

Μια Εναλλακτική Προσέγγιση Διδασκαλίας του Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση

Μαρία Σατρατζέμη

Επίκουρος Καθηγήτρια

Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας

Εγνατία 156, Τ.Θ. 1591, 54006, Θεσσαλονίκη, Ελλάδα

maya@uom.gr

Βασίλειος Δαγδιλέλης

Επίκουρος Καθηγητής

Τμήμα Εκπαιδευτικής & Κοινωνικής Πολιτικής, Πανεπιστήμιο Μακεδονίας

Εγνατία 156, Τ.Θ. 1591, 54006, Θεσσαλονίκη, Ελλάδα

dagdil@uom.gr

Γεώργιος Ευαγγελίδης

Επίκουρος Καθηγητής

Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας

Εγνατία 156, Τ.Θ. 1591, 54006, Θεσσαλονίκη, Ελλάδα

gevan@uom.gr

ΠΕΡΙΛΗΨΗ

Η εισαγωγή του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση χαρακτηρίστηκε από μια ασαφή πολιτική από τη μεριά της Πολιτείας. Η ασάφεια αυτή εντοπίζεται σε πολλαπλά επίπεδα: τόσο στο περιεχόμενο όσο και τους στόχους της διδασκαλίας του σχετικού γνωστικού αντικειμένου.

Τα τελευταία κυρίως χρόνια παρατηρείται ένας σχετικός περιορισμός της διδασκαλίας του προγραμματισμού θεωρουμένου ως ειδικού αντικειμένου. Υιοθετώντας την αντίθετη υπόθεση, ότι δηλαδή ο προγραμματισμός είναι ένα γενικού ενδιαφέροντος διανοητικό εργαλείο, παραθέτουμε ένα συγκεκριμένο παράδειγμα διδασκαλίας του προγραμματισμού, σε αρχάριους μαθητές της Δευτεροβάθμιας Εκπαίδευσης.

Η διδασκαλία αυτή στηρίζεται ακριβώς όχι στις τεχνικές ιδιαιτερότητες του προγραμματιστικού αυτού μοντέλου, αλλά στη γενικότερη παιδευτική και διανοητική του αξία.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: *Εκπαιδευτικά προγραμματιστικά περιβάλλοντα, Μικρόκοσμοι προγραμματισμού, δυσκολίες εκμάθησης προγραμματισμού*

ΕΙΣΑΓΩΓΗ

Η επίσημη διδασκαλία του προγραμματισμού στη δευτεροβάθμια Εκπαίδευση στην Ελλάδα άρχισε στα μέσα της δεκαετίας του '80. Παρά τις επανειλημμένες απόπειρες για την αναμόρφωση του προγράμματος σπουδών που σχετιζόταν με τον προγραμματισμό, το αντίστοιχο μάθημα

εξακολουθεί, ακόμη και σήμερα, να έχει ένα επαμφοτερίζον status. Η επίσημη εκπαιδευτική πολιτική στο θέμα της διδασκαλίας του προγραμματισμού, υπήρξε αρκετά ασαφής. Η τρέχουσα άποψη χαρακτηρίζεται από τη διπλή επιλογή μιας εισαγωγής στο Γυμνάσιο και μιας αυστηρότερης διδασκαλίας στην Τεχνολογική κατεύθυνση και στα ΤΕΕ. Η ουσία των επιλογών αυτών θα μπορούσε να συνοψιστεί στο ότι ο προγραμματισμός αντιμετωπίζεται ως ένα «ειδικού ενδιαφέροντος» μάθημα το οποίο ακριβώς δεν παρουσιάζει γενικότερο ενδιαφέρον.

Ωστόσο ο προγραμματισμός μπορεί να αποτελέσει ένα διανοητικό εργαλείο μεγάλης σημασίας. Ο προγραμματισμός είναι βέβαια μια εγγενώς πολύπλοκη διανοητική δραστηριότητα (Dagdilelis, 1986), (Dagdilelis, 1989). Σε όλα τα επίπεδα διδασκαλίας, είτε στο Γυμνάσιο, είτε στο Λύκειο, η διδασκαλία του προγραμματισμού συνοδεύεται από τα κλασσικά φαινόμενα που χαρακτηρίζουν τα «δύσκολα» μαθήματα: οι μαθητές δεν κατανοούν ακόμη και βασικές έννοιες του προγραμματισμού και δυσκολεύονται να επιλύσουν και τα πιο απλά προβλήματα.

Ωστόσο, και άλλοι επιστημονικοί κλάδοι παρουσιάζουν αυτή την εγγενή πολυπλοκότητα. Η αξία λοιπόν του προγραμματισμού δε συνδέεται άμεσα με τη «δυσκολία» του – αλλά με την ενδεχόμενη συνεισφορά του στο διανοητικό οπλοστάσιο των μαθητών. Το σημαντικό στοιχείο στην περίπτωση του προγραμματισμού είναι ότι ένα μεγάλο τμήμα των δυσκολιών του προέρχεται και από το «μέσο» - από τα περιβάλλοντα δηλαδή τα οποία χρησιμοποιούνται για τη διδασκαλία του. Πιο συγκεκριμένα, θεωρούμε ότι τα προγραμματιστικά περιβάλλοντα αποτελούν εξαιρετικής πολυπλοκότητας και ακριβείας εργαλεία και επομένως όλα τα προγραμματιστικά εργαλεία δεν είναι κατάλληλα για όλες τις περιπτώσεις. Θεωρούμε, μαζί με πολλούς ερευνητές της διεθνούς κοινότητας, ότι τουλάχιστον σε ορισμένες περιπτώσεις, η επιλογή του κατάλληλου προγραμματιστικού περιβάλλοντος είναι ουσιαστικός παράγοντας για μια αποτελεσματική εισαγωγή στον προγραμματισμό.

Στην εργασία αυτή λοιπόν, προτείνεται μια εναλλακτική προσέγγιση διδασκαλίας του προγραμματισμού, και στο Γυμνάσιο αλλά και στο Λύκειο, η οποία στηρίζεται σε ένα ειδικό περιβάλλον, γνωστό με την ονομασία Karel the Robot the Robot (Pattis & Roberts & Stehlik, 1995) και στην αντίστοιχη γλώσσα προγραμματισμού. Στην εργασία παρουσιάζονται συνοπτικά μερικές βασικές δυσκολίες που συναντούν όσοι προσεγγίζουν τις έννοιες του προγραμματισμού. Στη συνέχεια παρουσιάζεται η γενική ιδέα του Karel the Robot, αλλά και το αντίστοιχο προγραμματιστικό περιβάλλον. Στο τέλος προτείνονται σχέδια μαθημάτων τα οποία θα μπορούσαν να υιοθετηθούν από τους καθηγητές Πληροφορικής για τη διδασκαλία του Προγραμματισμού. Το τελευταίο μέρος της εργασίας περιλαμβάνει ορισμένα συμπεράσματα.

ΔΥΣΚΟΛΙΕΣ ΣΤΗ ΔΙΔΑΣΚΑΛΙΑ ΚΑΙ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η προσπάθεια εισαγωγής μαθημάτων Προγραμματισμού ΗΥ στα Πολυκλαδικά Λύκεια ξεκίνησε το 1985. Το 1997 το Παιδαγωγικό Ινστιτούτο (Π.Ι.) δημοσίευσε το Ενιαίο Πλαίσιο Σπουδών Πληροφορικής για το Γυμνάσιο και το Λύκειο (διαθέσιμο στη διεύθυνση <http://www.pi-schools.gr/greek/epps/but1.htm>) καθορίζοντας το γενικό σκοπό και τους άξονες περιεχομένου των μαθημάτων της Πληροφορικής. Ο τέταρτος άξονας περιεχομένου για το Γυμνάσιο είχε τον τίτλο «Ελέγχω - Προγραμματίζω τον Υπολογιστή» και καθόριζε ότι θα πρέπει να διδαχθούν οι μαθητές της Γ' Γυμνασίου σε 10 διδακτικές ώρες τα εξής:

- την έννοια του αλγόριθμου
- τον κύκλο ζωής ενός προγράμματος
- το περιβάλλον μιας γλώσσας προγραμματισμού
- τις βασικές δομές μιας συμβολικής γλώσσας προγραμματισμού.

Το αντίστοιχο βιβλίο που εκδόθηκε από το Π.Ι. και κάλυπτε την ύλη για όλο το Γυμνάσιο, μέσα σε 25 περίπου σελίδες περιελάμβανε την ύλη που αναφέρεται στο περιβάλλον μιας γλώσσας προγραμματισμού και στις βασικές δομές μιας συμβολικής γλώσσας προγραμματισμού.

Τέλος η συνήθης πρακτική που ακολουθήθηκε από τους καθηγητές Πληροφορικής για τη διδασκαλία του προγραμματισμού, ήταν η χρήση μιας γλώσσας γενικού σκοπού, όπως η Basic ή η Pascal και του αντίστοιχου προγραμματιστικού περιβάλλοντος, συνήθως ενός επαγγελματικού περιβάλλοντος. Σε ορισμένες περιπτώσεις χρησιμοποιήθηκε η γλώσσα Logo. Η Logo αν και αποτελεί μια γλώσσα με σαφή εκπαιδευτικό προσανατολισμό, ωστόσο δεν είναι πάντοτε η πλέον ενδεδειγμένη λύση για τη διδασκαλία του προγραμματισμού γιατί, όπως είναι γνωστό, η Logo αντιστοιχεί σε ένα πολύ συγκεκριμένο υπολογιστικό παράδειγμα, αυτό του συναρτησιακού προγραμματισμού. Έτσι, κυρίως σε σχολές της τριτοβάθμιας ή της δευτεροβάθμιας εκπαίδευσης που έχουν έναν προσανατολισμό περισσότερο εφαρμοσμένο παρά θεωρητικό, προτιμώνται τα περιβάλλοντα που συνδυάζουν την απλότητα της Logo, αλλά σχετίζονται με τον *επιτακτικό προγραμματισμό*.

Παρόλες τις σχετικές διαφοροποιήσεις, η διδασκαλία του προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση, παρουσίασε και παρουσιάζει μερικά κοινά χαρακτηριστικά. Κατά κανόνα, τα προβλήματα που προτείνονται είναι μικρού μεγέθους, αναφέρονται στην επεξεργασία αριθμών και συμβόλων και συνήθως επιλύονται με τη χρήση εντολών εισόδου/ εξόδου, εντολών απόδοσης τιμής και πιθανόν με εντολές ελέγχου. Έτσι η προσοχή των μαθητών επικεντρώνεται στην εκμάθηση της σύνταξης της γλώσσας και όχι στην ανάπτυξη ικανοτήτων για την επίλυση προβλημάτων με τη βοήθεια των αλγορίθμων και των προγραμμάτων.

Μια τέτοια προσέγγιση για τη διδασκαλία του προγραμματισμού αποδείχθηκε αναποτελεσματική, όπως αυτή τεκμηριώνεται εκτενώς στη σχετική αρθρογραφία. Πολλές εργασίες και μελέτες (Du Boulay, 1989), (Du Boulay & O'Shea & Monk, 1989), (Brusilovsky P. et al, 1997), (Dagdilelis, 1986), (Δαγδιλίλης, 1996) καταγράφουν τα προβλήματα που δημιουργήθηκαν από την κλασσική προσέγγιση διδασκαλίας των αλγορίθμων και του προγραμματισμού και τα οποία συνοψίζονται ως εξής:

- δυσκολίες που οφείλονται στην κατανόηση των γενικών ιδιοτήτων της «νοητής μηχανής» που μαθαίνει να ελέγχει ο μαθητής και της σχέσης της με τη φυσική μηχανή.
- η σύνταξη και η σημασιολογία ως επέκταση των ιδιοτήτων και της συμπεριφοράς της νοητής μηχανής.
- η απόκτηση της ικανότητας καθορισμού, ανάπτυξης, ελέγχου και εκσφαλμάτωσης ενός προγράμματος με τα διαθέσιμα εργαλεία.
- η εκτέλεση ενός προγράμματος είναι ένα είδος μηχανισμού και ο μαθητής χρειάζεται πολύ χρόνο για να εντοπίσει τη σχέση ανάμεσα στο πρόγραμμα και στο μηχανισμό που περιγράφει.
- ο υπολογιστής μαζί με τον προγραμματιστικό περιβάλλον μπορεί να θεωρηθεί ως ένας μηχανισμός, που χρησιμοποιείται στη συνέχεια για τη δημιουργία άλλων μηχανισμών, δηλαδή των προγραμμάτων. Η περιγραφή του υπολογιστικού συστήματος ως μιας μηχανής θα πρέπει να πραγματοποιηθεί στο κατάλληλο επίπεδο, έτσι ώστε ο μαθητής να μπορεί να αντιλαμβάνεται τι συμβαίνει στο εσωτερικό του ΗΥ, γιατί διαφορετικά μπορεί να αναπτύξει δική του θεωρία που τις περισσότερες φορές είναι ανεπαρκής και ίσως λανθασμένη.
- οι μαθητές δυσκολεύονται να κατανοήσουν τι παρουσιάζεται κάθε φορά στην οθόνη. Στην οθόνη του υπολογιστή μπορεί να εμφανίζονται πληροφορίες που αφορούν σε προηγούμενη αλληλεπίδραση του χρήστη – υπολογιστή, ή εμφάνιση δεδομένων εισόδου που πληκτρολόγησε ο χρήστης ή εμφάνιση αποτελεσμάτων συγκεκριμένων εντολών.
- η διαχείριση της νοητής μηχανής, όπως η ανάκληση, αποθήκευση και έκδοση προγραμμάτων.
- πολλές σημαντικές αλλαγές στην εσωτερική κατάσταση της μηχανής δεν παρουσιάζονται κατά την εκτέλεση ενός προγράμματος.

- το πρόγραμμα γίνεται κατανοητό από το σύστημα βάσει πολύ αυστηρών κανόνων πράγμα που προκαλεί δυσκολίες στους μαθητές μια και προσδίδουν στο σύστημα ανθρωπομορφικά χαρακτηριστικά.
- η χρήση αγγλικών λέξεων στις γλώσσες προγραμματισμού δημιουργεί προβλήματα, αφού:
 - ο προσδίδει στο σύστημα μεγαλύτερη νοημοσύνη
 - ο στη γλώσσα προγραμματισμού έχουν διαφορετικό νόημα απ' ότι στην Αγγλική γλώσσα.
- τα επαγγελματικά προγραμματιστικά περιβάλλοντα παρέχουν μικρή στήριξη κατά την εκσφαλμάτωση, αφού συνήθως τα μηνύματα λάθους δεν είναι και τόσο φιλικά προς το χρήστη.
- οι γλώσσες προγραμματισμού γενικού σκοπού διαθέτουν κατά κανόνα ένα μεγάλο ρεπερτόριο εντολών και είναι πολύπλοκες.
- η επίλυση ενδιαφερόντων προβλημάτων απαιτεί την εκμάθηση ενός μεγάλου υποσυνόλου της γλώσσας και την ανάπτυξη αρκετά μεγάλων προγραμμάτων, απαιτεί δηλαδή τη επικέντρωση της προσοχής στην εκμάθηση της γλώσσας.
- η προσοχή των μαθητών επικεντρώνεται στην εκμάθηση της σύνταξης της γλώσσας και όχι στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων.
- δεν υπάρχει, κατά κανόνα, επαρκής στήριξη του σπουδαστή στην κατανόηση των βασικών ενεργειών και δομών ελέγχου, αφού το περιβάλλον προγραμματισμού συνήθως δεν παρέχει δυνατότητες οπτικοποίησης (η διαδικασία εκτέλεσης ενός προγράμματος δεν είναι ορατή από τον μαθητή).
- η διανοητική πολυπλοκότητα που απαιτεί η εκφορά ενός αλγορίθμου σε μια γλώσσα προγραμματισμού είναι μεγάλη, λόγω της «φύσης» της γλώσσας.

Είναι βέβαια φυσικό να διατυπώσει κανείς την άποψη ότι, αν οι υποθέσεις αυτές αληθεύουν, τότε η σχεδίαση νέων προγραμματιστικών περιβαλλόντων με στόχους διδακτικούς θα έπρεπε να τις λάβει υπόψη της.

KAREL THE ROBOT

Τα παραπάνω ευρήματα, σχετικά με τις δυσκολίες που αντιμετωπίζουν οι αρχάριοι κατά την εκμάθηση του προγραμματισμού, οδήγησε στην επινόηση μικρο-γλωσσών προγραμματισμού και προγραμματιστικών περιβαλλόντων προκειμένου να ξεπεραστούν οι δυσκολίες που εντοπίστηκαν. Ανάμεσα σ' αυτές τις προσπάθειες συγκαταλέγεται και η ιδέα του *Karel the Robot* σχεδιάστηκε από τον Richard Pattis (Pattis & Roberts & Stehlik, 1995), προκειμένου να χρησιμοποιηθεί για το μάθημα της εισαγωγής στον προγραμματισμό. Ο Karel είναι ένα ρομπότ που εκτελεί διάφορες αποστολές (προγράμματα) σε ένα κόσμο που αποτελείται από οριζόντιους δρόμους και κατακόρυφες λεωφόρους. Στον κόσμο του Karel μπορούν να τοποθετηθούν τμήματα τοίχου μεταξύ των διασταυρώσεων δημιουργώντας εμπόδια (π.χ. λαβύρινθους), που καλείται να ξεπεράσει ο Karel, και beepers, μικροί πλαστικοί κώνοι, που παράγουν ένα ήχο (μπιπ). Ο Karel έχει τη δυνατότητα να κινείται προς την τρέχουσα κατεύθυνση κατά 1 μπλοκ, να στρίβει αριστερά κατά 90 μοίρες, να εντοπίζει beepers που βρίσκονται στην ίδια διασταύρωση μ' αυτόν, να εντοπίζει τοίχους που βρίσκονται μπροστά του σε απόσταση μισού μπλοκ χρησιμοποιώντας μια κάμερα, να σηκώνει και να κατεβάζει beepers με το μηχανικό του χέρι και τέλος μπορεί να καθορίζει προς ποια κατεύθυνση βλέπει χρησιμοποιώντας την πυξίδα του. Ο Karel μπορεί και ανταποκρίνεται στις εξής 5 βασικές εντολές: move, turnleft, pickbeeper, putbeeper τις οποίες χρησιμοποιούμε για να «διατάξουμε» τον Karel να κινηθεί μέσα στον κόσμο και να πραγματοποιήσει αποστολές. Ακόμη μπορούμε να εφοδιάσουμε τον Karel με ένα λεξικό νέων εντολών που περιλαμβάνει το όνομά τους καθώς και τον ορισμό τους. Δηλαδή μπορούμε να επεκτείνουμε το λεξιλόγιο του Karel με νέες εντολές οι οποίες υλοποιούνται με τη βοήθεια των 5

βασικών εντολών που αντιλαμβάνεται ο Karel. Επιπλέον οι εντολές if/then και if/then/else επιτρέπουν στον Karel να ελέγχει τον κόσμο του και στη συνέχεια να αποφασίζει ποια εντολή θα εκτελέσει στη συνέχεια. Τέλος οι εντολές iterate και while επιτρέπουν στο ρομπότ να επαναλαμβάνει μια εντολή για ένα συγκεκριμένο αριθμό επαναλήψεων.

Για την υλοποίηση της γλώσσας Karel the Robot έχουν κατασκευασθεί διάφορα προγραμματιστικά περιβάλλοντα που προσομοιώνουν την λειτουργία του ρομπότ. Τα πιο πολλά απ' αυτά είναι διαθέσιμα ελεύθερα στο Διαδίκτυο, πράγμα που διευκολύνει τους καθηγητές Πληροφορικής για την ελεύθερη απόκτηση και διάθεση του προγραμματιστικού περιβάλλοντος, αφού δεν απαιτούνται έξοδα για την απόκτηση και διάθεσή τους. Το λογισμικό MacGnome Genie αναπτύχθηκε στο Carnegie Mellon University και είναι ένα από τα πλέον επιτυχημένα εκπαιδευτικά περιβάλλοντα προγραμματισμού, το οποίο προσομοιώνει τον κόσμο του Karel. Είναι υλοποιημένο σε πλατφόρμα Mac και μπορεί κανείς να το αγοράσει μαζί με το βιβλίο Karel the Robot (Pattis 1995).

Το λογισμικό που εμείς προτείνουμε για χρήση (το οποίο λειτουργεί στην περισσότερη διαδεδομένη πλατφόρμα PC, Intel-HY) είναι διαθέσιμο στη διεύθυνση:

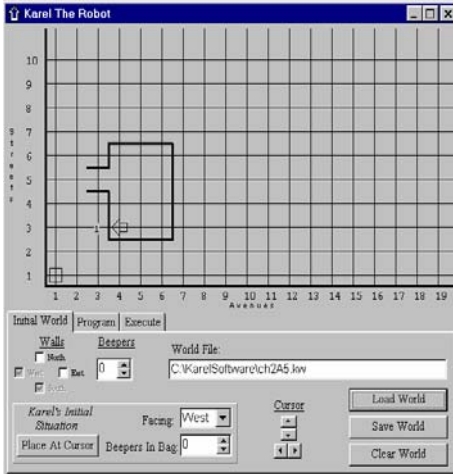
<http://ocweb.otterbein.edu/csc/cs115/web/Karel/KarelsWorld.htm>.

Θα πρέπει να σημειωθεί ότι και στον ελλαδικό χώρο έχουν πραγματοποιηθεί μια σειρά από πιλοτικές κατασκευές του Karel όπως και δοκιμαστικές διδασκαλίες του προγραμματισμού με το περιβάλλον αυτό. Τα τελευταία χρόνια στο Πανεπιστήμιο Μακεδονίας, για παράδειγμα, έχει αναπτυχθεί έντονη δραστηριότητα γύρω από την ανάπτυξη εκπαιδευτικών περιβαλλόντων και εκπαιδευτικό υλικού για τη διδασκαλία του προγραμματισμού. Στα πλαίσια των ερευνητικών αυτών προγραμμάτων έχουν υλοποιηθεί τόσο προγραμματιστικά περιβάλλοντα, όσο και σχετικό διδακτικό υλικό. Στη φάση της ολοκλήρωσης βρίσκεται λογισμικό που προσομοιώνει την γλώσσα Karel++ (Bergin, Stehlik, Roberts, & Pattis, 1996), με την οποία επιχειρείται η διδασκαλία του αντικειμενοστραφούς προγραμματισμού επεκτείνοντας τη γλώσσα Karel. Επίσης, στα πλαίσια πτυχιακών εργασιών φοιτητών έχει αναπτυχθεί ένας δικτυακός τόπος που βρίσκεται στη διεύθυνση www.it.uom.gr/Karel2001, στον οποίον παρουσιάζεται το λογισμικό MacGnome Genie (Miller & Pane & Meter & Vorhmann, 1994), προτείνονται λύσεις σε ασκήσεις που παρουσιάζονται στο βιβλίο του Pattis όπως και χρήσιμοι σύνδεσμοι που οδηγούν σε άλλους δικτυακούς τόπους όπου μπορεί κανείς να βρει διδακτικό υλικό βασισμένο στη γλώσσα Karel. Εξάλλου σε παλαιότερη πτυχιακή εργασία είχε αναπτυχθεί και ένα περιβάλλον Karel (Σατρατζέμη, Δαγδιλέλης, Μαργαρίτης, Μιχαήλ & Παπανικολάου, 1999).

Από τα υφιστάμενα περιβάλλοντα, αυτό που προτείνουμε παραπάνω αποτελείται από 3 βασικά παράθυρα. Το παράθυρο του Σχήματος 1 παρουσιάζει τον συντάκτη του κόσμου του Karel, όπου ο μαθητής με τη βοήθεια του ποντικιού και δίνοντας τιμές στα διάφορα πλαίσια κειμένου, καθορίζει την αρχική κατάσταση του κόσμου του Karel, τοποθετώντας τοίχους ή beepers καθώς και την αρχική θέση από την οποία θα ξεκινήσει το ρομπότ την αποστολή του. Στο Σχήμα 2 παρουσιάζεται ο συντάκτης των προγραμμάτων, ενώ στο Σχήμα 3 βλέπουμε ένα στιγμιότυπο της εκτέλεσης του προγράμματος που παρουσιάζεται στο Σχήμα 2.

Από τη σύντομη παρουσίαση της γλώσσας του Karel the Robot αλλά και του αντίστοιχου προγραμματιστικού περιβάλλοντος, γίνεται φανερό ότι ο μαθητής μαθαίνει τις έννοιες του προγραμματισμού ελέγχοντας με τις διαθέσιμες εντολές το ρομπότ. Οι εντολές είναι απλές ενώ το αποτέλεσμα της εκτέλεσης είναι άμεσα ορατό στην οθόνη. Ο μαθητής βλέπει το ρομπότ, δηλαδή τον πρωταγωνιστή του μικρόκοσμου, να εκτελεί μια προς μια τις εντολές, οι οποίες μεταβάλλουν την αρχική του κατάσταση και την κατάσταση του περιβάλλοντος που ζει. Ακόμη η δυνατότητα δημιουργίας νέων εντολών επιτρέπει πολύ εύκολα στο καθηγητή να παρουσιάσει την έννοια του υποπρογράμματος και να δείξει την αναγκαιότητα χρήσης υποπρογραμμάτων για την ανάπτυξη προγραμμάτων. Δηλαδή η δείξει ότι με τα υποπρογράμματα μπορούμε να επαναχρησιμοποιήσουμε κώδικα αλλά και να αντιμετωπίσουμε ένα πρόβλημα εφαρμόζοντας τμηματικό (δομημένο) προγραμματισμό (modular programming). Αντίθετα, χρησιμοποιώντας μια

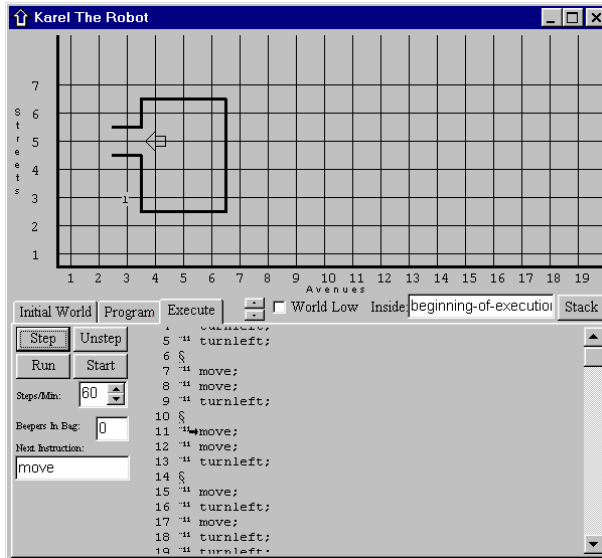
γλώσσα γενικού σκοπού, συνήθως, είναι δύσκολο να πείσουμε τους αρχάριους μαθητές για την αναγκαιότητα χρήσης υποπρογραμμάτων.



Σχήμα 1: Ο συντάκτης του κόσμου του Karel



Σχήμα 2: Ο συντάκτης του προγράμματος του Karel



Σχήμα 3: Η εκτέλεση ενός προγράμματος

Συνήθως τα προβλήματα που προτείνονται για την εισαγωγή της έννοιας του υποπρογράμματος είναι αριθμητικής φύσης, όπως για παράδειγμα ο υπολογισμός της δύναμης ενός αριθμού ή ο υπολογισμός του παραγοντικού, δηλαδή προβλήματα μικρού μεγέθους, τα οποία αδυνατούν να

καταστήσουν φανερή την αναγκαιότητα χρήσης υποπρογραμμάτων κατά την ανάπτυξη ενός προγράμματος. Έτσι για άλλη μια φορά η προσοχή επικεντρώνεται στην μάθηση της σύνταξης για τη δήλωση και κλήση των υποπρογραμμάτων και όχι στην ανάπτυξη ικανοτήτων για την επίλυση ενός προβλήματος βασισμένης στον τμηματικό προγραμματισμό.

Η δύναμη του Karel the Robot ως γλώσσας για τη διδασκαλία του προγραμματισμού, στηρίζεται στο γεγονός ότι καθιστά αόρατα στοιχεία τα οποία δεν είναι σημαντικά για μια εισαγωγή στον προγραμματισμό και επιτρέπει στους μαθητές να σχηματίσουν ευκολότερα ένα διαισθητικό εννοιολογικό μοντέλο για τις βασικές δομές μιας γλώσσας προγραμματισμού. Για παράδειγμα, η έλλειψη χειρισμού μεταβλητών, δηλαδή χειρισμού της νοητής μηχανής, και η αντικατάστασή τους με την κατάσταση του κόσμου του ρομπότ την οποία το πρόγραμμα που εκτελεί το ρομπότ μεταβάλλει, διευκολύνει τους μαθητές. Δηλαδή το πρόγραμμα που εκτελεί το ρομπότ διαχειρίζεται την κατάσταση που βλέπουν οι μαθητές, δηλαδή μια πραγματική κατάσταση, και όχι αφαιρετικές τιμές που αποθηκεύονται σε θέσεις μνήμης στις οποίες αποδίδουν συμβολικά ονόματα. Το πλεονέκτημα του Karel the Robot έγκειται στο γεγονός ότι η κατάσταση του κόσμου την οποία διαχειρίζεται ο Karel είναι πιο οικεία στους μαθητές απ' ό,τι η αφηρημένη έννοια της μεταβλητής, και η απλοποίηση των μεταβλητών διευκολύνει την κατανόηση των βασικών δομών μιας γλώσσας καθώς και την ανάπτυξη ικανοτήτων επίλυσης προβλημάτων.

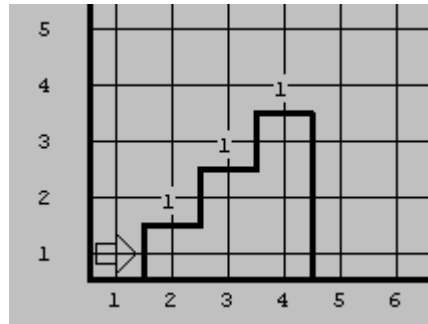
ΣΧΕΔΙΑ ΜΑΘΗΜΑΤΩΝ

Χρησιμοποιώντας τον Karel the Robot είναι δυνατόν να διδαχθούν και να γίνουν κατανοητά από τους μαθητές όλα όσα προτείνονται στον τέταρτο άξονα του ΕΠΣ για το μάθημα της Πληροφορικής της Γ' Γυμνασίου. Στη συνέχεια προτείνουμε ένα πλάνο μαθημάτων καθώς και προβλήματα τα οποία θα μπορούσαν να ακολουθηθούν από τους καθηγητές Πληροφορικής για τη διδασκαλία του προγραμματισμού στην Γ' Γυμνασίου.

Την 1η ώρα θα μπορούσαν οι μαθητές να εξοικειωθούν με το προγραμματιστικό περιβάλλον κατασκευάζοντας αρχικές καταστάσεις του κόσμου του Karel χρησιμοποιώντας το συντάκτη κόσμων και στη συνέχεια να εξοικειωθούν με την ανάκληση, αποθήκευση και τροποποίηση των ήδη κατασκευασμένων κόσμων.

Στην 2^η και 3^η διδακτική ώρα θα μπορούσαν να εξοικειωθούν με τις βασικές εντολές του ρομπότ αναθέτοντας στο ρομπότ να πραγματοποιήσει απλές αποστολές. Στη φάση αυτή οι μαθητές θα εξοικειωθούν με την έννοια συντάκτης προγράμματος και τις δυνατότητες που προσφέρει για ανάκληση υπαρχόντων προγραμμάτων καθώς και την αποθήκευση και τροποποίηση ήδη δημιουργημένων προγραμμάτων. Στη συνέχεια θα μπορούσαν να δοθούν έτοιμα προγράμματα και να ζητηθεί από τους μαθητές να τα εκτελέσουν ώστε να αντιληφθούν την έννοια της εκτέλεσης μιας αποστολής του ρομπότ, δηλαδή ενός προγράμματος. Τέλος θα μπορούσε να προταθεί για λύση το παρακάτω πρόβλημα:

Πρόβλημα 1: Στον κόσμο του Karel υπάρχει η σκάλα που φαίνεται στο διπλανό σχήμα. Ο Karel πρέπει να ξεκινήσει από τη διασταύρωση (1,1) με κατεύθυνση προς τα ανατολικά και να ανέβει τη σκάλα σκουπίζοντας τα σκουπίδια (beepers) που βρίσκονται σε κάθε σκαλοπάτι. Κατασκευάσε ένα πρόγραμμα που να καθοδηγεί το ρομπότ ώστε να εκτελέσει την παραπάνω αποστολή.

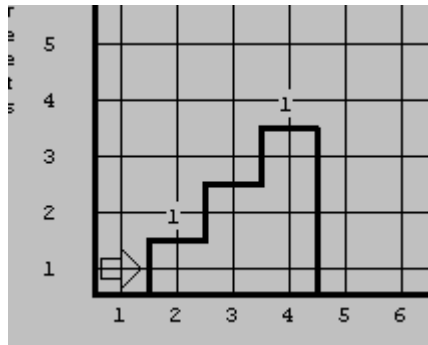


Η 4^η ώρα θα μπορούσε να αφιερωθεί στην παρουσίαση των προγραμματιστικών λαθών, προτείνοντας στους μαθητές προγράμματα που παρουσιάζουν συντακτικά λάθη, λάθη εκτέλεσης και λογικά λάθη.

Στην 5^η και 6^η να παρουσιασθεί η δυνατότητα προσθήκης νέων εντολών στο ρεπερτόριο του ρομπότ, καταδεικνύοντας, την αναγκαιότητα της ύπαρξης ενός μηχανισμού που μπορεί και ορίζει μια νέα εντολή, η οποία χρησιμοποιεί βασικές εντολές που αντιλαμβάνεται το ρομπότ. Για παράδειγμα η αδυναμία του ρομπότ να αντιλαμβάνεται μια εντολή που να τον οδηγεί να στρίψει κατά 90 μοίρες στα δεξιά μπορεί να «βελτιωθεί» με την προσθήκη στο λεξιλόγιο του ρομπότ της νέας εντολής turnright που κατασκευάζεται από τη χρήση 3 διαδοχικών turnleft. Ως άσκηση θα μπορούσε να προταθεί η ανάπτυξη νέου προγράμματος το οποίο να λύνει το πρόβλημα 1 ορίζοντας νέες εντολές.

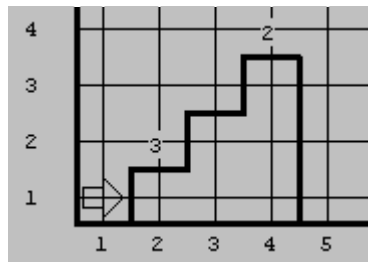
Οι 2 επόμενες ώρες μπορούν να χρησιμοποιηθούν για να λυθούν προβλήματα που απαιτούν τη χρήση if/then/else. Ως τυπικό πρόβλημα θα μπορούσε να προταθεί το παρακάτω.

Πρόβλημα 2: Στον κόσμο του Karel υπάρχει η σκάλα που φαίνεται στο διπλανό σχήμα. Ο Karel πρέπει να ξεκινήσει από τη διασταύρωση (1,1) με κατεύθυνση προς τα ανατολικά και να ανέβει τη σκάλα σκουπίζοντας τα σκουπίδια (beepers) που βρίσκονται σ' αυτή. Κατασκευάσε ένα πρόγραμμα που να καθοδηγεί το ρομπότ ώστε να εκτελέσει την παραπάνω αποστολή.



Και οι 2 τελευταίες ώρες να αφιερωθούν στην αντιμετώπιση προβλημάτων που απαιτούν τη χρήση επαναληπτικών εντολών. Ως πρόβλημα μπορεί να προταθεί το εξής:

Πρόβλημα 3: Στον κόσμο του Karel υπάρχει η σκάλα που φαίνεται στο διπλανό σχήμα. Ο Karel πρέπει να ξεκινήσει από τη διασταύρωση (1,1) με κατεύθυνση προς τα ανατολικά και να ανέβει τη σκάλα σκουπίζοντας τα σκουπίδια (beepers) που βρίσκονται σ' αυτή. Κατασκεύασε ένα πρόγραμμα που να καθοδηγεί το ρομπότ ώστε να εκτελέσει την παραπάνω αποστολή.



Στη συνέχεια θα μπορούσε να προταθεί για λύση το εξής πρόβλημα:

Πρόβλημα 4: Στον κόσμο του Karel υπάρχει η σκάλα που φαίνεται στο παραπάνω σχήμα. Ο Karel πρέπει να ξεκινήσει από τη διασταύρωση (1,1) με κατεύθυνση προς τα ανατολικά και να ανέβει τη σκάλα σκουπίζοντας τα σκουπίδια (beepers) που βρίσκονται σ' αυτή. Σε κάθε σκαλοπάτι υπάρχει άγνωστος αριθμός από beepers. Στο παραπάνω σχήμα φαίνεται μία από τις πιθανές αρχικές καταστάσεις. Κατασκεύασε ένα πρόγραμμα που δίνει λύση στο πρόβλημα οποιαδήποτε και αν είναι η αρχική κατάσταση (αριθμός σκουπιδιών στα σκαλοπάτια).

Στη διάρκεια των μαθημάτων καλό είναι να χρησιμοποιείται το ίδιο ή ελαφρώς τροποποιημένο πρόβλημα προς λύση, ώστε οι μαθητές να αντιλαμβάνονται την αξία της νέας έννοιας που διδάσκονται κάθε φορά αλλά και επιπλέον ο προσδιορισμός του προβλήματος να μην απαιτεί ιδιαίτερη προσπάθεια.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Το προτεινόμενο περιβάλλον του Karel the Robot, ή ακριβέστερα η οικογένεια των περιβαλλόντων αυτών – όπως δείχνουν μέχρι στιγμής τα ερευνητικά ή εμπειρικά δεδομένα – φαίνονται να είναι κατάλληλα για μια εισαγωγή στον προγραμματισμό.

Οι μικρόκοσμοι αυτής της κατηγορίας επιτρέπουν ουσιαστικά την εξοικείωση των αρχαρίων με την αλγοριθμική κυρίως όψη του προγραμματισμού HY - και μάλιστα με έναν τρόπο που είναι ακόμη απλούστερος και της Logo.

Η ανυπαρξία διαχείρισης δεδομένων και αποτελεσμάτων (δηλαδή εισόδου και εξόδου), ο άμεσος και «προφανής» τρόπος ανταπόκρισης του συστήματος συνιστούν στοιχεία τα οποία, τουλάχιστον σε μια αρχική φάση, μπορούν να συντελέσουν με ουσιαστικό τρόπο στην υπέρβαση προβλημάτων κατανόησης που αντιμετωπίζουν οι αρχάριοι προγραμματιστές.

Βέβαια το περιβάλλον Karel μπορεί να παρουσιάζει πολλά πλεονεκτήματα, αλλά τούτο δε σημαίνει ότι δεν παρουσιάζει και προβλήματα. Για παράδειγμα η γνώση που θα αποκτήσουν οι σπουδαστές μπορεί να είναι υπερβολικά πλαισιοποιημένη, δηλαδή υπερβολικά εξαρτώμενη από το περιβάλλον Karel. Όπως μάλιστα τείνουν να υποστηρίζουν σχετικές εργασίες (Dagdilelis, Balasheff, Carroni, 1990), οι σπουδαστές δύσκολα γενικεύουν ή μεταφέρουν σε άλλα περιβάλλοντα προγραμματιστικές γνώσεις που έχουν αποκτηθεί σε συγκεκριμένα περιβάλλοντα. Η διαδικασία της αποπλαισιοποίησης της γνώσης και της επαναπλαισιοποίησης της δεν είναι αυτόματη.

Επίσης, το πρόβλημα της διαχείρισης μεταβλητών και της εισόδου-εξόδου τιμών παραμένει ανοικτό, αλλά τα προβλήματα αυτής της κατηγορίας φαίνονται να είναι λιγότερο άμεσα συνδεδεμένα με την αλγοριθμική επίλυση των σχετικών προβλημάτων.

Οι μικρόκοσμοι Karel αποτελούν λοιπόν μια πρόταση για τη διδασκαλία του προγραμματισμού. Τόσο η θεωρητική a priori ανάλυση, όσο και τα εμπειρικά δεδομένα από την πολυετή χρήση των μικροκόσμων (Brusilovsky P. et al, 1997), (Miller, Pane, Meter & Vorthmann, 1994) φαίνονται να συγκλίνουν στην άποψη ότι αποτελούν μια αξιοπρόσεκτη δυνατότητα, κυρίως σε ότι αφορά τη

Δευτεροβάθμια Εκπαίδευση. Στη συνέχεια παρουσιάζουμε κάποια αποτελέσματα από ερωτηματολόγια που συγκεντρώθηκαν στο Carnegie Mellon University και σε πέντε άλλα πανεπιστήμια από μαθητές και καθηγητές που χρησιμοποίησαν τα περιβάλλοντα Genies. Η ανάλυση των ερωτηματολογίων έδειξε ότι:

1. Οι σπουδαστές που χρησιμοποιούν τα περιβάλλοντα Genies δεν παρουσιάζουν τις τυπικές δυσκολίες που αντιμετωπίζουν συνήθως οι αρχάριοι, όπως προβλήματα με τη σύνταξη της Pascal και τη χρήση υποπρογραμμάτων σε μεγάλα προγράμματα.
2. Οι σπουδαστές που χρησιμοποιούν τα περιβάλλοντα Genies είναι πιο ικανοποιημένοι με το περιβάλλον τους από άλλους σπουδαστές που ερωτήθηκαν, ενώ εκείνοι που έχουν χρησιμοποιήσει περισσότερα από ένα περιβάλλοντα προτιμούν τα περιβάλλοντα Genies.
3. Οι σπουδαστές που υιοθέτησαν ένα τμηματικό τρόπο προγραμματισμού είχαν καλύτερη επίδοση.
4. Η δυναμική εκτέλεση των προγραμμάτων ωφελεί ακόμα και τους πιο «ταλαντούχους» σπουδαστές.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Du Boulay, B., (1989), Some Difficulties Of Learning To Program, *In Studying The Novice Programmer*, Soloway, E., Sprohrer, J. (Eds.) Lawrence Erlbaum Associates, 283-300.
- Du Boulay, B., O'Shea, T. and Monk, J., (1989b), The Black Box Inside the Glass Box: Presenting Computing Concepts to Novices, *In Studying The Novice Programmer*, Soloway, E., Sprohrer, J. (Eds.) Lawrence Erlbaum Associates, 431-446.
- Bergin J., Stehlik M., Roberts J., & Pattis R., (1996), *Karel++*, *A Gentle Introduction to the Art of Object-Oriented Programming*, New York, Wiley.
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A. & Miller P., (1997), Mini-languages: a way to learn programming principle, *Education and Information Technologies*, 2, 65-83
- Dagdilelis V., (1986), *Conceptions des élèves a propos des notions fondamentales de la programmation informatique en classe de Troisième*, Mémoire D.E.A., Université Joseph FOURIER, Grenoble, France.
- Dagdilelis V., (1989), *La validation en programmation : a propos de conceptions des étudiants*, actes V Ecole d'été de Didactique des Mathématiques et de l'Informatique, Plestin-les-Grèves, France.
- Dagdilelis V., Balasheff N., Capponi B., (1990), L' apprentissage de l' itération dans deux environnements informatiques, *ASTER*, 1990.
- Pattis, R. E., Roberts, J. & Stehlik, M. (1995), *Karel - The Robot, A Gentle Introduction to the Art of Programming*, 2nd edn. New York, Wiley.
- Δαγδiléλης, 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.
- Miller, P., Pane, J., Meter, G. & Vorthmann. S., (1994) Evolution of Novice Programming Environments: The Structure Editors of Carnegie Mellon University, *Interactive Learning Environments* 4 (2), 140-158.
- Σατρατζέμη Μ., Δαγδiléλη Β., Μαργαρίτη Κ., Μιχαήλ Σ., Παπανικολάου Σ., (1999), Ενας Διδακτικός Μικρόκοσμος για μια Εισαγωγή στον προγραμματισμό: το Ρομπότ Karel, *Πρακτικά του 4ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή - Διδακτική των Μαθηματικών & Πληροφορική στην Εκπαίδευση*, 250-255.