατή η υπολογιστική εκτέλεσή του. Ένας αλγόριθμος δεν εκτελείται, ενώ με το πρόγραμμα ο υπολογιστής "λαμβάνει" τον κώδικα του προγράμματος και τα δεδομένα, τα επεξεργάζεται και επιστρέφει τα αποτελέσματα. Δεν είναι όμως σκοπός η εξέταση της δυνατότητας του υπολογιστή να εκτελέσει εντολές. Σκοπός είναι ο μαθητής να έχει τη δυνατότητα να καταγράψει τα λογικά βήματα του αλγορίθμου και να ελέγχει την αποδοτικότητά του. Είναι πολύ σημαντικό ο διδασκόμενος να μπορεί να "εκτελεί" στο "χαρτί" τον αλγόριθμο για να γνωρίζει τι αποτελέσματα θα δώσει για διαφορετικά δεδομένα εισόδου. Τα λογικά λάθη δεν μπορούν να διαπιστωθούν από την εκτέλεση του

προγράμματος με τη βοήθεια του υπολογιστή, αλλά μόνο με συγκεκριμένα δεδομένα ελέγχου και "τρέξιμο" με το "χέρι", τα οποία και βοηθούν στη διαπίστωσή τους.

Ο αλγόριθμος παρέχει τη λογική δομή του προγράμματος. Για το λόγο αυτό, η μεταφορά του αλγόριθμου σε πρόγραμμα είναι μία απλή διαδικασία. Οι ενέργειες του αλγορίθμου μεταφράζονται άμεσα σε εντολές προγράμματος σε οποιαδήποτε γλώσσα προγραμματισμού. Ο αλγόριθμος περιγράφει τη λύση και η κωδικοποίηση της λύσης σε μία γλώσσα προγραμματισμού γίνεται με σκοπό να εκφράζουμε τις ενέργειες και να επικοινωνούμε με τη μηχανή.

Έτσι, η διδασκαλία ενός μαθήματος αλγοριθμικής σχεδίασης στη δευτεροβάθμια εκπαίδευση πρέπει να προεκπαιδεύει τους μαθητές στον προγραμματισμό και να τους βοηθάει να μεταβούν εύκολα σε ένα προγραμματιστικό περιβάλλον στο Πανεπιστήμιο. Θα ήταν εξειδίκευση η εκμάθηση μιας γλώσσας προγραμματισμού στη δευτεροβάθμια η οποία μπορεί και να μην διδάσκεται στο Πανεπιστήμιο.

Από την άλλη, η επιλογή της κατασκευής μιας ελληνικής γλώσσας προγραμματισμού με συντακτικούς κανόνες και ελληνικό λεξιλόγιο και η χρήση της για την εκπαίδευση, δεν θα αποφέρει ουσιαστικό κέρδος στους εκπαιδευόμενους. Αυτή η γλώσσα θα είναι μία γλώσσα για τη συγκεκριμένη ομάδα εκπαιδευόμενων και η χρήση της θα έχει τοπική εμβέλεια.

3.2. Προβλήματα κατά τη διδασκαλία του μαθήματος ΑΕΠΠ στο εργαστήριο

Το μάθημα ΑΕΠΠ είναι το βασικότερο μάθημα Πληροφορικής που διδάσκεται στη δευτεροβάθμια εκπαίδευση και είναι αναγκαίο -ως μάθημα πληροφορικής- να εκμεταλλεύεται όσο το δυνατόν καλύτερα τις σύγχρονες τεχνολογίες. Το διδακτικό πακέτο παρουσιάζει την επίλυση των προβλημάτων χρησιμοποιώντας ψευδογλώσσα και διάγραμμα ροής και στη συνέχεια η ψευδογλώσσα κωδικοποιείται με τη χρήση της ΓΛΩΣΣΑΣ. Η ΓΛΩΣΣΑ δεν θα πρέπει να αποτελεί μέρος της εξεταστέας ύλης, αλλά τη γλώσσα με την οποία θα γίνεται η επικοινωνία του μαθητή με το εκπαιδευτικό λογισμικό του μαθήματος. Αυτό το εκπαιδευτικό λογισμικό πρέπει να περιλαμβάνει εκείνες τις προδιαγραφές που θα το κατατάξουν στα διερευνητικά λογισμικά. Το λογισμικό πρέπει να προσφέρει τη δυνατότητα δυναμικού χειρισμού εργαλείων οπτικοποίησης και διαχείρισης δεδομένων και πρέπει να προσφέρει συνδυαστική χρήση πολλαπλών αναπαραστάσεων, με δυνατότητα αμφίδρομου χειρισμού τους. Αυτό σημαίνει ότι το λογισμικό πρέπει να επιτρέπει στο μαθητή, είτε να αναπτύσσει με τη χρήση της ψευδογλώσσας, είτε να γράφει σε πρόγραμμα με τη ΓΛΩΣΣΑ ή κάποια γλώσσα προγραμματισμού, είτε να δημιουργεί ένα διάγραμμα ροής. Στη συνέχεια, πρέπει το υλοποιημένο πρόγραμμα με τον τρόπο που επέλεξε ο μαθητής να μετατρέπεται στις άλλες δύο μορφές αναπαράστασης αλγορίθμου. Το λογισμικό πρέπει να υποστηρίζει τη συνεργατική μάθηση. Το λογισμικό πρέπει να προσφέρει εργαλεία διατύπωσης δομημένων ερωτήσεων, πιθανώς με γραφικό τρόπο. Το λογισμικό πρέπει να συνοδεύεται από προσεκτικά σχεδιασμένα "σενάρια" χρήσης και οδηγίες παιδαγωγικής αξιοποίησής του. Τέλος, το λογισμικό πρέπει να προσφέρει τη δυνατότητα δοκιμής υποθέσεων και σεναρίων του τύπου "τι θα γινόταν εάν", με στόχο την επαλήθευσή τους, την καταγραφή της αλληλουχίας των δοκιμών και την εξαγωγή συμπερασμάτων με τη μορφή κανόνων [Κυνηγός].

4. Προτάσεις

Η διδασκαλία του μαθήματος στην τάξη και η συμμετοχή μας στη διόρθωση των γραπτών των πανελληνίων μας οδήγησαν στις ακόλουθες παρατηρήσεις:

1. Το διδακτικό πακέτο αρχικά παρουσιάζει την επίλυση των προβλημάτων χρησιμοποιώντας ψευδογλώσσα (κεφάλαια 2 έως 5) και στη συνέχεια η ψευδογλώσσα κωδικοποιείται με τη χρήση της ΓΛΩΣΣΑΣ (κεφάλαια 7 έως 14). Έτσι, η παρουσίαση των αλγοριθμικών δομών και των δομών δεδομένων με ψευδογλώσσα και ΓΛΩΣΣΑ, είναι αρωγός στο έργο του εκπαιδευτικού, αφού μπορεί να διδάξει στην τάξη αλγόριθμο και να παρουσιάσει στο εργαστήριο το πρόγραμμα καλύπτοντας με τον τρόπο αυτό τους στόχους του μαθήματος.

Η πραγματικότητα όμως είναι διαφορετική. Το πρόγραμμα σπουδών περιορίζει το μάθημα σε δύο διδακτικές ώρες, με αποτέλεσμα οι εκπαιδευτικοί να μην έχουν τον απαραίτητο χρόνο να παρουσιάσουν την ύλη του μαθήματος με πληρότητα. Ακόμα και στην περίπτωση που υπήρχε μια επιπλέον διδακτική ώρα, όπως είχε προταθεί από το Παιδαγωγικό Ινστιτούτο, η παρουσίαση και χρήση της ΓΛΩΣΣΑΣ στο εργαστήριο δεν θα ήταν εφικτή, αφού δεν παρέχεται κατάλληλο λογισμικό ως εργαλείο προγραμματισμού. Θα μπορούσε βέβαια να χρησιμοποιηθεί το προγραμματιστικό περιβάλλον της Basic ή της Pascal για την εξοικείωση των μαθητών στον προγραμματισμό. Η επιλογή αυτή επιβαρύνει τους καθηγητές με τη διδασκαλία μιας τεχνητής γλώσσας και τους μαθητές με την εκμάθηση της γλώσσας αυτής. Επιπλέον, μια τέτοια επιλογή θα ήταν πέρα από τους στόχους του μαθήματος, αφού σύμφωνα με την εξεταστέα ύλη των δύο τελευταίων χρόνων, οι μαθητές μπορούν να αναπτύξουν τα θέματα των εξετάσεων είτε σε οποιαδήποτε μορφή παράστασης αλγορίθμου, είτε σε ΓΛΩΣΣΑ, αποκλείοντας έτσι τη δυνατότητα, που οι μαθητές είχαν τις προηγούμενες χρονιές, να αναπτύξουν τα θέματα σε Basic ή Pascal.

Με δεδομένο ότι οι ώρες διδασκαλίας δεν είναι αρκετές και ότι οι μαθητές μπορούν να επιλέξουν οποιαδήποτε μορφή αναπαράστασης αλγορίθμου για την ανάπτυξη των θεμάτων, γίνεται προσπάθεια κάλυψης της ύλης δίνοντας μεγαλύτερη βαρύτητα σε μια από τις δύο μορφές κωδικοποίησης (ψευδογλώσσα ή ΓΛΩΣΣΑ). Με την παρούσα κατάσταση, η ΓΛΩΣΣΑ θα πρέπει να διδαχθεί στον "πίνακα" και να εξετασθεί στο "χαρτί", περιορισμός που δεν θυμίζει καθόλου προγραμματισμό. Η επιλογή να δοθεί βαρύτητα στην ψευδογλώσσα καλύπτει σε μεγάλο βαθμό τους στόχους του μαθήματος και χρήζει των πλεονεκτημάτων αλγοριθμικού σχεδιασμού που προαναφέραμε σε σχέση με τον προγραμματισμό στη ΓΛΩΣΣΑ. Άρα, οι μαθητές πρέπει πρώτα να εκπαιδεύονται στην ανάλυση προβλημάτων και στην αλγοριθμική επίλυσή τους και ύστερα στη μετάβαση από τον αλγόριθμο σε πρόγραμμα με τη χρήση μιας γλώσσας προγραμματισμού.

2. Με τη σπειροειδή προσέγγιση που προτείνεται από τους συγγραφείς του διδακτικού πακέτου, οι έννοιες να μην προσεγγίζονται επαναληπτικά, αλλά οι διαφορετικές παρουσιάσεις του ίδιου θέματος, οδηγούν σε σύγχυση μαθητές και συναδέλφους. Έτσι, παρουσιάζονται προβλήματα κατά τη διδασκαλία, τη διόρθωση και την αξιολόγηση των γραπτών των πανελληνίων. Προτείνουμε, οι μαθητές να εκπαιδεύονται και να εξετάζονται στην αλγοριθμική επίλυση προβλημάτων με τη χρήση ψευδογλώσσας ή διαγραμματικών τεχνικών και η ΓΛΩΣΣΑ να αποτελεί τη γλώσσα που θα χρησιμοποιηθεί στο εργαστήριο. Οι βαθμολογητές όμως, θα πρέπει να αξιολογούν τις αλγοριθμικές λύσεις των θεμάτων δίνοντας ιδιαίτερη έμφαση στον τρόπο επίλυσης, επιτρέποντας πολλαπλές αλγοριθμικές εκφράσεις για την ίδια αλγοριθμική εντολή.
3. Θεωρούμε ότι πρέπει να υπάρξει ένας σαφής διαχωρισμός μεταξύ ψευδογλώσσας και ΓΛΩΣΣΑΣ για την ορθότητα της εξέτασης του μαθήματος. Στις εξετάσεις, ο μαθητής πρέπει να έχει τη δυνατότητα να επιλέξει οποιαδήποτε από τους τέσσερις τρόπους αναπαράστασης αλγορίθμων, δηλαδή Ελεύθερο κείμενο, Φυσική γλώσσα με βήματα, Διαγραμματικές τεχνικές ή Κωδικοποίηση (ψευδογλώσσα ή πρόγραμμα στη ΓΛΩΣΣΑ) χωρίς να υποχρεώνεται να χρησιμοποιήσει μία από αυτές. Για να επιτρέπεται στο μαθητή να αναπτύξει τον αλγόριθμο σε όποια μορφή επιθυμεί, η ιδανική εκφώνηση στις εξετάσεις πρέπει να έχει την παρακάτω μορφή: "Να αναπτύξετε αλγόριθμο ο οποίος...". Είναι χρέος μας όμως, να τονίζουμε στους μαθητές ότι ο καλύτερος τρόπος αναπαράστασης είναι η ψευδογλώσσα. Αυτή η δυνατότητα επιλογής αφορά μόνο την ανάπτυξη αλγορίθμων και όχι τη θεωρητική εξέταση του προγραμματισμού ή του "τρεξίματος" ενός προγράμματος. Για παράδειγμα οι μαθητές πρέπει να είναι σε θέση να απαντήσουν ερωτήσεις σχετικά με τον τύπο των μεταβλητών ή με τα αποτελέσματα της εκτέλεσης ενός προγράμματος.

5. Το μέλλον του μαθήματος

Μέσα από αυτή την εργασία αναδείχτηκε η σημασία της αλγοριθμικής επίλυσης προβλημάτων και η αναγκαιότητα του μαθήματος της Ανάπτυξης Εφαρμογών σε Προγραμματιστικό Περιβάλλον. Αν και από τα παραπάνω εντοπίστηκαν ορισμένα προβλήματα που διέπουν το μάθημα, η χρησιμότητά του και η ανταπόκριση των μαθητών είναι οι καλύτεροι σύμμαχοί μας για να αγωνιστούμε για τη βελτίωσή του.

Μέσα από τις προτάσεις μας φαίνεται ξεκάθαρα η αναγκαιότητα αύξησης των ωρών διδασκαλίας και ύπαρξης κατάλληλου διερευνητικού λογισμικού το οποίο να έχει πρόσθετη αξία. Με τον τρόπο αυτό, το μάθημα θα γίνει πραγματικά εργαστηριακό, αφού ο διδάσκων, θα ενισχύσει τη διδασκαλία των αλγορίθμων με εργαστηριακά παραδείγματα.

Η χρήση του διδακτικού πακέτου τα τέσσερα αυτά χρόνια, ανέδειξε ελλείψεις και παραβλέψεις. Πολλοί συνάδελφοι σε προσωπικούς δικτυακούς τόπους ή μέσω του σχολικού δικτύου, καθώς και η συγγραφική ομάδα έχουν δημοσιεύσει παροράματα. Φέτος, το Παιδαγωγικό Ινστιτούτο προχώρησε σε κάποιες διορθώσεις του διδακτικού πακέτου, όμως η συνεχής ανανέωσή του πρέπει να γίνει στόχος όλων μας.

Τέλος, θεωρούμε ότι η συνεχής βελτίωση και εξέλιξη του μαθήματος μπορεί και πρέπει να οδηγήσει την ΕΠΥ καθώς και σχολές - τμήματα Πληροφορικής Πανεπιστημίων - ΑΤΕΙ να απαιτήσουν να γίνει το μάθημα της Ανάπτυξης Εφαρμογών σε Προγραμματιστικό Περιβάλλον, μάθημα αυξημένης βαρύτητας για την εισαγωγή σε σχολές και τμήματα Πληροφορικής.

6. Βιβλιογραφία

1. Βακάλη Α., Γιαννόπουλος Η., Ιωαννίδης Ν., Κοΐλιας Χ., Μάλαμας Κ., Μανωλόπουλος Ι., Πολίτης Π. (1999), Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων.
2. Δουκάκης Σ. (2002), Παρατηρήσεις για τις Εξετάσεις στο Μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, <http://users.ntua.gr/sdoukakis>.
3. Δουκάκης Σ., Κοΐλιας Χ., Ψαλτίδου Α. (2003), Η σημασία του αλγορίθμου και τα πλεονεκτήματα της αλγοριθμικής επίλυσης στο μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (υπό έκδοση, 2ο Πανελλήνιο Συνέδριο των Εκπαιδευτικών στις ΤΠΕ "Αξιοποίηση των Τεχνολογιών της Πληροφορίας & Επικοινωνίας στη Διδακτική πράξη").
4. Δουκάκης Σ., Ψαλτίδου Α. (2002), Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, Εκδόσεις Νέων Τεχνολογιών.
5. Κοΐλιας Χ., (2003), Αναπαράσταση αλγορίθμων με ψευδογλώσσα (υπό έκδοση, 2ο Πανελλήνιο Συνέδριο των Εκπαιδευτικών στις ΤΠΕ "Αξιοποίηση των Τεχνολογιών της Πληροφορίας & Επικοινωνίας στη Διδακτική πράξη").
6. Κυνηγός Χ., Κουτλής Μ., (), Λογισμικό υπό ... συνθήκες
7. Τζιμογιάννης Α., Κόμης Β. (2000), Η έννοια της μεταβλητής στο προγραμματισμό: Δυσκολίες και παρανοήσεις μαθητών του Ενιαίου Λυκείου, 2^ο Πανελλήνιο Συνέδριο: Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση, Εκδόσεις Νέων Τεχνολογιών.
8. Beekman, G. (1997), Computer Confluence, Addison - Wesley.
9. Papert S. (1980), Νοητικές Θύελλες: Παιδιά, ηλεκτρονικοί υπολογιστές και δυναμικές ιδέες, Εκδόσεις Οδυσσέας.
10. Siegler R., (1998), Πώς σκέφτονται τα παιδιά, Ελληνική Μετάφραση (2002), Εκδόσεις Gutenberg.