

Ε.Π. Εκπαίδευση και Δια Βίου Μάθηση, ΕΣΠΑ (2007 – 2013)

ΕΠΙΜΟΡΦΩΣΗ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΓΙΑ ΤΗΝ ΑΞΙΟΠΟΙΗΣΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΤΩΝ ΤΠΕ ΣΤΗ ΔΙΔΑΚΤΙΚΗ ΠΡΑΞΗ

Επιμορφωτικό υλικό
για την επιμόρφωση των εκπαιδευτικών στα
Κέντρα Στήριξης Επιμόρφωσης

ΤΕΥΧΟΣ 6B: ΕΙΔΙΚΟ ΜΕΡΟΣ ΚΛΑΔΩΝ ΠΕ19/20 ΔΙΔΑΚΤΙΚΑ ΣΕΝΑΡΙΑ

Α' έκδοση



Διεύθυνση Επιμόρφωσης & Πιστοποίησης

Πάτρα, Απρίλιος 2013



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα

Προοίμιο.....	8
Ενότητα Σεναρίου 1.....	10
Το ρομπότ KAREL: εισαγωγή στο δομημένο προγραμματισμό	10
1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	10
2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	10
3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ.....	10
4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	11
5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	12
6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	15
7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία») και αντίλογος, επιφυλάξεις, προβλήματα).....	16
8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ .	19
9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ	23
10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ.....	23
11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ.....	24
12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ	24
13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ	25
14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ	25
15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ	26
16. ΑΞΙΟΛΟΓΗΣΗ	31
17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ	31
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - Εξοικείωση με τον μικρόκοσμο του ρομπότ Karel	32
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 – Δημιουργώντας νέες εντολές για τον Karel	40
18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ	48
Ενότητα Σεναρίου 2	49
Το ρομπότ objectKAREL: εισαγωγή στον αντικειμενοστραφή προγραμματισμό	49
1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	49
2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	49
3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ.....	49
4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	50
5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	51
6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	57

7.	ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα).....	58
8.	ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ .	61
9.	ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ	63
10.	ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ.....	63
11.	ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ.....	66
12.	ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ	66
13.	ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ	66
14.	ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ	66
15.	ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ	67
16.	ΑΞΙΟΛΟΓΗΣΗ	71
17.	ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ	71
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - Εξοικείωση με τα αντικείμενα και τις κλάσεις	72
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Προγραμματίζοντας ρομπότ.....	79
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 – Τεστ αξιολόγησης.....	89
18.	ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ	90
	Ενότητα Σεναρίου 3	91
	Διδασκαλία των δομών επανάληψης	91
2.	ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	91
3.	ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ.....	91
4.	ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	92
5.	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	93
6.	ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	96
7.	ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα).....	100
8.	ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ	101
9.	ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ	103
10.	ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ.....	103
11.	ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ.....	104
12.	ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ	104
13.	ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ	104
14.	ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ	104
15.	ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ	106
16.	ΑΞΙΟΛΟΓΗΣΗ	106
17.	ΓΙΑ ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ	106

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - ΟΣΟ...ΕΠΑΝΑΛΑΒΕ	107
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Επανάλαβε... μέχρις_ότου	109
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 - Για....από...μέχρι...	111
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 - ΕΠΑΝΑΛΗΨΗ ΕΜΠΕΔΩΣΗ	112
18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ	115
Ενότητα Σεναρίου 4	116
Από τη μια μορφή επανάληψης σε μια άλλη.....	116
2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	116
3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ.....	116
4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	116
5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	117
6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	118
7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα).....	118
8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ	119
9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ	120
10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ.....	121
11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ.....	121
12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ	122
13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ	122
14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ	122
15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ	122
16. ΑΞΙΟΛΟΓΗΣΗ	122
17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ	122
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1	123
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2	126
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3	129
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 - ΕΠΑΝΑΛΗΨΗ ΕΜΠΕΔΩΣΗ.....	132
Ενότητα Σεναρίου 5	133
Ταξινόμηση με τη μέθοδο φυσαλίδας	133
2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	133
3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ.....	133
4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	133
5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	134

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	138
7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα).....	138
8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ	138
9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ	139
10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ.....	140
11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ.....	141
12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ	141
13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ	141
14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ	141
15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ	142
16. ΑΞΙΟΛΟΓΗΣΗ	142
17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ	142
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1	143
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2	144
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3	146
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 - ΕΠΑΝΑΛΗΨΗ ΕΜΠΕΔΩΣΗΣ.....	147
18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ.....	148
Ενότητα Σεναρίου 6	149
Εισαγωγή στη γλώσσα Logo	149
1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	149
2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	149
3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ	149
4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	149
5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	149
6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	150
7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)	153
8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ / ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ	155
9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ	155
10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ	156
11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ	156
12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ	157
13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ	157
14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ.....	157

15.	ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ	157
16.	ΑΞΙΟΛΟΓΗΣΗ	160
17.	ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ	160
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1	163
	Γνωριμία με το περιβάλλον MSW Logo – Οι πρώτες εντολές.....	163
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2	166
	Φτιάχνοντας σχήματα	166
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3	168
	Φτιάχνοντας Διαδικασίες.....	168
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4	171
	Τα προγράμματα ως Συναρτήσεις.....	171
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5	174
	Τα προγράμματα ως Συναρτήσεις: αναδρομή.....	174
18.	ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ.....	178
	Οπτικός προγραμματισμός για μικρές ηλικίες.....	178
	ΠΑΡΑΡΤΗΜΑ 1: ο Χελωνόκοσμος.....	186
	Τα ψηφιδωτά.....	186
	Ο Χελωνόκοσμος και η Κερήθρα.....	192
	Φωνάζοντας το όνομά μου (εισαγωγή στην αναδρομή – recursion)	201
	ΠΑΡΑΡΤΗΜΑ 2: Easy Logo	209
	Ενότητα Σεναρίου 7	213
	Εννοιολογική χαρτογράφηση.....	213
1.	ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ.....	213
2.	ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	213
3.	ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ	213
4.	ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	214
5.	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	215
6.	ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ	216
7.	ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)	219
8.	ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ .	220
9.	ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ	221
10.	ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ	221
11.	ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ	222
12.	ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ	222
13.	ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ	222

14.	ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Ή ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ.....	222
15.	ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ	223
16.	ΑΞΙΟΛΟΓΗΣΗ	229
17.	ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ	229
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 – Βασικές Έννοιες & Κατηγορίες Δικτύων Υπολογιστών	230
	ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 – Αξιολόγηση	235
18.	ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ.....	236
	ΠΑΡΑΡΤΗΜΑ.....	237

Προοίμιο

Το επιμορφωτικό υλικό για την επιμόρφωση των εκπαιδευτικών στα Κέντρα Στήριξης Επιμόρφωσης εκπονήθηκε αρχικά στο πλαίσιο της Πράξης «Επιμόρφωση Εκπαιδευτικών στη χρήση και αξιοποίηση των ΤΠΕ στην εκπαιδευτική διδακτική διαδικασία» του ΕΠΕΑΕΚ ΙΙ, Γ' ΚΠΣ και αφορούσε στους κλάδους εκπαιδευτικών ΠΕ02 (Φιλολογοί), ΠΕ03 (Μαθηματικοί), ΠΕ04 (Φυσικές Επιστήμες) και ΠΕ60/70 (Δάσκαλοι και Νηπιαγωγοί). Στο πλαίσιο της Πράξης «Επιμόρφωση των Εκπαιδευτικών για την Αξιοποίηση και Εφαρμογή των ΤΠΕ στη Διδακτική Πράξη» του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» του ΕΣΠΑ (2007-2013), επικαιροποιήθηκε το παραπάνω υλικό, ενώ εκπονήθηκε το επιμορφωτικό υλικό για τους κλάδους ΠΕ19-20 (καθηγητές Πληροφορικής).

Το επιμορφωτικό υλικό για την επιμόρφωση των εκπαιδευτικών περιλαμβάνει 6 τεύχη: το τεύχος 1, που περιέχει το λεγόμενο Γενικό Μέρος (αφορά όλους τους κλάδους εκπαιδευτικών), και πέντε ακόμη τεύχη του Ειδικού μέρους, ένα για καθένα από τους 5 κλάδους. Το επιμορφωτικό υλικό που αφορά στο ειδικό μέρος της επιμόρφωσης των εκπαιδευτικών ΠΕ19/20 αποτελεί το τεύχος 6, διαιρεμένο στα υποτεύχη 6Α, 6Β, 6Γ.

Το επιμορφωτικό υλικό του παρόντος τεύχους 6B (τμήμα του οποίου παρήχθη χρησιμοποιώντας ως πρωτογενές υλικό το αντίστοιχο υλικό της εκπαίδευσης επιμορφωτών ΠΕ19/20¹) δημιουργήθηκε από συγγραφική ομάδα, με επικεφαλής τον Καθηγητή του Πανεπιστημίου Μακεδονίας Βασίλειο Δαγδιλέλη και συμμετέχοντες τους εξής:

- Καζανίδη Ιωάννη
- Μαλλιαράκη Χρήστο

¹ Το αντίστοιχο Επιμορφωτικό Υλικό για την εκπαίδευση επιμορφωτών στα Πανεπιστημιακά Κέντρα Επιμόρφωσης (ΠΑΚΕ) δημιουργήθηκε από συγγραφική ομάδα αποτελούμενη από τον Καθηγητή του ΕΚΠΑ Χρόνη Κυνηγό, την Καθηγήτρια του ΕΚΠΑ Μαρία Γρηγοριάδου, τον Καθηγητή του Πανεπιστημίου Μακεδονίας Βασίλειο Δαγδιλέλη, τον Καθηγητή του Πανεπιστημίου Πατρών Βασίλειο Κόμη και τον Αναπληρωτή Καθηγητή του Πανεπιστημίου Πελοποννήσου Αθανάσιο Τζιμογιάννη. Στο υλικό συνεισέφεραν οι Σοφία Αγγέλαινα, Αγορίτσα Γόγουλου, Ευαγγελία Γουλή, Αναστάσιος Λαδιάς, Δημήτριος Νικολός, Ηλίας Νίτσος, Στυλιανός Ξυνόγαλος, Γεώργιος Πανσεληνάς, Κυπαρισσία Παπανικολάου, Δημήτριος Ρούσσινος, Ευθύμιος Ταμπούρης, Γραμματική Τσαγκάνου, Παναγιώτης Τσιωτάκης, Νίκος Φαχαντίδης, Στασινή Φράγκου.

- Ξυνόγαλο Στυλιανό
- Παπαδόπουλο Ιωάννη
- Πανσεληνά Γεώργιο
- Χατζηφωτεινού Αικατερίνη

Ένα τμήμα του επιμορφωτικού υλικού του Τεύχους 1 (Γενικό Μέρος) χρησιμοποιήθηκε ως πρωτογενές για την παραγωγή του παρόντος τεύχους 6B. Το υλικό του τεύχους 1 είχε δημιουργηθεί και επικαιροποιηθεί από συγγραφική ομάδα με επικεφαλής τον Καθηγητή του Πανεπιστημίου Μακεδονίας Βασίλειο Δαγδιλέλη και συνεργάτη τον Παπαδόπουλο Ιωάννη, Λέκτορα του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης².

Το επιμορφωτικό υλικό αυτό συντάχθηκε στο πλαίσιο των αρμοδιοτήτων της ειδικής επιστημονικής επιτροπής, αποτελούμενης από τους:

- Χαράλαμπος Ζαγούρα, Καθηγητή του Πανεπιστημίου Πατρών, ο οποίος έχει την ευθύνη συντονισμού των εργασιών της επιτροπής
- Βασίλειο Δαγδιλέλη, Καθηγητή Πανεπιστημίου Μακεδονίας
- Βασίλειο Κόμη, Καθηγητή Πανεπιστημίου Πατρών
- Δημήτριο Κουτσογιάννη, Αναπληρωτή Καθηγητή Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης
- Χρόνη Κυνηγό, Καθηγητή Πανεπιστημίου Αθηνών
- Δημήτριο Ψύλλο, Καθηγητή Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης

Η ως άνω ειδική επιστημονική επιτροπή λειτουργεί στην παρούσα Πράξη ως Επιστημονική Επιτροπή του ΙΤΥΕ-Διόφαντος, συμπράττοντος επιστημονικού φορέα υλοποίησης του έργου.

Το επιμορφωτικό υλικό για την επιμόρφωση των εκπαιδευτικών αποτελεί ιδιοκτησία του ΥπΔΒΜΘ και καλύπτεται από την ισχύουσα νομοθεσία για την προστασία των πνευματικών δικαιωμάτων των δημιουργών.

² Για το Γενικό Μέρος του Επιμορφωτικού Υλικού για τα ΠΑΚΕ συνεργάστηκαν και οι Καψάλης Αχιλλέας, πρώην Καθηγητής στο Πανεπιστήμιο Μακεδονίας, Παπαδόπουλος Ιωάννης, Λέκτορας του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, Φαχαντίδης Νικόλαος, Αναπληρωτής Καθηγητής στο Πανεπιστήμιο Δυτικής Μακεδονίας.

ΔΙΔΑΚΤΙΚΑ ΣΕΝΑΡΙΑ & ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ

Ενότητα Σεναρίου 1

Το ρομπότ KAREL: εισαγωγή στο δομημένο προγραμματισμό

Karel – Δομημένος προγραμματισμός

1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Δημιουργία νέων εντολών στον μικρόκοσμο του ρομπότ Karel

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 4-5 διδακτικές ώρες, δηλαδή περίπου 2 ώρες για την εξοικείωση με τον μικρόκοσμο και τη δομή ακολουθίας και 2-3 ώρες για την εξοικείωση με τη δημιουργία νέων εντολών (διαδικασιών). Βέβαια, πρέπει να επισημανθεί ότι η διάρκεια αυτή είναι ενδεικτική και εξαρτάται από την ηλικία, την προηγούμενη εμπειρία και τον αριθμό των μαθητών.

3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Πρόκειται για διδακτικό σενάριο που μπορεί να ενταχθεί κατά κύριο λόγο στο μάθημα Πληροφορικής του Γυμνασίου για τη διδασκαλία των αρχών του δομημένου-διαδικαστικού προγραμματισμού. Τα ιδιαίτερα χαρακτηριστικά του διδακτικού μικρόκοσμου Karel καθιστούν τη διδασκαλία τόσο των βασικών αλγοριθμικών δομών όσο και της έννοιας της διαδικασίας ιδιαίτερα εύκολη.

Το σενάριο δεν προϋποθέτει κάποιες γνώσεις προγραμματισμού, αλλά η ύπαρξή τους μπορεί να επηρεάσει την εφαρμογή του σεναρίου. Η πρότερη εμπειρία των μαθητών μπορεί να αξιοποιηθεί και να εμπλουτιστούν κατάλληλα οι προτεινόμενες δραστηριότητες ώστε να περιλαμβάνουν, για παράδειγμα, την επίλυση προβλημάτων με μεγαλύτερη

αλγοριθμική πολυπλοκότητα.

4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Βασικός σκοπός του σεναρίου είναι να εισαχθούν οι μαθητές στην έννοια της διαδικασίας μέσω της δημιουργίας νέων εντολών για το ρομπότ Karel. Αυτό βέβαια προϋποθέτει ότι οι μαθητές έχουν εισαχθεί προηγουμένως στην έννοια της γλώσσας προγραμματισμού, του προγράμματος και της δομής ακολουθίας, καθώς επίσης και με τις βασικές λειτουργίες του περιβάλλοντος. Για λόγους πληρότητας, στο επιμορφωτικό σενάριο περιλαμβάνονται και τα φύλλα εργασίας που μπορούν να χρησιμοποιηθούν για την διδασκαλία όλων των προαναφερθεισών εννοιών και όχι μόνο της έννοιας της διαδικασίας.

Στους στόχους του σεναρίου περιλαμβάνονται η κατανόηση και η εφαρμογή των σχετικών εννοιών και η αποφυγή της δημιουργίας των παρανοήσεων (ελλιπής/λανθασμένη κατανόηση εννοιών) που έχουν καταγραφεί στη βιβλιογραφία.

Αναλυτικά, οι στόχοι του σεναρίου είναι

... όσον αφορά στις βασικές έννοιες προγραμματισμού και τη δομή ακολουθίας:

- Η εξοικείωση των μαθητών με το μοντέλο αναφοράς που χρησιμοποιείται στον μικρόκοσμο του ρομπότ Karel (δρόμοι, λεωφόροι, τμήματα τοίχου, beepers, ρομπότ Karel).
- Η κατανόηση της έννοιας της γλώσσας προγραμματισμού και του προγράμματος. Ένα πρόγραμμα παρουσιάζεται ως μια σειρά οδηγιών/εντολών που κατευθύνουν το ρομπότ Karel στην επίτευξη μιας αποστολής, ενώ η γλώσσα προγραμματισμού παρουσιάζεται ως μια ειδική γλώσσα την οποία κατανοεί ο Karel.
- Η κατανόηση της έννοιας του καθορισμού των δεδομένων ενός προβλήματος. Στον μικρόκοσμο του Karel ο καθορισμός των δεδομένων αναφέρεται στον καθορισμό της αρχικής κατάστασης του Karel (δρόμος, λεωφόρος, κατεύθυνση, αριθμός beepers στην τσάντα) και της κατάστασης του κόσμου (τυχόν τμήματα τοίχου ή/και beepers).
- Η εξοικείωση με τις βασικές εντολές που περιλαμβάνει η γλώσσα προγραμματισμού του Karel, η κατανόηση της σημασίας τους και η ανάπτυξη απλών προγραμμάτων που περιλαμβάνουν μια ακολουθία αυτών των βασικών εντολών.
- Η κατανόηση του γεγονότος ότι η εκτέλεση μιας εντολής από τον Karel μπορεί να έχει αποτέλεσμα την μεταβολή - αλλαγή της κατάστασης του (δρόμος, λεωφόρος, κατεύθυνση, αριθμός beepers στην τσάντα), η οποία αποθηκεύεται σε

προκαθορισμένες θέσεις της μνήμης που διαθέτει ο Karel και ονομάζονται *μεταβλητές*.

- Η κατανόηση των βασικών τύπων λαθών – συντακτικά λάθη, λογικά λάθη και λάθη εκτέλεσης – που μπορεί να προκύψουν κατά την ανάπτυξη ενός προγράμματος και η διόρθωσή τους.
- Η εξοικείωση των μαθητών με τις βασικές λειτουργίες του περιβάλλοντος: περιβάλλον μαθημάτων, περιβάλλον προγραμματισμού (ανάπτυξη, μεταγλώττιση και εκτέλεση προγράμματος).

... *όσον αφορά στην έννοια της διαδικασίας:*

- Η κατανόηση της ανάγκης ύπαρξης ενός μηχανισμού για τη *δημιουργία νέων εντολών (διαδικασιών)* για το ρομπότ Karel.
- Η κατανόηση των πλεονεκτημάτων που παρέχει η χρήση διαδικασιών σε ένα πρόγραμμα.
- Η εξοικείωση με τη *δήλωση και τον ορισμό νέων εντολών (διαδικασιών)*.
- Η κατανόηση της *ροής εκτέλεσης* ενός προγράμματος που χρησιμοποιεί διαδικασίες και η αντιμετώπιση των σχετικών δυσκολιών των μαθητών.
- Η εξοικείωση με την κλήση διαδικασιών μέσα από άλλες διαδικασίες.

5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το παρόν σενάριο για την εισαγωγή των μαθητών στον δομημένο-διαδικαστικό προγραμματισμό με τον μικρόκοσμο του ρομπότ Karel, βασίζονται στο διδακτικό υλικό που ενσωματώνεται στο ίδιο το περιβάλλον και στη διδακτική στρατηγική που υιοθετήθηκε και διέπει τη σχεδίαση του ίδιου του περιβάλλοντος. Κρίνεται σκόπιμο λοιπόν, πριν την παρουσίαση των σχετικών φύλλων εργασίας, να αναφερθούν τα βασικά χαρακτηριστικά του περιβάλλοντος μιας και αυτά καθορίζουν και τη διδακτική στρατηγική που υιοθετείται σε μια διδασκαλία που βασίζεται σε αυτό.

Οι λειτουργίες που ενσωματώνει ο προγραμματιστικός μικρόκοσμος Karel είναι οι εξής:

- Μια **σειρά μαθημάτων**, κάθε ένα από τα οποία περιλαμβάνει σύντομη και περιεκτική θεωρία και μία ή περισσότερες δραστηριότητες.
- Ένα **συντάκτη δομής**, ο οποίος επιτρέπει την ανάπτυξη προγραμμάτων μέσω ενός

μενού με τις διαθέσιμες εντολές και πλαισίων διαλόγου. Το μενού ενημερώνεται αυτόματα με τις νέες εντολές που ορίζει ο μαθητής.

- Ένα σύστημα δυναμικής προσομοίωσης εκτέλεσης των προγραμμάτων που παρέχει τη δυνατότητα της **βηματικής εκτέλεσης** και της **επεξηγηματικής οπτικοποίησης**, εμφάνισης δηλαδή επεξηγήσεων για τη σημασία της τρέχουσας εντολής χρησιμοποιώντας φυσική γλώσσα.
- **Καταγραφή των ενεργειών των μαθητών** και παρουσίαση τους σε ένα διαφορετικό παράθυρο.

Τα παιδαγωγικά οφέλη που απορρέουν από τα προαναφερθέντα χαρακτηριστικά του μικρόκοσμου Karel παρουσιάζονται στον Πίνακα 1.

Χαρακτηριστικό	Παιδαγωγικά οφέλη
Χρήση της προσέγγισης των μικρόκοσμων	εξαλείφει το πρόβλημα του εκτεταμένου ρεπερτορίου εντολών και την πολυπλοκότητα των γλωσσών προγραμματισμού γενικού σκοπού μειώνει τη διανοητική “απόσταση” ανάμεσα στα νοητικά μοντέλα ή τις περιγραφές των αλγορίθμων σε φυσική γλώσσα και την περιγραφή τους στη γλώσσα προγραμματισμού επιλύονται προβλήματα που προκαλούν το ενδιαφέρον των μαθητών
Ενσωμάτωση τμήματος με μαθήματα (θεωρία, δραστηριότητες)	όλο το απαραίτητο υλικό για το μαθητή συμπεριλαμβάνεται στο περιβάλλον οι δραστηριότητες βοηθάνε τους μαθητές να εξοικειωθούν με τις έννοιες πριν να τους ζητηθεί να τις υλοποιήσουν
Συντάκτης δομής	δεν χρειάζεται να απομνημονευθούν οι συντακτικές λεπτομέρειες της γλώσσας προγραμματισμού και δίνεται η δυνατότητα επικέντρωσης στις έννοιες
Φιλικά προς το χρήστη μηνύματα λάθους	αποφεύγεται η χρονοβόρα διαδικασία εντοπισμού και διόρθωσης των λαθών, η οποία συνήθως προκαλεί την απογοήτευση των μαθητών

	οι λανθασμένες αντιλήψεις, στις οποίες συνήθως οφείλονται τα λάθη, αποκαλύπτονται στους μαθητές
<i>Βηματική εκτέλεση και επεξηγηματική οπτικοποίηση</i>	αποκαλύπτει τη δυναμική φύση της εκτέλεσης προγραμμάτων στηρίζει τους μαθητές στην κατανόηση της σημασιολογίας των δομών ελέγχου και της ροής εκτέλεσης υποστηρίζει τους μαθητές στον εντοπισμό λογικών λαθών δίνει τη δυνατότητα στο διδάσκοντα να καλύψει περισσότερη ύλη σε μικρότερο χρονικό διάστημα
<i>Καταγραφή των ενεργειών των μαθητών</i>	στηρίζει τους διδάσκοντες στη μελέτη των δυσκολιών και των λανθασμένων αντιλήψεων, καθώς και των τεχνικών επίλυσης προβλημάτων που χρησιμοποιούν οι μαθητές

Πίνακας 1. Τα παιδαγωγικά οφέλη του μικρόκοσμου προγραμματισμού Karel

Στο παρόν σενάριο υιοθετείται μια ενδεικτική οργάνωση των μαθημάτων, η οποία εφαρμόστηκε για τη διδασκαλία του προγραμματισμού στη Γ' τάξη του 2^{ου} Πειραματικού Γυμνασίου Θεσσαλονίκης τη σχολική χρονιά 2008-2009 (Ξυνόγαλος, 2009; Χινογαλος, 2011). Φυσικά δεν είναι απαραίτητο να διδαχθεί και πάλι με την ίδια ακριβώς σειρά, αλλά αυτή έχει ήδη δοκιμαστεί. Στον Πίνακα 2 παρουσιάζεται συνοπτικά η οργάνωση και το περιεχόμενο της προτεινόμενης σειράς μαθημάτων. Όπως ήδη αναφέρθηκε η οργάνωση των μαθημάτων είναι ενδεικτική. Ο εκπαιδευτικός μπορεί ανάλογα με την ηλικία και την προηγούμενη εμπειρία των μαθητών, καθώς επίσης και τις διαθέσιμες διδακτικές ώρες να προσαρμόσει κατάλληλα τα μαθήματα. Θα μπορούσε για παράδειγμα να επικεντρωθεί κανείς στις βασικές αλγοριθμικές δομές (ακολουθίας, επιλογής και επανάληψης) και να μην διδάξει τη δημιουργία νέων εντολών (υποπρογραμμάτων). Επίσης, θα μπορούσε κανείς να αφιερώσει στα 3 τελευταία μαθήματα περισσότερο χρόνο και να εμβαθύνει στις συγκεκριμένες έννοιες. Στην περίπτωση που ο εκπαιδευτικός αποφασίσει να διδάξει τη - θεμελιώδη για το διαδικαστικό προγραμματισμό – έννοια της διαδικασίας καλό θα ήταν αυτό να γίνει στην αρχή των μαθημάτων. Εξάλλου, ο εκδότης δομής του περιβάλλοντος, η απουσία των μεταβλητών ως παραμέτρων και η βηματική εκτέλεση των προγραμμάτων καθιστούν τη διδασκαλία της έννοιας της διαδικασίας πολύ ευκολότερη.

Διδακτική ώρα	Τίτλος μαθήματος	Περιεχόμενο
1 ^η – 2 ^η	Βασικές εντολές	Περιγραφή του μικρόκοσμου, πρόγραμμα, βασικές εντολές, μπλοκ εντολών, κυρίως πρόγραμμα
3 ^η – 4 ^η	Δημιουργία νέων εντολών	Ορισμός νέας εντολής, τα πλεονεκτήματα της δημιουργίας νέων εντολών
5 ^η – 6 ^η	Δομή επιλογής	Συνθήκη, λογικές τιμές, χρησιμότητα των εντολών υπό συνθήκη, if, if/else
7 ^η – 8 ^η	Δομές επανάλ.	while, loop

Πίνακας 2. Ενδεικτική οργάνωση των μαθημάτων στον μικρόκοσμο Karel

Ένα εναλλακτικό σενάριο, στο οποίο προτείνεται η διδασκαλία των δομών επιλογής πριν από την έννοια της διαδικασίας, παρουσιάζεται στην εργασία (Κοτίνη, 2011).

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Η βασικότερη έννοια του δομημένου-διαδικαστικού προγραμματισμού είναι η έννοια του υποπρογράμματος (διαδικασίας, συνάρτησης). Σε ένα τυπικό μάθημα εισαγωγής στον διαδικαστικό προγραμματισμό βέβαια, η έννοια της διαδικασίας διδάσκεται σε ένα αρκετά προχωρημένο στάδιο και αφού πρώτα παρουσιαστούν στους μαθητές οι τρεις βασικές αλγοριθμικές δομές: η δομή ακολουθίας, η δομή επιλογής και η δομή επανάληψης.

Στον μικρόκοσμο του ρομπότ Karel, ωστόσο, προτείνεται η διδασκαλία της έννοιας της διαδικασίας να γίνει από τα πρώτα μαθήματα, αμέσως μετά από τη δομή ακολουθίας. Αυτό μπορεί να επιτευχθεί εύκολα στον μικρόκοσμο του ρομπότ Karel, μιας και η έννοια της διαδικασίας παρουσιάζεται με σαφή και κατανοητό τρόπο. Η πολυπλοκότητα εκμάθησης της συγκεκριμένης έννοιας περιορίζεται σημαντικά από το γεγονός μη χρήσης λίστας παραμέτρων τόσο στις ήδη υλοποιημένες διαδικασίες που καλούν οι μαθητές από το πρώτο μάθημα, όσο και σε αυτές που αναπτύσσουν οι ίδιοι οι μαθητές. Επιπλέον, η γλώσσα προγραμματισμού που χρησιμοποιείται στον μικρόκοσμο και οι δραστηριότητες που έχουν ενσωματωθεί στο ίδιο το περιβάλλον έχουν ως αποτέλεσμα οι μαθητές να οδηγούνται γρήγορα σε αδιέξοδο κατά την επίλυση προβλημάτων και να αναζητούν από μόνοι τους ένα μηχανισμό δημιουργίας νέων εντολών (υποπρογραμμάτων). Συγκεκριμένα,

οι βασικές εντολές που απαρτίζουν το ρεπερτόριο εντολών του ρομπότ Karel είναι τόσο περιορισμένες, ώστε οι μαθητές αναγκάζονται ακόμα και για απλά προβλήματα να αναπτύσσουν μακροσκελή προγράμματα. Αυτό έχει ως αποτέλεσμα, όπως προαναφέρθηκε, οι μαθητές να αναζητούν ένα μηχανισμό δημιουργίας νέων εντολών και να κατανοούν ευκολότερα τη σημασία της έννοιας της διαδικασίας όταν αυτή τους παρουσιάζεται.

7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Το διδακτικό σενάριο προβλέπεται να υλοποιηθεί στο εργαστήριο Πληροφορικής.

Απαιτείται το περιβάλλον Karel. Το περιβάλλον δεν απαιτεί κάποια ιδιαίτερη διαδικασία εγκατάστασης. Αποσυμπιέζετε το αρχείο Karel.zip (<http://users.uom.gr/~stelios/Karel.zip>), οπότε δημιουργείται ένας ομώνυμος φάκελος και εκτελείτε το αρχείο Karel.exe που υπάρχει σε αυτόν.

«Προστιθέμενη αξία»

Το περιβάλλον Karel αποτελεί ένα διδακτικό μικρόκοσμο και συνεπώς παρουσιάζει τα γνωστά πλεονεκτήματα που χαρακτηρίζουν γενικά τους προγραμματιστικούς μικρόκοσμους:

- Η γλώσσα προγραμματισμού αποτελείται από ένα περιορισμένο ρεπερτόριο εντολών με απλή σύνταξη και σημασιολογία.
- Βασίζονται σε υπαρκτά μοντέλα που είναι ήδη γνωστά στο μαθητή, μειώνοντας έτσι δραματικά τη διανοητική «απόσταση» ανάμεσα στα νοητά μοντέλα ή την περιγραφή σε φυσική γλώσσα των αλγορίθμων και στην περιγραφή τους στη γλώσσα προγραμματισμού.
- Τα προβλήματα που καλούνται να λύσουν οι μαθητές παρουσιάζουν ιδιαίτερο ενδιαφέρον.
- Η εκτέλεση ενός προγράμματος είναι βηματική και ορατή, αποκαλύπτοντας έτσι τη σημασία των διδασκόμενων δομών, καθώς και τις έννοιες που σχετίζονται με τη δομή και την εκτέλεση των προγραμμάτων.
- Υπάρχει δυνατότητα προσαρμογής του μικρόκοσμου στις ανάγκες του κοινού στο οποίο απευθύνεται.

Επιπλέον, όπως αναφέρθηκε στην ενότητα 5, ο μικρόκοσμος Karel ενσωματώνει:

- το απαραίτητο *διδακτικό υλικό* (σύντομη και περιεκτική θεωρία, εγχειρίδιο χρήσης)
- *δραστηριότητες* για την εξοικείωση των μαθητών με τις έννοιες πριν την εφαρμογή τους στα πλαίσια ανάπτυξης προγραμμάτων
- ένα συντάκτη δομής για την εύκολη ανάπτυξη προγραμμάτων και την αποφυγή επικέντρωσης στις λεπτομέρειες της γλώσσας προγραμματισμού
- τη δυνατότητα της επεξηγηματικής οπτικοποίησης, εμφάνισης δηλαδή μηνυμάτων σε φυσική γλώσσα για την τρέχουσα κάθε φορά εντολή κατά τη βηματική εκτέλεση των προγραμμάτων.

Αντίλογος, επιφυλάξεις, προβλήματα

Εντολές στα Αγγλικά. Υπάρχει δυνατότητα επιλογής εμφάνισης του διδακτικού υλικού που είναι ενσωματωμένο στον μικρόκοσμο Karel, καθώς και πραγματοποίησης του διαλόγου χρήστη-συστήματος είτε στα Αγγλικά είτε στα Ελληνικά. Ωστόσο, για το ρεπερτόριο εντολών χρησιμοποιείται η αγγλική γλώσσα, γεγονός που ενδεχομένως να θεωρηθεί από ορισμένους εκπαιδευτικούς ως πρόβλημα. Ωστόσο:

- το ρεπερτόριο εντολών είναι περιορισμένο
- χρησιμοποιούνται σαφή ονόματα εντολών που φανερώνουν τη σημασία τους και δεν δημιουργούν παρανοήσεις για τη σημασία τους
- οι εντολές δεν χρειάζεται να απομνημονευθούν, αφού η ανάπτυξη των προγραμμάτων γίνεται χρησιμοποιώντας τον συντάκτη δομής.

Έλλειψεις στη γλώσσα προγραμματισμού. Στον μικρόκοσμο Karel το ομόνυμο ρομπότ που εκτελεί τις διάφορες αποστολές μπορεί να εκτελέσει: (1) τις εντολές (διαδικασίες) `move`, `turnLeft`, `pickBeeper` και `putBeeper`, (2) να ελέγξει την κατάστασή του (π.χ. αν έχει beeper στην τσάντα του, αν βλέπει προς μια συγκεκριμένη κατεύθυνση κτλ) και την κατάσταση του περιβάλλοντος του (π.χ. αν υπάρχει τοίχος μπροστά) χρησιμοποιώντας κάποια κατηγορήματα (λογικές συναρτήσεις) και φυσικά την κατάλληλη δομή επιλογής ή επανάληψης (`if`, `if/else`, `while`). Στο σύνολο των βασικών εντολών υπάρχει η εντολή `turnLeft` για στροφή 90 μοιρών προς τα αριστερά, αλλά απουσιάζει μια εντολή, έστω `turnRight`, για στροφή 90 μοιρών προς τα δεξιά. Είναι βέβαια προφανές ότι σε αρκετές περιπτώσεις το ρομπότ Karel χρειάζεται να στρίψει προς τα δεξιά και για να γίνει αυτό οι μαθητές αναγκάζονται να καλούν κάθε φορά τη μέθοδο `turnLeft` 3 φορές. Αυτό με μια πρώτη ματιά μπορεί να θεωρηθεί ως έλλειψη της γλώσσας προγραμματισμού που χρησιμοποιείται, ή

ακόμα και ως πρόβλημα. Ωστόσο, η επιλογή των δημιουργών της γλώσσας να μην συμπεριληφθεί η εντολή `turnRight` στις βασικές εντολές του ρομπότ ήταν απόλυτα συνειδητή. Οι μαθητές αναγνωρίζουν από την αρχή το γεγονός ότι το ρομπότ Karel έχει πολύ περιορισμένες δυνατότητες και αναζητούν τρόπους δημιουργίας νέων εντολών (διαδικασιών). Με αυτό τον τρόπο οι μαθητές κατανοούν πολύ ευκολότερα την ανάγκη ύπαρξης ενός μηχανισμού δημιουργίας νέων εντολών (διαδικασιών). Η «έλλειψη» λοιπόν μιας εντολής `turnRight`, για παράδειγμα, εξυπηρετεί ένα συγκεκριμένο διδακτικό σκοπό και δεν οφείλεται σε μια προβληματική σχεδίαση της γλώσσας.

Συντάκτης δομής. Οι συντάκτες δομής είναι γνωστό ότι στηρίζουν ουσιαστικά τους αρχάριους προγραμματιστές στην ανάπτυξη προγραμμάτων, στην αποφυγή της επικέντρωσης στην εκμάθηση της σύνταξης της γλώσσας προγραμματισμού και των συντακτικών λαθών. Ωστόσο, είναι επίσης γνωστό ότι η διόρθωση ενός προγράμματος, η οποία πολλές φορές συνεπάγεται την προσωρινή παραβίαση των συντακτικών κανόνων, είναι αρκετές φορές πιο δύσκολη χρησιμοποιώντας ένα συντάκτη δομής που απαιτεί ένα πρόγραμμα να παραμένει καθ' όλη τη διάρκεια ανάπτυξής του συντακτικά σωστό. Ο συντάκτης δομής του Karel βέβαια, ακριβώς όπως και ο συντάκτης δομής του `objectKarel`, είναι λιγότερο «αυστηρός» σε σχέση με έναν τυπικό εκδότη δομής, και καθιστά τη διόρθωση των προγραμμάτων εύκολη. Εξάλλου, οι αρχάριοι προγραμματιστές - σε αντίθεση με τους έμπειρους προγραμματιστές - προτιμούν πάντα τη χρήση ενός συντάκτη δομής για την ανάπτυξη των προγραμμάτων τους, ακόμα και με τους περιορισμούς που μπορεί να έχει ένας τέτοιος συντάκτης δομής.

Διδασκαλία της έννοιας της διαδικασίας σε αρχικό στάδιο. Όπως ήδη αναφέρθηκε, προτείνεται η διδασκαλία της έννοιας της διαδικασίας να πραγματοποιηθεί στην αρχή των μαθημάτων, μιας και αποτελεί θεμελιώδη έννοια για τον δομημένο-διαδικαστικό προγραμματισμό και επιπλέον η διδασκαλία της καθίσταται σημαντικά ευκολότερη στον μικρόκοσμο του ρομπότ Karel. Επιπλέον, τα πρώτα στοιχεία από την πιλοτική εφαρμογή της προτεινόμενης διδακτικής προσέγγισης σε μαθητές Γυμνασίου ήταν ιδιαίτερα ενθαρρυντικά (Ξυνόγαλος, 2010). Ωστόσο ο εκπαιδευτικός, ανάλογα και με τις ιδιαίτερες συνθήκες μιας συγκεκριμένης διδασκαλίας/τάξης, μπορεί αν επιθυμεί να διδάξει, όπως συνήθως, πρώτα τις βασικές αλγοριθμικές δομές και στη συνέχεια την έννοια της διαδικασίας.

8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

Γενικά...

Η έννοια της διαδικασίας κατέχει κεντρικό ρόλο στον δομημένο-διαδικαστικό προγραμματισμό. Ωστόσο, η διδασκαλία και εκμάθησή της συνοδεύεται από ποικίλες δυσκολίες. Οι δυσκολίες που αφορούν στην έννοια της διαδικασίας μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες:

- *Δυσκολίες που σχετίζονται με τη σύνταξη και τη σημασία της διαδικασίας σε ένα πρόγραμμα:* καθορισμός της λίστας παραμέτρων (πλήθος και τύπος παραμέτρων), επικοινωνία κυρίως προγράμματος και υποπρογραμμάτων, κλήση διαδικασίας από άλλη διαδικασία, στοίβα κλήσεων και γενικά δυσκολίες που αφορούν στη ροή εκτέλεσης ενός προγράμματος με διαδικασίας.
- *Δυσκολίες αξιοποίησης της διαδικασίας ως εργαλείο ανάπτυξης επαναχρησιμοποιήσιμων- αυτόνομων τμημάτων κώδικα & υλοποίησης των επιμέρους τμημάτων επίλυσης ενός αλγορίθμου.*

Η δεύτερη κατηγορία δυσκολιών σχετίζεται με τον μεγάλο βαθμό αφαίρεσης που χαρακτηρίζει την έννοια της διαδικασίας. Οι δυσκολίες εντοπίζονται όχι μόνο στην υλοποίηση των διαδικασιών, αλλά πολύ περισσότερο στον συνδυασμό τους για την επίλυση ενός δεδομένου προβλήματος. Θεωρώντας τη διαδικασία ως “κομμάτι” ενός προγράμματος, αξίζει να αναφερθούμε σε μια μελέτη των Sprohner και Soloway (1986), τα βασικά σημεία της οποίας αναλύονται στο (Ξυνόγαλος, 2002) και παρατίθενται στη συνέχεια:

Οι Sprohner και Soloway διεξήγαγαν μια εμπειρική μελέτη (Sprohner & Soloway, 1986) προκειμένου να διαπιστώσουν την ορθότητα των παρακάτω ευρέως αποδεκτών υποθέσεων σχετικά με τα λάθη των αρχάριων προγραμματιστών:

- (1) Λίγοι τύποι λαθών ευθύνονται για την πλειονότητα των λαθών στα προγράμματα των σπουδαστών.
- (2) Τα περισσότερα λάθη οφείλονται σε λανθασμένες αντιλήψεις των σπουδαστών για τις δομές της γλώσσας προγραμματισμού.

Η μελέτη διεξήχθη το 1984 στο Yale University, στο πλαίσιο ενός μαθήματος εισαγωγής στον προγραμματισμό με Pascal. Από τις 10 εργασίες επιλέχθηκαν τρεις για περαιτέρω

μελέτη, από τις οποίες η πρώτη απαιτούσε τη χρήση (για πρώτη φορά) φωλιασμένων IF-THEN-ELSE δομών, η δεύτερη βρόχων και η τρίτη διαδικασιών. Μιας και σκοπός των ερευνητών ήταν η μελέτη μη συντακτικών λαθών επιλέχθηκαν 158 συντακτικά σωστά προγράμματα. Συγκεκριμένα, επιλέχθηκε η πρώτη συντακτικά σωστή έκδοση κάθε προγράμματος, αφού αυτή περιέχει κατά κανόνα τα περισσότερα μη συντακτικά λάθη\σφάλματα σε σχέση με τις μεταγενέστερες μερικώς αποσφαλματωμένες εκδόσεις. Για την ανάλυση των δεδομένων χρησιμοποιήθηκε η μέθοδος Goal/Plan Analysis των Sprohrer, Soloway και Porpe.

Σύμφωνα με τα αποτελέσματα της εμπειρικής μελέτης η πρώτη υπόθεση είναι αληθής σε αντίθεση με τη δεύτερη. Συγκεκριμένα, έγινε φανερό ότι οι λανθασμένες αντιλήψεις σχετικά με τις δομές της γλώσσας προγραμματισμού δεν είναι τόσο ευρέως διαδεδομένες ή προβληματικές όσο γενικά πιστεύεται. Τα περισσότερα λάθη είναι αποτέλεσμα των δυσκολιών συνδυασμού σχεδίων ή αλλιώς των «κομματιών» ενός προγράμματος. Στον Πίνακα 3 φαίνονται συνοπτικά τα αποτελέσματα της μελέτης. Περισσότερα στοιχεία για την ανάλυση και τα αποτελέσματα υπάρχουν στο (Sprohrer & Soloway, 1986).

Πρόβλημα	Λάθη που δεν οφείλονται στις δομές	Λάθη που ίσως οφείλονται στις δομές	Λάθη που οφείλονται στις δομές
If-Then-Else (N=28)	19 (68%)	9 (32%)	0 (0%)
Βρόχοι (N=46)	20 (43%)	21 (46%)	5 (11%)
Διαδικασίες (N=27)	13 (48%)	10 (37%)	4 (15%)
Σύνολο (N=101)	52 (52%)	40 (39%)	9 (9%)

Πίνακας 3. Τα αποτελέσματα του πειράματος.

Με βάση τα αποτελέσματα των μελετών οι παραπάνω ερευνητές κάνουν μια ταξινόμηση των λαθών των αρχάριων προγραμματιστών. Όπως και οι ίδιοι επισημαίνουν οι κατηγορίες που θα αναφερθούν στη συνέχεια δεν είναι ούτε αμοιβαία αποκλειόμενες ούτε και πλήρεις.

Προβλήματα που οφείλονται στις δομές της γλώσσας προγραμματισμού (construct-based problems). Πρόκειται για προβλήματα που δυσκολεύουν τους αρχάριους προγραμματιστές

στην εκμάθηση της σωστής σημασιολογίας των δομών της γλώσσας προγραμματισμού:

- *Πρόβλημα φυσικής γλώσσας (natural-language problem):* πολλές δομές των γλωσσών προγραμματισμού έχουν ονόματα που σχετίζονται με λέξεις της φυσικής γλώσσας δημιουργώντας έτσι προβλήματα κατανόησης των δομών ή/και λανθασμένες αντιλήψεις.
- *Πρόβλημα ανθρώπινης μετάφρασης (Human interpreter problem):* οι αρχάριοι προγραμματιστές πολλές φορές υποθέτουν ότι ο υπολογιστής θα μεταφράσει τις δομές με τον ίδιο τρόπο που τις μεταφράζουν και οι ίδιοι.
- *Πρόβλημα μη συνοχής (inconsistency problem):* αρκετές φορές οι αρχάριοι προγραμματιστές αφού καταλάβουν πώς «δουλεύει» μια δομή σε μια συγκεκριμένη κατάσταση, υποθέτουν ότι η δομή θα «δουλέψει» με τον ίδιο τρόπο και σε μια άλλη παρόμοια κατάσταση.

Προβλήματα συνδυασμού σχεδίων (plan composition problems). Πρόκειται για προβλήματα που δυσκολεύουν τους αρχάριους προγραμματιστές στο σωστό συνδυασμό σχεδίων:

- *Πρόβλημα περίληψης (summarization problem):* περίληψη πολύπλοκων συνδυασμών σχεδίων σε μια κύρια συνάρτηση, παραβλέποντας τυχόν αλληλεξαρτήσεις μεταξύ των δευτερευουσών συναρτήσεων.
- *Πρόβλημα βελτιστοποίησης (optimization problem):* οι αρχάριοι προγραμματιστές παρασύρονται τόσο από την επιθυμία τους να βελτιστοποιήσουν τα σχέδια τους ώστε δεν ελέγχουν επαρκώς αν η βελτιστοποίηση μπορεί όντως να πραγματοποιηθεί.
- *Πρόβλημα προηγούμενης εμπειρίας (previous-experience problem):* η ανάπτυξη, ανακύκλωση και προσαρμογή σχεδίων βασίζεται πολύ συχνά σε προϋπάρχουσες εμπειρίες. Συχνά, ακατάλληλες απόψεις ήδη αναπτυγμένων και χρησιμοποιημένων σχεδίων «παραβιάζουν» ή αλλιώς «μολύνουν» ένα σχετικό σχέδιο που χρησιμοποιείται σε μια νέα κατάσταση. Αυτό το πρόβλημα είναι γνωστό και ως plan pollution problem.
- *Πρόβλημα προσαρμογής (specification problem):* οι αρχάριοι προγραμματιστές αναπτύσσουν αφηρημένα σχέδια που πρέπει να προσαρμοστούν σε μια δεδομένη νέα περίπτωση. Συχνά, ένα τέτοιο σχέδιο είναι ακατάλληλο ή λανθασμένο για την περίπτωση.
- *Πρόβλημα φυσικής γλώσσας (natural-language problem):* οι αρχάριοι συνηθίζουν να

αντιστοιχούν σχέδια που κατέχουν ή δημιουργούν σε φυσική γλώσσα σε μια γλώσσα προγραμματισμού. Αυτή η αντιστοίχιση τις περισσότερες φορές οδηγεί σε λάθη.

- *Πρόβλημα μετάφρασης (interpretation problem)*: οι αρχάριοι βασίζονται εν μέρη στις υπάρχουσες γνώσεις για τους στόχους και τα σχέδια όταν «μεταφράζουν» μια εργασία, όταν μεταφράζουν δηλαδή τον ορισμό του προς επίλυση προβλήματος. Απαιτήσεις που δεν εκφράζονται «ξεκάθαρα» αγνοούνται ή λαμβάνονται υπόψη μόνο όταν σχέδια για την επίτευξή τους μπορούν εύκολα να ανακληθούν.
- *Πρόβλημα απρόσμενων περιπτώσεων (unexpected cases problem)*: οι αρχάριοι συχνά γράφουν προγράμματα που «δουλεύουν» σωστά για συγκεκριμένες συνηθισμένες περιπτώσεις, αλλά όχι για όλες τις περιπτώσεις.
- *Πρόβλημα οριακών περιπτώσεων (Boundary problem)*: οι αρχάριοι αντιμετωπίζουν δυσκολίες στην διαχείριση των οριακών σημείων κατά την εξειδίκευση των σχεδίων τους.
- *Πρόβλημα μνημονικού φορτίου (cognitive load problem)*: οι αρχάριοι συχνά δεν αντιλαμβάνονται την απώλεια πληροφοριών από τη μνήμη τους (working memory). Πολλές από τις πληροφορίες αυτές αντιπροσωπεύουν μικρά αλλά σημαντικά τμήματα σχεδίων ή αλληλεπιδράσεις μεταξύ σχεδίων, οι οποίες παραβλέπονται.

Ειδικά στον μικρόκοσμο του ρομπότ Karel...

Ο μικρόκοσμος του ρομπότ Karel, όπως τα περισσότερα εκπαιδευτικά περιβάλλοντα, έχει σχεδιαστεί με στόχο την εξάλειψη ή τουλάχιστον την ευκολότερη αντιμετώπιση των δυσκολιών/παρανοήσεων που έχουν καταγραφεί στη βιβλιογραφία. Η μη χρήση λίστας παραμέτρων στις διαδικασίες που αναπτύσσονται στον μικρόκοσμο Karel, περιορίζει σημαντικά τη σχετική πολυπλοκότητα και βοηθάει τους μαθητές να κατανοήσουν ευκολότερα τον βασικό ρόλο των διαδικασιών. Επιπλέον, η βηματική εκτέλεση των προγραμμάτων με ταυτόχρονη οπτικοποίηση των αποτελεσμάτων στον μικρόκοσμο και την επεξήγηση τους σε φυσική γλώσσα (επεξηγηματική οπτικοποίηση) στηρίζουν ουσιαστικά τους μαθητές στην κατανόηση της ροής εκτέλεσης προγραμμάτων που περιλαμβάνουν διαδικασίες και στην αποφυγή των σχετικών παρανοήσεων.

Επιπλέον, η φύση των προβλημάτων που ανατίθενται στους μαθητές στο πλαίσιο του μικρόκοσμου ευνοεί τη χρήση διαδικασιών και αναδεικνύει τη σημασία και την αποτελεσματικότητα χρήσης τους στην ανάπτυξη προγραμμάτων.

Αποτελέσματα από την χρήση του μικρόκοσμου Karel σε μαθητές της Γ' τάξης του

Γυμνασίου, ειδικά για την έννοια της διαδικασίας, παρουσιάζονται στην εργασία (Ξυνόγαλος, 2010).

9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ

Μετά από έναν αρχικό χρόνο για εξοικείωση των μαθητών με το περιβάλλον KAREL και την ιδέα του «ρομπότ», αναμένεται οι μαθητές να εργαστούν χωρίς προβλήματα – ίσως μάλιστα να το κάνουν και με ευχαρίστηση δεδομένης της έλξης που μπορεί να ασκεί το περιβάλλον εξαιτίας των «παιγνιδιών» του χαρακτηριστικών.

Δεν αναμένεται πολύς χρόνος για την εξοικείωση των μαθητών με το περιβάλλον, αφού είναι σχετικά απλό, ούτε και με τη γλώσσα προγραμματισμού, η οποία, αν και στα Αγγλικά, αναμένεται να γίνει εύκολα κατανοητή.

Το περιβάλλον είναι μάλλον εύρωστο (robust) και δοκιμασμένο, άρα δεν αναμένονται «καταρρεύσεις» λόγω σφαλμάτων λογισμικού, ασυμβατότητας με το λειτουργικό σύστημα κλπ. Άρα, συνολικά, δεν αναμένεται διδακτικός θόρυβος πιο ισχυρός από τα συνήθη περιβάλλοντα.

10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Pattis, R. E., Roberts, J. & Stehlik, M. (1995) *Karel - The Robot, A Gentle Introduction to the Art of Programming*. 2nd edn. Wiley, New York.

Spohrer, J. C. & Soloway, E. (1986) Novice Mistakes: Are the Folk Wisdoms Correct? *Communications of the ACM*, Vol. 29, No. 7, pp. 624-632.

Xinogalos, S. (2011), Teaching Programming to Secondary Education Students with a Learning Environment Based on “Karel the Robot”: A Pilot Study in a Greek High School, *Chapter in “Horizons in Computer Science Research”, Vol. 2*, Thomas S. Clary (Ed.), New York: Nova Science, 67-92.

Κοτίνη, Ι. (2012), Δομημένος Προγραμματισμός με το Ρομπότ Karel – Δομή Επιλογής, *Πρακτικά Ημερίδας Πληροφορικής “Η πληροφορική στην εποχή του Νέου Σχολείου”, ΠΑΚΕ Κ. Μακεδονίας, 27 Μαρτίου 2012, Θεσσαλονίκη.*

Ξυνόγαλος, Σ. (2002), «Εκπαιδευτική Τεχνολογία»: Ένας Διδακτικός Μικρόκοσμος για την Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό, *Διδακτορική Διατριβή, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας, σελ. 569.*

Ξυνόγαλος, Σ. (2003), Σενάρια Διδασκαλίας του Προγραμματισμού στη Δευτεροβάθμια Εκπαίδευση, *Πρακτικά του 2^{ου} Πανελληνίου Συνεδρίου των Εκπαιδευτικών για τις ΤΠΕ «Αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στη Διδακτική Πράξη»*, Σύρος, 9-11 Μαΐου 2003, Α' τόμος, 783-795.

Ξυνόγαλος, Σ. (2009), Πρόταση για τη Διδασκαλία του Προγραμματισμού στο Γυμνάσιο με Χρήση του Ρομπότ Karel, *Πρακτικά του 5^{ου} Πανελληνίου Συνεδρίου των Εκπαιδευτικών για τις ΤΠΕ «Αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στη Διδακτική Πράξη»*, Σύρος, 953-963.

Ξυνόγαλος, Σ. (2010), Η Διδασκαλία της Έννοιας της Διαδικασίας με Χρήση του Ρομπότ Karel σε Μαθητές Γυμνασίου: μια μελέτη περίπτωσης, *Πρακτικά 5^{ου} Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής»*, Αθήνα 9-11 Απριλίου, 105-114.

11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Οι προτεινόμενες δραστηριότητες για τους μαθητές εγγράφονται σε μια τυπική κονστрукτιβιστική προσέγγιση: στους μαθητές προτείνονται μια σειρά από προβλήματα τα οποία καλούνται να επιλύσουν στο περιβάλλον KAREL. Οι μαθητές, μπορούν, εργαζόμενοι σε ομάδες, να διερευνήσουν τις δυνατότητες του περιβάλλοντος και να τις συνδυάσουν για την επίλυση των προβλημάτων.

Η επίλυση των προβλημάτων που προτείνονται εξάλλου, καθιστά εμφανή τη λειτουργία και αξία των χρησιμοποιούμενων μεθόδων, εννοιών, τεχνικών κλπ.

Η επιλογή των καταλληλότερων μεθόδων για την επίλυση των προβλημάτων μπορεί να προκύψει μέσα από κοινωνιογνωστικές συγκρούσεις, εφόσον οι μαθητές εργάζονται σε ομάδες.

12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ

Το περιβάλλον KAREL, αποτελεί ένα μικρόκοσμο που προσομοιώνει ένα υποθετικό ρομπότ με σχετικά περιορισμένο ρεπερτόριο δυνατοτήτων.

Ωστόσο, βασικές έννοιες του προγραμματισμού (όπως το πεπερασμένο των βημάτων των διαδικασιών, η διαδοχή των εντολών, οι επαναληπτικές δομές και οι δομές επιλογής) αποκτούν ένα συγκεκριμένο νόημα στο πλαίσιο του περιβάλλοντος: αλλαγές στη σειρά εκτέλεσης μπορεί να προκαλέσουν μη-επιθυμητή «συμπεριφορά» του ρομπότ* επίσης η δομή επιλογής προβάλλει ξεκάθαρα ως μία επιλογή εξοικονόμησης εντολών.

Το συγκεκριμένο περιβάλλον μπορεί να νοηματοδοτήσει με ορθό τρόπο αυτά τα βασικά χαρακτηριστικά των προγραμμάτων (και των αλγορίθμων) και η σχετική γνώση μπορεί να είναι διαθέσιμη και εφαρμόσιμη, από την πλευρά των μαθητών και σε άλλα προγραμματιστικά περιβάλλοντα.

13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Προτείνεται οι μαθητές να οργανωθούν σε ομάδες των 2-3 (το πολύ) ατόμων.

Στους υπολογιστές πρέπει να είναι εγκατεστημένο το λογισμικό Karel (<http://users.uom.gr/~stelios/Karel.zip>). Ένας βιντεοπροβολέας θα ήταν χρήσιμος, χωρίς όμως να είναι απαραίτητος. Τα φύλλα εργασίας θα πρέπει να μοιραστούν στους μαθητές τόσο για την αποτελεσματικότερη διεξαγωγή των δραστηριοτήτων στην τάξη, όσο και για τη στήριξη των μαθητών στη μελέτη στο σπίτι.

14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Ο μικρόκοσμος του ρομπότ Karel μπορεί να χρησιμοποιηθεί, όπως έχει ήδη αναφερθεί, για τη διδασκαλία των δομών επιλογής και επανάληψης. Σχετικό υλικό (θεωρία και δραστηριότητες) υπάρχει ενσωματωμένο στην περιοχή των μαθημάτων του μικρόκοσμου.

Επίσης, οι εισαγωγικές δραστηριότητες που παρουσιάζονται μπορούν να εμπλουτιστούν κατάλληλα ανάλογα με την ηλικία των μαθητών, την προηγούμενη εμπειρία τους και των διαθέσιμο χρόνο. Ενδεικτικά, αναφέρονται οι παρακάτω επεκτάσεις:

Ο εκπαιδευτικός μπορεί να δώσει στους μαθητές τον κώδικα ενός προγράμματος (με τις γραμμές αριθμημένες) και να τους ζητήσει να τον εκτελέσουν στο χαρτί, προσδιορίζοντας το αποτέλεσμα της κάθε εντολής στην κατάσταση του κόσμου και τις μεταβολές στις μεταβλητές του ρομπότ Karel.

	<i>Μεταβολές τιμών μεταβλητών</i>			
<i>Εκτέλεση εντολής</i>	<i>Δρόμος</i>	<i>Λεωφόρος</i>	<i>Κατεύθυνση</i>	<i>Αριθμός beepers στην τσάντα</i>

Το πρόγραμμα που θα δοθεί μπορεί να περιλαμβάνει μόνο τις βασικές εντολές του Karel, αλλά μπορεί να περιλαμβάνει και τον ορισμό νέων εντολών (διαδικασιών). Η δραστηριότητα αυτή θα βοηθήσει τον εκπαιδευτικό να διαγνώσει τυχόν δυσκολίες που αφορούν στη ροή εκτέλεσης ενός προγράμματος με χρήση νέων εντολών. Επίσης, στο πλαίσιο της δραστηριότητας αυτής μπορεί να ζητηθεί από τους μαθητές να προσδιορίσουν τις μεταβολές στις μεταβλητές του ρομπότ Karel που προκαλεί η εκτέλεση κάθε αριθμημένης εντολής.

Επίσης, μπορεί να δοθεί η *εκφώνηση ενός προβλήματος και το πρόγραμμα σε χαρτί και να ζητηθεί από τους μαθητές να εντοπίσουν τα λάθη, να αναφέρουν τον τύπο τους και να τα διορθώσουν*. Αφού οι μαθητές γράψουν τις απαντήσεις τους σε χαρτί, ακολουθεί συζήτηση στην τάξη σχετικά με τις προτεινόμενες από τους μαθητές διορθώσεις. Στη συνέχεια, μπορεί να δοθεί στους μαθητές το αρχείο του προγράμματος, προκειμένου να επιβεβαιώσουν τις απαντήσεις τους. Οι μαθητές αποσφαλτώνουν το πρόγραμμα με τη χρήση του υπολογιστή και σημειώνουν στο χαρτί τα λάθη τους.

15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

Ανάλυση φύλλου εργασίας 1

Βήμα 1^ο - Δραστηριότητα 1: εκκίνηση του περιβάλλοντος. Το περιβάλλον δεν απαιτεί κάποια ιδιαίτερη διαδικασία εγκατάστασης. Αποσυμπιέζετε το αρχείο Karel.zip (<http://users.uom.gr/~stelios/Karel.zip>) και εκτελείτε το αρχείο Karel.exe.

Βήμα 2^ο: Παρουσιάζεται ο προγραμματιστικός μικρόκοσμος του ρομπότ Karel χρησιμοποιώντας τη θεωρία και τη δραστηριότητα της ενότητας «Εισαγωγή». Γίνεται αναφορά στην έννοια των εντολών και της γλώσσα προγραμματισμού.

Βήμα 3^ο - Δραστηριότητα 2: Επεξηγούνται οι 4 βασικές εντολές (move(), turnLeft(), putBeeper(), pickBeeper()) τις οποίες αναγνωρίζει και εκτελεί το ρομπότ Karel (θεωρία ενότητας «Βασικές εντολές»). Η εντολή turnOff(), χρειάζεται – όπως αναφέρεται και στη θεωρία – σε ελάχιστες περιπτώσεις και δεν χρειάζεται να δοθεί ιδιαίτερη βαρύτητα.

Βήμα 4^ο: Οι μαθητές κατευθύνουν το ρομπότ Karel στην εκτέλεση των απαραίτητων εντολών για την εκτέλεση μιας αποστολής πατώντας κουμπιά με τα ονόματα των εντολών, παρατηρούν το αποτέλεσμα στην κατάσταση του κόσμου και στο ίδιο το ρομπότ και τη σύνταξη των εντολών στη γλώσσα προγραμματισμού (δραστηριότητα ενότητας «Βασικές

εντολές»). Επισημαίνεται ότι το ρομπότ Karel διατηρεί σε ειδικές μεταβλητές τις απαραίτητες πληροφορίες για τη θέση, την κατεύθυνση και τον αριθμό των beepers που έχει στην τσάντα του και ζητείται από τους μαθητές να προσδιορίσουν ποια ή ποιες μεταβλητές επηρεάζονται από την εκτέλεση κάθε μιας από τις διαθέσιμες βασικές εντολές.

Δραστηριότητα 3: στο πλαίσιο της δραστηριότητας αυτής οι μαθητές εξοικειώνονται με τον κύκλο ανάπτυξης ενός προγράμματος, τις λειτουργίες του προγραμματιστικού περιβάλλοντος και την ανάπτυξη απλών προγραμμάτων που βασίζονται στη δομή ακολουθίας.

Στη συγκεκριμένη δραστηριότητα προτείνεται μία απλή άσκηση, η οποία βέβαια μπορεί να αντικατασταθεί από οποιαδήποτε άσκηση πιστεύει ο εκπαιδευτικός ότι μπορεί να προκαλέσει το ενδιαφέρον των μαθητών. Θα πρέπει βέβαια ο εκπαιδευτικός να φροντίσει η λύση της πρώτης άσκησης να περιλαμβάνει και τις τέσσερις βασικές εντολές που αναγνωρίζει το ρομπότ Karel.

Αρχικά, δίνονται στους μαθητές συνοπτικές οδηγίες σχετικά με τη διαδικασία ανάπτυξης, μεταγλώττισης και εκτέλεσης ενός προγράμματος με την οποία θα πρέπει να εξοικειωθούν οι μαθητές. Επισημαίνεται στους μαθητές ότι στο εγχειρίδιο του προγραμματιστικού περιβάλλοντος (που τους δίνεται προαιρετικά) υπάρχουν πιο αναλυτικές οδηγίες για κάθε βήμα της συγκεκριμένης διαδικασίας, στις οποίες οι μαθητές μπορούν καταφύγουν για την επίλυση αποριών κατά την ανάπτυξη προγραμμάτων στο σπίτι. Επίσης, ενημερώνονται οι μαθητές ότι στο περιβάλλον υπάρχουν αναλυτικές οδηγίες για όλες τις λειτουργίες που υποστηρίζει (Βοήθεια ► Περιεχόμενα).

Όταν η εισαγωγή στον διαδικαστικό προγραμματισμό πραγματοποιείται με μια συμβατική γλώσσα, η διαδικασία ανάπτυξης προγραμμάτων περιγράφεται στους μαθητές ως μια διαδικασία τριών φάσεων:

1^η φάση: είσοδος δεδομένων

2^η φάση: επεξεργασία

3^η φάση: έξοδος αποτελεσμάτων

Στο πλαίσιο της 1^{ης} φάσης οι μαθητές καλούνται να μελετήσουν τον προσδιορισμό του προβλήματος, να καθορίσουν ποια είναι τα δεδομένα εισόδου, να δηλώσουν τις απαραίτητες μεταβλητές και να διαβάσουν σε αυτές τιμές. Στον μικρόκοσμο του Karel, η φάση αυτή φαίνεται να απουσιάζει μιας και η γλώσσα του Karel δεν υποστηρίζει τη δήλωση μεταβλητών. Όπως αναφέρθηκε ήδη όμως, γίνεται αναφορά στην έννοια της μεταβλητής: το ρομπότ Karel κρατάει τις απαραίτητες πληροφορίες (θέση, κατεύθυνση, αριθμός

beepers) σε ειδικές θέσεις στη μνήμη που διαθέτει, οι οποίες ονομάζονται μεταβλητές. Οι μεταβλητές του ρομπότ Karel αλλάζουν τιμή (μεταβάλλονται), καθώς ο Karel εκτελεί διάφορες εντολές. Επίσης, ο Karel καλείται κάθε φορά να φέρει εις πέρας μια αποστολή θεωρώντας ως δεδομένη μια συγκεκριμένη αρχική κατάσταση του κόσμου. Στον μικρόκοσμο του Karel λοιπόν, η φάση της εισόδου περιλαμβάνει τον καθορισμό από τους μαθητές της κατάλληλης αρχικής κατάστασης του Karel και της κατάστασης του κόσμου όπου θα εκτελέσει την αποστολή του.

Στο πλαίσιο της 3^{ης} φάσης χρησιμοποιώντας μια συμβατική γλώσσα οι μαθητές πρέπει να χρησιμοποιήσουν τις κατάλληλες εντολές για την εμφάνιση των αποτελεσμάτων που προέκυψαν από την επεξεργασία των δεδομένων. Στον μικρόκοσμο του Karel, τα αποτελέσματα του προγράμματος στον ίδιο τον Karel και στον κόσμο του είναι άμεσα ορατά. Ωστόσο, ο εκπαιδευτικός θα πρέπει να επισημάνει στους μαθητές ότι θα πρέπει τα αποτελέσματα αυτά να ελέγχονται. Το γεγονός ότι ένα πρόγραμμα εκτελείται χωρίς να προκληθεί λάθος εκτέλεσης, δεν σημαίνει απαραίτητα ότι δίνει σωστά αποτελέσματα.

Μετά την ολοκλήρωση του προγράμματος, ο εκπαιδευτικός μπορεί να ζητήσει από τους μαθητές να κάνουν συγκεκριμένες αλλαγές στο πρόγραμμα, έτσι ώστε να προκληθούν συντακτικά λάθη, λογικά λάθη και λάθη εκτέλεσης και να αποσαφηνιστεί η αιτία που προκαλεί κάθε ένας από αυτούς τους τύπους λαθών και ο τρόπος με τον οποίο αντιμετωπίζονται.

Δραστηριότητες 4-9: οι δραστηριότητες αυτές ασκήσεις που μπορούν να χρησιμοποιηθούν για τη διδασκαλία και εμπέδωση του γνωστικού αντικείμενου, αλλά και στην προετοιμασία φύλλων αξιολόγησης. Ανάλογα με τις διδακτικές ώρες που θα αποφασίσει να αφιερώσει ο εκπαιδευτικός και το βάθος στο οποίο θα αναλύσει τις σχετικές έννοιες, οι προτεινόμενες δραστηριότητες μπορούν να εμπλουτιστούν, όπως και στις προτεινόμενες επεκτάσεις.

Ανάλυση φύλλου εργασίας 2

Στο πλαίσιο των δραστηριοτήτων του 2^{ου} φύλλου εργασίας οι μαθητές εξοικειώνονται με την έννοια και τα πλεονεκτήματα της δημιουργίας νέων εντολών, καθώς και με τον τρόπο υλοποίησής τους στο περιβάλλον του Karel. Αφού γίνει μια σύντομη σύνδεση με το προηγούμενο μάθημα, εκπονούνται οι δραστηριότητες του φύλλου εργασίας. Ο εκπαιδευτικός συντονίζει τη διεξαγωγή των δραστηριοτήτων και επισημαίνει ότι το φύλλο εργασίας μπορεί να χρησιμοποιηθεί από τους μαθητές για τη μελέτη στο σπίτι. Στη συνέχεια, αναλύεται σύντομα η σημασία των βημάτων που πρέπει να ακολουθηθούν για

την εκπόνηση των δραστηριοτήτων του φύλλου εργασίας και την επίτευξη των στόχων του σεναρίου.

Βήμα 1^ο – Δραστηριότητα 1: Χρησιμοποιείται το παράδειγμα του ρομπότ-ταξιδιώτη (των αποστολών μεγάλων αποστάσεων) της ενότητας «Συναρτήσεις» των μαθημάτων του προγραμματιστικού περιβάλλοντος. Στόχος του παραδείγματος αυτού είναι να κατανοήσουν οι μαθητές ότι πολλές φορές, ακόμη και για απλά προβλήματα, απαιτείται η ανάπτυξη προγραμμάτων με πολύ μεγάλη έκταση, με αποτέλεσμα να είναι δύσκολη η κατανόηση, η αποσφαλμάτωση και η τροποποίησή τους για την επίλυση παρόμοιων προβλημάτων. Ακολουθεί συζήτηση στην τάξη προκειμένου να εντοπιστούν οι λόγοι για τους οποίους για ένα τόσο απλό πρόβλημα απαιτείται η ανάπτυξη ενός τόσο μεγάλου προγράμματος. Ο εκπαιδευτικός, εφόσον αυτό δεν προκύψει από την συζήτηση στην τάξη, επισημαίνει ότι το πρόβλημα που αναφέρθηκε για το παράδειγμα του ρομπότ-ταξιδιώτη έγκειται στο γεγονός ότι οι λύσεις μας πρέπει να μεταφράζονται στις στοιχειώδεις εντολές της γλώσσας προγραμματισμού του ρομπότ Karel. Αναφέρεται η δυνατότητα δημιουργίας νέων εντολών που παρέχουν νέες δυνατότητες στον Karel.

Βήμα 2^ο – Δραστηριότητα 2: Ο εκπαιδευτικός, εφόσον αυτό δεν προκύψει από την συζήτηση στην τάξη, επισημαίνει ότι το πρόβλημα που αναφέρθηκε για το παράδειγμα του ρομπότ-ταξιδιώτη έγκειται στο γεγονός ότι οι λύσεις μας πρέπει να μεταφράζονται στις στοιχειώδεις εντολές της γλώσσας προγραμματισμού του ρομπότ Karel. Αναφέρεται η δυνατότητα δημιουργίας νέων εντολών που παρέχουν νέες δυνατότητες στον Karel.

Χρησιμοποιείται το πρόγραμμα του ρομπότ-ταξιδιώτη με τη δήλωση και τον ορισμό της νέας εντολής `moveKlm` (θεωρία της ενότητας «Συναρτήσεις») προκειμένου να εξηγηθούν οι έννοιες της δημιουργίας νέας εντολής (διαδικασίας), του λεξικού και ο τρόπος ορισμού νέων εντολών. Έμφαση δίνεται στο γεγονός ότι μέσα από μια εντολή μπορεί να γίνει κλήση οποιασδήποτε άλλης ήδη γνωστής ή νέας εντολής ορίζεται μέσα στο πρόγραμμα.

Βήμα 3^ο – Δραστηριότητα 3: Παρουσιάζεται το πρόβλημα του «σκουπίσματος στις σκάλες» (δραστηριότητα της ενότητας «Συναρτήσεις») και γίνεται συζήτηση με τους μαθητές για τη λύση του. Οι μαθητές μελετούν και εκτελούν δύο προγράμματα για το παραπάνω πρόβλημα – στο 1^ο χρησιμοποιούνται μόνο οι βασικές εντολές του ρομπότ Karel, ενώ στο 2^ο χρησιμοποιούνται νέες εντολές. Οι μαθητές παροτρύνονται να εκτελέσουν τη δεύτερη έκδοση του προγράμματος βηματικά και να δώσουν ιδιαίτερη προσοχή στη ροή εκτέλεσης του προγράμματος. Ο εκπαιδευτικός κάνει ερωτήσεις στους μαθητές προκειμένου να διαπιστώσει αν κατανοούν:

- το γεγονός ότι μέσα από μια νέα εντολή που ορίζουμε μπορεί να γίνει κλήση άλλων εντολών, είτε αυτές είναι οι βασικές εντολές του ρομπότ Karel είτε νέες εντολές, καθώς επίσης και ποια είναι η ροή εκτέλεσης ενός προγράμματος σε αυτή την περίπτωση
- το γεγονός ότι κάθε νέα εντολή που ορίζουμε μπορεί να κληθεί όσες φορές χρειάζεται (και όχι μόνο μία φορά – *παρανόηση*), καθώς επίσης και ότι οι νέες εντολές που ορίζονται σε ένα πρόγραμμα δεν εκτελούνται με τη σειρά που εμφανίζονται μέσα σε αυτό (*παρανόηση*)
- ορισμένες εντολές έχουν βοηθητικό ρόλο, ορίζονται δηλαδή απλά για να κληθούν μέσα από άλλες εντολές και όχι από το κυρίως πρόγραμμα. Οι βοηθητικές εντολές μπορούν να κληθούν στα σημεία όπου θέλουμε να εκτελεστεί η λειτουργία που αντιπροσωπεύουν και συντελούν στην ανάπτυξη πιο ευανάγνωστου κώδικα και φυσικά στην ανάπτυξη επαναχρησιμοποιήσιμου κώδικα.

Τέλος, οι δύο εκδόσεις του προγράμματος συγκρίνονται και σχολιάζονται με στόχο την ανάδειξη των πλεονεκτημάτων που απορρέουν από τη δημιουργία και χρήση νέων εντολών.

Βήμα 4^ο – Δραστηριότητα 4: Στο πλαίσιο της δραστηριότητας αυτής οι μαθητές καλούνται να τροποποιήσουν ένα πρόγραμμα που αναπτύχθηκε στο μάθημα εξοικείωσης με τις βασικές εντολές που αναγνωρίζει το ρομπότ Karel. Στη συγκεκριμένη περίπτωση χρησιμοποιείται το πρόβλημα της αναρρίχησης στο βουνό και τοποθέτησης της σημαίας.

Ο εκπαιδευτικός ρωτάει τους μαθητές αν αναρωτήθηκαν κατά την ανάπτυξη του προγράμματός τους, ποιες εντολές - εκτός από τις τέσσερις βασικές εντολές που αναγνωρίζει και εκτελεί ο Karel – θα ήταν χρήσιμες σε αυτή την αποστολή. Οι μαθητές συζητάνε στην τάξη και προτείνουν τρόπους βελτίωσης του υπάρχοντος προγράμματος ορίζοντας νέες εντολές. Ο εκπαιδευτικός σημειώνει στον πίνακα τις προτάσεις των μαθητών. Συμπερασματικά, προτείνεται ο ορισμός τριών νέων εντολών τις οποίες υλοποιούν οι μαθητές:

- στροφή δεξιά κατά 90 μοίρες,
- ανέβασμα στο επόμενο επίπεδο του βουνού και
- κατέβασμα στο επόμενο επίπεδο του βουνού.

Στη συνέχεια, οι μαθητές χρησιμοποιώντας τις οδηγίες για τον τρόπο δήλωσης και ορισμού νέων εντολών στο περιβάλλον του Karel και με καθοδήγηση από τον εκπαιδευτικό, αναπτύσσουν το πρόγραμμα. Οι μαθητές εκτελούν βηματικά το πρόγραμμα προκειμένου

να κατανοήσουν τη ροή εκτέλεσης του προγράμματος και να αποφευχθεί η δημιουργία των παρανοήσεων που αφορούν στη ροή εκτέλεσης των προγραμμάτων που χρησιμοποιούν διαδικασίες.

Οι μαθητές σχολιάζουν τα πλεονεκτήματα και μειονεκτήματα των δύο προγραμμάτων που αναπτύχθηκαν για το ίδιο πρόβλημα.

Δραστηριότητες 5-6: μετά την εξοκείωση των μαθητών με την έννοια της διαδικασίας και την ανάπτυξη προγραμμάτων που κάνουν χρήση διαδικασιών, οι μαθητές αναπτύσσουν προγράμματα για την επίλυση απλών προβλημάτων με χρήση διαδικασιών.

Ανάλογα με τις διδακτικές ώρες που θα αποφασίσει να αφιερώσει ο εκπαιδευτικός και το βάθος στο οποίο θα αναλύσει τις σχετικές έννοιες, οι προτεινόμενες δραστηριότητες μπορούν να εμπλουτιστούν, όπως αναφέρεται στην ενότητα 18.

16. ΑΞΙΟΛΟΓΗΣΗ

Για την αξιολόγηση των μαθητών μπορεί να ετοιμαστεί ένα σχετικό φύλλο εργασίας που θα περιλαμβάνει:

- Ερωτήσεις κλειστού τύπου (σωστού/λάθους, πολλαπλής επιλογής κτλ) προκειμένου να ελέγξει ο εκπαιδευτικός αν οι μαθητές κατανόησαν τις σχετικές έννοιες, καθώς επίσης και αν έχουν δημιουργηθεί κάποιες παρανοήσεις. Οι ερωτήσεις και οι απαντήσεις που θα συμπεριληφθούν θα πρέπει να αντιπροσωπεύουν συνήθεις δυσκολίες και παρανοήσεις που έχουν καταγραφεί για τις σχετικές έννοιες.
- Σταυρόλεξα, SUDOKU. Αξίζει να σημειωθεί ότι τα σταυρόλεξα και γενικά τα ερωτήματα κλειστού τύπου λειτουργούν σε ένα συμπεριφοριστικό πλαίσιο (μψηχεβιοριστικό, όπως λέγεται μερικές φορές) και βοηθούν στην απομνημόνευση ή αποτελούν συμβολικές «επιβραβεύσεις». Όσο για τα SUDOKU, αποτελούν δυνατότητα που προσφέρεται σε περιβάλλοντα όπως το Moodle (επομένως μπορούν να χρησιμοποιηθούν στις περιπτώσεις που υπάρχουν ανάλογες δυνατότητες)
- Εκτέλεση έτοιμων προγραμμάτων στο χαρτί για τη διάγνωση δυσκολιών/παρανοήσεων που αφορούν στη ροή εκτέλεσης.
- Ανάπτυξη προγραμμάτων.

17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ

Θα δοθούν 2 φύλλα εργασίας, τα οποία παρατίθενται παρακάτω.

Αναλυτική περιγραφή και ανάλυση των φύλλων εργασίας δίνεται στην ενότητα 15.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - Εξοικείωση με τον μικρόκοσμο του ρομπότ Karel

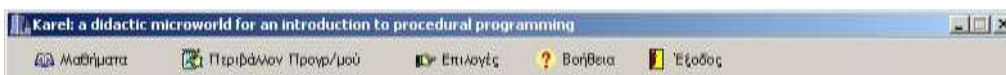


Δραστηριότητα 1 – γνωριμία με το περιβάλλον Karel

- (1) Εκκινήστε το περιβάλλον Karel
- (2) Πατήστε το κουμπί Options και επιλέξτε Language ► Greek (Εικόνα 1), προκειμένου να εμφανίζεται όλο το υλικό και να πραγματοποιείται ο διάλογος χρήστη-συστήματος στα Ελληνικά (Εικόνα 2).



Εικόνα 1. Το βασικό μενού του περιβάλλοντος Karel.



Εικόνα 2. Το βασικό μενού του περιβάλλοντος Karel.

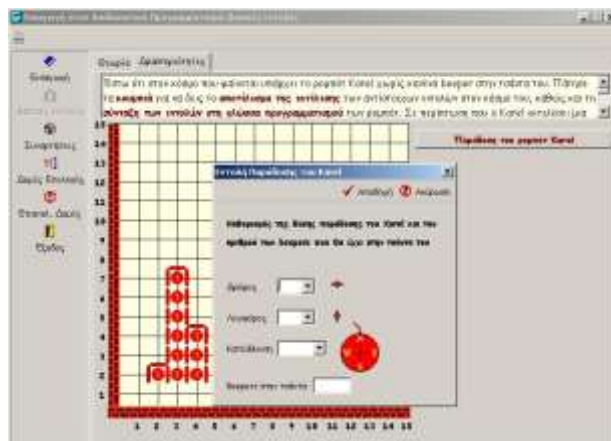
Οι επιλογές του βασικού μενού (Εικόνα 2) είναι οι εξής:

- **Μαθήματα:** σε κάθε ενότητα παρουσιάζεται η σχετική θεωρία και μια ή περισσότερες δραστηριότητες για την εξοικείωση με τις έννοιες της ενότητας.
- **Περιβάλλον Προγραμματισμού:** περιβάλλον για την ανάπτυξη, μεταγλώττιση, αποσφαλμάτωση και εκτέλεση προγραμμάτων.
- **Επιλογές:** επιλογή της γλώσσας (Ελληνική ή Αγγλική) που θα χρησιμοποιείται στο περιβάλλον.
- **Βοήθεια:** παρέχεται βοήθεια για όλες τις λειτουργίες του περιβάλλοντος.
- **Έξοδος:** έξοδος από το περιβάλλον.



Δραστηριότητα 2 – εκτέλεση των βασικών εντολών

- (1) Στον χώρο των μαθημάτων κάντε κλικ στην καρτέλα «Δραστηριότητες» της ενότητας «Βασικές εντολές» και διαβάστε την περιγραφή της (Εικόνα 3).



Εικόνα 3. Η δραστηριότητα της ενότητας «Βασικές εντολές».

- (2) Πατήστε το κουμπί «Παράδοση του ρομπότ Karel» και καθορίστε στη συνέχεια τις αρχικές τιμές των μεταβλητών που προσδιορίζουν τη θέση όπου θα παραδοθεί ο Karel και τον αριθμό των beepers που θα έχει στην τσάντα του. Πατήστε το κουμπί «Αποδοχή» και ο Karel θα εμφανιστεί έτοιμος για δουλειά!
- (3) Καθοδηγήστε το ρομπότ Karel στην εκτέλεση των κατάλληλων εντολών προκειμένου να καθαρίσει τους διαδρόμους από τα σκουπίδια και να τα μεταφέρει όλα στη θέση (1,9).

Παρατηρήσεις:



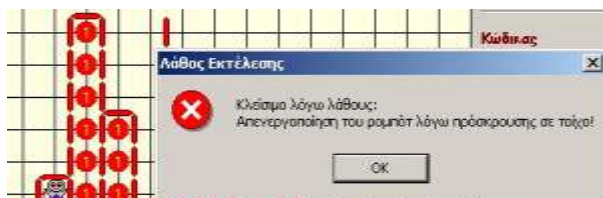
Μπορείτε να καθοδηγήσετε το ρομπότ Karel στην εκτέλεση των κατάλληλων εντολών πατώντας τα κουμπιά με τις ετικέτες `move()`, `turnLeft()`, `pickBeeper()` και `putBeeper()`.



Κάθε φορά που ο Karel εκτελεί μια εντολή, παρατηρήστε ποιο είναι το αποτέλεσμα της στην κατάσταση του κόσμου, αλλά και του ίδιου του ρομπότ. Παρατηρήστε πώς αλλάζουν τιμές οι μεταβλητές στις οποίες αποθηκεύονται οι πληροφορίες για τον Karel. Επίσης, παρατηρήστε ποια είναι η μορφή του προγράμματος στη γλώσσα προγραμματισμού του Karel (στο πλαίσιο Κώδικας).



Προσπαθήστε να κατευθύνετε τον Karel ώστε να ολοκληρώσει την αποστολή του χωρίς να προκληθεί κάποιο λάθος εκτέλεσης (για παράδειγμα, αν ενώ μπροστά από τον Karel υπάρχει τμήμα τοίχου και εσείς στέλνοντας του το μήνυμα `move()` του πείτε να πέσει πάνω στον τοίχο!). Ωστόσο, αν προκύψει κάποιο λάθος εκτέλεσης, διαβάστε το σχετικό μήνυμα που εμφανίζεται και προσπαθήστε να καταλάβετε γιατί προέκυψε το λάθος (Εικόνα 4).



Εικόνα 4. Πλαίσιο ενημέρωσης για λάθος εκτέλεσης.



Αφού ολοκληρώσετε την δραστηριότητα προσπαθήστε να απαντήσετε στις παρακάτω ερωτήσεις:

- (4) Ποιες είναι οι βασικές εντολές που καταλαβαίνει ο Karel, ή αλλιώς οι εντολές που ανήκουν στο λεξιλόγιό του;
- (5) Πώς ανταποκρίνεται ο Karel σε κάθε μία από αυτές τις εντολές;
- (6) Ο Karel ανταποκρίνεται πάντα με τον ίδιο τρόπο σε μια εντολή, έστω στην εντολή `move`;
- (7) Με ποιο τρόπο μπορούμε να κατευθύνουμε τον Karel προκειμένου να φέρει εις πέρας μια αποστολή με επιτυχία; Τι πρέπει να προσέχουμε;
- (8) Υπάρχει περίπτωση αποτυχίας του ρομπότ Karel στην εκτέλεση μιας αποστολής; Δώστε ένα παράδειγμα.

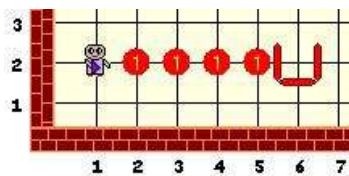


Αν δυσκολεύεστε να απαντήσετε τις παραπάνω ερωτήσεις μελετήστε τη θεωρία των ενοτήτων «Εισαγωγή» και «Βασικές εντολές».

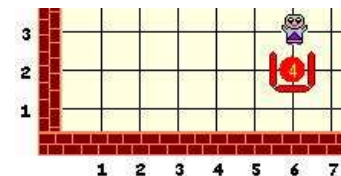


Δραστηριότητα 3 – καθοδηγώντας το ρομπότ Karel στην πρώτη του αποστολή!

Ο Karel ξεκινώντας από τη θέση που φαίνεται στην αρχική κατάσταση της παρακάτω εικόνας πρέπει να μαζέψει τα σκουπίδια και να τα τοποθετήσει μέσα στον κάδο απορριμμάτων. Γράψε ένα πρόγραμμα που θα κατευθύνει τον Karel στην εκτέλεση της αποστολής του. Παρακάτω φαίνεται η αρχική (πριν να ξεκινήσει η εκτέλεση) και η τελική κατάσταση (μετά την ολοκλήρωση της εκτέλεσης του προγράμματος) του κόσμου και του Karel.

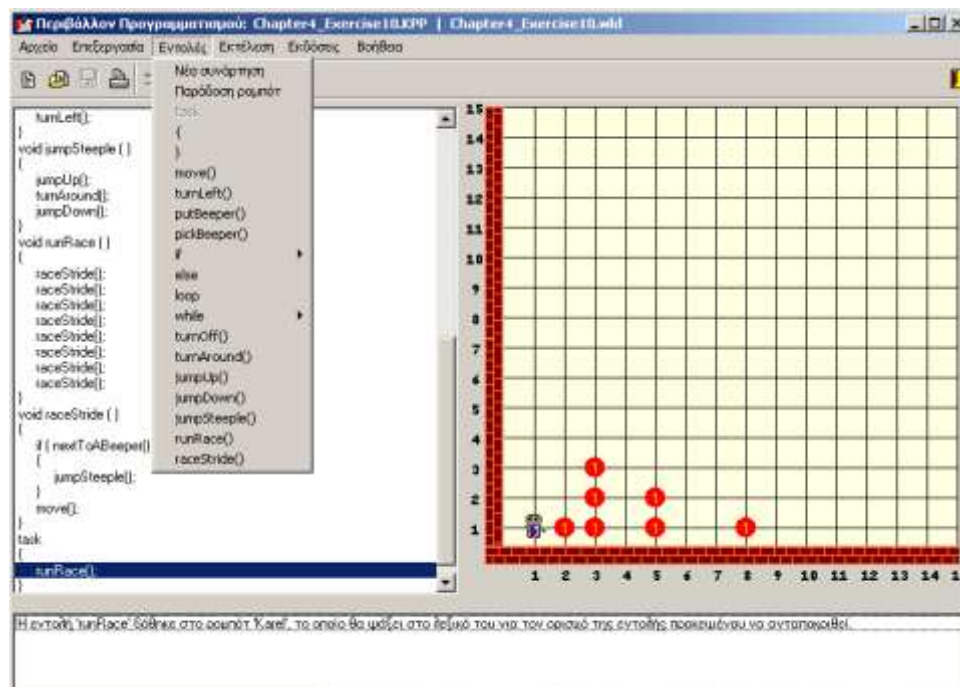


Αρχική κατάσταση του κόσμου.



Τελική κατάσταση του κόσμου.

Για την ανάπτυξη, μεταγλώττιση και εκτέλεση ενός προγράμματος ακολουθούμε τα παρακάτω βήματα, στο περιβάλλον προγραμματισμού:



Το βασικό παράθυρο του Περιβάλλοντος Προγραμματισμού.

Βήμα 1ο	Δημιουργούμε την αρχική κατάσταση του κόσμου
Βήμα 2ο	Ζητάμε την παράδοση του ρομπότ Karel στην κατάλληλη θέση, με την κατάλληλη κατεύθυνση, και τον απαιτούμενο αριθμό beepers στην τσάντα του.
Βήμα 3ο	Αναπτύσσουμε το κυρίως πρόγραμμα , επιλέγοντας από το μενού <i>Εντολές</i> : <ul style="list-style-type: none"> ▪ αρχικά, τη δεσμευμένη λέξη task με την οποία ξεκινάει πάντα το κυρίως πρόγραμμα. ▪ στη συνέχεια, τις εντολές που θέλουμε να εκτελέσει ο Karel ▪ τέλος, το σύμβολο '}' με το οποίο κλείνουμε το μπλοκ εντολών του κυρίως προγράμματος.
Βήμα 4ο	Μεταγλωττίζουμε το πρόγραμμα , επιλέγοντας <i>Μεταγλώττιση</i> από το μενού <i>Εκτέλεση</i> . Σε περίπτωση που η μεταγλώττιση δεν είναι επιτυχής, πρέπει να εντοπίσουμε τα λάθη να τα διορθώσουμε και να μεταγλωττίσουμε ξανά το πρόγραμμα.
Βήμα 5ο	Εκτελούμε βηματικά το πρόγραμμα , επιλέγοντας <i>Εκτέλεση Βήμα προς Βήμα</i> από το μενού <i>Εντολές</i> ή πατώντας το F8 από το πληκτρολόγιο για την εκτέλεση της επόμενης κάθε φορά εντολής. Παρατηρούμε την τρέχουσα εντολή του προγράμματος που εκτελείται, το αποτέλεσμα στον κόσμο του Karel και στο ίδιο το ρομπότ και διαβάζουμε τις επεξηγήσεις στο κάτω μέρος του παραθύρου. Το γεγονός ότι ένα πρόγραμμα μεταγλωττίζεται και εκτελείται χωρίς λάθη εκτέλεσης, δεν σημαίνει απαραίτητα ότι είναι σωστό! Ελέγξτε την τελική κατάσταση του κόσμου και του Karel.
Βήμα 6ο	Αποθηκεύουμε το αρχείο του πηγαίου κώδικα (.kpp) και της κατάστασης του κόσμου (.wld) με το ίδιο όνομα στον προεπιλεγμένο φάκελο projects, επιλέγοντας <i>Αποθήκευση ► Πηγαίου Κώδικα – Κατάστασης του κόσμου</i> από το μενού <i>Αρχείο</i> . Το βήμα αυτό βέβαια μπορεί να είναι και το πρώτο. Δεν πρέπει να περιμένουμε να τελειώσει το πρόγραμμα για να το αποθηκεύσουμε – αν για οποιοδήποτε λόγο κλείσει ο υπολογιστής μας θα χάσουμε τη δουλειά μας!

Αν αντιμετωπίζετε πρόβλημα σε οποιοδήποτε από τα παραπάνω βήματα μελετήστε το εγχειρίδιο χρήσης του περιβάλλοντος.



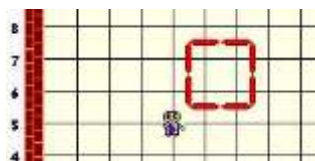
Δραστηριότητα 4

Στις πόλεις μπορούμε να διασχίσουμε περιμετρικά ένα οικοδομικό τετράγωνο επαναλαμβάνοντας τις παρακάτω ενέργειες τέσσερις φορές:

Περπάτησε μέχρι την κοντινότερη γωνία

Στρίψε δεξιά ή αριστερά (το ίδιο κάθε φορά)

Αν οι παραπάνω ενέργειες εφαρμοσθούν σωστά θα επιστρέψουμε στο σημείο που ξεκινήσαμε. Προγραμμάτισε τον Karel να διασχίσει περιμετρικά το τετράγωνο.

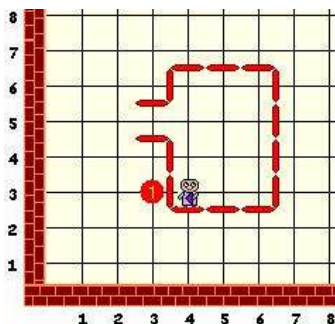


Διάσχιση ενός οικοδομικού
τετραγώνου



Δραστηριότητα 5

Κάθε πρωί ο Karel ξυπνάει στο κρεβάτι του όταν ένα ρομπότ-εφημεριδοπώλης του πετάει την εφημερίδα, που αναπαριστάται με ένα beeper, στην μπροστινή βεράντα του σπιτιού του. Προγραμμάτισε τον Karel ώστε να μαζέψει την εφημερίδα και να την φέρει πίσω στο κρεβάτι του. Η αρχική κατάσταση φαίνεται στην παρακάτω εικόνα, ενώ στην τελική κατάσταση ο Karel πρέπει να βρίσκεται πίσω στο κρεβάτι του (ίδια διασταύρωση & κατεύθυνση) με την εφημερίδα.

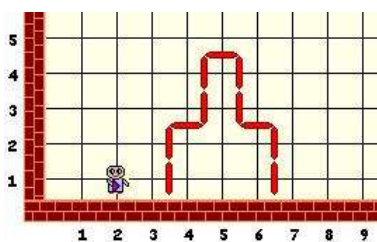


Η μεταφορά της εφημερίδας από τη βεράντα στο κρεβάτι.

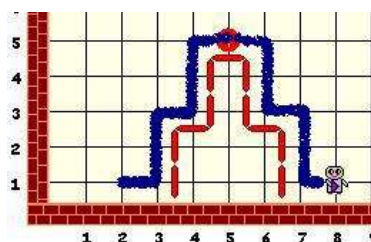


Δραστηριότητα 6

Τα τμήματα τοίχου στην παρακάτω εικόνα αναπαριστούν ένα βουνό. Προγραμμάτισε τον Karel ώστε να ανέβει στην κορυφή του βουνού και να τοποθετήσει μια σημαία, που αναπαριστάται με ένα beeper. Στη συνέχεια ο Karel πρέπει να κατέβει από την άλλη μεριά του βουνού. Να υποθέσεις ότι ο Karel ξεκινάει με μία σημαία-beeper στην τσάντα του και ακολουθεί τη διαδρομή που φαίνεται στην τελική κατάσταση της εικόνας.



Αρχική κατάσταση.



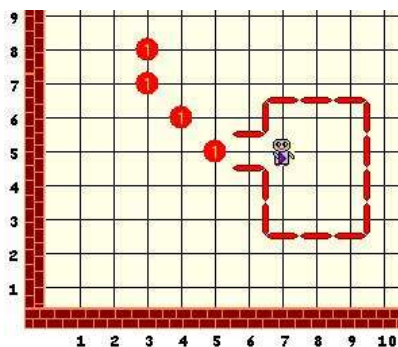
Η τελική κατάσταση και η διαδρομή που ακολουθεί το ρομπότ.

Η αποστολή αναρρίχησης στο βουνό και τοποθέτησης της σημαίας.



Δραστηριότητα 7

Επιστρέφοντας στο σπίτι από το super market, η σακούλα του Karel σκίστηκε και κάποια από τα ψώνια του έπεσαν στο δρόμο. Η αρχική κατάσταση όταν ο Karel ανακάλυψε ότι η σακούλα του έχει σκιστεί φαίνεται στην παρακάτω εικόνα. Προγραμμάτισε τον Karel να συγκεντρώσει όλα τα αντικείμενα (αναπαριστούνται με beepers) που έπεσαν από την σακούλα και να επιστρέψει στην θέση που ήταν όταν ανακάλυψε την διαρροή.

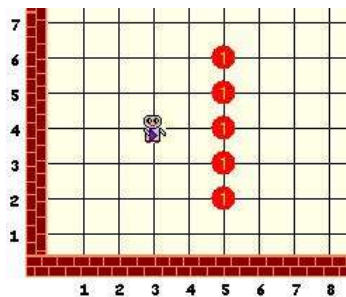


Η αρχική κατάσταση της αποστολής αποκομιδής των χαμένων αντικειμένων.

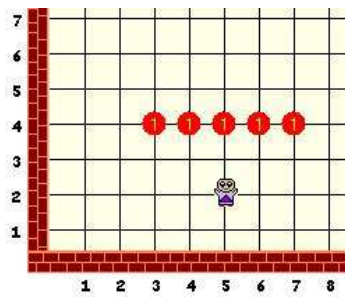


Δραστηριότητα 8

Προγραμματίσε τον Karel να αναδιατάξει τα beepers όπως φαίνεται στην παρακάτω εικόνα.



Αρχική κατάσταση.



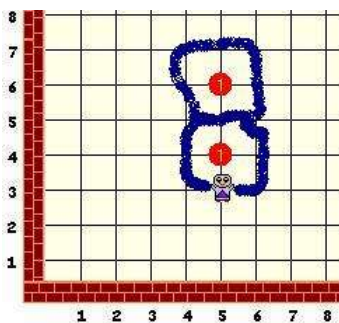
Τελική κατάσταση.

Αναδιάταξη των beepers.



Δραστηριότητα 9

Ο Karel προπονείτε για τους Ολυμπιακούς αγώνες των Ρομπότ. Ένα από τα αγωνίσματα απαιτεί από τους αθλητές να κινούνται γύρω από δύο beepers σχηματίζοντας οχτάρια, όπως φαίνεται στην παρακάτω εικόνα. Γράψε ένα πρόγραμμα που καθοδηγεί τον Karel στην εκτέλεση αυτού του αγωνίσματος όσο το δυνατό γρηγορότερα (εκτελώντας όσο το δυνατό λιγότερες εντολές). Το ρομπότ πρέπει να σταματήσει στο ίδιο σημείο από όπου ξεκίνησε και να βλέπει προς την ίδια κατεύθυνση.



Το αγώνισμα των Ολυμπιακών αγώνων.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 – Δημιουργώντας νέες εντολές για τον Karel



Δραστηριότητα 1 – αξιολόγηση της λύσης για το πρόβλημα των μεγάλων αποστάσεων

Ας υποθέσουμε ότι το ρομπότ Karel ξεκινώντας από την 1^η λεωφόρο και τον 2^ο δρόμο πρέπει να κινηθεί ανατολικά κατά μήκος του 2^{ου} δρόμου για 10 χιλιόμετρα (1 χλμ = 8 μπλοκ), να σηκώσει ένα beeper και στη συνέχεια να κινηθεί 5 χλμ βόρεια. Πρόκειται για μια απλή αποστολή που ανήκει στην κατηγορία των προβλημάτων που είναι γνωστή ως προβλήματα του ρομπότ-ταξιδιώτη, μιας και απαιτούν από το ρομπότ να ταξιδεύει σε μεγάλες αποστάσεις.

Παρακάτω παρουσιάζεται τμήμα του μακροσκελούς προγράμματος για την εκτέλεση της συγκεκριμένης αποστολής:

task

```
{  
    move();          // η εντολή εμφανίζεται 80 φορές  
    ...  
    pickBeeper();  
    turnLft();  
    move();          // η εντολή εμφανίζεται 80 φορές  
    ...  
}
```



- (1) Παρόλο που το πρόβλημα είναι απλό η έκταση του προγράμματος είναι μεγάλη. Πού οφείλεται αυτό;
- (2) Πιστεύετε ότι το πρόγραμμα αυτό μπορεί να αποσφαλματωθεί εύκολα στην περίπτωση που υπάρχουν ένα ή περισσότερα λάθη; Μπορεί να τροποποιηθεί εύκολα προκειμένου να επιλυθεί ένα παρόμοιο πρόβλημα για ένα ρομπότ ταξιδιώτη;
- (3) Μπορείτε να σκεφτείτε με ποιο τρόπο θα μπορούσαμε να αντιμετωπίσουμε πιο αποτελεσματικά το παραπάνω πρόβλημα, αλλά και γενικότερα τις αποστολές όπου ο Karel καλείται να μετακινείται σε μεγάλες αποστάσεις; Ποια ή ποιες εντολές θα ήταν χρήσιμες;



Δραστηριότητα 2 – δημιουργία νέων εντολών για αποστολές μεγάλων αποστάσεων...

Αφού το ρομπότ Karel κινείται κατά ένα μπλοκ και δεν καταλαβαίνει την έννοια του χιλιομέτρου πρέπει να μεταφράσουμε τη λύση μας σε όλα αυτά τα προβλήματα σε εντολές που μετακινούν τον Karel κατά ένα μπλοκ κάθε φορά. Για το συγκεκριμένο πρόβλημα το πρόγραμμά μας αποτελείται από 120 εντολές `move()`!!



Λύση στο πρόβλημα μπορεί να μας δώσει η δυνατότητα δημιουργίας νέων εντολών, οι οποίες παρέχουν στον Karel νέες δυνατότητες. Για παράδειγμα, για τα προβλήματα όπου ο Karel χρειάζεται να μετακινείται σε μεγάλες αποστάσεις μπορούμε να δημιουργήσουμε μια νέα εντολή, έστω `moveKlm()`, για την κίνησή του κατά 1 χλμ.

Πρότυπο δήλωσης και ορισμού νέας εντολής	Εφαρμογή της δημιουργίας νέων εντολών στο πρόβλημα του ρομπότ ταξιδιώτη
<pre>void <όνομα-εντολής> () { <λίστα-εντολών> }</pre> <p>όπου: void = δεσμευμένη λέξη που δηλώνει ότι η εντολή δεν επιστρέφει καμία τιμή όταν εκτελείται.</p>	<pre>void moveKlm() { move(); move(); move(); move(); move(); move(); move(); move(); move(); }</pre>

Έχοντας ορίσει τη νέα εντολή το κυρίως πρόγραμμα που περιλάμβανε περισσότερες από 120 γραμμές κώδικα, γίνεται:

task

```
{
    moveKlm();
    moveKlm();
}
```

```
moveKlm();  
moveKlm();  
moveKlm();  
moveKlm();  
moveKlm();  
moveKlm();  
moveKlm();  
moveKlm();  
pickBeeper();  
turnLeft();  
moveKlm();  
moveKlm();  
moveKlm();  
moveKlm();  
moveKlm();
```

}



(9) Μπορείτε να αναφέρετε κάποια πλεονεκτήματα της δημιουργίας νέων εντολών;

(10) Μπορείτε να σκεφτείτε κάποιες εντολές που χρειάζεται να εκτελεί συχνά ο Karel και θα ήταν χρήσιμο να τις ορίσετε σε ένα αρχείο, το οποίο θα χρησιμοποιείται ως πρότυπο για κάθε πρόγραμμα που αναπτύσσετε;



Αν δυσκολεύεστε να απαντήσετε τις παραπάνω ερωτήσεις, ολοκληρώστε και την επόμενη δραστηριότητα και προσπαθήστε ξανά.



Δραστηριότητα 3 – τα πλεονεκτήματα της δημιουργίας νέων εντολών

Κάντε κλικ στην καρτέλα «Δραστηριότητες» της ενότητας «Συναρτήσεις» και διαβάστε την περιγραφή της δραστηριότητας. Μελετήστε και εκτελέστε βηματικά και τις δύο εκδόσεις του προγράμματος.



- (1) Και τα δύο προγράμματα λύνουν το ίδιο πρόβλημα σωστά. Ποια από τις δύο εκδόσεις θεωρείς καλύτερη;
- (2) Σε ποια εντολές μπορεί να ανταποκριθεί ο Karel σε κάθε μία από τις δύο λύσεις;
- (3) Τι αποτέλεσμα έχει η κλήση μιας εντολής μέσα από κάποια άλλη εντολή; Είναι κατανοητή η ροή εκτέλεσης του προγράμματος στη λύση με τη χρήση των νέων εντολών (διαδικασιών);
- (4) Ποια λύση θα επέλεγες αν η σκάλα είχε 10 ή και περισσότερα σκαλοπάτια;
- (5) Αν αναθέταμε στον Karel να καθαρίζει τις σκάλες σε κτίρια με ένα συγκεκριμένο αριθμό σκαλοπατιών, για παράδειγμα 100, ποια ή ποιες εντολές θα έπρεπε να δίνουμε στον Karel; Είναι αποτελεσματικός ο τρόπος επικοινωνίας μας (το πρόγραμμα που θα του δώσουμε να εκτελέσει) με τον Karel ή μήπως θα μπορούσαμε να τον βελτιώσουμε;
- (6) Ποια από τις δύο λύσεις θεωρείς ότι μπορεί να γίνει πιο εύκολα κατανοητή από ένα μαθητή (αν του δοθούν και οι δύο και ζητηθεί να τις σχολιάσει);
- (7) Ποια από τις δύο λύσεις θεωρείς ότι μπορεί πιο εύκολα να τροποποιηθεί για την επίλυση ενός παρόμοιου προβλήματος (όπως αυτά που αναφέρονται παραπάνω);



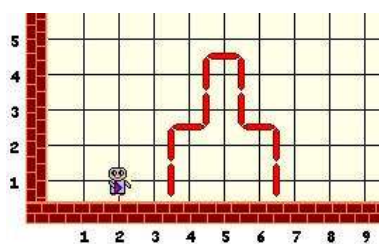
Αν δυσκολεύεστε να απαντήσετε τις ερωτήσεις των δύο τελευταίων δραστηριοτήτων, μελετήστε τη θεωρία της ενότητας «Συναρτήσεις».



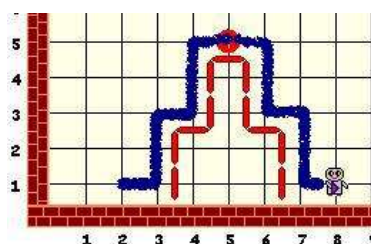
Δραστηριότητα 4 – εξοικείωση με τη δημιουργία νέων εντολών

Τροποποιήστε το πρόγραμμα της αναρρίχησης στο βουνό και τοποθέτησης της σημαίας (φύλλο εργασίας 1, δραστηριότητα 6), το οποίο λύθηκε με τη χρήση μόνο των βασικών εντολών που αναγνωρίζει ο Karel, έτσι ώστε να χρησιμοποιεί νέες εντολές. Οι νέες εντολές που θα ορίσετε θα μπορούν να χρησιμοποιηθούν σε ένα βουνό με οποιοδήποτε ύψος, αλλά με το γενικό σχήμα που φαίνεται στην εικόνα.

Τα τμήματα τοίχου στην παρακάτω εικόνα αναπαριστούν ένα βουνό. Προγραμματίστε τον Karel ώστε να ανέβει στην κορυφή του βουνού και να τοποθετήσει μια σημαία, που αναπαριστάται με ένα *beeper*. Στη συνέχεια ο Karel πρέπει να κατέβει από την άλλη μεριά του βουνού. Υπέθεσε ότι ο Karel ξεκινάει με μια σημαία-beeper στην τσάντα του και ακολουθεί τη διαδρομή που φαίνεται στην τελική κατάσταση της εικόνας.



Αρχική κατάσταση.



Η τελική κατάσταση και η διαδρομή που ακολουθεί το ρομπότ.

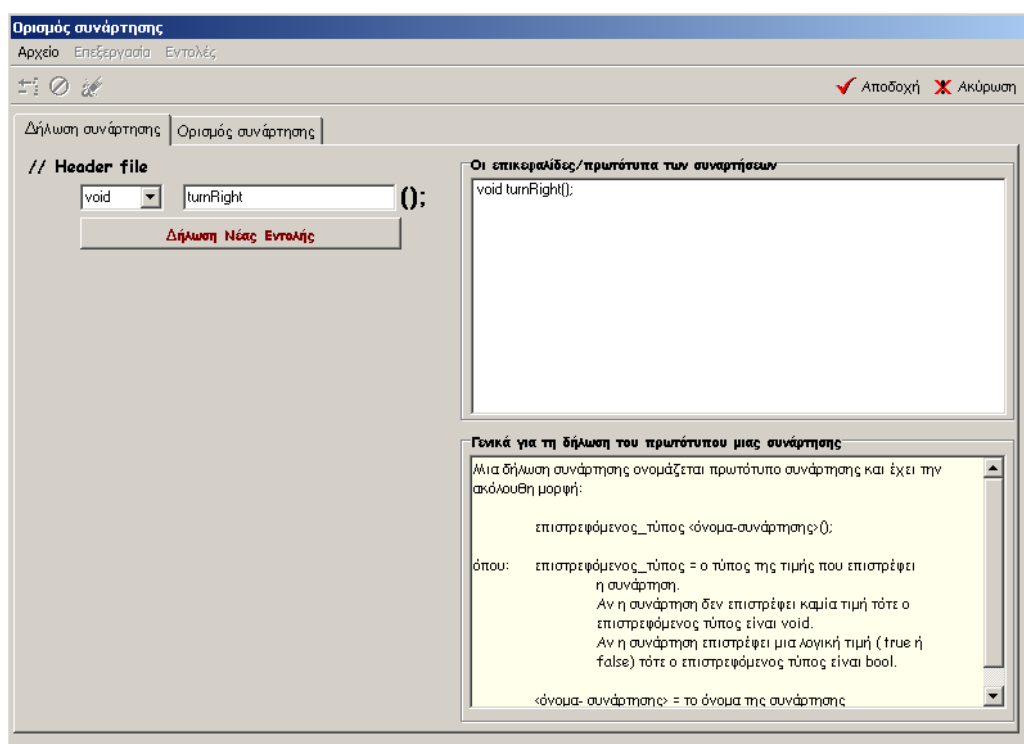
Η αποστολή αναρρίχησης στο βουνό και τοποθέτησης της σημαίας.

Για την ανάπτυξη ενός προγράμματος, στο οποίο ορίζονται νέες εντολές, ακολουθούμε τα παρακάτω βήματα στο περιβάλλον προγραμματισμού:

Δήλωση νέας συνάρτησης

- (1) Επιλέγουμε **Νέα συνάρτηση** από το μενού **Εντολές**, οπότε εμφανίζεται το παρακάτω παράθυρο που φαίνεται στην παρακάτω Εικόνα.
- (2) Στην καρτέλα **Δήλωση συνάρτησης** δηλώνουμε τη νέα συνάρτηση χρησιμοποιώντας το πρότυπο που υπάρχει: αφού προσδιορίσουμε τον επιστρεφόμενο τύπο - void στην περίπτωση που η συνάρτηση δεν επιστρέφει τιμή - δηλώνουμε το όνομα της νέας συνάρτησης. Οι συναρτήσεις δηλώνονται πατώντας το κουμπί **Δήλωση Νέας Εντολής**.

- (3) Στην καρτέλα **Ορισμός Συνάρτησης** ορίζουμε το σώμα των συναρτήσεων χρησιμοποιώντας το μενού εντολής, το οποίο ενημερώνεται αυτόματα με τα ονόματα όλων των συναρτήσεων.
- (4) Επιλέγουμε **Αποδοχή-Επιστροφή** από το μενού **Αρχείο** ή πατάμε το ομώνυμο κουμπί και επιστρέφουμε στο περιβάλλον προγραμματισμού, οπότε ενημερώνεται ο πηγαίος κώδικας και το μενού Εντολής.



Δήλωση και ορισμός νέων εντολών.

Διόρθωση συνάρτησης

Παρατήρηση: μπορούμε ανά πάσα στιγμή να μεταβαίνουμε από την μία καρτέλα στην άλλη και να κάνουμε διορθώσεις:

- (1) Επιλέγουμε **Διόρθωση** ► **Συνάρτησης** από το μενού **Επεξεργασία** οπότε εμφανίζεται το γνωστό παράθυρο με τις καρτέλες Δήλωση Συνάρτησης & Ορισμός Συναρτήσεων, ενημερωμένες με τη δήλωση και τον ορισμό όλων των συναρτήσεων του προγράμματος.
- (2) Κάνουμε τις αλλαγές που θέλουμε μεταβαίνοντας ελεύθερα από τη μία καρτέλα στην άλλη:
- (3) Η **αλλαγή του ονόματος μιας συνάρτησης ή/και του επιστρεφόμενου τύπου** γίνεται στο πρότυπο που βρίσκεται στο αριστερό μέρος της καρτέλας Δήλωση Συνάρτησης.

Για την **διαγραφή μιας συνάρτησης**:

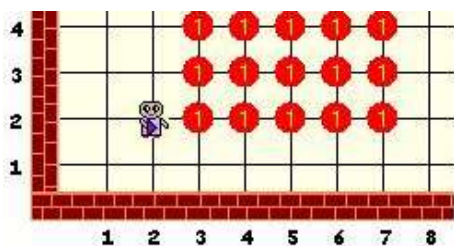
- (1) Επιλέγουμε με το ποντίκι τη συνάρτηση στο πλαίσιο «Οι επικεφαλίδες/υπογραφές των συναρτήσεων».
- (2) Επιλέγουμε **Διαγραφή** ► **Γραμμής** από το μενού **Επεξεργασία**.

Για την εισαγωγή σχολίου, κενής γραμμής και εντολής μεταξύ άλλων στον ορισμό των συναρτήσεων ισχύει ότι και για το κυρίως πρόγραμμα.

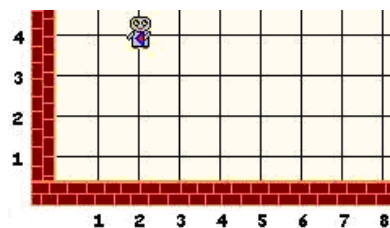


Δραστηριότητα 5

Ο Karel πρέπει να θερίσει το χωράφι με τα beeper που φαίνεται στην παρακάτω Εικόνα. Σε κάθε διασταύρωση υπάρχει ένα beeper. Προγραμματίσει τον Karel να φέρει εις πέρας την αποστολή του θερίσματος. Η αρχική και η τελική κατάσταση φαίνονται στην Εικόνα. Επειδή το χωράφι αποτελείται από οριζόντιες γραμμές beeper ίδιου μήκους και το ρομπότ χρειάζεται να εκτελέσει τις ίδιες ενέργειες για το θέρισμα της κάθε γραμμής μπορείς να ορίσεις μια νέα εντολή με όνομα `harvestOneRow`. Την εντολή αυτή θα την καλέσεις στη συνέχεια τρεις φορές από το κυρίως πρόγραμμα (task) για το θέρισμα όλου του χωραφιού.



Αρχική κατάσταση.



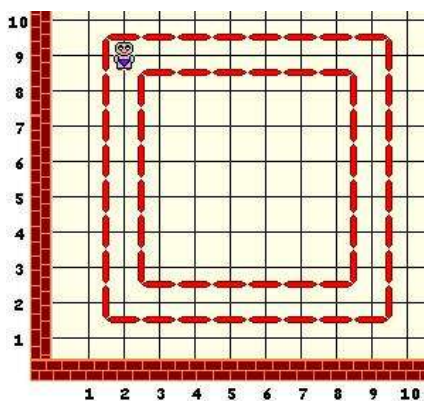
Τελική κατάσταση.

Το θέρισμα του χωραφιού.

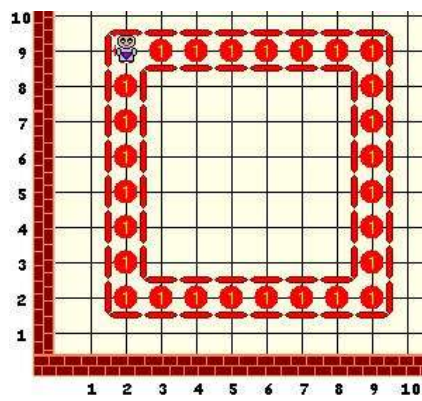


Δραστηριότητα 6

Μια εταιρία καθαριότητας προσέλαβε τον Karel για το στρώσιμο χαλιών (που είναι φτιαγμένα από beepers) σε κτίρια που έχουν το σχήμα που φαίνεται παρακάτω. Γράψε ένα πρόγραμμα προκειμένου να κατευθύνεις το ρομπότ να στρώσει το χαλί στο κτίριο της παρακάτω Εικόνας. Δεν πρέπει να υπάρχουν «εξογκώματα» στο χαλί, γεγονός που σημαίνει ότι το ρομπότ πρέπει να τοποθετεί μόνο ένα beeper σε κάθε γωνία του διαδρόμου. Το ρομπότ ξεκινάει την αποστολή έχοντας στην τσάντα του όσα ακριβώς beepers χρειάζονται. Επειδή όλοι οι διάδρομοι έχουν το ίδιο μήκος, μπορείς να ορίσεις μια νέα εντολή, έστω `installOneCarpet`, που θα καθοδηγεί τον Karel να στρώσει το χαλί σε ένα διάδρομο. Την εντολή αυτή θα την καλέσεις στη συνέχεια τέσσερις φορές από το κυρίως πρόγραμμα (task) για το στρώσιμο του χαλιού σε όλο το κτίριο.



Αρχική κατάσταση.



Τελική κατάσταση

Το στρώσιμο των χαλιών.

18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

- Ολοκληρώστε τη μελέτη των φύλλων εργασίας που προορίζονται για τους μαθητές και σχολιάστε κάθε ένα από αυτά, αναφέροντας τυχόν αδύναμα σημεία και προτείνοντας αλλαγές, βελτιώσεις ή και προσθήκες (θα μπορούσε να αποτελέσει και κοινή δραστηριότητα με wiki ή με ένα ανάλογο περιβάλλον/εργαλείο).
- Ετοιμάστε ένα φύλλο εργασίας που θα έχει ως στόχο την (αυτό)αξιολόγηση των μαθητών, αναφέροντας σύντομα τον στόχο της κάθε δραστηριότητας που προτείνετε.
- Μελετήστε το διαθέσιμο υλικό για τις δομές επιλογής και επανάληψης και σχεδιάστε δραστηριότητες (ή φύλλα εργασίας) για τη διδασκαλία τους.
- Προτείνετε μια οργάνωση των μαθημάτων για τη διδασκαλία σε μαθητές Γυμνασίου των δομών ελέγχου και της έννοιας της διαδικασίας. Να αναφέρετε τη σειρά διδασκαλίας των διάφορων εννοιών, την προτεινόμενη διάρκεια κάθε ενότητας και το υλικό που θα χρησιμοποιηθεί (από το ήδη διαθέσιμο ή νέο).
- Μελετήστε τη βιβλιογραφία.

Ενότητα Σεναρίου 2

Το ρομπότ objectKAREL: εισαγωγή στον αντικειμενοστραφή προγραμματισμό

Αντικειμενοστρεφής προγραμματισμός - σενάρια

1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προγραμματίζοντας ρομπότ στον μικρόκοσμο objectKarel

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 4 διδακτικές ώρες, δηλαδή περίπου 2 ώρες για την εξοικείωση με τις σχετικές έννοιες και 2 ώρες για την υλοποίησή τους μέσω δραστηριοτήτων ανάπτυξης προγραμμάτων. Βέβαια, πρέπει να επισημανθεί ότι η διάρκεια αυτή είναι ενδεικτική και εξαρτάται από την ηλικία, την προηγούμενη εμπειρία και τον αριθμό των μαθητών.

3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Πρόκειται για διδακτικό σενάριο που μπορεί να ενταχθεί κατά κύριο λόγο στο μάθημα επιλογής του Λυκείου «Εφαρμογές Υπολογιστών», αλλά και σε Τεχνικά ή Επαγγελματικά Λύκεια ή Σχολές. Επίσης, εφόσον υπάρχει εμπειρία των μαθητών στην αλγοριθμική και στον (διαδικαστικό) προγραμματισμό από το Δημοτικό και τις δύο πρώτες τάξεις του Γυμνασίου, θα μπορούσε να πραγματοποιηθεί η εισαγωγή στον αντικειμενοστρεφή προγραμματισμό στη Γ' Γυμνασίου. Τα ιδιαίτερα χαρακτηριστικά του διδακτικού μικρόκοσμου objectKarel καθιστούν τη διδασκαλία των σχετικών εννοιών πολύ πιο εύκολη σε σχέση με άλλα αντικειμενοστρεφή περιβάλλοντα.

Το σενάριο δεν προϋποθέτει κάποιες γνώσεις προγραμματισμού, αλλά η ύπαρξή τους μπορεί να επηρεάσει την εφαρμογή του σεναρίου, όπως ήδη αναφέρθηκε. Στο σενάριο που περιγράφεται η προσοχή επικεντρώνεται στην εκμάθηση βασικών εννοιών του

αντικειμενοστρεφούς προγραμματισμού, όπως αντικείμενο, κλάση και μήνυμα, και όχι στην εκμάθηση των καθιερωμένων αλγοριθμικών δομών, δηλαδή της ακολουθιακής δομής, της δομής επιλογής και επανάληψης. Ωστόσο, η πρότερη εμπειρία των μαθητών μπορεί να αξιοποιηθεί και να εμπλουτιστούν κατάλληλα οι προτεινόμενες δραστηριότητες ώστε να περιλαμβάνουν τη χρήση δομών επιλογής και επανάληψης.

Για λόγους πληρότητας, αναφέρεται ότι το σενάριο θα μπορούσε επίσης να ενταχθεί σε πρόγραμμα Πληροφορικού Γραμματισμού σε ΣΔΕ (Σχολεία Δεύτερης Ευκαιρίας για ενηλίκους), ακόμη και στην περίπτωση που οι διδασκόμενοι δεν έχουν γνώσεις στον προγραμματισμό.

4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Βασικός σκοπός του σεναρίου είναι να εισαχθούν οι μαθητές στις έννοιες του αντικειμένου και της κλάσης. Οι μαθητές θα πρέπει να είναι σε θέση να δημιουργούν αντικείμενα (ρομπότ) από μια υπάρχουσα κλάση και να αποστέλλουν σε αυτά τα κατάλληλα μηνύματα για την επίτευξη μιας συγκεκριμένης αποστολής στον κόσμο των ρομπότ (επίλυση προβλήματος). Επίσης, σκοπός του σεναρίου είναι να εξοικειωθούν σταδιακά οι μαθητές με τη διαδικασία ανάπτυξης και αποσφαλμάτωσης προγραμμάτων.

Στους στόχους του σεναρίου περιλαμβάνονται η κατανόηση και η εφαρμογή των σχετικών εννοιών και η αποφυγή της δημιουργίας των παρανοήσεων (ελλιπής/λανθασμένη κατανόηση εννοιών) που έχουν καταγραφεί στη βιβλιογραφία. Αναλυτικά, οι στόχοι του σεναρίου είναι:

- Η εξοικείωση των μαθητών με τις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού, δηλαδή τις έννοιες Αντικείμενο, Μήνυμα και Κλάση.
- Η κατανόηση της σχέσης που συνδέει τις έννοιες αντικείμενο και κλάση, καθώς και του ρόλου τους στη σχεδίαση της λύσης ενός προβλήματος. Οι μαθητές πρέπει να κατανοήσουν ότι μια κλάση περιγράφει με γενικό τρόπο ένα συγκεκριμένο τύπο αντικειμένων και αποτελεί το πρότυπο με βάση το οποίο μπορούμε να δημιουργήσουμε αντικείμενα του συγκεκριμένου τύπου.
- Η κατανόηση της εντολής δημιουργίας αντικειμένων (στιγμιότυπων) από μια κλάση. Οι μαθητές πρέπει να κατανοήσουν ότι μπορούμε να δημιουργήσουμε οποιοδήποτε αριθμό αντικειμένων από μια κλάση, αρκεί να δώσουμε ένα όνομα σε κάθε ένα από αυτά τα αντικείμενα και αρχικές τιμές στις ιδιότητες που καθορίζονται στην κλάση του αντικειμένου.

- Η εξοικείωση με την αποστολή μηνυμάτων σε αντικείμενα και η κατανόηση του γεγονότος ότι ένα αντικείμενο ανταποκρίνεται σε ένα μήνυμα με τον τρόπο που καθορίζει η κλάση στην οποία ανήκει.
- Η κατανόηση του γεγονότος ότι η εκτέλεση μιας μεθόδου από ένα αντικείμενο μπορεί να έχει ως αποτέλεσμα την αλλαγή της τιμής μιας ή περισσότερων ιδιοτήτων του, ή με άλλα λόγια την αλλαγή της κατάστασής του.
- Η κατανόηση του γεγονότος ότι ένα αντικείμενο μπορεί να ανταποκριθεί διαφορετικά στο ίδιο μήνυμα ανάλογα με την κατάστασή του.
- Να μπορούν οι μαθητές να σχεδιάσουν λύσεις σε απλά προβλήματα που ανατίθενται στο περιβάλλον του objectKarel.
- Να είναι σε θέση οι μαθητές να δημιουργούν πολλαπλά αντικείμενα (στιγμιότυπα) από μια κλάση και να τα αρχικοποιούν κατάλληλα ανάλογα με τις απαιτήσεις του προβλήματος.
- Να αντιληφθούν οι μαθητές ότι η κλάση από την οποία δημιουργείται ένα αντικείμενο καθορίζει το σύνολο των μηνυμάτων στα οποία μπορεί να ανταποκριθεί, και κατ' επέκταση να είναι σε θέση να στέλνουν σε ένα αντικείμενο τα κατάλληλα μηνύματα για την επίλυση ενός προβλήματος.
- Να είναι σε θέση οι μαθητές να χρησιμοποιούν το περιβάλλον του objectKarel για την ανάπτυξη, αποθήκευση, ανάκληση, διόρθωση, μεταγλώττιση, αποσφαλμάτωση & εκτέλεσης ενός προγράμματος.
- Να κατανοήσουν οι μαθητές τους διάφορους τύπους λαθών – συντακτικά, λογικά, εκτέλεσης – που μπορεί να εμφανιστούν σε ένα πρόγραμμα και να είναι σε θέση να τα διορθώνουν.
- Να κατανοήσουν οι μαθητές την αξία των δυνατοτήτων της βηματικής εκτέλεσης και της επεξηγηματικής οπτικοποίησης τόσο στην κατανόηση της σημασίας των δομών της γλώσσας όσο και στον εντοπισμό λογικών λαθών. Οι μαθητές πρέπει να κατανοήσουν ότι ένα πρόγραμμα που μεταγλωττίζεται επιτυχώς και εκτελείται δεν είναι απαραίτητα σωστό.

5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Βασική ιδέα είναι η αξιοποίηση της διδακτικής λογικής που υιοθετεί ο μικρόκοσμος objectKarel για την εκμάθηση των βασικών εννοιών του αντικειμενοστρεφούς προγραμματισμού μέσω μιας σειράς δραστηριοτήτων, από τις οποίες αρκετές

ενσωματώνονται στο ίδιο το περιβάλλον.

Η ιδιαιτερότητά του έγκειται στο γεγονός ότι, εκτός από το περιβάλλον προγραμματισμού, ενσωματώνει και όλο το απαραίτητο διδακτικό υλικό για τη διδασκαλία του αντικειμενοστρεφούς προγραμματισμού. Συγκεκριμένα, στον μικρόκοσμο ενσωματώνεται σύντομη και περιεκτική θεωρία και μια σειρά δραστηριοτήτων για την εξοικείωση των μαθητών με τις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού. Η διδασκαλία μιας έννοιας του αντικειμενοστρεφούς προγραμματισμού ή και των βασικών δομών ελέγχου στον μικρόκοσμο του objectKarel περιλαμβάνει τις εξής φάσεις (Ξυνόγαλος, 2002; 2003):

Φάση 1^η: παρουσίαση των προγραμματιστικών εννοιών χρησιμοποιώντας τη θεωρία που υπάρχει στα μαθήματα που ενσωματώνονται στον μικρόκοσμο.

Φάση 2^η: εξοικείωση των μαθητών με τις σχετικές έννοιες χρησιμοποιώντας τις ενσωματωμένες στο περιβάλλον δραστηριότητες

Φάση 3^η: υλοποίηση των εννοιών, αναπτύσσοντας πρόγραμμα με τον εκδότη δομής

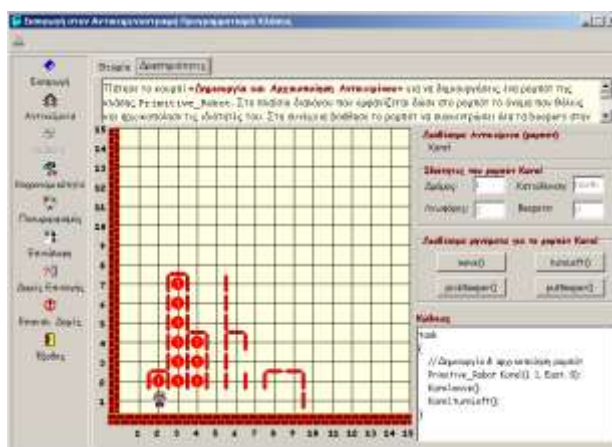
Φάση 4^η: μεταγλώττιση του προγράμματος

Φάση 5^η: βηματική εκτέλεση του προγράμματος και εντοπισμός λογικών λαθών (αποσφαλμάτωση)

Στη συνέχεια παρέχονται λεπτομέρειες σχετικά με τις παραπάνω φάσεις.

Φάσεις 1 και 2: εξοικείωση με τις διδασκόμενες έννοιες

Ο μικρόκοσμος objectKarel ενσωματώνει μια σειρά μαθημάτων, τα οποία περιλαμβάνουν σύντομη και περιεκτική θεωρία, καθώς και δραστηριότητες για τη διδασκαλία των σχετικών εννοιών.



Εικόνα 1. Το περιβάλλον των Δραστηριοτήτων του objectKarel

Η *θεωρία* κάθε ενότητας επικεντρώνεται στη διδασκαλία μιας ή περισσότερων εννοιών του αντικειμενοστρεφούς προγραμματισμού ή μιας δομής ελέγχου. Οι βασικότερες έννοιες του αντικειμενοστρεφούς προγραμματισμού παρουσιάζονται στους μαθητές πριν από τις δομές ελέγχου, ενώ σε όλα τα μαθήματα δίνεται ιδιαίτερη βαρύτητα στις έννοιες αυτές. Οι μαθητές διαχειρίζονται αντικείμενα (ρομπότ) από την αρχή και δημιουργούν τις δικές τους κλάσεις από τα πρώτα μαθήματα.

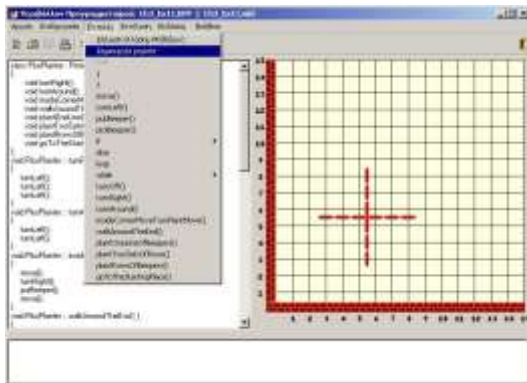
Οι *δραστηριότητες* βρίσκονται στο επίκεντρο των εννοιών που στοχεύει να διδάξει το αντίστοιχο μάθημα και ενσωματώθηκαν στο προγραμματιστικό περιβάλλον έχοντας την πεποίθηση ότι η παρουσίαση των εννοιών του αντικειμενοστρεφούς προγραμματισμού παρέχοντας μόνο κειμενικές και προφορικές περιγραφές δεν έχει το επιθυμητό αποτέλεσμα. Η σχεδίαση των δραστηριοτήτων βασίστηκε στις βασικές αρχές της θεωρίας του *εποικοδομητισμού* και της *διερευνητικής μάθησης*. Στόχος τους είναι η εξοικείωση των μαθητών με τις διδασκόμενες έννοιες και τις λειτουργίες του προγραμματιστικού περιβάλλοντος πριν να ξεκινήσουν να αναπτύσσουν προγράμματα. Για την επίτευξη του στόχου αυτού ο μαθητής *προγραμματίζει με τεχνικές άμεσης διαχείρισης* και μελετά/εκτελεί βηματικά έτοιμα προγράμματα. Για παράδειγμα, ο μαθητής μπορεί να δει στην οθόνη το αποτέλεσμα της αποστολής ενός μηνύματος σε ένα αντικείμενο, τον τρόπο με τον οποίο η εκτέλεση της αντίστοιχης μεθόδου μεταβάλλει τις τιμές των ιδιοτήτων του και τη σύνταξη της εντολής στη γλώσσα προγραμματισμού, πατώντας απλά ένα κουμπί (που έχει ως ετικέτα το όνομα του μηνύματος), όπως φαίνεται στην Εικόνα 1.

Τα μαθήματα που ενσωματώνονται στο περιβάλλον παρέχουν τη δυνατότητα *διδασκαλίας περισσότερης ύλης σε λιγότερο χρόνο*, ενώ οι δραστηριότητες στηρίζουν τους μαθητές στην *κατανόηση και αφομοίωση των διδασκόμενων εννοιών πριν να προχωρήσουν στην ανάπτυξη των προγραμμάτων τους*.

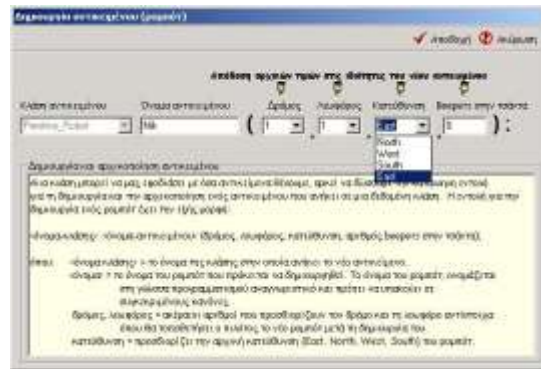
Φάση 3: ανάπτυξη προγράμματος

Μετά τη διδασκαλία και την εξοικείωση με τις σχετικές έννοιες ο μαθητής αναπτύσσει προγράμματα χρησιμοποιώντας τον εκδότη δομής (structure editor) του προγραμματιστικού περιβάλλοντος. Σε ένα τυπικό εκδότη δομής η ανάπτυξη των προγραμμάτων γίνεται με τη χρήση προτύπων και την επιλογή από μενού συντακτικά σωστών εντολών για κάθε ημιτελές τμήμα ενός προγράμματος. Στο περιβάλλον του objectKarel, ο μαθητής αναπτύσσει τα προγράμματά του μέσω του μενού Εντολές του

βασικού παραθύρου (Εικόνα 2) και πλαισίων διαλόγου (Εικόνα 3).



Εικόνα 2. Το μενού που χρησιμοποιείται για την ανάπτυξη των προγραμμάτων.



Εικόνα 3. Πλαίσιο διαλόγου για τη δημιουργία και αρχικοποίηση ενός αντικειμένου (ρομπότ).

Παρόλο που η γλώσσα προγραμματισμού του μικρόκοσμου αποτελείται από ένα περιορισμένο ρεπερτόριο εντολών, η χρήση ενός εκδότη δομής βοηθάει στην περαιτέρω αντιμετώπιση του προβλήματος της επικέντρωσης στις συντακτικές λεπτομέρειες της γλώσσας προγραμματισμού και παρέχει τη *δυνατότητα επικέντρωσης στις έννοιες και στην ανάπτυξη ικανοτήτων επίλυσης προβλημάτων*. Επίσης, καθιστά δυνατή τη *διδασκαλία των βασικών εννοιών του αντικειμενοστρεφούς προγραμματισμού και των δομών ελέγχου σε μικρότερο χρονικό διάστημα*, αφενός γιατί οι μαθητές δεν χρειάζεται να αφιερώσουν χρόνο στην εκμάθηση της σύνταξης της γλώσσας και αφετέρου γιατί ο περιορισμός των συντακτικών λαθών μειώνει δραματικά το χρόνο που αφιερώνεται στην αποσφαλμάτωση των προγραμμάτων. Τέλος, είναι προφανές ότι η χρήση του εκδότη δομής παρέχει τη δυνατότητα διδασκαλίας των βασικών αρχών του αντικειμενοστρεφούς προγραμματισμού σε *μαθητές μικρότερης ηλικίας*.

Βέβαια, ο συγκεκριμένος εκδότης είναι λιγότερο «αυστηρός» σε σχέση με ένα τυπικό εκδότη δομής, ο οποίος επιτρέπει την ανάπτυξη μόνο συντακτικά σωστού πηγαίου κώδικα. Στο συγκεκριμένο εκδότη δομής επιτρέπεται η παραβίαση κάποιων συντακτικών κανόνων, προκειμένου η διόρθωση ενός προγράμματος να μην συνεπάγεται μια εκτεταμένη σειρά ενεργειών. Αυτό βέβαια σημαίνει ότι υπάρχει περίπτωση να προκύψουν κάποια συντακτικά λάθη. Ωστόσο, το γεγονός αυτό δεν δημιουργεί πρόβλημα, αφενός γιατί ο αριθμός των συντακτικών λαθών που μπορεί να προκύψουν είναι μικρός και αφετέρου για τα λάθη που μπορεί να προκύψουν *αναφέρει φιλικά προς το χρήστη μηνύματα λάθους*.

Φάση 4: μεταγλώττιση

Τα αποτελέσματα της μεταγλώττισης ενός προγράμματος (λάθη – προειδοποιήσεις) παρουσιάζονται στην περιοχή που εκτείνεται στο κάτω μέρος του βασικού παραθύρου. Η παρουσίαση των αποτελεσμάτων της μεταγλώττισης δεν είναι στατική. Αντίθετα, παρέχεται μια *δυναμική οπτικοποίηση των αποτελεσμάτων της μεταγλώττισης*. Όταν ο μαθητής κάνει διπλό κλικ σε κάποιο μήνυμα λάθους, αλλάζει το χρώμα του υποβάθρου της αντίστοιχης γραμμής, καθώς επίσης και της γραμμής του πηγαίου κώδικα όπου εντοπίστηκε το λάθος. Η ταυτόχρονη σηματοδότηση του μηνύματος λάθους και της γραμμής του πηγαίου κώδικα όπου εντοπίζεται το λάθος βοηθάει το μαθητή να επικεντρώσει την προσοχή του στο σωστό σημείο. Επίσης, ο μαθητής δεν χρειάζεται να μετράει τις γραμμές του πηγαίου κώδικα προκειμένου να εντοπίσει τη γραμμή στην οποία βρίσκεται το λάθος, με αποτέλεσμα ο χρόνος του να αφιερώνεται σε πιο ουσιαστικά θέματα.

Επιπλέον, τα *μηνύματα λάθους* και οι *προειδοποιήσεις* που αναφέρονται: (1) είναι εξελληνισμένα, (2) χρησιμοποιούν φυσική γλώσσα και όχι κωδικούς, (3) η γραμμή που αναφέρεται είναι η πραγματική γραμμή στην οποία βρίσκεται το λάθος, και (4) προσδιορίζουν όχι μόνο τι είναι λάθος, αλλά και γιατί είναι λάθος (όπου δεν είναι εμφανές). Ο μαθητής λοιπόν δεν δυσκολεύεται στην κατανόηση των μηνυμάτων λάθους, η οποία έχει συνήθως ως αποτέλεσμα την πραγματοποίηση μιας αλληλουχίας αλλαγών σε τυχαία σημεία του προγράμματος και αλληπάλληλων μεταγλωττίσεων.

Φάση 5: βηματική εκτέλεση και αποσφαλμάτωση

Εφόσον ένα πρόγραμμα έχει μεταγλωττιστεί επιτυχώς ο μαθητής έχει τη δυνατότητα εκτέλεσής του με τους εξής τρόπους:

Εκτέλεση: παρουσιάζεται το αποτέλεσμα της εκτέλεσης του προγράμματος στον κόσμο.

Ανίχνευση: το πρόγραμμα εκτελείται βήμα προς βήμα με προκαθορισμένη ταχύτητα, την οποία επιλέγει ο μαθητής. Η τρέχουσα εντολή σηματοδοτείται με αλλαγή του χρώματος του υποβάθρου από άσπρο σε μπλε και των χαρακτήρων από μαύρο σε άσπρο, ενώ ταυτόχρονα φαίνεται το αποτέλεσμα της εκτέλεσής της στην κατάσταση του κόσμου.

Βήμα προς Βήμα Εκτέλεση: σε αντίθεση με τους προηγούμενους τρόπους εκτέλεσης, ο μαθητής αναλαμβάνει ενεργό ρόλο και αποφασίζει πότε θα εκτελεστεί η επόμενη εντολή.

Οι δύο τελευταίες επιλογές εκτέλεσης ενός προγράμματος, γνωστές ως *δυναμική προσομοίωση της εκτέλεσης ενός προγράμματος* (program animation), παρέχουν ουσιαστική βοήθεια στον αρχάριο προγραμματιστή. Ιδιαίτερα η βήμα προς βήμα εκτέλεση

αποκαλύπτει το δυναμικό χαρακτήρα της εκτέλεσης ενός προγράμματος, στηρίζει ουσιαστικά το μαθητή στην κατανόηση της σημασίας των διδασκόμενων εννοιών, βοηθάει στην κατανόηση της ροής ελέγχου και στην εξάλειψη των σχετικών παρανοήσεων, βοηθάει στον εντοπισμό, αλλά και την κατανόηση της πηγής των λογικών λαθών. Επίσης, παρέχει στο διδάσκοντα τη δυνατότητα να καλύψει περισσότερη ύλη σε λιγότερο χρόνο, ενώ αποφεύγονται και τα λάθη που γίνονται στην προσπάθεια παρουσίασης δυναμικών εννοιών χρησιμοποιώντας το στατικό μέσο του πίνακα.

Επιπλέον, όταν ο μαθητής χρησιμοποιεί την ανίχνευση ή τη βήμα προς βήμα εκτέλεση, παρέχονται επεξηγήσεις για τη σημασία της τρέχουσας κάθε φορά εντολής. Οι επεξηγήσεις χρησιμοποιούν φυσική γλώσσα και παρουσιάζονται στην περιοχή που εκτείνεται κατά μήκος του κάτω μέρους του βασικού παραθύρου του προγραμματιστικού περιβάλλοντος. Η δυνατότητα αυτή, γνωστή ως *επεξηγηματική οπτικοποίηση* (explanatory visualization), παρέχει τα εξής πλεονεκτήματα: (1) ένα εναλλακτικό μέσο παρουσίασης της σημασίας των εντολών και παρακολούθησης της ροής εκτέλεσης ενός προγράμματος. Ο μαθητής μπορεί να συμβουλευτεί τις επεξηγήσεις, όταν δεν μπορεί να αντιληφθεί τι συμβαίνει παρακολουθώντας μόνο τη δυναμική οπτικοποίηση της εκτέλεσης ενός προγράμματος, και (2) πιστεύουμε ότι μπορεί να βοηθήσει ιδιαίτερα στην κατανόηση και κυρίως στην εξάλειψη παρανοήσεων που αφορούν έννοιες και δομές που είναι δύσκολες για τους μαθητές, όπως για παράδειγμα οι δομές επιλογής. Παρόλο που με τη βήμα προς βήμα εκτέλεση σηματοδοτείται η εντολή που εκτελείται κάθε φορά, ο μαθητής σε ορισμένες περιπτώσεις δεν μπορεί να «ακολουθήσει» την εκτέλεση ενός προγράμματος. Αν, για παράδειγμα, πιστεύει ότι το τμήμα then μιας δομής επιλογής εκτελείται ανεξάρτητα από το αν αληθεύει η συνθήκη, είναι πολύ πιθανό σε μια ανάλογη περίπτωση να μην καταλάβει γιατί δεν εκτελείται το τμήμα then και να δημιουργηθεί σύγχυση όσον αφορά στη ροή εκτέλεσης. Η επεξήγηση, σε φυσική γλώσσα, του λόγου για τον οποίο δεν θα εκτελεστεί το τμήμα then, βοηθάει το μαθητή να κατανοήσει τη σημασία της εντολής και να εξαλείψει τη σχετική παρανόηση ευκολότερα.

Επίσης, στην περίπτωση που προκύψει λάθος εκτέλεσης το προγραμματιστικό περιβάλλον παρουσιάζει στο μαθητή τις εξής πληροφορίες για τον εντοπισμό του λογικού λάθους που το προκάλεσε: (i) την εντολή που προκάλεσε το λάθος εκτέλεσης, (ii) την ακριβή θέση του ρομπότ και της κατάστασης του κόσμου εκείνη τη στιγμή, και (iii) την επεξήγηση του λόγου που οδήγησε στο λάθος εκτέλεσης.

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Οι δύο βασικότερες έννοιες του αντικειμενοστρεφούς προγραμματισμού είναι η έννοια της κλάσης και του αντικειμένου, οι οποίες μάλιστα σύμφωνα με τη διεθνή βιβλιογραφία πολλές φορές συγχέονται. Η διδασκαλία της συγκεκριμένης τεχνικής προγραμματισμού πρέπει να επικεντρωθεί στις παρακάτω έννοιες:

- Η αναπαράσταση των οντοτήτων/τιμημάτων ενός συστήματος του κόσμου για το οποίο δημιουργούμε ένα μοντέλο στον υπολογιστή γίνεται με τα **αντικείμενα**. Στον μικρόκοσμο `objectKarel` τα μοναδικά αντικείμενα είναι τα ρομπότ, στα οποία ανατίθενται διάφορες αποστολές.
 - Τα αντικείμενα ανάλογα με τις ιδιότητες που τα χαρακτηρίζουν και τις λειτουργίες που μπορούν να εκτελούν κατηγοριοποιούνται και περιγράφονται με ένα γενικό τρόπο σε μια **κλάση**. Για κάθε διαφορετικό είδος αντικειμένων πρέπει να ορίσουμε μία ξεχωριστή κλάση. Στον μικρόκοσμο `objectKarel` υπάρχουν διαθέσιμες δύο κλάσεις, ή αλλιώς μοντέλα ρομπότ: (1) η κλάση `Primitive_Robot`, τα ρομπότ της οποίας εκτελούν πέντε μόνο βασικές εντολές, (2) η κλάση `Robot`, τα ρομπότ της οποίας μπορούν επιπλέον να ελέγχουν διάφορες συνθήκες και να εκτελούν εντολές κατ' επιλογή και κατ' επανάληψη.
 - Μια κλάση περιγράφει με ένα γενικό (αφηρημένο) τρόπο:
- (3) τις *ιδιότητες* που χαρακτηρίζουν ένα είδος αντικειμένων ή αλλιώς τις τιμές δεδομένων που χαρακτηρίζουν την *κατάστασή* τους. Οι τιμές των ιδιοτήτων αυτών αποθηκεύονται σε ειδικές μεταβλητές που δηλώνονται στην κλάση και ονομάζονται **πεδία**. Οι τιμές αυτές μπορούν να αλλάζουν κατά την εκτέλεση ενός προγράμματος, αλλάζοντας ταυτόχρονα και την κατάσταση του αντικειμένου. Τα ρομπότ των κλάσεων `Primitive_Robot` και `Robot` διαθέτουν 4 πεδία για την αποθήκευση του δρόμου, της λεωφόρου, της κατεύθυνσης και του αριθμού των `beepers` που κουβαλάει ένα ρομπότ στην τσάντα του. Όλες οι κλάσεις (μοντέλα ρομπότ) που δημιουργούν οι μαθητές κληρονομούν αυτά τα 4 πεδία, αλλά δεν υπάρχει η δυνατότητα προσθήκης νέων πεδίων.
- (4) τις *λειτουργίες* που καθορίζουν τη *συμπεριφορά* ενός είδους αντικειμένων και προσδιορίζουν, μεταξύ άλλων, τον τρόπο επεξεργασίας των δεδομένων που περικλείει το κάθε αντικείμενο. Η περιγραφή του τρόπου εκτέλεσης μιας συγκεκριμένης λειτουργίας γίνεται γράφοντας μια ομάδα εντολών στην οποία δίνουμε ένα συγκεκριμένο όνομα και μπορούμε να εκτελούμε για κάθε αντικείμενο της κλάσης,

όποτε χρειάζεται. Η υλοποίηση κάθε τέτοιας λειτουργίας ονομάζεται **μέθοδος**. Συχνά επίσης αναφέρουμε ότι στέλνουμε σε ένα αντικείμενο ένα **μήνυμα** και αυτό ανταποκρίνεται εκτελώντας την αντίστοιχη μέθοδο.

- Από τη στιγμή που θα ορίσουμε μία κλάση, *μπορούμε να δημιουργήσουμε όσα αντικείμενα, ή αλλιώς **στιγμιότυπα**, χρειαζόμαστε* για την επίλυση ενός δεδομένου προβλήματος. Η κλάση αποτελεί το καλούπι βάσει του οποίου κατασκευάζονται τα αντικείμενα. Κάθε στιγμιότυπο/αντικείμενο μιας κλάσης έχει τις ιδιότητες (πεδία) και μπορεί να εκτελέσει τις λειτουργίες (μεθόδους) της κλάσης από την οποία προέρχεται. Όλα τα στιγμιότυπα μιας κλάσης έχουν τα ίδια πεδία, αλλά οι τιμές των πεδίων αυτών και συνεπώς η κατάσταση των αντικειμένων μιας δεδομένης κλάσης μπορεί να διαφέρει.
- Όταν δημιουργούμε ένα νέο αντικείμενο του δίνουμε ένα *όνομα* προκειμένου να μπορούμε στη συνέχεια να αναφερθούμε σε αυτό μέσω του προγράμματός μας και να του στείλουμε μηνύματα, και επίσης *αρχικοποιούμε* τα πεδία του καθορίζοντας έτσι την αρχική του κατάσταση. Η δημιουργία ενός νέου αντικειμένου γίνεται καλώντας μια ειδική μέθοδο που ορίζεται στην κλάση του αντικειμένου και ονομάζεται **κατασκευαστής**.

7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Το διδακτικό σενάριο προβλέπεται να υλοποιηθεί στο εργαστήριο Πληροφορικής.

Απαιτείται το περιβάλλον objectKarel. Το περιβάλλον δεν απαιτεί κάποια ιδιαίτερη διαδικασία εγκατάστασης. Αποσυμπίεζετε το αρχείο objectKarel.zip (<http://users.uom.gr/~stelios/objectKarel.zip>), οπότε δημιουργείται ένας ομώνυμος φάκελος και εκτελείτε το αρχείο objectKarel.exe που υπάρχει σε αυτόν.

«Προστιθέμενη αξία»

Το περιβάλλον objectKarel αποτελεί ένα διδακτικό μικρόκοσμο και συνεπώς παρουσιάζει τα γνωστά πλεονεκτήματα που χαρακτηρίζουν γενικά τους προγραμματιστικούς μικρόκοσμους:

- Η γλώσσα προγραμματισμού αποτελείται από ένα περιορισμένο ρεπερτόριο εντολών με απλή σύνταξη και σημασιολογία.

- Βασίζονται σε υπαρκτά μοντέλα που είναι ήδη γνωστά στο μαθητή, μειώνοντας έτσι δραματικά τη διανοητική «απόσταση» ανάμεσα στα νοητά μοντέλα ή την περιγραφή σε φυσική γλώσσα των αλγορίθμων και στην περιγραφή τους στη γλώσσα προγραμματισμού.
- Τα προβλήματα που καλούνται να λύσουν οι μαθητές παρουσιάζουν ιδιαίτερο ενδιαφέρον.
- Η εκτέλεση ενός προγράμματος είναι βηματική και ορατή, αποκαλύπτοντας έτσι τη σημασία των διδασκόμενων δομών, καθώς και τις έννοιες που σχετίζονται με τη δομή και την εκτέλεση των προγραμμάτων.
- Υπάρχει δυνατότητα προσαρμογής του μικρόκοσμου στις ανάγκες του κοινού στο οποίο απευθύνεται.

Επιπλέον, όπως αναφέρθηκε στην ενότητα 5, ο μικρόκοσμος objectKarel ενσωματώνει:

- το απαραίτητο διδακτικό υλικό (σύντομη και περιεκτική θεωρία, εγχειρίδιο χρήσης)
- δραστηριότητες για την εξοικείωση των μαθητών με τις έννοιες πριν την εφαρμογή τους στα πλαίσια ανάπτυξης προγραμμάτων
- ένα συντάκτη δομής για την εύκολη ανάπτυξη προγραμμάτων και την αποφυγή επικέντρωσης στις λεπτομέρειες της γλώσσας προγραμματισμού
- τη δυνατότητα της επεξηγηματικής οπτικοποίησης, εμφάνισης δηλαδή μηνυμάτων σε φυσική γλώσσα για την τρέχουσα κάθε φορά εντολή κατά τη βηματική εκτέλεση των προγραμμάτων.

Αντίλογος, επιφυλάξεις, προβλήματα

Εντολές στα Αγγλικά. Υπάρχει δυνατότητα επιλογής εμφάνισης του διδακτικού υλικού που είναι ενσωματωμένο στον μικρόκοσμο objectKarel, καθώς και πραγματοποίησης του διαλόγου χρήστη-συστήματος είτε στα Αγγλικά είτε στα Ελληνικά. Ωστόσο, για το ρεπερτόριο εντολών χρησιμοποιείται η αγγλική γλώσσα, γεγονός που ενδεχομένως να θεωρηθεί από ορισμένους εκπαιδευτικούς ως πρόβλημα. Ωστόσο:

- το ρεπερτόριο εντολών είναι περιορισμένο
- χρησιμοποιούνται σαφή ονόματα εντολών που φανερώνουν τη σημασία τους και δεν δημιουργούν παρανοήσεις για τη σημασία τους
- οι εντολές δεν χρειάζεται να απομνημονευθούν, αφού η ανάπτυξη των προγραμμάτων γίνεται χρησιμοποιώντας τον συντάκτη δομής.

Ελλείψεις στη γλώσσα προγραμματισμού. Στον μικρόκοσμο `objectKarel` υπάρχουν δύο έτοιμες κλάσεις ρομπότ: (1) η κλάση `Primitive_Robot` που μας εφοδιάζει με ρομπότ ικανά να ανταποκριθούν στα μηνύματα `move`, `turnLeft`, `pickBeeper` και `putBeeper`, (2) η κλάση `Robot`, τα ρομπότ της οποίας έχουν όλες τις δυνατότητες των ρομπότ της κλάσης `Primitive_Robot` και επιπλέον έχουν τη δυνατότητα να ελέγχουν την κατάστασή τους (π.χ. αν έχουν beeper στην τσάντα τους, αν βλέπουν προς μια συγκεκριμένη κατεύθυνση κτλ) και την κατάσταση του περιβάλλοντος τους (π.χ. αν υπάρχει τοίχος μπροστά) χρησιμοποιώντας δομές επιλογής και επανάληψης (`if`, `if/else`, `while`). Όσον αφορά στην κλάση `Primitive_Robot` που χρησιμοποιείται στα αρχικά μαθήματα, υπάρχει διαθέσιμη η μέθοδος `turnLeft` για στροφή 90 μοιρών προς τα αριστερά, αλλά απουσιάζει μια μέθοδος, έστω `turnRight`, για στροφή 90 μοιρών προς τα δεξιά. Είναι βέβαια προφανές ότι σε αρκετές περιπτώσεις ένα ρομπότ χρειάζεται να στρίψει προς τα δεξιά και για να γίνει αυτό οι μαθητές αναγκάζονται να καλούν κάθε φορά τη μέθοδο `turnLeft` 3 φορές. Αυτό με μια πρώτη ματιά μπορεί να θεωρηθεί ως έλλειψη της γλώσσας προγραμματισμού που χρησιμοποιείται, η ακόμα και ως πρόβλημα. Ωστόσο, η επιλογή των δημιουργών της γλώσσας να μην συμπεριληφθεί η μέθοδος `turnRight` στη βασική κλάση ρομπότ ήταν απόλυτα συνειδητή. Οι μαθητές αναγνωρίζουν από την αρχή το γεγονός ότι το βασικό μοντέλο ρομπότ έχει πολύ περιορισμένες δυνατότητες και αναζητούν τρόπους δημιουργίας νέων μοντέλων ρομπότ (κλάσεων) με περισσότερες δυνατότητες (μεθόδους). Με αυτό τον τρόπο οι μαθητές κατανοούν πολύ ευκολότερα την ανάγκη ύπαρξης ενός μηχανισμού δημιουργίας νέων μοντέλων ρομπότ (κλάσεων) που θα έχουν όλες τις δυνατότητες του βασικού μοντέλου (κληρονομικότητα) και επιπλέον δυνατότητες ανάλογα με το σενάριο. Η «έλλειψη» λοιπόν μιας μεθόδου `turnRight`, για παράδειγμα, εξυπηρετεί ένα συγκεκριμένο διδακτικό σκοπό και δεν οφείλεται σε μια προβληματική σχεδίαση της γλώσσας.

Συντάκτης δομής. Οι συντάκτες δομής είναι γνωστό ότι στηρίζουν ουσιαστικά τους αρχάριους προγραμματιστές στην ανάπτυξη προγραμμάτων, στην αποφυγή της επικέντρωσης στην εκμάθηση της σύνταξης της γλώσσας προγραμματισμού και των συντακτικών λαθών. Ωστόσο, είναι επίσης γνωστό ότι η διόρθωση ενός προγράμματος, η οποία πολλές φορές συνεπάγεται την προσωρινή παραβίαση των συντακτικών κανόνων, είναι αρκετές φορές πιο δύσκολη χρησιμοποιώντας ένα συντάκτη δομής που απαιτεί ένα πρόγραμμα να παραμένει καθ' όλη τη διάρκεια ανάπτυξής του συντακτικά σωστό. Ο συντάκτης δομής του `objectKarel`, όπως αναφέρθηκε στην ενότητα 5 είναι λιγότερο

«αυστηρός», και καθιστά τη διόρθωση των προγραμμάτων εύκολη. Εξάλλου, οι αρχάριοι προγραμματιστές - σε αντίθεση με τους έμπειρους προγραμματιστές - προτιμούν πάντα τη χρήση ενός συντάκτη δομής για την ανάπτυξη των προγραμμάτων τους, ακόμα και με τους περιορισμούς που μπορεί να έχει ένας τέτοιος συντάκτης δομής (Xinogalos et al, 2006b).

8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

Οι συνηθέστερες δυσκολίες ενός αρχάριου προγραμματιστή, όσον αφορά στις βασικές αντικειμενοστρεφείς έννοιες, εντοπίζονται στα εξής:

Κλάση και αντικείμενο

- Αρκετοί μαθητές/σπουδαστές συγχέουν τις έννοιες της κλάσης και του αντικειμένου και δεν μπορούν να τις διαχωρίσουν (Holland et al., 1997; Sanders & Thomas, 2007).
- Μια κλάση είναι μια συλλογή αντικειμένων και όχι ένα πρότυπο δημιουργίας αντικειμένων (Ragonis & Ben-Ari, 2005; Sanders & Thomas, 2007; Thomasson et al., 2006).
- Δυσκολία κατανόησης της στατικής φύσης μιας κλάσης και της δυναμικής φύσης ενός αντικειμένου (Ragonis & Ben-Ari, 2005).

Δημιουργία αντικειμένων (Ragonis & Ben-Ari, 2005)

- Δυσκολία κατανόησης της διαδικασίας δημιουργίας ενός αντικειμένου.
- Τα αντικείμενα δημιουργούνται από μόνα τους.
- Η χρήση του κατασκευαστή είναι προαιρετική. Η δημιουργία ενός αντικειμένου μπορεί να επιτευχθεί και με μια μέθοδο, έστω `set_values`, που δίνει αρχικές τιμές στις μεταβλητές στιγμιοτύπου του νέου αντικειμένου (Fleury, 2000).
- Η χρήση πολλαπλών κατασκευαστών προκαλεί σύγχυση (Carter & Fowler, 1998). Αρκετοί δυσκολεύονται να ορίσουν δύο ή και περισσότερους κατασκευαστές στην ίδια κλάση και ορίζουν ένα μόνο κατασκευαστή ή δεν δίνουν το ίδιο όνομα στους κατασκευαστές όπως απαιτείται (Ξυνόγαλος, 2008).
- Οι κατασκευαστές μπορούν να περιλαμβάνουν μόνο εντολές ανάθεσης για την αρχικοποίηση των πεδίων.
- Η δημιουργία ενός στιγμιοτύπου (αντικειμένου) μιας κλάσης αφορά μόνο στην εκτέλεση του κατασκευαστή και όχι στην κατανομή μνήμης.

- Δυσκολία κατανόησης ενός κενού κατασκευαστή.
- Η αρχικοποίηση ενός πεδίου με μια σταθερά κατά τη δήλωσή του προκαλεί δυσκολία στο διαχωρισμό μεταξύ της κλάσης και του αντικειμένου.
- Αν τα πεδία αρχικοποιούνται στη δήλωση της κλάσης, τότε δεν υπάρχει ανάγκη να δημιουργηθούν αντικείμενα.

Ιδιότητες και κατάσταση ενός αντικειμένου

Η δυσκολία κατανόησης του ρόλου των πεδίων μιας κλάσης έχει ως συνέπεια τη δημιουργία διάφορων παρανοήσεων:

- Κάθε αντικείμενο αποτελεί απλά ένα «περιτύλιγμα» μιας μεταβλητής (Holland et al., 1997).
- Δυσκολία κατανόησης της επίδρασης που έχει η εκτέλεση μιας μεθόδου στην κατάσταση ενός αντικειμένου (Ragonis & Ben-Ari, 2005). Σε αρκετές περιπτώσεις δεν γίνεται αντιληπτό ότι η αρχική κατάσταση ενός αντικειμένου καθορίζεται όταν αυτό δημιουργείται και στη συνέχεια ενημερώνεται με την εκτέλεση μεθόδων από το αντικείμενο (Ξυνόγαλος, 2008).
- Δύο αντικείμενα της ίδιας κλάσης δεν μπορούν να έχουν τις ίδιες τιμές πεδίων (Ragonis & Ben-Ari, 2005).

Λειτουργίες και συμπεριφορά ενός αντικειμένου

- Σε αρκετές περιπτώσεις δεν γίνεται αντιληπτό ότι η συμπεριφορά ενός αντικειμένου μπορεί να αλλάξει ουσιαστικά ανάλογα με την κατάστασή του (Holland et al., 1997).
- Δεν αξιοποιείται η λειτουργικότητα των μεθόδων και επαναλαμβάνεται ένα τμήμα κώδικα αντί να ορίζεται μία μέθοδος (Fleury, 2001).
- Μπορούμε να ορίσουμε μια μέθοδο που προσθέτει μια ιδιότητα/πεδίο σε μία κλάση (Ragonis & Ben-Ari, 2005).
- Δυσκολία κατανόησης της κατηγοριοποίησης των μεθόδων σε κατασκευαστές, μεθόδους πρόσβασης και μετάλλαξης (Ragonis & Ben-Ari, 2005).
- Δυσκολία κατανόησης του γεγονότος ότι μία μέθοδος μπορεί να κληθεί για οποιοδήποτε αντικείμενο μιας κλάσης (Ragonis & Ben-Ari, 2005).
- Οι μέθοδοι εκτελούνται σύμφωνα με τη σειρά που εμφανίζονται στον ορισμό της κλάσης (Ragonis & Ben-Ari, 2005).

- Κάθε μέθοδος μπορεί να κληθεί μόνο μία φορά (Ragonis & Ben-Ari, 2005).
- Δυσκολία κατανόησης της κλήσης μιας μεθόδου από άλλη μέθοδο (Ragonis & Ben-Ari, 2005).
- Δυσκολία κατανόησης της προέλευσης των τιμών των παραμέτρων σε μια μέθοδο (Ragonis & Ben-Ari, 2005).
- Δυσκολία κατανόησης σχετικά με το πού καταλήγει η επιστρεφόμενη τιμή μιας μεθόδου (Ragonis & Ben-Ari, 2005).

9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΥΡΥΒΟΣ

Όταν οι μαθητές έχουν προηγούμενη εμπειρία σε κάποια διαδικαστική γλώσσα (ή ψευδογλώσσα), η εκμάθηση των αρχών του αντικειμενοστρεφούς προγραμματισμού καθίσταται δυσκολότερη. Οι σημαντικότερες δυσκολίες που παρουσιάζονται είναι οι εξής:

- οι μαθητές *δυσκολεύονται να αλλάξουν τρόπο σκέψης* και να αρχίσουν να σχεδιάζουν αντικειμενοστρεφείς λύσεις σε προβλήματα. Παρά τους ισχυρισμούς ότι η αντικειμενοστρεφής προσέγγιση επίλυσης προβλημάτων είναι πιο εύκολη (φυσική για την ακρίβεια), ο νέος τρόπος σκέψης που απαιτεί δεν γίνεται εύκολα κατανοητός από μαθητές που έχουν εμπειρία στην επίλυση προβλημάτων με μια διαδικαστική γλώσσα προγραμματισμού (Tempete, 1991).
- *δεν αξιοποιούνται τα πλεονεκτήματα του αντικειμενοστρεφούς προγραμματισμού.* Μαθητές με προηγούμενη εμπειρία σε διαδικαστικές γλώσσες προγραμματισμού, κατά την εισαγωγή τους στη Java αντιμετώπισαν δυσκολίες με τις έννοιες του αντικειμενοστρεφούς προγραμματισμού και συγκεκριμένα με την αξιοποίησή τους. Για παράδειγμα, οι μαθητές παρουσίασαν την τάση να χρησιμοποιούν τις μεθόδους ως διαδικασίες, αγνοώντας το ρόλο τους στο πλαίσιο του αντικειμενοστρεφούς προγραμματισμού (Handjerrouit, 1998; Handjerrouit, 1999).

Η αντιμετώπιση των παραπάνω δυσκολιών βέβαια είναι πολύ ευκολότερη σε ένα διδακτικό μικρόκοσμο, όπως ο objectKarel.

10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Bergin, J., Stehlik, M., Roberts, J. & Pattis, R. Karel++ - A Gentle Introduction to the Art of Object-Oriented Programming (John Wiley and Sons. New York, 2nd edition, 1997)

- Carter, J. & Fowler, A. (1998), Object Oriented Students? *SIGCSE Bulletin*, 30(3), 271.
- Fleury, A. (2000). Programming in Java: student-constructed rules. *SIGCSE Bulletin*, 32(1), 197-201.
- Fleury, A. E. (2001), Encapsulation and reuse as viewed by java students, *ACM SIGCSE Bulletin*, Vol. 33, Issue 1, 189-193.
- Hadjerrouit, S. (1998), A Constructivist Framework for Integrating the Java Paradigm into the Undergraduate Curriculum, *ACM SIGCSE Bulletin*, Vol. 30, Issue 3, 105-107.
- Hadjerrouit, S. (1999), A constructivist approach to object-oriented design and programming, *ACM SIGCSE Bulletin*, Vol. 31, Issue 3, 171-174.
- Holland, S. Griffiths, R., Woodman, M. (1997), Avoiding object misconceptions. *SIGCSE Bulletin*, 29(1), 1997, 131-134.
- Ragonis, N. & Ben-Ari, M. (2005), On Understanding the Statics and Dynamics of Object-Oriented Programs. *SIGCSE Bulletin*, 37(1), 226-230.
- Sanders, K. & Thomas, L. (2007), Checklists for grading object-oriented CS1 programs: concepts and misconceptions, *SIGCSE Bull.* 39(3), 166-170.
- Tempte, M C. (1991), Let's Begin Introducing the Object-Oriented Paradigm, *ACM SIGCSE Bulletin*, Vol. 23, No. 1, 338-342.
- Thomasson, B., Ratcliffe, M. & Thomas, L. (2006), Identifying novice difficulties in object oriented design. *SIGCSE Bulletin*, 38(3), 28-32.
- Xinogalos, S. (2011), Object-Oriented Programming – What Do Students Think of Objects and Classes?, *Proceedings of the 14th IASTED International Conference on Computers and Advanced Technology in Education (CATE 2011)*, 11-13 July 2011, Cambridge, UK, 181-186.
- Xinogalos, S., Satratzemi, M. & Dagdilelis, V. (2006a). An Introduction to object-oriented programming with a didactic microworld: objectKarel, *Computers & Education*, Volume 47, Issue 2, September 2006, 148-171, Elsevier Publishers.
- Xinogalos, S., Satratzemi, M. (2005) Using Hands-on Activities for Motivating Students with OOP Concepts Before They Are Asked to Implement Them, *ACM SIGCSE Bulletin*, Vol. 37, Number 3, September 2005, 380.
- Xinogalos, S., Satratzemi, M. (2005), The Hands-on Activities of the Programming Microworld objectKarel, *ACM SIGCSE Bulletin*, Vol. 37, Number 3, September 2005, 384.
- Xinogalos, S., Satratzemi, M., & Dagdilelis, V. (2006b), Evaluating objectKarel - an educational

programming environment for object oriented programming, In A. Mendez-Vilas et al. (eds) *"Current Developments in Technology-Assisted Education"*, vol. 2, 821-825, Formatex press

Ξυνόγαλος, Σ. & Σατρατζέμη, Μ. (2004), Η Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό: Προβλήματα και Μεθοδολογίες για την Αντιμετώπισή τους, *Πρακτικά του 4ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση»*, Αθήνα, 29 Σεπτεμβρίου-3 Οκτωβρίου 2004, 133-142 (τόμος Β').

Ξυνόγαλος, Σ. (2002), «Εκπαιδευτική Τεχνολογία»: Ένας Διδακτικός Μικρόκοσμος για την Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό, *Διδακτορική Διατριβή*, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας, σελ. 569.

Ξυνόγαλος, Σ. (2003), Η διδακτική και μαθησιακή δραστηριότητα στον προγραμματιστικό μικρόκοσμο objectKarel, *Πρακτικά του 6ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Διδακτική των Μαθηματικών και Πληροφορική στην Εκπαίδευση»*, Βόλος, 10-13 Οκτωβρίου 2003, 317-326.

Ξυνόγαλος, Σ. (2006), Η διδασκαλία του αντικειμενοστρεφούς προγραμματισμού με τον μικρόκοσμο objectKarel: εμπειρίες, προβληματισμοί και προτάσεις, *Περιοδικό «ΘΕΜΑΤΑ στην Εκπαίδευση»* ειδικό αφιέρωμα: «Σύγχρονη έρευνα στη Διδακτική της Πληροφορικής: ερευνητικοί άξονες, μέθοδοι, τεχνικές, εργαλεία», 7:3, 305-327, Ελληνικά γράμματα.

Ξυνόγαλος, Σ. (2008), Μελέτη των Δυσκολιών των Φοιτητών για την Έννοια του «Αντικειμένου» στον Αντικειμενοστραφή Προγραμματισμό, *Πρακτικά 4ου Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής»*, Πάτρα 28-30 Μαρτίου, 91-100.

Ξυνόγαλος, Σ. (2012), Διδακτική Προσέγγιση για τον Αντικειμενοστρεφή Προγραμματισμό: εφαρμογή στο BlueJ, *Πρακτικά 6ου Πανελληνίου Συνεδρίου "Διδακτική της Πληροφορικής"*, Φλώρινα, 20-22 Απριλίου, 63-72.

Σατρατζέμη, Μ., Ξυνόγαλος, Σ. & Δαγδιλέλης, Β. (2006), «Εκπαιδευτικά Προγραμματιστικά Περιβάλλοντα για τη Διδασκαλία του Αντικειμενοστραφούς Προγραμματισμού: μια επισκόπηση», *Πρακτικά 5ου Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση»*, Θεσσαλονίκη, 5-8 Οκτωβρίου 2006, 899-906.

11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Οι προτεινόμενες δραστηριότητες για τους μαθητές εγγράφονται σε μια τυπική κονστρουκτιβιστική προσέγγιση: στους μαθητές προτείνονται μια σειρά από προβλήματα τα οποία καλούνται να επιλύσουν με τη στήριξη του διδάσκοντος.

Αναμένεται να παρατηρηθούν δυσκολίες από αυτές που περιγράφονται στις προηγούμενες παραγράφους της παρούσας ενότητας.

12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ

Δεν φαίνεται να έχουμε.

13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Προτείνεται οι μαθητές να οργανωθούν σε ομάδες των 2-3 (το πολύ) ατόμων.

Στους υπολογιστές πρέπει να είναι εγκατεστημένο το λογισμικό objectKarel (<http://users.uom.gr/~stelios/objectKarel.zip>). Ένας βιντεοπροβολέας θα ήταν χρήσιμος, χωρίς όμως να είναι απαραίτητος. Τα φύλλα εργασίας θα πρέπει να μοιραστούν στους μαθητές τόσο για την αποτελεσματικότερη διεξαγωγή των δραστηριοτήτων στην τάξη, όσο και για τη στήριξη των μαθητών στη μελέτη στο σπίτι.

14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Ανάλογα με τις διδακτικές ώρες που θα αποφασίσει να αφιερώσει ο εκπαιδευτικός και το βάθος στο οποίο θα αναλύσει τις σχετικές έννοιες, οι προτεινόμενες δραστηριότητες μπορούν να εμπλουτιστούν, όπως ενδεικτικά αναφέρεται στη συνέχεια.

Ο εκπαιδευτικός μπορεί να ζητήσει από τους μαθητές, στο πλαίσιο μιας άσκησης, να δημιουργήσουν έναν πίνακα (όπως αυτός που φαίνεται παρακάτω), στον οποίο να καταγράψουν για κάθε μέθοδο που εκτελεί το ρομπότ πώς μεταβάλλεται η κατάσταση του.

	<i>Κατάσταση αντικειμένου</i>			
<i>Εκτέλεση μεθόδου</i>	<i>Δρόμος</i>	<i>Λεωφόρος</i>	<i>Κατεύθυνση</i>	<i>Αριθμός beepers στην τσάντα</i>

Στο πλαίσιο της ίδιας ή διαφορετικής άσκησης μπορεί να ζητηθεί από τους μαθητές να αναφέρουν ένα παράδειγμα όπου ένα ρομπότ ανταποκρίνεται με διαφορετικό τρόπο στο ίδιο μήνυμα, καθώς επίσης και να προσδιορίσουν το λόγο γι' αυτό. Είναι σημαντικό να κατανοήσουν οι μαθητές ότι ένα αντικείμενο μπορεί να ανταποκριθεί διαφορετικά στο ίδιο

μήνυμα ανάλογα με την κατάσταση του και την κατάσταση του περιβάλλοντος στο οποίο δραστηριοποιείται. Τα αντικείμενα έχουν *συμπεριφορά* και δεν είναι απλά εγγραφές για την αποθήκευση κάποιων δεδομένων.

Τέλος, μπορεί να δοθεί η εκφώνηση ενός προβλήματος και το πρόγραμμα σε χαρτί και να ζητηθεί από τους μαθητές να εντοπίσουν τα λάθη, να αναφέρουν τον τύπο τους και να τα διορθώσουν. Αφού οι μαθητές γράψουν τις απαντήσεις τους σε χαρτί, ακολουθεί συζήτηση στην τάξη σχετικά με τις προτεινόμενες από τους μαθητές διορθώσεις. Στη συνέχεια, μπορεί να δοθεί στους μαθητές το αρχείο του προγράμματος, προκειμένου να επιβεβαιώσουν τις απαντήσεις τους. Οι μαθητές σημειώνουν στο χαρτί τα λάθη τους.

15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

Ανάλυση φύλλου εργασίας 1

Βήμα 1^ο - Δραστηριότητα 1: εκκίνηση του περιβάλλοντος. Το περιβάλλον δεν απαιτεί κάποια ιδιαίτερη διαδικασία εγκατάστασης. Αποσυμπίεζετε το αρχείο objectKarel.zip (<http://users.uom.gr/~stelios/objectKarel.zip>) και εκτελείτε το αρχείο objectKarel.exe.

Βήμα 2^ο: Περιγράφεται το αντικειμενοστραφές παράδειγμα προγραμματισμού που βασίζεται στις έννοιες Αντικείμενο και Μήνυμα (θεωρία ενότητας «Εισαγωγή»).

Παρουσιάζεται ο προγραμματιστικός μικρόκοσμος objectKarel και επισημαίνεται ότι τα μοναδικά αντικείμενα είναι τα ρομπότ (δραστηριότητα ενότητας «Εισαγωγή»).

Βήμα 3^ο: Επεξηγούνται τα 4 μηνύματα (move(), turnLeft(), putBeeper(), pickBeeper()) στα οποία ανταποκρίνεται κάθε ρομπότ (θεωρία ενότητας «Αντικείμενα»).

Βήμα 4^ο – Δραστηριότητα 2: Οι μαθητές στέλνουν τα 4 μηνύματα στο ρομπότ Karel, παρατηρούν το αποτέλεσμα στην κατάσταση του κόσμου και τη σύνταξη τους στη γλώσσα προγραμματισμού (δραστηριότητα ενότητας «Αντικείμενα»). Επισημαίνεται ότι κάθε αντικείμενο είναι αυτόνομο, έχει δηλαδή τη δική του «φυσική» υπόσταση και ταυτότητα, και γι' αυτό χρησιμοποιούμε πάντα το όνομα του αντικειμένου στο οποίο στέλνουμε το μήνυμα: <Όνομα-αντικειμένου>.Μήνυμα()

Βήμα 5^ο: Επισημαίνεται ότι τα ρομπότ που χρησιμοποίησαν οι μαθητές αποτελούν στιγμιότυπα ενός βασικού μοντέλου – κλάσης με όνομα Primitive_Robot. Γίνεται σύντομη αναφορά στη δήλωση κλάσης χρησιμοποιώντας ως παράδειγμα τη δήλωση της κλάσης Primitive_Robot (θεωρία ενότητας «Κλάσεις»).

Βήμα 6^ο: Επισημαίνεται ότι μια κλάση μπορεί να μας εφοδιάσει με όσα αντικείμενα θέλουμε, αρκεί να δώσουμε την κατάλληλη εντολή για τη δημιουργία και την αρχικοποίησή τους. Γίνεται αναφορά στις έννοιες:

- ιδιότητες, οι τιμές των οποίων μεταβάλλονται (μεταβάλλεται η κατάσταση του αντικειμένου) με την εκτέλεση των μεθόδων, και
- συμπεριφορά, επισημαίνοντας ότι ένα αντικείμενο μπορεί να ανταποκριθεί με διαφορετικό τρόπο στο ίδιο μήνυμα ανάλογα με την κατάστασή του .

Βήμα 7^ο – Δραστηριότητα 3: Οι μαθητές εξοικειώνονται με τις έννοιες που αναφέρονται στο προηγούμενο βήμα μέσω της δραστηριότητας 3.

Βήμα 8^ο – Δραστηριότητα 4: Δίνεται στους μαθητές η αρχική κατάσταση του κόσμου και ο πηγαίος κώδικας ενός προγράμματος και ζητείται από τους μαθητές να προσδιορίσουν:

- (1) Το αποτέλεσμα της εκτέλεσης του προγράμματος στην κατάσταση του κόσμου, ή με άλλα λόγια την τελική κατάσταση του κόσμου.

Προκειμένου να προσδιορίσουν οι μαθητές σωστά την τελική κατάσταση του κόσμου, θα πρέπει να έχουν κατανοήσει την εντολή δημιουργίας και αρχικοποίησης ρομπότ, το μηχανισμό της αποστολής μηνυμάτων σε αντικείμενα και τη σημασία των μηνυμάτων στα οποία μπορούν να ανταποκριθούν τα ρομπότ (αντικείμενα) του βασικού μοντέλου (κλάσης). Επίσης, οι μαθητές θα πρέπει να έχουν την ικανότητα ανίχνευσης της εκτέλεσης ενός προγράμματος στο χαρτί.

- (2) Τα ονόματα των αντικειμένων που υπάρχουν στο πρόγραμμα.

Οι μαθητές θα πρέπει να είναι σε θέση να διαχωρίζουν τις έννοιες κλάση και αντικείμενο. Απαντήσεις του τύπου “Primitive_Robot Nik” υποδηλώνουν την ύπαρξη της γνωστής παρανόησης ότι κλάση και αντικείμενο είναι έννοιες ταυτόσημες. Στην περίπτωση δε που η παρανόηση αυτή έχει εδραιωθεί μπορεί ακόμα και να θεωρηθεί από κάποιους μαθητές το πρόγραμμα λάθος, μιας και “κάθε κλάση μπορεί να χρησιμοποιηθεί για τη δημιουργία ενός αντικειμένου”.

- (3) Την αρχική και την τελική κατάσταση κάθε αντικειμένου.

Οι μαθητές πρέπει να γνωρίζουν ότι η κατάσταση ενός αντικειμένου καθορίζεται από τις τιμές των ιδιοτήτων του, καθώς επίσης και ότι αυτές αλλάζουν μέσω της εκτέλεσης μεθόδων.

- (4) Τα ονόματα των μηνυμάτων στα οποία μπορεί να ανταποκριθεί το κάθε αντικείμενο.

Οι μαθητές πρέπει να γνωρίζουν ότι κάθε αντικείμενο μπορεί να ανταποκριθεί στα μηνύματα για τα οποία έχουν οριστεί αντίστοιχες μέθοδοι στην κλάση από την οποία προέρχεται. Το γεγονός ότι μπορεί σε κάποια δεδομένη αποστολή ένα αντικείμενο μιας κλάσης να μην χρησιμοποιεί όλες τις διαθέσιμες μεθόδους – όπως στη συγκεκριμένη περίπτωση - δεν σημαίνει ότι δεν μπορεί να ανταποκριθεί στα αντίστοιχα μηνύματα. Η απάντηση των μαθητών σε αυτή την ερώτηση πρέπει να δοθεί σε συνάρτηση με τις γνώσεις τους για την κλάση που χρησιμοποιήθηκε ως πρότυπο για τη δημιουργία των 2 ρομπότ και όχι σε συνάρτηση με τα μηνύματα τα οποία στέλνονται σε αυτά στη συγκεκριμένη περίπτωση.

Ανάλυση φύλλου εργασίας 2

Δραστηριότητα 1: στο πλαίσιο αυτής της δραστηριότητας οι μαθητές εξοικειώνονται με τη διαδικασία ανάπτυξης, μεταγλώττισης, αποσφαλμάτωσης και εκτέλεσης ενός προγράμματος στο περιβάλλον του objectKarel.

Βήμα 1^ο: Περιγραφή του βασικού παραθύρου του περιβάλλοντος προγραμματισμού. Οι μαθητές είτε με καθοδήγηση είτε ακολουθώντας τις οδηγίες του φύλλου εργασίας προχωρούν στα παρακάτω βήματα. Οι οδηγίες που υπάρχουν στη 2^η στήλη του πίνακα είναι αναλυτικές και μπορούν να χρησιμοποιηθούν από τους μαθητές ως οδηγός αναφοράς/εγχειρίδιο του περιβάλλοντος και κατά την ανάπτυξη προγραμμάτων στο σπίτι.

Βήμα 2^ο: Ανάκληση ενός υπάρχοντος προγράμματος.

Βήμα 3^ο: Δημιουργία κατάστασης του κόσμου.

Βήμα 4^ο: Μεταγλώττιση του προγράμματος.

Βήμα 5^ο: Εκτέλεση του προγράμματος με όλους τους τρόπους. Έμφαση δίνεται στη βηματική εκτέλεση του προγράμματος. Οι μαθητές ενθαρρύνονται να εκτελούν βηματικά τα προγράμματά τους, να παρακολουθούν το αποτέλεσμα εκτέλεσης της κάθε εντολής χωριστά και να διαβάζουν τα μηνύματα που εμφανίζονται στο κάτω μέρος του παραθύρου για την επεξήγηση του αποτελέσματος της κάθε εντολής. Η βηματική εκτέλεση συγκρίνεται με την απευθείας εκτέλεση και επισημαίνεται ότι με τον δεύτερο τρόπο εκτέλεσης βλέπουμε μόνο το τελικό αποτέλεσμα και όχι τη διαδρομή που ακολουθεί ένα ρομπότ. Το ρομπότ μπορεί να έχει επιτύχει στην αποστολή που του ανατέθηκε, αλλά μπορεί να έχει κάνει περιττούς κύκλους, για παράδειγμα. Η βηματική εκτέλεση μας βοηθάει στην αξιολόγηση μιας λύσης σε ένα πρόβλημα. Επίσης, η βηματική εκτέλεση είναι απαραίτητη στην περίπτωση που το πρόγραμμά μας δεν δίνει σωστά αποτελέσματα, αφού μας βοηθάει να εντοπίσουμε το σημείο ή τα σημεία του προγράμματος όπου υπάρχει λάθος (το ρομπότ

μας δεν ενεργεί σωστά).

Βήμα 6^ο: Εισαγωγή σχολίου και κενής γραμμής στο πρόγραμμα. Επισημαίνεται η αξία των σχολίων, των κενών γραμμών και της στοίχισης και χρήσης κατάλληλων εσοχών σε ένα πρόγραμμα.

Βήμα 7^ο: Διαγραφή της εντολής δημιουργίας του ρομπότ. Πριν να μεταγλωττίσουν οι μαθητές το πρόγραμμα, ο εκπαιδευτικός ζητάει από τους μαθητές να προσδιορίσουν τι αποτέλεσμα θα έχει η διαγραφή της εντολής δημιουργίας του ρομπότ. Οι μαθητές θα πρέπει να είναι σε θέση να κατανοούν ότι δεν μπορούμε να στέλνουμε μήνυμα σε ένα ρομπότ που δεν υπάρχει. Γίνεται αναφορά στα σημασιολογικά λάθη.

Βήμα 8^ο: Μεταγλώττιση και αποσφαλμάτωση του προγράμματος. Εξηγούνται τα λάθη μεταγλώττισης που προκύπτουν και οι μαθητές καλούνται να διορθώσουν (δημιουργία του ρομπότ με όνομα Karel και κατάλληλη αρχική κατάσταση) και να μεταγλωττίσουν το πρόγραμμα.

Βήμα 9^ο: Διαγραφή του άγκιστρου '}' στο τέλος του προγράμματος, μεταγλώττιση και εκτέλεση. Γίνεται αναφορά στα συντακτικά λάθη και τον τρόπο αντιμετώπισής τους. Είναι πολύ σημαντικό να κατανοήσουν οι μαθητές ότι πρέπει να διαβάζουν προσεκτικά τα μηνύματα λάθους που αναφέρει ένα οποιοδήποτε προγραμματιστικό περιβάλλον, να προσπαθούν να τα κατανοήσουν και να εντοπίσουν την αιτία του λάθους και όχι να κάνουν - όπως συνηθίζεται - αλλαγές σε τυχαία σημεία του προγράμματος.

Βήμα 10^ο: Διαγραφή της τελευταίας εντολής, μεταγλώττιση και εκτέλεση. Επισημαίνεται ότι παρόλο που το πρόγραμμα εκτελείται επιτυχώς (χωρίς λάθη εκτέλεσης) δεν λύνει σωστά το πρόβλημα (το ρομπότ δεν έχει την ίδια κατεύθυνση). Αναφορά στα λογικά λάθη και τον έλεγχο ενός προγράμματος. Οι μαθητές πρέπει να μάθουν να ελέγχουν αν το τελικό αποτέλεσμα της εκτέλεσης ενός προγράμματος είναι το αναμενόμενο και όχι απλά να διαπιστώνουν αν ένα πρόγραμμα εκτελείται.

Βήμα 11^ο: Διαγραφή της εντολής move αμέσως πριν την εντολή pickBeeper και εκτέλεση του προγράμματος. Κατά την εκτέλεση του προγράμματος προκύπτει κλείσιμο λόγω λάθους, οπότε γίνεται αναφορά στα λάθη εκτέλεσης και τον τρόπο αντιμετώπισής τους. Γίνεται σύγκριση μεταξύ των διαφόρων τύπων λαθών – λάθη μεταγλώττισης (σημασιολογικά, συντακτικά), λογικά λάθη και λάθη εκτέλεσης – και του τρόπου εντοπισμού και διόρθωσής τους. Διόρθωση του προγράμματος, εισαγωγή δηλαδή της εντολής move που διαγράφηκε προηγουμένως, και αποθήκευση.

Βήμα 12^ο: Παρουσίαση του συστήματος βοήθειας του λογισμικού (επιλογή *Περιεχόμενα*

του μενού *Βοήθεια*).

Δραστηριότητες 2-7: μετά την εξοικείωση των μαθητών με τις σχετικές αντικειμενοστρεφής έννοιες και τη διαδικασία ανάπτυξης και αποσφαλμάτωσης προγραμμάτων, οι μαθητές αναπτύσσουν προγράμματα για την επίλυση προβλημάτων.

Συνιστάται να λυθεί τουλάχιστον μία από τις ασκήσεις στις οποίες χρησιμοποιούνται *περισσότερα από ένα αντικείμενα της βασικής κλάσης*, προκειμένου να αποφευχθεί η δημιουργία της παρανόησης ότι κλάση και αντικείμενο είναι έννοιες ταυτόσημες. Επίσης, οι μαθητές θα κατανοήσουν το λόγο για τον οποίο χρειάζεται να δίνουμε ένα διαφορετικό όνομα σε κάθε αντικείμενο που δημιουργούμε από κάθε κλάση. Επίσης, οι μαθητές με ενδεχόμενη εμπειρία σε μια διαδικαστική γλώσσα θα κατανοήσουν ευκολότερα ότι μια μέθοδος δεν ισοδυναμεί με ένα υποπρόγραμμα σε μια διαδικαστική γλώσσα προγραμματισμού. Δεν αρκεί να υπάρχει η κλάση για να καλέσουμε μια μέθοδο που έχει οριστεί σε αυτή. Πρέπει να έχουμε προηγουμένως δημιουργήσει ένα (τουλάχιστον) αντικείμενο μιας κλάσης και να του στείλουμε ένα μήνυμα, προκειμένου να εκτελέσει την αντίστοιχη μέθοδο.

Ανάλυση φύλλου εργασίας 3

Το φύλλο εργασίας 3 περιγράφεται στην αμέσως επόμενη ενότητα (16).

16. ΑΞΙΟΛΟΓΗΣΗ

Στο φύλλο εργασίας 3 περιλαμβάνεται – ενδεικτικά - μια σειρά ερωτήσεων πολλαπλής επιλογής, προκειμένου να ελέγξει ο εκπαιδευτικός αν οι μαθητές κατανόησαν τις έννοιες του αντικειμένου, της κλάσης και της αποστολής μηνυμάτων. Οι ερωτήσεις και οι απαντήσεις που έχουν συμπεριληφθεί στο φύλλο ελέγχου αντιπροσωπεύουν συνήθεις δυσκολίες και παρανοήσεις που έχουν καταγραφεί σε εισαγωγικά μαθήματα αντικειμενοστρεφούς προγραμματισμού.

Το τεστ (αυτό)αξιολόγησης μπορεί να συμπληρωθεί με σταυρόλεξα, SUDOKU, επίλυση προβλημάτων κτλ.

17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ

Θα δοθούν 3 φύλλα εργασίας, τα οποία παρατίθενται παρακάτω.

Αναλυτική περιγραφή και ανάλυση των φύλλων εργασίας δίνεται στην ενότητα 15.

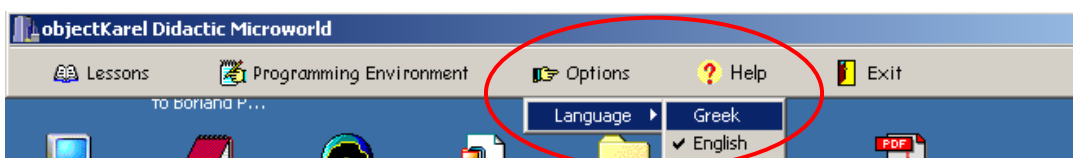
ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - Εξοικείωση με τα αντικείμενα και τις κλάσεις



Δραστηριότητα 1 – γνωριμία με το περιβάλλον objectKarel

(1) Εκκινήστε το περιβάλλον objectKarel

Πατήστε το κουμπί *Options* και επιλέξτε *Language* ▶ *Greek* (Εικόνα 1), προκειμένου να εμφανίζεται όλο το υλικό και να πραγματοποιείται ο διάλογος χρήστη-συστήματος στα Ελληνικά (Εικόνα 2).



Εικόνα 1. Το βασικό μενού του περιβάλλοντος objectKarel.



Εικόνα 2. Το βασικό μενού του περιβάλλοντος objectKarel.

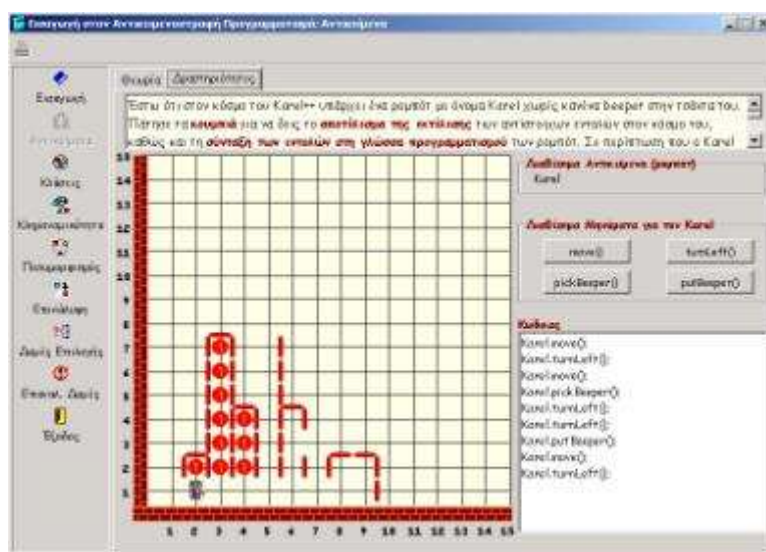
Οι επιλογές του βασικού μενού (Εικόνα 2) είναι οι εξής:

- **Μαθήματα:** σε κάθε ενότητα παρουσιάζεται η σχετική θεωρία και μια ή περισσότερες δραστηριότητες για την εξοικείωση με τις έννοιες της ενότητας.
- **Περιβάλλον Προγραμματισμού:** περιβάλλον για την ανάπτυξη, μεταγλώττιση, αποσφαλμάτωση και εκτέλεση προγραμμάτων.
- **Επιλογές:** επιλογή της γλώσσας (Ελληνική ή Αγγλική) που θα χρησιμοποιείται στο περιβάλλον.
- **Βοήθεια:** παρέχεται βοήθεια για όλες τις λειτουργίες του περιβάλλοντος.
- **Έξοδος:** έξοδος από το περιβάλλον.



Δραστηριότητα 2 – αποστολή μηνυμάτων σε ένα ρομπότ

- (1) Στον χώρο των μαθημάτων κάντε κλικ στην καρτέλα «Δραστηριότητες» της ενότητας «Αντικείμενα» και διαβάστε την περιγραφή της (Εικόνα 3).
- (2) Στείλτε στο ρομπότ Karel τα κατάλληλα μηνύματα προκειμένου να καθαρίσει τους διαδρόμους από τα σκουπίδια και να τα μεταφέρει όλα στη θέση (1,9).



Εικόνα 3. Η δραστηριότητα της ενότητας «Αντικείμενα».

Παρατηρήσεις:



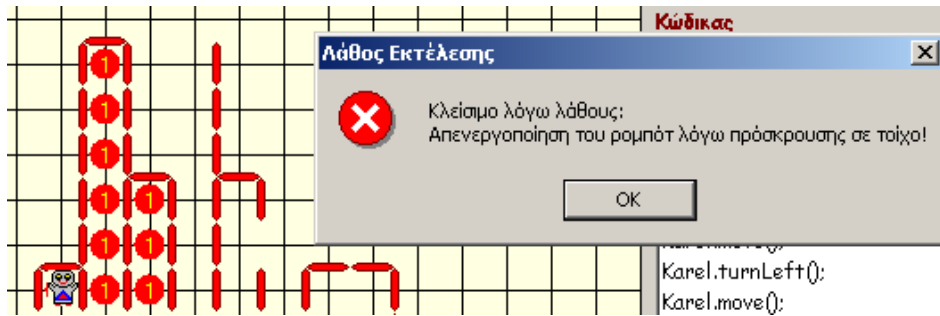
Μπορείτε να στείλετε στο ρομπότ Karel τα απαιτούμενα μηνύματα πατώντας τα κουμπιά με τις ετικέτες `move()`, `turnLeft()`, `pickBeeper()` και `putBeeper()`.



Κάθε φορά που στέλνεται στο ρομπότ Karel ένα μήνυμα παρατηρήστε πώς ανταποκρίνεται σε αυτό, καθώς επίσης και ποια είναι η μορφή της αντίστοιχης εντολής στη γλώσσα προγραμματισμού των ρομπότ (στο πλαίσιο Κώδικας).



Προσπαθήστε να κατευθύνετε τον Karel να ολοκληρώσει την αποστολή του χωρίς να προκληθεί κάποιο λάθος εκτέλεσης (για παράδειγμα, αν ενώ μπροστά από τον Karel υπάρχει τμήμα τοίχου και εσείς στέλνοντας του το μήνυμα `move()` του πείτε να πέσει πάνω στον τοίχο!). Ωστόσο, αν προκύψει κάποιο λάθος εκτέλεσης, διαβάστε το σχετικό μήνυμα που εμφανίζεται και προσπαθήστε να καταλάβετε γιατί προέκυψε το λάθος (Εικόνα 4).



Εικόνα 4. Πλαίσιο ενημέρωσης για λάθος εκτέλεσης.

Αφού ολοκληρώσετε τη δραστηριότητα προσπαθήστε να απαντήσετε στις παρακάτω ερωτήσεις:

- (1) Ποια είναι τα μηνύματα τα οποία αντιλαμβάνεται ένα ρομπότ του βασικού μοντέλου και πως ανταποκρίνεται σε κάθε ένα από αυτά;
- (2) Ένα ρομπότ του βασικού μοντέλου ανταποκρίνεται πάντα με τον ίδιο τρόπο σε ένα μήνυμα, έστω στο μήνυμα move;
- (3) Με ποιο τρόπο στέλνουμε ένα μήνυμα σε ένα ρομπότ του βασικού μοντέλου;
- (4) Υπάρχει περίπτωση αποτυχίας ενός ρομπότ στην εκτέλεση μιας αποστολής; Δώστε ένα παράδειγμα.



Αν δυσκολεύεστε να απαντήσετε τις παραπάνω ερωτήσεις μελετήστε τη θεωρία των ενοτήτων «Εισαγωγή» και «Αντικείμενα».



Δραστηριότητα 3 – δημιουργία ρομπότ

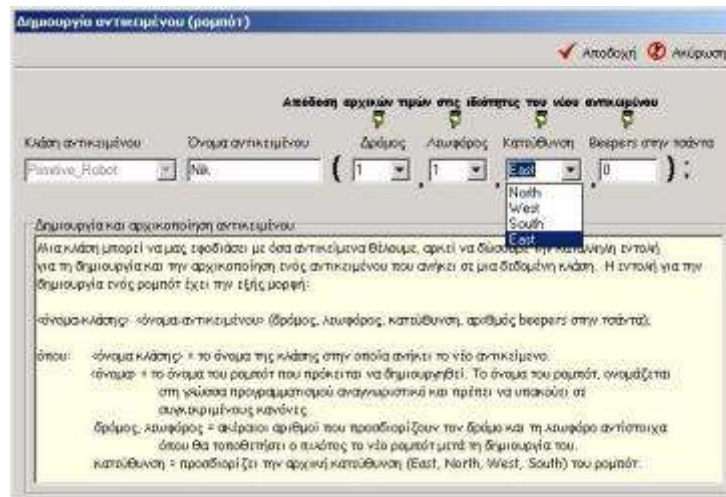
Στη δραστηριότητα υπήρχε διαθέσιμο ένα ρομπότ με όνομα Karel στο οποίο στείλατε τα κατάλληλα μηνύματα για την εκτέλεση της αποστολής. Στην πραγματικότητα, κάθε φορά που χρειαζόμαστε ένα ρομπότ του βασικού μοντέλου θα πρέπει να ζητήσουμε τη δημιουργία του με τον τρόπο που περιγράφεται στα βήματα που ακολουθούν.

- (1) Στο χώρο των μαθημάτων κάντε κλικ στην καρτέλα «Δραστηριότητες» της ενότητας «Κλάσεις» και διαβάστε την περιγραφή της δραστηριότητας (Εικόνα 5).



Εικόνα 5. Η δραστηριότητα της ενότητας «Κλάσεις».

- (2) Ακολουθώντας της οδηγίες της δραστηριότητας πατήστε το κουμπί «Δημιουργία και Αρχικοποίηση Αντικείμενου» προκειμένου να δημιουργήσετε ένα ρομπότ (αντικείμενο) του βασικού μοντέλου (κλάσης).
- (3) Στο πλαίσιο διαλόγου που εμφανίζεται (Εικόνα 6) δώστε στο πεδίο «Όνομα Αντικείμενου» το όνομα που θέλετε να έχει το ρομπότ που θα δημιουργηθεί και καθορίστε τις αρχικές τιμές των ιδιοτήτων του ρομπότ, ή αλλιώς την αρχική του κατάσταση. Στο κάτω μέρος του πλαισίου υπάρχουν σχετικές πληροφορίες.

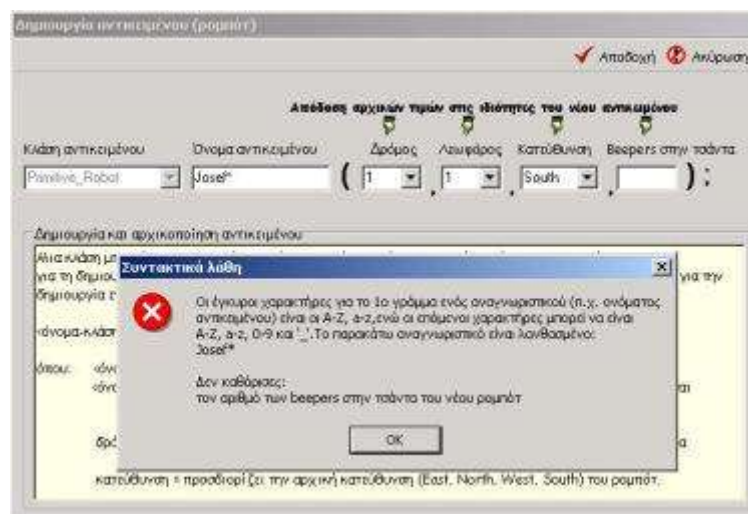


Εικόνα 6. Δημιουργία ρομπότ.

(4) Όταν συμπληρώσετε τα απαραίτητα στοιχεία για τη δημιουργία και την αρχικοποίηση του νέου ρομπότ πατήστε το κουμπί «Αποδοχή».



Αν υπάρχουν λάθη στην εντολή δημιουργίας του ρομπότ τότε διαβάστε προσεκτικά τις οδηγίες που εμφανίζονται (Εικόνα 7) και διορθώστε κατάλληλα.



Εικόνα 7. Αναφορά λάθους για την εντολή δημιουργίας ρομπότ.

Αν η εντολή είναι σωστή τότε το πλαίσιο διαλόγου κλείνει και η καρτέλα της δραστηριότητας έχει ενημερωθεί με (Εικόνα 8):

- το όνομα του νέου ρομπότ,
- τις αρχικές τιμές των ιδιοτήτων του,
- τα μηνύματα στα οποία μπορεί να ανταποκριθεί το ρομπότ, και
- τη μορφή των ενεργειών μας (δημιουργία αντικειμένου, αποστολή μηνυμάτων) στη

γλώσσα προγραμματισμού. Σ' αυτή τη δραστηριότητα παρουσιάζεται στον μαθητή και η μορφή του κυρίως προγράμματος.



Εικόνα 8. Παρουσίαση της κατάστασης (ιδιότητες) κατά την αποστολή μηνυμάτων σε ένα ρομπότ.

- (5) Στείλτε στο ρομπότ που δημιουργήσατε τα κατάλληλα μηνύματα προκειμένου να καθαρίσει τους διαδρόμους από τα σκουπίδια και να τα μεταφέρει όλα στη θέση (1,9).



Κάθε φορά που στέλνεται στο ρομπότ ένα μήνυμα παρατηρήστε πώς ανταποκρίνεται σε αυτό και πώς αλλάζουν οι τιμές των ιδιοτήτων του (η κατάστασή του).

Παρατηρήστε επίσης ποια είναι η μορφή του ολοκληρωμένου προγράμματος.



Αφού ολοκληρώσετε την δραστηριότητα προσπαθήστε να απαντήσετε στις παρακάτω ερωτήσεις:

- (1) Ποια είναι η σχέση μεταξύ ενός μοντέλου ρομπότ (κλάσης) και ενός συγκεκριμένου ρομπότ (αντικειμένου);
- (2) Πόσα αντικείμενα-ρομπότ μπορούμε να δημιουργήσουμε από μία κλάση;
- (3) Τι χρειάζεται να κάνουμε για να δημιουργήσουμε ένα αντικείμενο-ρομπότ;
- (4) Ποιες είναι οι ιδιότητες ενός ρομπότ του μοντέλου (κλάσης) Primitive_Robot;
- (5) Με ποιο τρόπο μεταβάλλονται οι τιμές των ιδιοτήτων ενός ρομπότ, ή με άλλα λόγια η κατάστασή του;
- (6) Ένα αντικείμενο (ρομπότ) ανταποκρίνεται σε οποιοδήποτε μήνυμα του στείλουμε; Ποιος καθορίζει σε ποια μηνύματα μπορεί να ανταποκριθεί ένα ρομπότ;

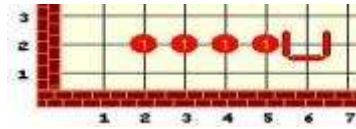


Αν δυσκολεύεστε να απαντήσετε τις παραπάνω ερωτήσεις μελετήστε τη θεωρία της ενότητας «Κλάσεις».



Δραστηριότητα 4 – κατανόηση προγράμματος

Θεωρώντας ως δεδομένα την αρχική κατάσταση του κόσμου και τον κώδικα του προγράμματος που παρουσιάζονται παρακάτω, απαντήστε στις ερωτήσεις που ακολουθούν:



Αρχική κατάσταση του κόσμου

Πηγαίος κώδικας:

```

1  task
2  {
3      Primitive_Robot Nik ( 2, 1, East, 0 );
4      Primitive_Robot Maria ( 3, 3, South, 0 );
5      Nik.move();
6      Nik.pickBeeper();
7      Nik.move();
8      Nik.putBeeper();
9      Maria.move();
10     Maria.pickBeeper();
11     Maria.pickBeeper();
12     Maria.turnLeft();
13     Maria.move();
14     Maria.pickBeeper();
15     Maria.move();
16     Maria.pickBeeper();
17     Maria.turnLeft();
18     Maria.move();
19     Maria.turnLeft();
20     Maria.turnLeft();
21     Maria.turnLeft();
22     Maria.move();
23     Maria.turnLeft();
24     Maria.turnLeft();
25     Maria.turnLeft();
26     Maria.move();
27     Maria.putBeeper();
28     Maria.putBeeper();
29     Maria.putBeeper();
30     Maria.putBeeper();
31     Maria.turnLeft();
32     Maria.turnLeft();
33     Maria.move();
34 }
    
```

Να προσδιορίσετε:

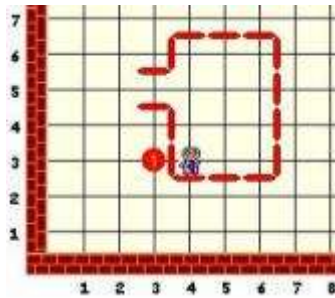
- (1) Το αποτέλεσμα της εκτέλεσης του προγράμματος στην κατάσταση του κόσμου, ή με άλλα λόγια την τελική κατάσταση του κόσμου.
- (2) Τα ονόματα των αντικειμένων που υπάρχουν στο πρόγραμμα.
- (3) Την αρχική και την τελική κατάσταση κάθε αντικειμένου.
- (4) Τα ονόματα των μηνυμάτων στα οποία μπορεί να ανταποκριθεί το κάθε αντικείμενο.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Προγραμματίζοντας ρομπότ



Δραστηριότητα 1 – ανάπτυξη, μεταγλώττιση, αποσφαλμάτωση και εκτέλεση προγράμματος

Κάθε πρωί ο Karel ξυπνάει στο κρεβάτι του όταν ένα ρομπότ-εφημεριδοπώλης του πετάει την εφημερίδα, που αναπαριστάμε με ένα beeper, στην μπροστινή βεράντα του σπιτιού του. Προγραμματίσε τον Karel ώστε να μαζέψει την εφημερίδα και να την φέρει πίσω στο κρεβάτι του. Η αρχική κατάσταση φαίνεται παρακάτω, ενώ στην τελική κατάσταση ο Karel πρέπει να βρίσκεται πίσω στο κρεβάτι του (ίδια διασταύρωση & κατεύθυνση) με την εφημερίδα.



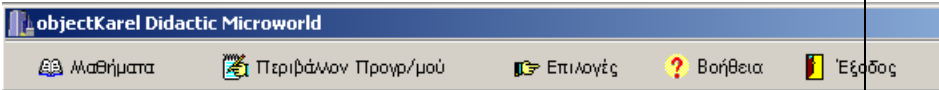
Αρχική κατάσταση.

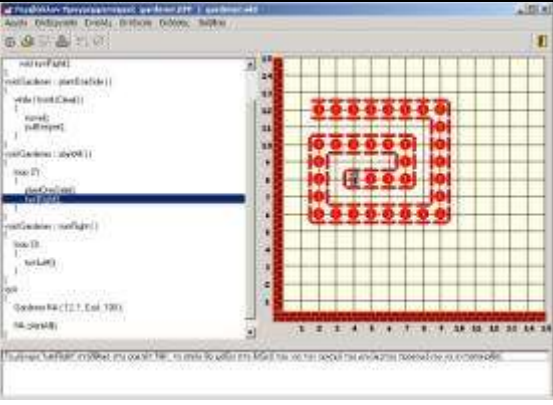
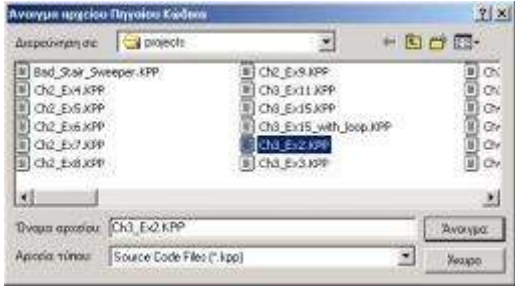
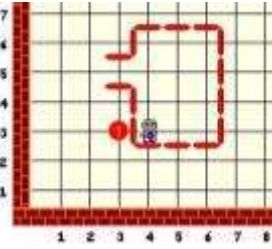
Ο πηγαίος κώδικας του προγράμματος:

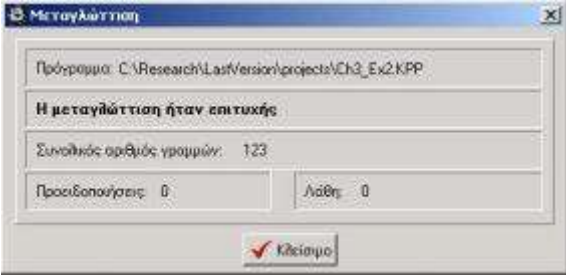
```
1 task
2 {
3     Primitive_Robot Karel ( 3, 4, West, 0 );
4     Karel.turnLeft();
5     Karel.turnLeft();
6     Karel.turnLeft();
7     Karel.move();
8     Karel.move();
9     Karel.turnLeft();
10    Karel.move();
11    Karel.move();
12    Karel.turnLeft();
13    Karel.move();
14    Karel.move();
15    Karel.turnLeft();
```

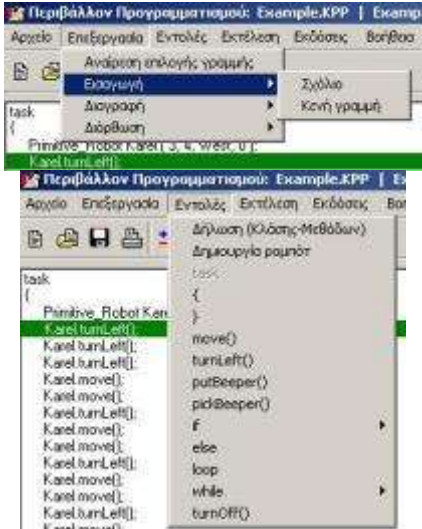




```
16     Karel.move();
17     Karel.pickBeeper();
18     Karel.turnLeft();
19     Karel.move();
20     Karel.turnLeft();
21     Karel.move();
22     Karel.turnLeft();
23     Karel.turnLeft();
24     Karel.turnLeft();
25     Karel.move();
26     Karel.turnLeft();
27     Karel.turnLeft();
28     Karel.turnLeft();
29     Karel.move();
30     Karel.move();
31     Karel.turnLeft();
32     Karel.turnLeft();
33     Karel.turnLeft();
34     Karel.move();
35     Karel.move();
36     Karel.turnLeft();
37     Karel.turnLeft();
38     Karel.turnLeft();
39 }
```

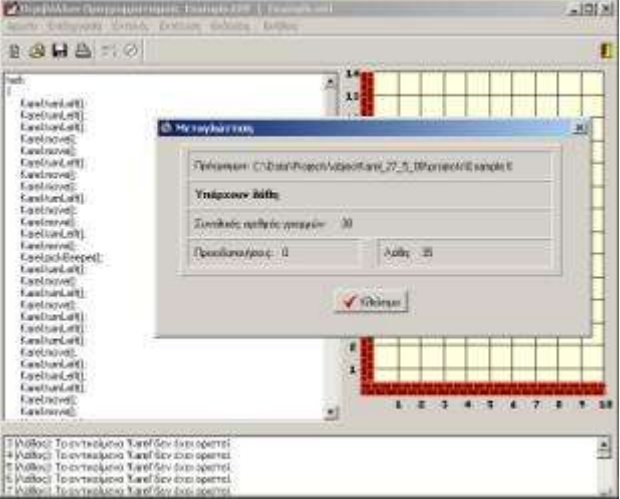

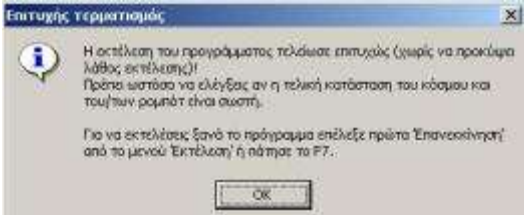

Εκτελέστε τα παρακάτω βήματα, ακολουθώντας τις οδηγίες που υπάρχουν στη 2^η στήλη του πίνακα:

<p>Βήμα 1^ο: Εκκινήστε το περιβάλλον προγραμματισμού</p>	<p>Από το βασικό μενού του περιβάλλοντος επιλέξτε «Περιβάλλον Προγραμματισμού».</p>  <p>Από το βασικό παράθυρο του περιβάλλοντος προγραμματισμού παρέχεται η δυνατότητα:</p> <ul style="list-style-type: none">▪ ανάπτυξης, αποθήκευσης, ανάκλησης και διόρθωσης,▪ μεταγλώττισης,▪ αποσφαλμάτωσης &▪ εκτέλεσης <p>ενός προγράμματος.</p>
---	---

	
<p>Βήμα 2^ο: Ανοίξτε το αρχείο με τον πηγαίο κώδικα του προγράμματος</p>	<p>(1) Από το μενού <i>Αρχείο</i> επιλέξτε <i>Άνοιγμα αρχείου</i> ► <i>Πηγαίου Κώδικα</i>, οπότε εμφανίζεται το παρακάτω πλαίσιο διαλόγου.</p>  <p>(2) Κάντε κλικ στο όνομα του αρχείου που θέλετε να ανοίξετε. (3) Πατήστε το κουμπί <i>Άνοιγμα</i>, οπότε εμφανίζεται ο πηγαίος κώδικας στο αριστερό μέρος του παραθύρου.</p> <p>Παρατήρηση: τα προγράμματα αποθηκεύονται εξ' ορισμού στον φάκελο <i>projects</i> (ο οποίος βρίσκεται μέσα στο φάκελο <i>objectKarel</i>) και έχουν επέκταση <i>kpp</i>.</p>
<p>Βήμα 3^ο: Δημιουργήστε την αρχική κατάσταση του κόσμου</p>	<ul style="list-style-type: none"> Δημιουργήστε στον κόσμο των ρομπότ την αρχική κατάσταση:  Από το μενού <i>Αρχείο</i> επιλέξτε <i>Αποθήκευση</i> ► <i>Πηγαίου Κώδικα – Κατάστασης του κόσμου</i>, έτσι ώστε να αποθηκευτεί η κατάσταση του κόσμου σε ένα αρχείο με το ίδιο όνομα με το αρχείο πηγαίου κώδικα (και επέκταση <i>wld</i>). <p>Παρατήρηση: αν για ένα πρόβλημα έχετε αποθηκεύει τον πηγαίο κώδικα και την κατάσταση του κόσμου με το ίδιο όνομα, για να τα ανοίξετε και τα δύο μαζί από το μενού <i>Αρχείο</i> επιλέξτε <i>Άνοιγμα αρχείου</i> ► <i>Πηγαίου Κώδικα- Κατάστασης του Κόσμου</i>.</p>

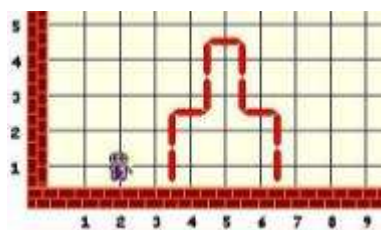
	<p>Για την προσθήκη:</p> <ul style="list-style-type: none"> ▪ οριζόντιου ή κατακόρυφου τμήματος τοίχου κάνουμε κλικ με το ποντίκι πάνω στη λεωφόρο ή το δρόμο αντίστοιχα, μεταξύ των κατάλληλων διασταυρώσεων. ▪ beepers κάνουμε κλικ με το ποντίκι πάνω στην κατάλληλη διασταύρωση τόσες φορές όσα είναι και τα beepers που θέλουμε να προσθέσουμε. <p>Για τη διαγραφή:</p> <ul style="list-style-type: none"> ▪ οριζόντιου ή κατακόρυφου τμήματος τοίχου κάνουμε κλικ με το ποντίκι πάνω του. ▪ beepers κάνουμε δεξί κλικ με το ποντίκι πάνω στην κατάλληλη διασταύρωση τόσες φορές όσα είναι και τα beepers που θέλουμε να διαγράψουμε.
<p>Βήμα 4^ο: Μεταγλώττιστε το πρόγραμμα</p>	<p>Για την μεταγλώττιση του τρέχοντος προγράμματος επιλέξτε <i>Μεταγλώττιση</i> από το μενού <i>Εκτέλεση</i>, οπότε εμφανίζεται ένα πλαίσιο διαλόγου με τα αποτελέσματα της μεταγλώττισης. Εφόσον η μεταγλώττιση είναι επιτυχής μπορείτε να εκτελέσετε το πρόγραμμα.</p>  <p>Παρατήρηση: Αν δεν έχουμε αποθηκεύσει τον πηγαίο κώδικα εμφανίζεται το παράθυρο «Αποθήκευση αρχείου πηγαίου κώδικα» προκειμένου να αποθηκεύσουμε τον πηγαίο κώδικα.</p>
<p>Βήμα 5^ο: Εκτελέστε το πρόγραμμα</p>	<p>(1) Επιλέξτε <i>Εκτέλεση Βήμα προς Βήμα</i> από το μενού <i>Εκτέλεση</i> ή πατήστε το <i>F8</i> από το πληκτρολόγιο για την εκτέλεση της επόμενης γραμμής κώδικα. Η τρέχουσα εντολή σηματοδοτείται με αλλαγή του χρώματος του υποβάθρου της, ενώ το αποτέλεσμα της εκτέλεσής φαίνεται στην κατάσταση του κόσμου. Επιπλέον, στο κάτω μέρος του παραθύρου επεξηγείται η εντολή.</p> <p>(2) Για να εκτελέσετε ξανά το πρόγραμμα επιλέξτε <i>Επανεκκίνηση</i> από το μενού <i>Εκτέλεση</i> ή πατήστε το <i>F7</i>.</p>
<p>Βήμα 6^ο: Εισάγετε μια κενή γραμμή και ένα σχόλιο μεταξύ της εντολής δημιουργίας του ρομπότ και των εντολών αποστολής μηνυμάτων σε αυτό</p>	<p>Η ανάπτυξη ενός προγράμματος στο περιβάλλον του objectKarel γίνεται επιλέγοντας την κατάλληλη εντολή από το μενού <i>Εντολές</i>. Κάθε εντολή προστίθεται στο τέλος του προγράμματος. Αν θέλετε να προσθέσετε εντολή μεταξύ άλλων τότε:</p> <p>(1) Επιλέξτε με το ποντίκι τη γραμμή στην οποία θέλετε να εισαχθεί η εντολή που θα προστεθεί. Η επιλογή της γραμμής σηματοδοτείται με αλλαγή του χρώματος του υποβάθρου της από λευκό σε πράσινο και των χαρακτήρων από μαύρο σε λευκό.</p> <p>(2) Επιλέξτε από το μενού <i>Επεξεργασία</i> να εισάγετε σχόλιο ή κενή γραμμή ή από το μενού <i>Εντολές</i> την εντολή που θέλετε να εισαχθεί στην επιλεγμένη γραμμή. Οι εντολές από την</p>

	<p>επιλεγμένη γραμμή και μετά μετακινούνται προς τα κάτω.</p>  <p>Παρατήρηση: Αν έχετε επιλέξει κάποια γραμμή και θέλετε να αναιρέσετε την ενέργεια, επιλέξτε <i>Αναίρεση επιλογής γραμμής</i> από το μενού <i>Επεξεργασία</i> ή απλά πατήστε το κουμπί  της γραμμής εργαλείων.</p>
<p>Βήμα 7^ο: Διαγράψτε την εντολή δημιουργίας του ρομπότ</p>	<p>(1) Επιλέξτε με το ποντίκι τη γραμμή που θέλετε να διαγράψετε (όπως αναφέρθηκε παραπάνω αλλάζει το χρώμα του υποβάθρου της από λευκό σε πράσινο, δείτε την εικόνα του βήματος 6).</p> <p>(2) Επιλέξτε <i>Διαγραφή Γραμμής</i> από το μενού <i>Επεξεργασία</i> ή απλά πατήστε το κουμπί  της γραμμής εργαλείων.</p>  <p>Τι αποτέλεσμα πιστεύετε ότι θα έχει η διαγραφή της εντολής δημιουργίας και αρχικοποίησης του ρομπότ;</p>
<p>Βήμα 8^ο: Μεταγλωττίστε και αποσφαλματώστε το πρόγραμμα</p>	 <p>Τι παρατηρείτε όταν μεταγλωττίσετε το πρόγραμμα; Στο πλαίσιο διαλόγου που εμφανίζεται αναφέρεται ότι υπάρχουν 35 λάθη. Είναι κατανοητό που οφείλονται τα λάθη αυτά; Διαβάστε τα μηνύματα λάθους που αναφέρονται στο κάτω μέρος του παραθύρου; Γιατί αναφέρει 35 φορές το ίδιο λάθος; Κάντε διπλό κλικ πάνω σε κάποιο μήνυμα λάθους. Τι παρατηρείτε; Σε τι αναφέρεται ο αριθμός με τον οποίο ξεκινάει ένα μήνυμα λάθους;</p>

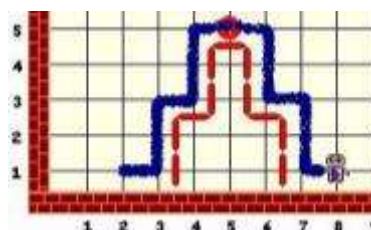
	 <p>Διορθώστε το πρόγραμμα, εισάγοντας ξανά την εντολή δημιουργίας και αρχικοποίησης του ρομπότ. Μπορούμε να δώσουμε στο ρομπότ οποιοδήποτε όνομα, για παράδειγμα Νίκος, ή στη συγκεκριμένη περίπτωση υπάρχει κάποιος περιορισμός; Μεταγλωττίστε και εκτελέστε το πρόγραμμα.</p>
<p>Βήμα 9^ο: Διαγράψτε το άγκιστρο '}' στο τέλος του προγράμματος και μεταγλωττίστε το πρόγραμμα</p>	<p>Διαβάστε το μήνυμα λάθους στο κάτω τμήμα του παραθύρου. Κάντε διπλό κλικ πάνω στο μήνυμα λάθους και παρατηρήστε σε ποια γραμμή του προγράμματος εντοπίζεται το λάθος. Διορθώστε το συντακτικό λάθος, μεταγλωττίστε και εκτελέστε το πρόγραμμα.</p> 
<p>Βήμα 10^ο: Διαγράψτε την τελευταία εντολή του προγράμματος, μεταγλωττίστε και εκτελέστε το πρόγραμμα</p>	<p>Τι παρατηρείτε μετά την ολοκλήρωση της εκτέλεσης του προγράμματος; Δίνει το προγράμμα μας σωστά αποτελέσματα; Διαβάστε τις οδηγίες που δίνονται στο ενημερωτικό πλαίσιο που εμφανίζεται και απαντήστε.</p> 
<p>Βήμα 11^ο: Διαγράψτε την εντολή move (γραμμή 16) αμέσως πριν την εντολή rickBeerer, μεταγλωττίστε και εκτελέστε το πρόγραμμα</p>	<p>Τι λάθος προκύπτει; Συγκρίνετε τα λάθη που προέκυψαν στα βήματα 8 έως 11. Κατανοείτε τους διαφορετικούς τύπου λαθών που μπορούν να προκύψουν σε ένα πρόγραμμα και τους τρόπους με τους οποίους τα αντιμετωπίζουμε;</p> 

Δραστηριότητα 2

Τα τμήματα τοίχου στην *Εικόνα 1* αναπαριστούν ένα βουνό. Γράψε ένα πρόγραμμα για τη δημιουργία ενός ρομπότ (στιγμιότυπου/αντικειμένου) της κλάσης `Primitive_Robot` και την αποστολή των κατάλληλων μηνυμάτων (ή αλλιώς την κλήση των κατάλληλων μεθόδων), έτσι ώστε το ρομπότ να ανέβει στην κορυφή του βουνού και να τοποθετήσει μια σημαία, που αναπαριστάται με ένα `beeper`. Στη συνέχεια το ρομπότ πρέπει να κατέβει από την άλλη μεριά του βουνού. Θυμήσου ότι ένα ρομπότ της κλάσης `Primitive_Robot` δεν είναι ένα `super-ρομπότ` για να πηδήξει με ένα άλμα στην κορυφή, να τοποθετήσει την σημαία και να πηδήξει κάτω με ένα άλμα. Όπως φαίνεται στην τελική κατάσταση το ρομπότ πρέπει να ακολουθήσει τη διαδρομή του βουνού ανεβαίνοντας και κατεβαίνοντας.



Αρχική κατάσταση.

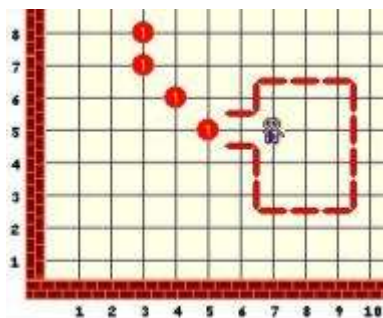


Η τελική κατάσταση και η διαδρομή που ακολουθεί το ρομπότ.

Εικόνα 1. Η αποστολή αναρρίχησης στο βουνό και τοποθέτησης της σημαίας.

Δραστηριότητα 3

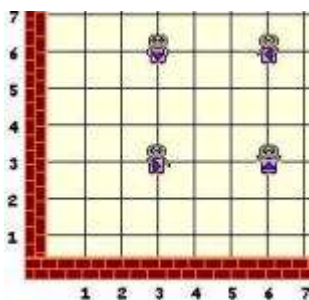
Επιστρέφοντας στο σπίτι από το `super market`, η σακούλα του ρομπότ της *Εικόνας 2* σκίστηκε και κάποια από τα ψώνια του έπεσαν στο δρόμο. Η αρχική κατάσταση όταν το ρομπότ ανακάλυψε ότι η σακούλα του έχει σχιστεί φαίνεται στην *Εικόνα 2*. Γράψε ένα πρόγραμμα για τη δημιουργία ενός ρομπότ (στιγμιότυπου/αντικειμένου) της κλάσης `Primitive_Robot` και την αποστολή των κατάλληλων μηνυμάτων (ή αλλιώς την κλήση των κατάλληλων μεθόδων), έτσι ώστε το ρομπότ να συγκεντρώσει όλα τα προϊόντα (αναπαριστώνται με `beepers`) που έπεσαν από την σακούλα και να επιστρέψει στη θέση που ήταν όταν ανακάλυψε τη διαρροή.



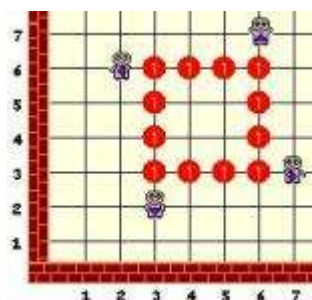
Εικόνα 2. Η αρχική κατάσταση της αποστολής αποκομιδής των χαμένων προϊόντων

Δραστηριότητα 4

Η ομάδα των ρομπότ της *Εικόνας 3* αποτελεί ένα συνεργείο που στρώνει πλακάκια. Η αποστολή τους στη συγκεκριμένη περίπτωση είναι να στρώσουν πλακάκια (beepers) σε ένα κήπο δημιουργώντας ένα διάδρομο με το μέγεθος και το σχήμα που φαίνεται στην τελική κατάσταση της *Εικόνας 3*. Γράψε ένα πρόγραμμα για τη δημιουργία τεσσάρων ρομπότ (στιγμιότυπων/αντικειμένων) της κλάσης `Primitive_Robot` και την αποστολή στη συνέχεια των κατάλληλων μηνυμάτων (ή αλλιώς την κλήση των κατάλληλων μεθόδων) σε κάθε ένα από τα ρομπότ για την εκτέλεση αυτής της αποστολής. Τελειώνοντας την αποστολή τα 4 ρομπότ δεν πρέπει να βρίσκονται πάνω στο διάδρομο με τα πλακάκια (όχι όμως απαραίτητα στις θέσεις που φαίνονται στην τελική κατάσταση της *Εικόνας 3*).



Αρχική κατάσταση.



Τελική κατάσταση.

Εικόνα 3. Δημιουργία διαδρόμου από πλακάκια.

Δραστηριότητα 5

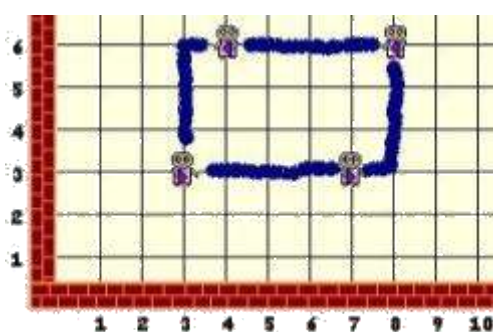
Μια ομάδα 4 ρομπότ προπονείται για το αγώνισμα της σκυταλοδρομίας. Η διαδρομή που ακολουθούν τα ρομπότ φαίνεται στην *Εικόνα 4*. Συγκεκριμένα:

- Τον αγώνα ξεκινάει το ρομπότ που βρίσκεται στη διασταύρωση (3, 3) έχοντας τη σκυτάλη (beeper) στην τσάντα του. Όταν το ρομπότ φτάσει στη διασταύρωση (3, 7)

αφήνει κάτω τη σκυτάλη

- Το 2^ο ρομπότ που βρίσκεται στη διασταύρωση (3, 7) σηκώνει τη σκυτάλη και συνεχίζει τη διαδρομή μέχρι τη διασταύρωση (6, 8) οπότε αφήνει κάτω τη σκυτάλη.
- Ο αγώνας συνεχίζεται με τον ίδιο τρόπο και τελειώνει όταν το 4^ο ρομπότ φτάσει στη διασταύρωση (3, 3) με τη σκυτάλη στην τσάντα του.

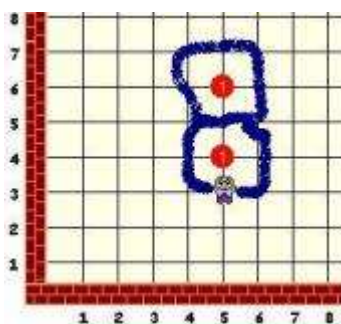
Γράψε ένα πρόγραμμα για τη δημιουργία τεσσάρων ρομπότ (στιγμιότυπων/αντικειμένων) της κλάσης `Primitive_Robot` και την αποστολή στη συνέχεια των κατάλληλων μηνυμάτων (ή αλλιώς την κλήση των κατάλληλων μεθόδων) σε κάθε ένα από τα ρομπότ για την εκτέλεση αυτής της αποστολής.



Εικόνα 4. Το αγώνισμα της σκυταλοδρομίας.

Δραστηριότητα 6

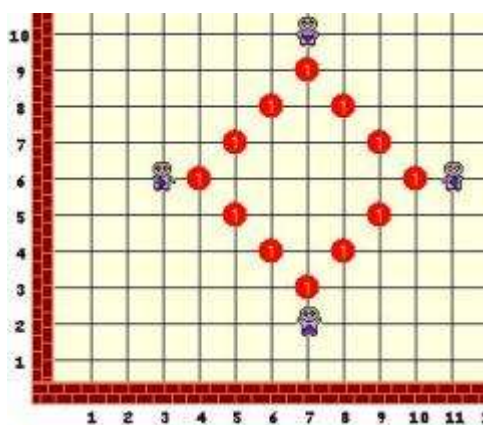
Το ρομπότ της Εικόνας 5 προπονείται για τους Ολυμπιακούς αγώνες των Ρομπότ. Ένα από τα αγωνίσματα απαιτεί από τους αθλητές να κινούνται γύρω από δύο beepers σχηματίζοντας οχτάρια, όπως φαίνεται στην Εικόνα 5. Γράψε ένα πρόγραμμα που καθοδηγεί το ρομπότ στην εκτέλεση αυτού του αγωνίσματος όσο το δυνατό γρηγορότερα (εκτελώντας όσο το δυνατό λιγότερες εντολές). Το ρομπότ πρέπει να σταματήσει στο ίδιο σημείο από όπου ξεκίνησε και να βλέπει προς την ίδια κατεύθυνση.



Εικόνα 5. Το αγώνισμα των Ολυμπιακών αγώνων.

Δραστηριότητα 7

Το συνεργείο των 4 ρομπότ της Εικόνας 6 ανέλαβε να ξηλώσει τα χαλασμένα πλακάκια ενός διαδρόμου που υπάρχει στον κήπο ενός ξενοδοχείου. Κάθε ένα από τα 4 ρομπότ ξεκινάει από τη θέση που φαίνεται στην Εικόνα 6 και ξηλώνει 3 πλακάκια (beepers), τα οποία τοποθετεί στην τσάντα που κουβαλάει μαζί του. Γράψε ένα πρόγραμμα για τη δημιουργία τεσσάρων στιγμιστύπων/αντικειμένων της κλάσης `Primitive_Robot`, δηλαδή τεσσάρων ρομπότ με τις βασικές δυνατότητες, και την αποστολή στη συνέχεια των κατάλληλων μηνυμάτων για την εκτέλεση αυτής της αποστολής.



Εικόνα 6. Ξήλωμα διαδρόμου από πλακάκια.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 – Τεστ αξιολόγησης

Πρόσθετες ασκήσεις, δραστηριότητες

Θέματα εμπέδωσης, αξιολόγησης και αυτοαξιολόγησης των μαθητών

ΕΡΩΤΗΣΕΙΣ ΚΛΕΙΣΤΟΥ ΤΥΠΟΥ

ΕΡΩΤΗΣΗ 1

Επιλέξτε τη μοναδική **σωστή** απάντηση.

- A) Η κλάση Primitive_Robot είναι μια συλλογή ρομπότ.
- B) Η κλάση Primitive_Robot είναι ένα πρότυπο δημιουργίας ρομπότ με κάποιες βασικές δυνατότητες.
- Γ) Οι έννοιες μοντέλο ρομπότ (κλάση) και ρομπότ (αντικείμενο) αναφέρονται στην ουσία στο ίδιο ακριβώς πράγμα.
- Δ) όλα τα παραπάνω.

ΕΡΩΤΗΣΗ 2

Επιλέξτε τη μοναδική **λάθος** απάντηση.

- A) Δύο ρομπότ της κλάσης Primitive_Robot μπορούν να έχουν τις ίδιες τιμές στις ιδιότητές τους.
- B) Μπορεί να χρησιμοποιηθεί το ίδιο όνομα (αναγνωριστικό) για δύο ρομπότ, εφόσον αυτά έχουν κάποια διαφορετική τιμή σε κάποια από τις ιδιότητές τους.
- Γ) Από την κλάση Primitive_Robot μπορούμε να δημιουργήσουμε όσα ρομπότ θέλουμε.

ΕΡΩΤΗΣΗ 3

Επιλέξτε τη μοναδική **σωστή** απάντηση

- A) Μια μέθοδος μπορεί να κληθεί για οποιοδήποτε αντικείμενο μιας κλάσης.
- B) Μπορούμε να καλέσουμε μια μέθοδο για ένα αντικείμενο μόνο μία φορά.
- Γ) Όλα τα παραπάνω.

ΕΡΩΤΗΣΗ 4

Επιλέξτε τη μοναδική **λάθος** απάντηση

- A) Αν δημιουργήσουμε ένα ρομπότ της κλάσης Primitive_Robot τότε αυτό θα πρέπει υποχρεωτικά να εκτελέσει όλες τις μεθόδους της κλάσης.
- B) Κάθε μέθοδος της κλάσης Primitive_Robot μπορεί να κληθεί μόνο μία φορά για ένα ρομπότ.

Γ) Είναι λανθασμένες και οι δύο προηγούμενες απαντήσεις

ΕΡΩΤΗΣΗ 5

Επιλέξτε τη μοναδική **σωστή** απάντηση

- A) Η εκτέλεση μιας μεθόδου μπορεί να αλλάξει την κατάσταση ενός αντικειμένου.
- B) Το αποτέλεσμα της εκτέλεσης μιας συγκεκριμένης μεθόδου είναι πάντα το ίδιο ανεξάρτητα από την κατάσταση του αντικειμένου.
- Γ) Όλα τα παραπάνω.

ΕΡΩΤΗΣΗ 6

Επιλέξτε τη μοναδική **σωστή** απάντηση

- A) Ένα αντικείμενο ανταποκρίνεται σε ένα μήνυμα μόνο εφόσον έχει οριστεί η αντίστοιχη μέθοδος στην κλάση από την οποία προέρχεται το αντικείμενο.
- B) Ένα αντικείμενο ανταποκρίνεται σε ένα μήνυμα πάντα με τον ίδιο ακριβώς τρόπο.
- Γ) Όλα τα παραπάνω.

18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

- Ολοκληρώστε τη μελέτη των φύλλων εργασίας που προορίζονται για τους μαθητές.
- Σχολιάστε κάθε ένα από τα 2 πρώτα φύλλα εργασίας, αναφέροντας τυχόν αδύναμα σημεία και προτείνοντας αλλαγές, βελτιώσεις ή και προσθήκες (κοινή δραστηριότητα με wiki).
- Συμπληρώστε κατάλληλα το 3^ο φύλλο εργασίας που έχει ως στόχο την (αυτο)αξιολόγηση των μαθητών, αναφέροντας σύντομα τον στόχο κάθε δραστηριότητας που προτείνετε.
- Δείτε τις άλλες επεκτάσεις.
- Προσαρμόστε τα φύλλα εργασίας, ή δημιουργήστε νέα φύλλα εργασίας για κάποιον άλλο μικρόκοσμο, όπως ο JKarelRobot ή ο Jeroo.
- Μελετήστε τη βιβλιογραφία.

Ενότητα Σεναρίου 3

Διδασκαλία των δομών επανάληψης

Δομές επανάληψης - εισαγωγή

1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Εισαγωγή σε 3 δομές επανάληψης

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 5 έως 8 διδακτικές ώρες, δηλαδή περίπου 1-3 ώρες για την κάθε δομή επανάληψης

3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Πρόκειται για διδακτικό σενάριο (για την ακρίβεια: 3 όμοια διδακτικά σενάρια) που μπορεί να ενταχθεί κατά κύριο λόγο - ίσως με μερικές παραλλαγές - στη Γ' Λυκείου στο μάθημα της «Ανάπτυξης Εφαρμογών σε Προγραμματιστικό Περιβάλλον».

Πιο συγκεκριμένα, το σενάριο προορίζεται για μαθητές με κάποιες στοιχειώδεις γνώσεις προγραμματισμού. Επομένως μπορεί κατά κύριο λόγο να προταθεί σε μαθητές της Γ' Λυκείου του Γενικού Λυκείου, αλλά και την αντίστοιχη τάξη (ή τάξεις) σε Τεχνικά ή Επαγγελματικά Λύκεια ή Σχολές. Οι σχετικές έννοιες αποτελούν τμήμα της ύλης που διδάσκεται κάθε χρόνο στο μάθημα της Γ' Λυκείου και εξετάζεται στις πανελλαδικές εξετάσεις για την εισαγωγή στην τριτοβάθμια Εκπαίδευση.

Η χρήση του διδακτικού σεναρίου σε επίπεδο Γυμνασίου μπορεί να γίνει με την επιφύλαξη ότι οι μαθητές πρέπει να έχουν το απαραίτητο γνωστικό υπόβαθρο (εξοικείωση με τις σχετικές έννοιες) και την απαραίτητη ωριμότητα ώστε να αντιμετωπίσουν σχετικά προηγμένες προγραμματιστικές έννοιες. Το σενάριο μπορεί λοιπόν να διδαχθεί και σε άλλες τάξεις, όπως η Γ' Γυμνασίου, εφόσον το χρησιμοποιούμενο προγραμματιστικό περιβάλλον, ή η γλώσσα προγραμματισμού, το «επιτρέπουν» (με την έννοια ότι οι δραστηριότητες που περιγράφονται έχουν νόημα μέσα στο πλαίσιο της χρησιμοποιούμενης γλώσσας).

Για λόγους πληρότητας, αναφέρεται ότι το σενάριο θα μπορούσε επίσης να ενταχθεί σε πρόγραμμα Πληροφορικού Γραμματισμού σε ΣΔΕ (Σχολεία Δεύτερης Ευκαιρίας για ενηλίκους), με την προϋπόθεση ότι οι διδασκόμενοι έχουν τις προαπαιτούμενες γνώσεις

(έννοια του προγράμματος, της μεταγλώττισης/διερμηνείας και εκτέλεσης ενός προγράμματος, της μεταβλητής και της εκχώρησης τιμών κλπ).

4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Ο βασικός στόχος είναι η εισαγωγή των μαθητών στη δομή επανάληψης – ακριβέστερα στις τρεις πλέον διαδεδομένες μορφές των δομών επανάληψης, αυτές που (συμβατικά) θα περιγράψαμε ως εξής: `while...do`, `repeat...until`, `for...to...do` (ΟΣΟ...ΕΠΑΝΑΛΑΒΕ, ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙΣ_ΟΤΟΥ και ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ...ΕΠΑΝΑΛΑΒΕ στην ελληνική ορολογία).

Ο σκοπός του σεναρίου είναι η πρώτη επαφή των μαθητών με τους τρεις μηχανισμούς επανάληψης. Οι μαθητές δεν αναμένεται βέβαια να κατανοήσουν πλήρως καθέναν από αυτούς και τις μεταξύ τους διαφορές, ούτε να είναι σε θέση να επιλέγουν την κατάλληλη δομή επανάληψης σε σχέση με το πρόβλημα που έχουν να αντιμετωπίσουν κάθε φορά.

Αναμένεται όμως, με το πέρας του μαθήματος οι μαθητές να είναι σε θέση να αναγνωρίζουν τον κάθε τύπο δομής επανάληψης, όχι μόνο μορφολογικά, αλλά και να κατανοούν τους λόγους για τους οποίους ένας δεδομένος τύπος είναι καταλληλότερος από κάποιον άλλο: οι μαθητές δηλαδή, δεν θα είναι ενδεχομένως σε θέση να επιλέγουν οι ίδιοι τον καταλληλότερο τύπο δομής επανάληψης, αλλά αναμένεται να είναι σε θέση να αναγνωρίζουν γιατί ένας τύπος που χρησιμοποιείται για τη λύση ενός προβλήματος είναι καταλληλότερος από τους άλλους.

Αναμένεται επίσης οι μαθητές να είναι σε θέση να μετατρέπουν έναν αλγόριθμο περιγραφόμενο σε «φυσική γλώσσα» σε μια μορφή προγράμματος με χρήση των τριών βασικών δομών επανάληψης – τουλάχιστον σε απλές, σχετικά, περιπτώσεις.

Οι μαθητές θα εξοικειωθούν με τη μετατροπή από μια μορφή δομής επανάληψης σε μια άλλη στο επόμενο επιμορφωτικό τρίωρο.

Εκτός του συγκεκριμένου αυτού στόχου οι μαθητές, μέσω του σεναρίου, θα εξοικειωθούν περισσότερο:

- Με τη χρήση μεταβλητών που έχουν έναν ιδιαίτερο ρόλο, που συναντάται συχνά στην επίλυση προβλημάτων (δημιουργία αλγορίθμων/προγραμμάτων), όπως οι μεταβλητές αθροιστή και μετρητή.
- Με τη θεώρηση των `strings` ως συμβολοσειρών (πολύ σημαντικό για τις δομές δεδομένων αλλά και για τη μοντελοποίηση διαφόρων οντοτήτων και επεξεργασίας τους)

Τα προτεινόμενα προβλήματα θα μπορούσαν οριακά να επιλυθούν και με άλλο είδος

δομής επανάληψης (από το προτεινόμενο) αλλά ο διδάσκων θα πρέπει να καταστήσει, κάθε φορά, σαφή τα πλεονεκτήματα της προτεινόμενης, δομής επανάληψης,. Το σημείο αυτό είναι σημαντικό, ώστε οι μαθητές να εξοικειωθούν με την αντίληψη ότι κάθε επιλεγόμενη δομή επανάληψης είναι η καταλληλότερη για την επίλυση του συγκεκριμένου προβλήματος.

5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Η βασική ιδέα είναι η επίλυση τριών προβλημάτων.

Και τα τρία προβλήματα απαιτούν για την επίλυσή τους μια δομή επανάληψης – διαφορετικού όμως είδους για κάθε πρόβλημα. Εκτός από τα αρχικά προτεινόμενα προβλήματα, προτείνονται και μερικά επιπλέον, τα οποία αποσκοπούν στην εξάσκηση, εμπέδωση και (αυτό)αξιολόγηση.

Εισαγωγή στις επαναληπτικές δομές: εφόσον πρόκειται για εισαγωγικές δραστηριότητες, καλό είναι πριν από τη χρήση κάθε δομής να παρουσιάζονται τα βασικά της χαρακτηριστικά, τα οποία

- να επαναλαμβάνονται και μετά τη λύση του προβλήματος και
- να επαναλαμβάνονται σε κάθε νέο πρόβλημα που θα προταθεί για λύση στους μαθητές – μέχρις ότου οι μαθητές μάθουν να αναγνωρίζουν με ευχέρεια τον κατάλληλο τύπο βρόχου (δομής επανάληψης).

Η γενική ιδέα μιας δομής επανάληψης είναι η ιδέα της επεξεργασίας ενός πλήθους ομοειδών δεδομένων. Για παράδειγμα, τα δεδομένα μπορεί να είναι:

- αριθμοί 1-20, όπως οι βαθμοί ενός μαθητή και θέλουμε να υπολογίσουμε τη μέση τιμή τους, ή το σύνολο των πωλήσεων σε ένα κατάστημα για να υπολογίσουμε τη συνολική είσπραξη,
- πολύπλοκα στοιχεία όπως ονοματεπώνυμο, διεύθυνση, τηλέφωνο κάθε συνδρομητή τα οποία καταχωρούνται για να ταξινομηθούν κατάλληλα ώστε να προστεθούν σε ένα τηλεφωνικό κατάλογο,
- αναλογικά δεδομένα, όπως στοιχεία από ένα μετεωρολογικό σταθμό, τα οποία συλλέγονται συνεχώς προκειμένου να υποστούν την κατάλληλη επεξεργασία

κλπ.

Επομένως στα παραπάνω παραδείγματα έχουμε μερικά κοινά στοιχεία:

(1) ένα πλήθος ομοειδών δεδομένων (βαθμοί, στοιχεία κάθε συνδρομητή κλπ).

(2) τη διαχείριση (μέση τιμή, ταξινόμηση, επεξεργασία δεδομένων).

Στις περιπτώσεις αυτές χρησιμοποιούμε τις λεγόμενες *επαναληπτικές διαδικασίες*, που μας επιτρέπουν να επαναλάβουμε όσες φορές είναι απαραίτητο, την επεξεργασία των δεδομένων.

Στην πραγματικότητα η συντριπτική πλειονότητα των εφαρμογών των Η.Υ. περιλαμβάνουν κάποιες επαναληπτικές δομές, γιατί σπάνια χρησιμοποιούμε έναν Η.Υ. για να επεξεργαστούμε μια και μοναδική φορά, ένα και μόνο δεδομένο.

Άρα, στη γενικότητά τους, οι δομές επανάληψης αφορούν στην επεξεργασία ενός πλήθους δεδομένων τα οποία υφίστανται την επεξεργασία αυτή σειριακά – το ένα μετά το άλλο.

Εισαγωγή στη δομή **όσο ισχύει μια συνθήκη επανάλαβε**

Είναι ίσως σκόπιμο ο διδάσκων να εισαγάγει αυτή την επαναληπτική δομή εξηγώντας την αναγκαιότητα για το συγκεκριμένο τύπο δομής με ένα παράδειγμα, όπως το παρακάτω:

Σε ορισμένες περιπτώσεις μπορεί στην επαναληπτική διαδικασία που εφαρμόζουμε, να θέλουμε να ελέγχουμε εκ των προτέρων αν το επόμενο στοιχείο θα υποστεί επεξεργασία ή όχι.

Για παράδειγμα, ένας υπάλληλος καταχωρεί βαθμούς από το μάθημα της Ιστορίας από τις Πανελλήνιες εξετάσεις, στη μορφή:

Εξαψήφιος κωδικός μαθητή - βαθμός πρώτου βαθμολογητή - βαθμός δεύτερου βαθμολογητή:

684468 18 20

976646 11 9

123545 12 14

856641 15 15

.....

Οι βαθμοί αυτοί καταχωρούνται προκειμένου να υποστούν περαιτέρω επεξεργασία. Αν ο καταχωρητής επιθυμεί να διακόψει προσωρινά την καταχώρηση (για να πάει να πάρει έναν καφέ;) μπορεί – για παράδειγμα – να πληκτρολογεί έναν προκαθορισμένο κωδικό (όπως ο αριθμός –999) και το σύστημα να διακόψει προσωρινά την καταχώρηση. Φυσικά δεν θέλουμε ο –999 να καταχωρηθεί ως κωδικός υποψηφίου, χωρίς βαθμούς, και να προκληθεί

κάποιου είδους πρόβλημα.

Άρα το σύστημα **πριν και καταχωρήσει** ένα κωδικό πρέπει να ελέγχει αν ο κωδικός είναι ο -999 που σημαίνει «όχι καταχώρηση – προσωρινή διακοπή».

Ο διδάσκων στη συνέχεια **εξηγεί τη σύνταξη** της δομής και **αναλύει τη λειτουργία** της.

Προσομοιώνει τη λειτουργία ενός πολύ μικρού προγράμματος που γράφει ο ίδιος (στον πίνακα ή στον Η.Υ. με βιντεοπρωτόκολλα). Εξηγεί τις διαδοχικές τιμές μεταβλητών και εκφράσεων.

Θέτει 2-3 ερωτήσεις στους μαθητές του είδους «τι θα συμβεί αν αλλάξουμε στο πρόγραμμα τη μεταβλητή... ή την έκφραση...» για να βεβαιωθεί ότι οι μαθητές έχουν κατανοήσει τα βασικά θέματα σύνταξης και λειτουργίας της δομής (syntax, semantics).

Στη συνέχεια μπορεί να χρησιμοποιηθεί το σχετικό φύλλο εργασίας.

Σε κάθε φύλλο εργασίας περιλαμβάνονται

- (1) ένα έτοιμο πρόγραμμα που εκτελούν οι μαθητές και παρατηρούν τις μεταβολές των τιμών των μεταβλητών και εκφράσεων.
- (2) ένα απλό πρόβλημα προς επίλυση
- (3) ένα πιο σύνθετο πρόβλημα προς επίλυση
- (4) μερικές ερωτήσεις αξιολόγησης
- (5) διάφορα τεστ και παιχνίδια.

Εισαγωγή στη δομή επανάλαβε...μέχρις ότου

Είναι ίσως σκόπιμο ο διδάσκων να εισαγάγει αυτή την επαναληπτική δομή εξηγώντας την αναγκαιότητα για το συγκεκριμένο τύπο δομής με ένα παράδειγμα, όπως το παρακάτω:

Σε ορισμένες περιπτώσεις μπορεί στην επαναληπτική διαδικασία που εφαρμόζουμε, να θέλουμε να ελέγχουμε εκ των υστέρων αν το επόμενο στοιχείο θα υποστεί επεξεργασία ή όχι.

Για παράδειγμα, θέλουμε να καταθέσουμε ένα αρχικό κεφάλαιο Κ ευρώ, με ένα επιτόκιο Ε και θα θέλαμε να γνωρίζουμε σε πόσα χρόνια το κεφάλαιο μαζί με τους τόκους θα υπερβεί ένα ορισμένο όριο Ο. Έτσι, το ποσόν κάθε χρόνου θα προκύπτει από το ποσόν του προηγούμενου χρόνου συν τον τόκο της τρέχουσας χρονιάς και η διαδικασία αυτή θα πρέπει να συνεχιστεί, έως ότι το ποσόν υπερβεί το επιθυμητό όριο.

Εισαγωγή στη δομή Για μεταβλητή από...μέχρι...

Είναι ίσως σκόπιμο ο διδάσκων να εισαγάγει αυτή την επαναληπτική δομή εξηγώντας την αναγκαιότητα για το συγκεκριμένο τύπο δομής με ένα παράδειγμα, όπως το παρακάτω:

Σε ορισμένες περιπτώσεις μπορεί στην επαναληπτική διαδικασία που εφαρμόζουμε, να είναι γνωστό από την αρχή το πλήθος των επαναλήψεων. Για παράδειγμα, θέλουμε να καταθέσουμε ένα αρχικό κεφάλαιο Κ ευρώ, με ένα επιτόκιο Ε και θα θέλαμε να γνωρίζουμε σε Ν χρόνια πόσο θα έχει γίνει το κεφάλαιο μαζί με τους τόκους (υποθέτουμε ότι κάθε χρόνο οι τόκοι ενσωματώνονται στο κεφάλαιο – ο λεγόμενος ανατοκισμός).

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Οι επαναληπτικές δομές υπήρξαν από τα πλέον διερευνημένα θέματα στη Διδακτική της Πληροφορικής, σε όλες τις βαθμίδες εκπαίδευσης, από πολλούς ερευνητές κι ερευνητικές ομάδες.

Οι επαναληπτικές δομές είναι πολλών τύπων: εκτός από τις συνήθεις τρεις που διδάσκονται τυπικά στα πλαίσια του δομημένου προγραμματισμού, υπάρχουν πολλοί άλλοι τύποι επαναληπτικών διαδικασιών. Θεωρητικά είναι γνωστό από το 1966 (*Θεώρημα των Boehm & Jacorini* [1966]) ότι η δομή **while συνθήκη do** είναι αρκετή από μόνη της για να καλύψει όλες τις περιπτώσεις που μπορούν να παρουσιαστούν.

Επίσης, όπως είναι γνωστό, ένα μικρό (μόλις 2 σελίδων κείμενο του E.W. Dijkstra, στο τέλος της δεκαετίας του 1960), σήμανε την αρχή του τέλους για τον προγραμματισμό «σπαγγέτι» όπως ονομαζότανε, επειδή η χωρίς περιορισμό χρήση της εντολής **go to** έκανε τα διαγράμματα ροής της εποχής εκείνης να μοιάζουν σα σπαγγέτι. Ο E.W. Dijkstra εισήγαγε την εποχή του δομημένου προγραμματισμού που ουσιαστικά καθιέρωσε τον προγραμματισμό που βασίζεται πάνω στις τρεις διαδοσόμενες μορφές επαναληπτικών δομών.

Θα πρέπει να σημειωθεί ότι το κίνητρο για την πρόταση του νέου προγραμματιστικού μοντέλου, ήταν ουσιαστικά η ανεύρεση λύσης σε δυο σημαντικά προβλήματα: το ένα είχε να κάνει με την ίδια τη φύση του προγραμματισμού. Ο προγραμματισμός για τους υποστηρικτές του δομημένου μοντέλου (E. W. Dijkstra, T. Hoare, N. Wirth κ. ά.) είναι ένας αυστηρός επιστημονικός κλάδος, μάλιστα ένας κλάδος των εφαρμοσμένων Μαθηματικών, και ως τέτοιος έπρεπε να διέπεται από μια επιστημονική μεθοδολογία. Εξάλλου, η προσέγγιση αυτή, θα διευκόλυνε τη διδασκαλία του. Στην άλλη όχθη, επιστήμονες όπως ο

D. E. Knuth, υποστήριξαν την ιδέα ενός προγραμματισμού που είναι ακόμη και τέχνη. Το δεύτερο πρόβλημα, άμεσα συνδεδεμένο με το πρώτο, ήταν η αποτελεσματικότητα και το (οικονομικό κόστος) των μεγάλων projects. Από τη μια το λάθος στον προγραμματισμό μπορούσε να κοστίζει ανθρώπινες ζωές (όπως η έκρηξη του διαστημοπλοίου «Απόλλων») και εκατομμύρια δολάρια. Από την άλλη, η εκσφαλμάτωση κόστιζε πολύ περισσότερο από την ίδια τη δημιουργία του κώδικα – γεγονός εξόφθαλμα παράλογο. Όλα αυτά δημιούργησαν (κατά την έκφραση του E. W. Dijkstra) *μια κρίση*, στην οποία ο δομημένος προγραμματισμός αποτελούσε μια απάντηση. Βέβαια, ο δομημένος προγραμματισμός, δεν σημαίνει απλά απαλοιφή του **go to** και χρήση των τριών γνωστών δομών επανάληψης, αλλά είναι ίσως το πιο πολυσυζητημένο χαρακτηριστικό του.

Όλη αυτή η προβληματική βέβαια δεν μπορεί να αναπαραχθεί μέσα στη σχολική τάξη, οπότε το τι αντιλαμβάνονται οι μαθητές για τους λόγους που επιβάλλουν τη χρήση αυτών των τριών δομών παραμένει ένα ερωτηματικό.

Ο ίδιος ο E. W. Dijkstra πάντως αναφέρει ότι η χρήση γλωσσών όπως η BASIC, που επιτρέπει βέβαια την εντολή **go to**, λειτουργεί ως «μόνιμη μόλυνση»: ο προγραμματιστής που θα αρχίσει να προγραμματίζει με **go to** είναι καταδικασμένος.

Η wikipedia αναφέρει (δεν υπάρχει επαλήθευση μέχρι στιγμής) ότι η ιστορία αυτής της διαμάχης έχει και μια χιουμοριστική πλευρά. Έτσι το θρυλικό άρθρο του Edsger W. Dijkstra “Go To Statement Considered Harmful”, στο περιοδικό *Communications of the ACM*, το διαδέχθηκε το 1987 το άρθρο “Go To Statement Considered Harmful” Considered Harmful (Frank Rubin, 1987 στο CACM, επίσης) και αυτό το διαδέχθηκε με τη σειρά του το κείμενο «“Go To Statement Considered Harmful” Considered Harmful’ Considered Harmful» (το 1987, στο CACM). Τη συνέχεια έκλεισε ο ίδιος ο E. W. Dijkstra με το κείμενο του *On a somewhat disappointing correspondence*.

Ανεξάρτητα από το θεωρητικό συμπέρασμα για την ισοδυναμία μεταξύ των διαφόρων δομών επανάληψης διακρίνονται αρκετές κατηγορίες επαναληπτικών δομών οι οποίες

α) παρουσιάζουν σε ορισμένες περιπτώσεις διαφορετικές δυνατότητες - από καθαρά θεωρητική άποψη

β) παρουσιάζουν διαφορές ως προς την καταλληλότητά τους για ορισμένο τύπο προβλημάτων

γ) ακόμη κι όταν είναι ισοδύναμες, θεωρητικά, μπορούν να είναι λιγότερο ή περισσότερο περίπλοκες για το χρήστη.

Σε γενικές γραμμές οι επαναληπτικές δομές ανήκουν στους εξής τύπους:

α) **repeat** N times action

Η διαδικασία ή το σύνολο εντολών action επαναλαμβάνεται N φορές - το N πρέπει βέβαια να είναι γνωστό εκ των προτέρων. Πρόκειται για τη συνήθη επαναληπτική δομή της LOGO.

β) **for** I:=N1 to N2 **do** action

Η διαδικασία ή το σύνολο εντολών action επαναλαμβάνεται N2-N1+1 φορές

γ) **while** συνθήκη **do** action

Η διαδικασία ή το σύνολο εντολών action επαναλαμβάνεται όσο η συνθήκη είναι αληθής. Το αντίστοιχο τεστ πραγματοποιείται πριν από το σώμα της επαναληπτικής διαδικασίας.

δ) **repeat** action **until** συνθήκη

Η διαδικασία ή το σύνολο εντολών action επαναλαμβάνεται όσο η συνθήκη είναι ψευδής. Το αντίστοιχο τεστ πραγματοποιείται μετά από το σώμα της επαναληπτικής διαδικασίας

Οι παραπάνω δομές μπορούν να θεωρηθούν σε δυο εναλλακτικούς τύπους ανάλογα με το αν προσφέρουν ή όχι τη δυνατότητα εξόδου από ένα ενδιάμεσο σημείο - κι όχι μόνο από το σημείο του τεστ. Με την έννοια αυτή η πιο γενική μορφή επαναληπτικής διαδικασίας είναι η εξής:

ε) **do**

```
action1  
if συνθήκη1 then exit  
action2  
if συνθήκη2 then exit  
.....  
action N  
if συνθήκηN then exit
```

loop

Η επαναληπτική αυτή διαδικασία μπορεί βέβαια να προσομοιώσει, κατά προφανή τρόπο, τη λειτουργία όλων των προηγούμενων. Αν για παράδειγμα παραληφθούν οι εντολές:

```
action2  
if συνθήκη2 then exit  
.....  
actionN  
if συνθήκηN then exit
```

τότε η δομή αυτή γίνεται ισοδύναμη με μια **while** *συνθήκη* **do** action.

Εκτός απο τη σημασία (semantics) κάθε τύπου επαναληπτικής δομής, σημαντικό ρόλο παίζουν και τα συστατικά στοιχεία των δομών αυτών. Έτσι σε κάθε επαναληπτική δομή διακρίνονται:

- ένα σώμα της επαναληπτικής δομής: το σύνολο των εντολών οι οποίες επανεπεξεργάζονται σε κάθε επανάληψη,
- ένας δείκτης της επαναληπτικής δομής: πρόκειται για το ιδιαίτερο εκείνο σύμβολο το οποίο επισημαίνει την απαρχή μιας επαναληπτικής δομής, όπως η λέξη while ή το repeat,
- τα σύμβολα ορίων της επαναληπτικής δομής: πρόκειται για τα ιδιαίτερα εκείνα σύμβολα τα οποία επισημαίνουν την αρχή και το τέλος του σώματος της επαναληπτικής δομής, όπως το ζεύγος do begin....end,
- η συνθήκη συνέχειας - ή τερματισμού - της επαναληπτικής δομής
- τέλος οι αποδόσεις αρχικών τιμών - όχι μόνο στη μεταβλητή που ελέγχει τη συνθήκη συνέχειας/τερματισμού, αλλά πιθανότατα κι άλλων μεταβλητών στο σώμα της επαναληπτικής δομής.

Σε ένα πιο προηγμένο επίπεδο θα μπορούσαμε ίσως να διακρίνουμε και τις αναλλοίωτες σχέσεις που διέπουν την επαναληπτική δομή.

Αν για έναν πεπειραμένο προγραμματιστή τα δομικά στοιχεία μιας επαναληπτικής δομής περνούν σχεδόν απαρατήρητα, ωστόσο για τους αρχάριους η διαχείριση τους μπορεί να αποβεί περίπλοκη. Θεωρώντας και μόνον το παράδειγμα της επαναληπτικής δομής **for** I:=N1 to N2 **do** action θα λέγαμε ότι τα στοιχεία τα οποία θα πρέπει να ληφθούν υπόψη είναι αρκετά (μερικά από τα οποία είναι εγγενή στην επαναληπτική δομή ενώ άλλα αποτελούν ιδιαιτερότητες του εκάστοτε μεταφραστή - compiler ή διερμηνευτή - interpreter):

- Το $N2 > N1$ αλλά είναι δυνατόν να ισχύει και το αντίστροφο: $N2 < N1$; Σε κάθε περίπτωση ποιο είναι το by default βήμα μεταβολής;
- Τι συμβαίνει αν $N1 = N2$; Τι συμβαίνει αν ακολουθηθεί η σύνταξη θετικού βήματος αλλά ισχύει $N2 < N1$;
- Τα $N1$, $N2$ και το βήμα μπορούν να είναι δεκαδικοί αριθμοί; Στην περίπτωση αυτή ο υπολογισμός του ακριβούς αριθμού των επαναλήψεων μπορεί να είναι δυσχερής για ένα αρχάριο προγραμματιστή.
- Σε κάθε περίπτωση (βήμα θετικό ή αρνητικό) ποια είναι η τιμή του I μετά την έξοδο από το σώμα της μεταβλητής;

- Ο έλεγχος για τη συνέχιση ή τον τερματισμό της επαναληπτικής δομής πραγματοποιείται πριν ή μετά το σώμα;

- Είναι δυνατόν τα N1, N2 να μην αποτελούν αριθμητικές σταθερές εκφρασμένες με ψηφία αλλά σταθερές γενικώς ή πολύπλοκες αριθμητικές εκφράσεις ή ακόμη και μεταβλητές ή συναρτήσεις των οποίων οι τιμές επαναπροσδιορίζονται σε κάθε κλήση της επαναληπτικής διαδικασίας.

Η χρήση λοιπόν ακόμη και μιας "τετριμμένης" δομής μπορεί να αποδειχθεί περίπλοκη.

7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Το διδακτικό σενάριο προβλέπεται να υλοποιηθεί στο εργαστήριο Πληροφορικής.

Απαιτείται ένα λογισμικό όπως η «Γλώσσα», το «ΔΕΛΥΣ» ή ο «Δημιουργός Διαγραμμάτων Ροής».

Κεντρικό σημείο της χρήσης των λογισμικών αυτών είναι:

(α) η εκτέλεση των προγραμμάτων σε ένα περιβάλλον με "διδακτικό προσανατολισμό", δηλαδή σε ένα περιβάλλον με απλή διεπιφάνεια χρήστη και με ένα περιορισμένο ρεπερτόριο εντολών.

(β) ένα πολύ ουσιαστικό σημείο επίσης της χρήσης των ψηφιακών αυτών περιβαλλόντων, είναι ότι επιτρέπουν τη σταδιακή (βηματική) εκτέλεση ενός προγράμματος. Κάθε ένα από τα περιβάλλοντα, έχει διαφορετικό τρόπο λειτουργίας από τα άλλα, αλλά συγκλίνουν όλα στη βηματική εκτέλεση. Η δυνατότητα αυτή επιτρέπει την ουσιαστική κατανόηση της λειτουργίας των προγραμμάτων από τους αρχάριους μαθητές, μέχρις ότου αποκτήσουν (δημιουργήσουν) ένα ικανοποιητικό νοητικό μοντέλο της εκτέλεσης των προγραμμάτων. Η βηματική εκτέλεση, σε συνδυασμό με τη δυνατότητα παρατήρησης των τιμών των μεταβλητών (με δυναμικό τρόπο), επιτρέπει στους αρχάριους προγραμματιστές να κατανοούν καλύτερα τα αποτελέσματα της εκτέλεσης των εντολών του προγράμματος. Σε κάθε περίπτωση, πάντως, ο διδάσκων έχει την επιλογή του τρόπου με τον οποίο θα εμφανιστούν τα αποτελέσματα.

Τα διδακτικά οφέλη (η «προστιθέμενη αξία») από τη χρήση του ψηφιακού μέσου μπορούν να αποδειχθούν πολύ σημαντικά, ιδιαίτερα στις περιπτώσεις στις οποίες τα αποτελέσματα είναι αναπάντεχα ή οι μαθητές δεν μπορούν πάντοτε να κατανοήσουν τον τρόπο λειτουργίας του Η.Υ. κατά την εκτέλεση ενός προγράμματος

8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

Οι συνηθέστερες δυσκολίες ενός αρχάριου προγραμματιστή εντοπίζονται στα εξής:

- Προτιμά να χρησιμοποιεί ακολουθίες επαναλαμβανόμενων εντολών στα προγράμματά του, αντί για βρόχους.
- Διαθέτει ανεπαρκή νοητικά μοντέλα για τις επαναληπτικές δομές.
- Δυσκολεύεται να επιλέξει την κατάλληλη, για κάθε πρόβλημα, επαναληπτική δομή.
- Αδυνατεί να ορίσει με επιτυχία τη συνθήκη εξόδου από ένα βρόχο.
- Δε μπορεί να καθορίσει την εμβέλεια του βρόχου και να προσδιορίσει την αρχή και το τέλος τους, καθώς και τις εντολές που επαναλαμβάνονται.

Ορισμένοι ερευνητές έδειξαν για παράδειγμα ότι η έκφραση της συνθήκης τερματισμού μπορεί να είναι πολύ δυσχερέστερη σε ορισμένες περιπτώσεις ανάλογα με το είδος της επαναληπτικής δομής που χρησιμοποιείται από τον προγραμματιστή.

Μ' ένα γενικότερο τρόπο τα βασικά συμπεράσματα από τις επαναληπτικές δομές θα μπορούσαν να συνοψιστούν ως εξής:

- Η χρήση των επαναληπτικών δομών δεν είναι τετριμμένη για τους αρχάριους προγραμματιστές. Κατά κανόνα η χρήση των επαναληπτικών δομών στηρίζεται στη "μεταφορά" μιας στρατηγικής επίλυσης από ένα άλλο, πιο οικείο περιβάλλον, στο προγραμματιστικό. Για να χρησιμοποιήσουμε ένα κλασσικό παράδειγμα, για τον υπολογισμό του αθροίσματος $1+2+\dots+300$ οι αρχάριοι προγραμματιστές προσπαθούν να μιμηθούν τη διαδικασία επίλυσης με χαρτί-μολύβι. Η τεχνική αυτή είναι παραγωγική, αφού δίνει μια βασική στρατηγική για την επίλυση του προβλήματος, είναι όμως ταυτόχρονα και ανασταλτική, γιατί η προγραμματιστική λύση δεν είναι συμβατή με τη λύση με χαρτί-μολύβι. Συγκεκριμένα ο αθροιστής παραμένει μια μεταβλητή η οποία είναι αναγκαία για την επίλυση, αλλά η οποία δεν αντιστοιχεί σε κανένα από τα δεδομένα ή τα ζητούμενα της εκφώνησης. Η λύση με χαρτί-μολύβι δεν έχει ανάγκη αποθήκευσης ενδιάμεσων αποτελεσμάτων αφού, κατά κάποιο τρόπο, η αναγραφή των αποτελεσμάτων αυτών στο χαρτί συνιστά ένα είδος αποθήκευσης. Όπως είπε κι ένας μαθητής "μπορούμε να το πούμε αλλά όχι να το φτιάξουμε..." (Dagdilelis [1987b]).

- Η επίλυση τέτοιου είδους προβλημάτων - στα οποία η προγραμματιστική λύση φαίνεται "απομακρυσμένη" από μια "φυσική" λύση με χαρτί-μολύβι - είναι ίσως εξαιρετικά

δυσχερής για αρχάριους προγραμματιστές. Ακόμη κι η επίλυση ενός προβλήματος σ' ένα προγραμματιστικό περιβάλλον δεν επαρκεί πάντοτε για την πλήρη κατανόηση της λύσης: η γνώση των αρχαρίων προγραμματιστών παραμένει "τοπική" κι εύθραυστη. Το ίδιο πρόβλημα επαναπροτεινόμενο αυτούσιο σε ένα άλλο προγραμματιστικό περιβάλλον φαίνεται να εγείρει τις ίδιες προγραμματιστικές δυσχέρειες (Carroni B., Dagdilelis V. [1988]).

- Εκτεταμένες έρευνες φαίνεται να δείχνουν ότι, αντίθετα από τους, έμμεσους ή άμεσους, ισχυρισμούς της σχολής του δομημένου προγραμματισμού, όλες οι επαναληπτικές δομές **δεν** είναι εξίσου περίπλοκες για τον προγραμματιστή. Αν θεωρήσουμε ότι μια επαναληπτική δομή είναι πιο εύκολη από μιαν άλλη, εφόσον ο χρήστης τη χρησιμοποιεί αυθόρμητα πιο συχνά από την δεύτερη και κάνει μ' αυτήν λιγότερα λάθη, τότε:

- η δομή **repeat** N times action είναι η πιο εύκολη απ' όλες. Η δομή αυτή πιθανά αντιστοιχεί σ' ένα "συνηθισμένο" τρόπο λειτουργίας στην καθημερινή ζωή.

- η δομή **while** συνθήκη **do** action φαίνεται να είναι πολύ πιο δύσκολη από την **repeat** action **until** συνθήκη, γεγονός που ερμηνεύεται από το γεγονός ότι η δεύτερη φαίνεται να αντιστοιχεί σε μια "φυσική" στρατηγική επαναληπτικής επεξεργασίας: επεξεργασία των δεδομένων κι ύστερα έλεγχος τερματισμού, ενώ αντίθετα στην πρώτη μορφή ο έλεγχος συνέχειας πραγματοποιείται πριν από οποιαδήποτε επεξεργασία (πλην της απόδοσης αρχικών τιμών, η οποία όμως δεν ανήκει στο σώμα της επαναληπτικής δομής).

- η δομή **for** I:=N1 to N2 **do** action φαίνεται επίσης να αποτελεί πηγή δυσχερειών για τους νέους προγραμματιστές, πιθανόν γιατί τόσο η μεταβολή της μεταβλητής μετρητή όσο και η πραγματοποίηση του τεστ ελέγχου είναι έμμεσες. Η διαπίστωση είναι σε αντίθεση με μια πολύ διαδεδομένη αντίληψη που θεωρεί τη δομή αυτή πιο απλή απ' όλες κι έτσι αποτελεί την κατ' εξοχή επαναληπτική δομή της BASIC.

Επίσης οι έρευνες φαίνεται να δείχνουν ότι η χρήση της εντολής Goto (τουλάχιστον σε ορισμένες περιπτώσεις) καταλήγει σε καλύτερης ποιότητας προγράμματα: πιο οικονομικά σε κώδικα, με μια πιο ξεκάθαρη δομή και κυρίως με λιγότερα λάθη.

Το πρόβλημα της επιλογής της καταλληλότερης επαναληπτικής δομής για την επίλυση συγκεκριμένων προβλημάτων φαίνεται να είναι αρκετά σημαντικό δεδομένου ότι ακόμη και σπουδαστές τριτοβάθμιας εκπαίδευσης δευτεροετείς ή και τριτοετείς βρίσκουν δυσκολίες στην έκφραση σύνθετων επαναληπτικών δομών.

9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΥΡΥΒΟΣ

Στο Λύκειο, τη χρονική στιγμή του σχολικού έτους κατά την οποία θα διδαχθούν οι σχετικές έννοιες, οι μαθητές θα έχουν ήδη εξοικειωθεί με τη λειτουργία του περιβάλλοντος και έτσι δεν αναμένονται δυσκολίες στο επίπεδο αυτό.

Εάν χρησιμοποιηθεί σε μικρότερες τάξεις, η χρήση ελληνικής γλώσσας πιθανότατα θα διευκολύνει τους μαθητές.

Η χρήση βοηθητικών μεταβλητών, η παρατήρηση των τιμών των μεταβλητών κατά την εκτέλεση (βηματική) του προγράμματος και άλλες τεχνικές θα βοηθήσουν τους μαθητές να κατανοήσουν τη λειτουργία των τριών τύπων βρόχων.

Το εκπαιδευτικό αυτό περιβάλλον είναι πάντως το πλησιέστερο, ως προς το είδος του προγραμματισμού που επιτρέπει, σε σχέση με τα περιβάλλοντα προγραμματισμού με τα οποία θα εργαστούν οι μαθητές είτε στο Πανεπιστήμιο, είτε στην επαγγελματική τους ζωή – εφόσον ασχοληθούν με τον προγραμματισμό. Σε πολλές περιπτώσεις βέβαια χρησιμοποιούνται απευθείας περιβάλλοντα επαγγελματικού τύπου.

10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Υπάρχει πολύ πλούσια διεθνής και ελληνική βιβλιογραφία – ακόμη και ελληνικές διδακτορικές διατριβές στο αντικείμενο. Παρόλα αυτά δεν αναφέρονται πηγές, γιατί η ταχύτητα ανανέωσης είναι πολύ μεγάλη.

Ενδεικτικός σύνδεσμος:

<http://www.helenemartin.com/2010-05-while-loops-gone-wild/>

Υπάρχουν πολλά διεθνή Συνέδρια, με προεξάρχοντα 2 Συνέδρια της ACM, το ένα στις Η.Π.Α., ετήσιο (γκρουπ) SIGCSE και ένα στην Ευρώπη (ITICSE – το 2003 μάλιστα διοργανώθηκε στη Θεσσαλονίκη). Η διεθνής ομάδα PPIG, ενδιαφέρεται ιδιαίτερα για την ψυχολογική διάσταση του προγραμματισμού και διοργανώνει επίσης διεθνή Συνέδρια.

Στην Ελλάδα, οι επαγγελματικές και ακαδημαϊκές ενώσεις Πληροφορικών και καθηγητών Πληροφορικής (ΕΠΥ και ΠΕΚΑΠ) διοργανώνουν τακτικά Συνέδρια αυτού ειδικά για τη διδασκαλία και τη Διδακτική της Πληροφορικής. Τα Συνέδρια της ΕΤΠΕ, συχνά έχουν ειδικές συνεδρίες για τη Διδακτική της Πληροφορικής και οι σχετικές εργασίες είναι άμεσα και ελεύθερα προσβάσιμες στην πύλη της ένωσης (<http://www.etpe.gr>)

Πολλοί διεθνείς οργανισμοί εξάλλου (όπως η ACM και η ISTE <http://www.iste.org>) έχουν τμήματα αφιερωμένα ειδικά στη διδασκαλία της Πληροφορικής. Οι περισσότερες

βιβλιογραφικές αναφορές στο θεωρητικό (ειδικό) μέρος του επιμορφωτικού υλικού περιλαμβάνουν τμήματα με ειδική αναφορά στη Διδακτική τη Πληροφορικής.

11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Οι προτεινόμενες δραστηριότητες για τους μαθητές εγγράφονται σε μια τυπική κονστрукτιβιστική προσέγγιση: στους μαθητές προτείνονται μια σειρά από προβλήματα τα οποία καλούνται να επιλύσουν.

Οι διαφορές ανάμεσα στους διάφορους τύπους δομών επανάληψης, δεν έχουν πλήρη αντιστοιχία στην καθημερινότητα – όχι μόνο οι μαθητές δεν εκφράζονται κατά κανόνα με τον τρόπο αυτό, αλλά πολλές φορές υπονοούνται ή είναι προφανείς οι προϋποθέσεις υπό τις οποίες πρέπει να εκτελεστεί μια εργασία πολλές φορές.

Αυτό αναμένεται να προκαλέσει διαφωνίες μεταξύ των μαθητών και ο εκπαιδευτικός μπορεί να κατευθύνει τη σχετική συζήτηση και τις διαφωνίες που θα προκύψουν προς την επιθυμητή πλευρά. Οι κοινωνιογνωστικές συγκρούσεις μπορούν να αποδειχθούν σημαντικές για την καλύτερη κατανόηση των διαφόρων δομών.

12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ

Δεν αναμένεται να παρατηρηθούν αξιολογες μεταβολές.

13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Προτείνεται οι μαθητές να οργανωθούν σε ομάδες των 2-3 (το πολύ) ατόμων.

Η οργάνωση αυτή ευνοεί την αλληλεπίδραση μεταξύ των μαθητών οι οποίοι αντιμετωπίζουν την επίλυση των προτεινομένων προβλημάτων ως ένα κοινό project, ως μια κοινή προσπάθεια και όχι ως ατομικές δράσεις. Αυτό δεν εξασφαλίζεται πάντοτε αυτόματα, καθώς για διάφορους λόγους, είναι δυνατόν ένας από τους μαθητές να παίξει ένα ρόλο «ηγετικό» -ιδιαίτερα αν έχει πρότερες σχετικές γνώσεις.

Ο προβλεπόμενος χρόνος κρίνεται ότι θα είναι επαρκής, καθώς δεν αναμένεται σημαντικός διδακτικός θόρυβος,

14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Αναφέρεται ένα πολύ γνωστό πρόβλημα, που αναφέρεται σε εμφωλευμένες δομές επανάληψης και μπορεί να χρησιμοποιηθεί από τον εκπαιδευτικό ως πρόσθετη δραστηριότητα:

ΠΩΣ ΕΛΕΓΧΕΤΑΙ ΕΝΑΣ ΑΥΤΟΜΑΤΟΣ ΜΗΧΑΝΙΣΜΟΣ ΠΟΥ ΚΑΘΑΡΙΖΕΙ ΠΑΤΑΤΕΣ

Διαθέτουμε ένα ρομπότ το οποίο μπορεί να καθαρίσει μια ποσότητα από πατάτες που βρίσκονται μέσα σ' ένα πανέρι. Αν το πανέρι είναι άδειο, το ρομπότ μπορεί να

δώσει εντολές, ώστε να ξαναγεμίσει το πανέρι. Το ρεπερτόριο των εντολών προς το ρομπότ είναι γνωστό και μπορεί να επεκταθεί από τον ίδιο τον χρήστη - ο χρήστης δηλαδή μπορεί να προσθέσει τις διαδικασίες ή συναρτήσεις που επιθυμεί (Dijkstra 1969, Lucas & Scholl 1975 και Lucas et als 1984). Στη διάθεσή του χρήστη υπάρχουν οι εξής εντολές:

Γέμισε_πανέρι: όταν καλείται η διαδικασία αυτή, ο αυτόματος μηχανισμός ξαναγεμίζει το πανέρι με πατάτες από την αποθήκη.

Καθάρισε_1_πατάτα: όταν καλείται η διαδικασία αυτή, ο αυτόματος μηχανισμός καθαρίζει μια πατάτα.

Αρκετές_πατάτες: μια boolean συνάρτηση με τιμή TRUE αν οι καθαρισμένες πατάτες είναι αρκετές και τιμή FALSE στην αντίθετη περίπτωση.

Άδειο_πανέρι: μια boolean συνάρτηση με τιμή TRUE αν το πανέρι είναι άδειο και τιμή FALSE στην αντίθετη περίπτωση.

Το πρόβλημα αυτό προτάθηκε αρχικά από τον Dijkstra E. W. (1969), και οι παρατηρήσεις του συμπίπτουν με τα αποτελέσματα των Lucas et als (1984). Μερικοί από τους προταθέντες αλγόριθμους ήταν οι εξής κι οι μαθητές καλούνται να τους σχολιάσουν:

Αλγόριθμος 1

```
While πανέρι_όχι_άδειο  
Καθάρισε_1_πατάτα
```

Αλγόριθμος 2

```
while Πανέρι_όχι_άδειο  
  while not Αρκετές_πατάτες  
    Καθάρισε_1_πατάτα
```

Αλγόριθμος 3

```
While not Αρκετές_πατάτες  
if πανέρι_όχι_άδειο then Καθάρισε_1_πατάτα  
  else Γέμισε_πανέρι
```

Αλγόριθμος 4

```
if πανέρι_όχι_άδειο then  
  while not Αρκετές_πατάτες Καθάρισε_1_πατάτα  
  else Γέμισε_πανέρι
```

Αλγόριθμος 5

```
while not Αρκετές_πατάτες  
  while Πανέρι_όχι_άδειο  
    Καθάρισε_1_πατάτα
```

Αλγόριθμος 6

```
while not Αρκετές_πατάτες  
  if Πανέρι_όχι_άδειο then Καθάρισε_1_πατάτα
```

Αλγόριθμος 7

```
while Πανέρι_όχι_άδειο Καθάρισε_1_πατάτα  
  if not Αρκετές_πατάτες then Γέμισε_πανέρι
```

Αλγόριθμος 8

```
if Πανέρι_όχι_άδειο then  
  while not Αρκετές_πατάτες Καθάρισε_1_πατάτα  
  else  
    begin  
      Γέμισε_πανέρι;  
      while not Αρκετές_πατάτες Καθάρισε_1_πατάτα  
    end
```

Η ανάλυση των απαντήσεων των μαθητών παρουσιάζει αρκετό ενδιαφέρον (αλλά ξεπερνάει τα όρια αυτού του επιμορφωτικού υλικού). Ο εκπαιδευτικός μπορεί να εξηγήσει στους μαθητές τα σημασιολογικά σφάλματα που περιέχουν οι λανθασμένοι αλγόριθμοι.

15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

Τα 3 φύλλα εργασίας είναι περιγραμμένα πολύ αναλυτικά και δεν χρειάζονται περαιτέρω διευκρινίσεις..

16. ΑΞΙΟΛΟΓΗΣΗ

Δεν προτείνονται ειδικά θέματα για την αξιολόγηση. Το γενικό σχήμα είναι η επίλυση προβλημάτων που απαιτούν τη χρήση μιας από τις δομές αυτές .

17. ΓΙΑ ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ

Θα πρέπει να εξηγηθεί το θέμα των παρανοήσεων, αυθόρμητων αντιλήψεων κλπ των μαθητών. Ενδεχομένως οι επιμορφούμενοι θα έχουν και προσωπική εμπειρία, αλλά κατά κύριο λόγο όσοι έχουν διδάξει στο Λύκειο.

Θα πρέπει να τονιστεί επίσης το γεγονός ότι τα παραδείγματα είναι για «διδασκτική χρήση». Για παράδειγμα οι καταχωριστές δεν καταχωρούν με τέτοιο πρωτόγονο τρόπο τους βαθμούς (προφανώς θα έχουν ένα γραφικό interface και τα δεδομένα θα περνιούνται με το συνήθη, σύγχρονο τρόπο). Η «διδασκτική απλοποίηση» είναι αναγκαία για να νοηματοδοτηθεί κατάλληλα μία έννοια.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 - ΟΣΟ...ΕΠΑΝΑΛΑΒΕ

Η δομή **Όσο ισχύει_μια_συνθήκη επανάλαβε... Τέλος_επανάληψης**

Δραστηριότητα 1

Το παρακάτω πρόγραμμα επιτρέπει στο χρήστη να «καταχωρεί» τριάδες αριθμών που αντιπροσωπεύουν κωδικούς υποψηφίων στις Πανελλαδικές, και 2 βαθμούς στο μάθημα της Ιστορίας (βαθμός 1^{ου} και 2^{ου} βαθμολογητή για κάθε υποψήφιο).

Αλγόριθμος ΒΑΘΜΟΛΟΓΙΕΣ_ΣΕ_ΕΝΑ_ΜΑΘΗΜΑ

Εκτύπωσε «ΔΩΣΕ ΜΟΥ ΚΩΔΙΚΟ ΓΙΑ ΤΟΝ ΕΠΟΜΕΝΟ ΜΑΘΗΤΗ ή -999 ΓΙΑ ΔΙΑΚΟΠΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ»

Διάβασε ΚΩΔΙΚΟΣ

Όσο ΚΩΔΙΚΟΣ \neq -999 **επανάλαβε**

Εκτύπωσε «ΔΩΣΕ ΜΟΥ ΤΟΝ ΠΡΩΤΟ ΒΑΘΜΟ»

Διάβασε ΒΑΘΜΟΣ_1

Εκτύπωσε «ΔΩΣΕ ΜΟΥ ΤΟ ΔΕΥΤΕΡΟ ΒΑΘΜΟ»

Διάβασε ΒΑΘΜΟΣ_2

Εκτύπωσε «ΔΩΣΕ ΜΟΥ ΚΩΔΙΚΟ ΓΙΑ ΤΟΝ ΕΠΟΜΕΝΟ ΜΑΘΗΤΗ ή -999 ΓΙΑ ΔΙΑΚΟΠΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ»

Διάβασε ΚΩΔΙΚΟΣ

Τέλος_επανάληψης

Τέλος

- Εκτέλεσε το πρόγραμμα για μερικές τριάδες, όπως

684468	18	20
976646	11	9
123545	12	14
856641	15	15
- Προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής, των λογικών εκφράσεων κλπ).
- Παρατήρησε ταυτόχρονα στο παράθυρο «παρατήρησης» τις τιμές των μεταβλητών και των διαφόρων εκφράσεων.
- Συμφωνούν με τις προβλέψεις σου;
- Διάκοψε τη λειτουργία του προγράμματος δίνοντας τον κωδικό -999 σε κάποια στιγμή.
- Δοκίμασε και άλλες εκτελέσεις με περισσότερες ή λιγότερες τριάδες.
- Τι συμβαίνει αν δοθεί κατευθείαν κωδικός -999 (στην αρχή του προγράμματος);

Δραστηριότητα 2

- Προσπάθησε, τροποποιώντας το πρόγραμμα που δοκίμασες παραπάνω, να

δημιουργήσεις ένα πρόγραμμα με το οποίο να διαβάσεις, ένα προς ένα, τα γράμματα μιας λέξης. Το πρόγραμμα θα τερματίζεται όταν θα εισαχθεί ως χαρακτήρας μια τελεία «.».

- Προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής, των λογικών εκφράσεων κλπ).
- Παρατήρησε ταυτόχρονα στο παράθυρο «παρατήρησης» τις τιμές των μεταβλητών και των διαφόρων εκφράσεων. Συμφωνούν με τις προβλέψεις σου;

Το πρόγραμμα στο αρχείο file_while_2 κάνει αυτό που ζητείται παραπάνω. Μπορείς, αν θέλεις, να συγκρίνεις το πρόγραμμα σου με αυτό.

Δραστηριότητα 3

Προσπάθησε, τροποποιώντας το πρόγραμμα που δοκίμασες παραπάνω, να δημιουργήσεις ένα πρόγραμμα με το οποίο να «καταχωρούνται» οι βαθμοί ενός μαθητή. Η διαδικασία θα τερματίζεται όταν δοθεί ο αριθμός «-9» που δεν είναι πιθανό να είναι βαθμός κάποιου μαθητή σε οποιοδήποτε μάθημα.

- Προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής, των λογικών εκφράσεων κλπ).
- Παρατήρησε ταυτόχρονα στο παράθυρο «παρατήρησης» τις τιμές των μεταβλητών και των διαφόρων εκφράσεων. Συμφωνούν με τις προβλέψεις σου;
- Διάκοψε τη λειτουργία του προγράμματος δίνοντας τον κωδικό -9 σε κάποια στιγμή.

Στη συνέχεια:

- Προσπάθησε να τροποποιήσεις το παραπάνω πρόγραμμα ώστε να εκτυπώνεται στο τέλος **το πλήθος** των βαθμών του μαθητή (δηλαδή πόσους βαθμούς «καταχώρησες»). Οι μαθητές δεν έχουν βέβαια το ίδιο πλήθος βαθμών (άλλοι- για παράδειγμα – είναι ενδεχόμενο να έχουν απαλλαγή σε κάποιο μάθημα κλπ). Το πρόγραμμα σου δίνει σωστό αποτέλεσμα αν δεν καταχωρήσεις κανένα βαθμό;
- Προσπάθησε να τροποποιήσεις το πρόγραμμα ώστε να υπολογίζει **το άθροισμα** των βαθμών του μαθητή. Το πρόγραμμα σου δίνει σωστό αποτέλεσμα αν δεν καταχωρήσεις κανένα βαθμό;
- Προσπάθησε να συνδυάσεις τα δυο παραπάνω προγράμματα και να υπολογίζεις τη **μέση τιμή** (αυτό που ονομάζουμε συνήθως *μέσο όρο*) των βαθμών του μαθητή. Το πρόγραμμα σου δίνει σωστό αποτέλεσμα αν δεν καταχωρήσεις κανένα βαθμό;

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 - Επανάλαβε... μέχρις_ότου

Η δομή Επανάλαβε... μέχρις_ότου

Δραστηριότητα 1

Το παρακάτω πρόγραμμα επιτρέπει την καταμέτρηση των χαρακτήρων μιας πρότασης που τελειώνει με τελεία "." (είναι μια δυνατότητα ενσωματωμένη σε πολλούς επεξεργαστές κειμένου και σε πολλές περιπτώσεις είναι χρήσιμη).

Αλγόριθμος ΜΕΤΡΑ_ΧΑΡΑΚΤΗΡΕΣ_ΣΕ_ΜΙΑ_ΠΡΟΤΑΣΗ

ΜΕΤΡΗΤΗΣ ← 0

Επανάλαβε

ΜΕΤΡΗΤΗΣ ← ΜΕΤΡΗΤΗΣ + 1

Διάβασε ΓΡΑΜΜΑ

Μέχρις_ότου ΓΡΑΜΜΑ = '.'

ΜΕΤΡΗΤΗΣ ← ΜΕΤΡΗΤΗΣ - 1

Εκτύπωσε ΜΕΤΡΗΤΗΣ

Τέλος

- Εκτέλεσε το πρόγραμμα για μερικές προτάσεις, όπως
ΑΓΑΠΩ ΠΑΡΑ ΠΟΛΥ ΤΟ ΣΧΟΛΕΙΟ.
ΑΓΑΠΩ ΠΟΛΥ ΤΟΥΣ ΚΑΘΗΓΗΤΕΣ ΜΟΥ.
ΛΕΩ ΨΕΜΜΑΤΑ.
Αλλά δεν τα λέω πάντοτε.
...
ΑΑ.
- Προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής, των λογικών εκφράσεων κλπ).
- Παρατήρησε ταυτόχρονα στο παράθυρο «παρατήρησης» τις τιμές των μεταβλητών και των διαφόρων εκφράσεων.
- Συμφωνούν με τις προβλέψεις σου;
- Δοκίμασε και άλλες εκτελέσεις με πολύ «μεγάλες» ή «μικρές» προτάσεις.
- Τι συμβαίνει αν δοθεί κατευθείαν μια τελεία «.», στην αρχή του προγράμματος;

Δραστηριότητα 2

Προτείνεται το εξής πρόβλημα: από το πληκτρολόγιο δίνουμε θετικούς αριθμούς και υπολογίζουμε το άθροισμα τους. Θέλουμε το πρόγραμμα να τερματιστεί μόλις το συγκεκριμένο άθροισμα υπερβεί το 200 και να εκτυπωθεί.

- Προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής, των λογικών εκφράσεων κλπ).
- Παρατήρησε ταυτόχρονα στο παράθυρο «παρατήρησης» τις τιμές των μεταβλητών και των διαφόρων εκφράσεων. Συμφωνούν με τις προβλέψεις σου;
- Μπορείς να τροποποιήσεις το πρόγραμμά σου έτσι ώστε το «όριο» 200 να μην είναι σταθερό, αλλά να μπορεί να το προσδιορίζει ο χρήστης κάθε φορά που ξεκινάει να εκτελεί το πρόγραμμα;

Το πρόγραμμα στο αρχείο file_repeat_2 κάνει αυτό που ζητείται παραπάνω. Μπορείς, αν θέλεις, να συγκρίνεις το πρόγραμμά σου με αυτό.

Δραστηριότητα 3

Προτείνεται το εξής πρόβλημα: μια πιο εξελιγμένη μορφή τηλεγραφημάτων είναι τα telex. Τα telex επιτρέπουν μικρά και κεφαλαία γράμματα, τόνους και μπορούν να πληκτρολογηθούν προκειμένου το κείμενο να αποσταλεί όλο μαζί. Θέλουμε να μετρούμε πόσες λέξεις έχει ένα telex (για να «χρεώσουμε» τον πελάτη).

Ένα telex είναι καθορισμένο με αυστηρό τρόπο: αποτελείται από ένα απροσδιόριστο αριθμό λέξεων (τουλάχιστον μία λέξη σε κάθε telex) και η τελευταία λέξη ακολουθείται πάντοτε από μια τελεία («.»). Ένα telex ξεκινάει πάντα με μια λέξη και ανάμεσα σε δυο λέξεις μεσολαβεί πάντοτε ένας κενός χαρακτήρας (space).

Προσπάθησε να γράψεις ένα πρόγραμμα που να «μετράει» τις λέξεις σε ένα telex, που εισάγεται χαρακτήρα-προς-χαρακτήρα από το πληκτρολόγιο.

Υπόδειξη: για να μετρηθούν οι λέξεις, μπορούν να μετρηθούν οι κενοί χαρακτήρες (space) και η τελεία στο τέλος «.».

- Προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής, των λογικών εκφράσεων κλπ).
- Παρατήρησε ταυτόχρονα στο παράθυρο «παρατήρησης» τις τιμές των μεταβλητών και των διαφόρων εκφράσεων. Συμφωνούν με τις προβλέψεις σου;

- Αν εισαχθεί μια και μόνη λέξη σε ένα telex, το πρόγραμμα σου είναι «σωστό» (δηλαδή μετράει σωστά μια και μόνη λέξη);
- Αν εισαχθεί μόνο μια τελεία «.» στην αρχή του προγράμματος, τι θα συμβεί;

Τα προγράμματα στα αρχεία file_repeat_3a και file_repeat_3b κάνουν αυτό που ζητείται παραπάνω. Μπορείς, αν θέλεις, να συγκρίνεις το πρόγραμμα σου με αυτά. Είναι τα προγράμματα αυτά ισοδύναμα (δηλαδή δίνουν ακριβώς το ίδιο αποτέλεσμα); Αν υποθέσουμε ότι μετρούμε τις εντολές που εκτελούνται σε κάθε επανάληψη, τα δυο προγράμματα είναι ισότιμα από την άποψη αυτή;

Υπάρχει τρόπος να εξετάσουμε το τι συμβαίνει στην πραγματικότητα, με τη βοήθεια του ίδιου του προγράμματος;

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3 - Για...από...μέχρι...

- Η δομή Για...από...μέχρι...

Δραστηριότητα 1

Ένα πρόβλημα «κατάλληλο» για την εισαγωγή στο FOR I=1 to N STEP K

Να βρεθεί το άθροισμα των αριθμών 1+...+300

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 - ΕΠΑΝΑΛΗΨΗ ΕΜΠΕΔΩΣΗ

Πρόσθετες ασκήσεις, δραστηριότητες

Θέματα εμπέδωσης, αξιολόγησης και αυτοαξιολόγησης των μαθητών

ΕΡΩΤΗΣΕΙΣ ΚΛΕΙΣΤΟΥ ΤΥΠΟΥ

- (1) Οι διάφοροι τύποι επαναληπτικών δομών (ΟΣΟ...ΕΠΑΝΑΛΑΒΕ... κλπ) είναι ουσιαστικά ισοδύναμοι, δηλαδή οποιοδήποτε πρόβλημα μπορεί να επιλυθεί και με τις τρεις επαναληπτικές δομές.

ΣΩΣΤΟ **ΛΑΘΟΣ**

Υπάρχουν κατηγορίες προβλημάτων οι οποίες λύνονται με ένα είδος επαναληπτικής δομής, αλλά όχι με άλλο. Η δομή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ... είναι, ωστόσο, η πιο «ισχυρή», γιατί με αυτήν μπορούμε να επιλύσουμε όλα τα προβλήματα – αν και η χρήση των άλλων τύπων, μπορεί να κάνει πιο εύκολη τη λύση ενός προβλήματος.

- (2) Κάθε πρόβλημα που χρειάζεται για την επίλυσή του μια επαναληπτική δομή τύπου ΓΙΑ...ΜΕΧΡΙ..., μπορεί να επιλυθεί και με μια δομή τύπου ΟΣΟ...ΕΠΑΝΑΛΑΒΕ

ΣΩΣΤΟ ΛΑΘΟΣ

Η δομή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ μπορεί πάντοτε να χρησιμοποιηθεί.

- (3) Σε μια δομή του είδους ΟΣΟ...ΕΠΑΝΑΛΑΒΕ... οι εντολές που είναι στο «σώμα» (στο εσωτερικό της, από την εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ, μέχρι το ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ) θα εκτελεστούν οπωσδήποτε, τουλάχιστον μία φορά.

ΣΩΣΤΟ **ΛΑΘΟΣ**

Αν η αρχική συνθήκη είναι ψευδής, τότε οι εντολές μέσα στο «σώμα» της δομής δεν θα εκτελεστούν καμία φορά.

- (4) Σε μια δομή του είδους ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙΣ_ΟΤΟΥ οι εντολές που είναι στο «σώμα» (στο εσωτερικό της, από την εντολή ΕΠΑΝΑΛΑΒΕ..., μέχρι το ΜΕΧΡΙΣ_ΟΤΟΥ) θα εκτελεστούν τουλάχιστον μία φορά.

ΣΩΣΤΟ ΛΑΘΟΣ

- (5) σε μια δομή τύπου ΓΙΑ μετ ΑΠΟ ΑΤ ΜΕΧΡΙ ΤΤ ΜΕ_ΒΗΜΑ Β, αν γνωρίζουμε την αρχική τιμή ΑΤ, την τελική ΤΙΜΗ ΤΤ και το «βήμα» Β, μπορούμε να υπολογίσουμε το πλήθος των επαναλήψεων, με τον εξής τρόπο:

ΑΝ ΑΤ>ΤΤ ΤΟΤΕ

ΑΡΙΘΜΟΣ-ΕΠΑΝΑΛΗΨΕΩΝ $\leftarrow 0$

ΑΛΛΙΩΣ

ΑΡΙΘΜΟΣ-ΕΠΑΝΑΛΗΨΕΩΝ $\leftarrow 1 + \text{ΑΚΕΡΑΙΟ ΜΕΡΟΣ } ((ΤΤ-ΑΤ)/Β)$

ΤΕΛΟΣ_ΑΝ

ΣΩΣΤΟ ΛΑΘΟΣ

- (6) σε μια δομή τύπου ΓΙΑ μετ ΑΠΟ ΑΤ ΜΕΧΡΙ ΤΤ ΜΕ_ΒΗΜΑ Β, αν γνωρίζουμε την αρχική τιμή ΑΤ, την τελική ΤΙΜΗ ΤΤ και το «βήμα» Β, μπορούμε να υπολογίσουμε το πλήθος των επαναλήψεων, με τον αλγόριθμο που περιγράφεται με τον παρακάτω ψευδοκώδικα:

ΑΡΙΘΜΟΣ-ΕΠΑΝΑΛΗΨΕΩΝ $\leftarrow 0$

ΟΣΟ ΑΤ<=ΤΤ ΕΠΑΝΑΛΑΒΕ

ΑΡΙΘΜΟΣ-ΕΠΑΝΑΛΗΨΕΩΝ \leftarrow ΑΡΙΘΜΟΣ-ΕΠΑΝΑΛΗΨΕΩΝ + 1

ΑΤ \leftarrow ΑΤ + Β

ΤΕΛΟΣ_ΕΠΑΝΑΛΑΒΕ

ΕΚΤΥΠΩΣΕ ΑΡΙΘΜΟΣ-ΕΠΑΝΑΛΗΨΕΩΝ

ΣΩΣΤΟ ΛΑΘΟΣ

- (7) σε μια δομή τύπου ΓΙΑ μετ ΑΠΟ ΑΤ ΜΕΧΡΙ ΤΤ ΜΕ_ΒΗΜΑ Β, αν γνωρίζουμε την αρχική τιμή ΑΤ, την τελική ΤΙΜΗ ΤΤ και το «βήμα» Β, μπορούμε να υπολογίσουμε το πλήθος των επαναλήψεων, με τον τύπο που ακολουθεί:

Αριθμός επαναλήψεων = $1 + (ΤΤ-ΑΤ)/Β$

ΣΩΣΤΟ ΛΑΘΟΣ

ΠΡΟΒΛΗΜΑ

Σε ένα συρτάρι έχουμε 100 κάλτσες κόκκινες και 100 κάλτσες μαύρες.

Εφαρμόζουμε τον ακόλουθο αλγόριθμο:

ΟΣΟ ΔΕΝ_ΕΧΕΙ_ΣΧΗΜΑΤΙΣΤΕΙ_ΕΝΑ_ΖΕΥΓΑΡΙ_ΚΑΛΤΣΕΣ ΕΠΑΝΑΛΑΒΕ

ΒΓΑΛΕ_ΜΙΑ_ΚΑΛΤΣΑ_ΑΠΟ_ΤΟ_ΣΥΡΤΑΡΙ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Ερώτηση: ποιος είναι ο μέγιστος αριθμός φορών που θα εκτελεστεί αυτή η επανάληψη;

ΠΡΟΒΛΗΜΑ

Ο παρακάτω αλγόριθμος περιγράφεται σε «φυσική γλώσσα» και υποτίθεται ότι υπολογίζει το αποτέλεσμα ύψωσης ενός αριθμού σε δύναμη που είναι ακέραιος αριθμός: A^V

Είναι σωστός ο ισχυρισμός αυτός; Μπορείς να εξηγήσεις γιατί;

Μετάτρεψε τον αλγόριθμο αυτό σε πρόγραμμα εκφρασμένο στη γλώσσα προγραμματισμού ΓΛΩΣΣΑ.

ΑΛΓΟΡΙΘΜΟΣ

Χρησιμοποιούμε μια βοηθητική μεταβλητή Z, με αρχική τιμή 1.

Μεινουμε σταδιακά το v (που υποτίθεται ότι είναι ακέραιος αριθμός) κατά μία μονάδα, έως ότου γίνει μηδέν.

Σε κάθε επανάληψη, πολλαπλασιάζουμε το Z με το A.

ΠΡΟΒΛΗΜΑ

Ο παρακάτω αλγόριθμος σε «φυσική γλώσσα» υποτίθεται ότι υπολογίζει το αποτέλεσμα ύψωσης ενός αριθμού σε δύναμη που είναι ακέραιος αριθμός: A^V

Είναι σωστός ο ισχυρισμός αυτός; Μπορείς να εξηγήσεις γιατί;

Μετάτρεψε τον αλγόριθμο αυτό σε πρόγραμμα εκφρασμένο στη γλώσσα

προγραμματισμού ΓΛΩΣΣΑ.

ΑΛΓΟΡΙΘΜΟΣ

Χρησιμοποιούμε μια βοηθητική μεταβλητή Z , με αρχική τιμή 1.

Αν το n είναι άρτιος αριθμός, τότε το διαιρούμε δια δυο και πολλαπλασιάζουμε το A με τον εαυτό του.

Αν όμως δεν είναι, τότε το n ελαττώνεται κατά μία μονάδα και το Z πολλαπλασιάζεται με το A και το αποτέλεσμα εκχωρείται στο Z .

Ο αλγόριθμος τερματίζεται όταν το N μηδενιστεί.

Το ζητούμενο είναι στη μνήμη Z .

18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

Ολοκληρώστε τη μελέτη των φύλλων εργασίας που προορίζονται για τους μαθητές. Θα μπορούσατε να επινοήσετε ανάλογα προβλήματα για την εισαγωγή στις διάφορες μορφές επανάληψης;

Ενότητα Σεναρίου 4

Από τη μια μορφή επανάληψης σε μια άλλη

ΜΕΤΑΤΡΟΠΕΣ ΔΟΜΩΝ ΕΠΑΝΑΛΗΨΗΣ

1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Μετατροπές δομών επανάληψης.

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 4 διδακτικές ώρες.

3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Το παρόν σενάριο εντάσσεται στο μάθημα Τεχνολογικής Κατεύθυνσης «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» της Γ' Λυκείου και σύμφωνα με το ΦΕΚ 345/13-4-1999, στη 2η ενότητα «Βασικές Έννοιες Αλγορίθμων».

Το σενάριο απευθύνεται σε μαθητές που έχουν ήδη διδαχθεί τις θεμελιώδεις έννοιες του προγραμματισμού, καθώς και τις βασικές δομές ακολουθίας, επιλογής και επανάληψης. Οι έννοιες αυτές διδάσκονται υποχρεωτικά και εξετάζονται στις πανελλαδικές εξετάσεις της Γ' Λυκείου για την εισαγωγή στην τριτοβάθμια Εκπαίδευση. Το διδακτικό σενάριο μπορεί να χρησιμοποιηθεί και στην τάξη της Γ' Γυμνασίου αν έχουν διδαχθεί οι μαθητές τις κύριες έννοιες του προγραμματισμού με κάποιο γραφικό περιβάλλον.

Κατ' ουσίαν, η ανάλυση του σεναρίου αυτού (όπως και η βιβλιογραφία και τα λοιπά στοιχεία της) είναι η ίδια με αυτήν της προηγούμενης ενότητας.

4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Σκοπός της δραστηριότητας είναι οι μαθητές να μάθουν να μετατρέπουν μια δομή επανάληψης σε άλλη όπου αυτό είναι εφικτό.

Στόχοι του σεναρίου είναι να μπορούν οι μαθητές:

- να περιγράψουν τη λειτουργία των τριών εντολών της δομής επανάληψης
- να μετατρέπουν την ΟΣΟ στη ΜΕΧΡΙΣ_ΟΤΟΥ και αντίστροφα
- να μετατρέπουν τη ΓΙΑ στις άλλες δύο δομές επανάληψης και αντίστροφα όπου αυτό είναι εφικτό
- να αναγνωρίζουν τη σημαντικότητα των εντολών της δομής επανάληψης και τη διαφορετικότητά τους.

Αναμένεται ακόμη οι μαθητές να είναι σε θέση να διακρίνουν πότε πρέπει να χρησιμοποιήσουν μια δομή επανάληψης και ποια είναι η καταλληλότερη για την επίλυση του συγκεκριμένου προβλήματος.. Επίσης, οι μαθητές μετά τη χρήση του σεναρίου θα εξοικειωθούν ακόμη περισσότερο με τη δομή ακολουθίας και τη χρήση των μεταβλητών.

5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το σενάριο εστιάζει στο 2^ο κεφάλαιο του μαθήματος Τεχνολογικής Κατεύθυνσης «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» που αναφέρεται στις δομές επανάληψης. Ειδικότερα, οι μαθητές θα εξοικειωθούν με τη μετατροπή μιας δομής επανάληψης σε άλλη και θα αξιολογήσουν τις δραστηριότητές τους.

Ο εκπαιδευτικός παρουσιάζει τις δομές επανάληψης με συγκεκριμένα παραδείγματα, στη συνέχεια χωρίζει τους μαθητές σε ομάδες και τους δίνει τη δυνατότητα να λύσουν προβλήματα μόνο με συγκεκριμένες επιτρεπόμενες εντολές. Στην πρώτη ομάδα δίνεται η δυνατότητα να χρησιμοποιήσει μόνο την εντολή ΟΣΟ, στη δεύτερη αντίστοιχα τη ΜΕΧΡΙΣ_ΟΤΟΥ και στην τρίτη τη ΓΙΑ. Στο τέλος γίνεται αντιπαραβολή των λύσεων και οι μαθητές καλούνται να τις αξιολογήσουν ως προς την ορθότητα και την απόδοσή τους.

Τα βήματα των φύλλου δραστηριότητας εμπεριέχουν:

- λεκτική και διαγραμματική περιγραφή των δομών επανάληψης
- μελέτη-συζήτηση σχετικά με τις αλλαγές από τη μια δομή επανάληψης στην άλλη

και περιλαμβάνουν:

- ποικιλομορφία με την ενίσχυση πολλών διαφορετικών αναπαραστάσεων ενός αλγορίθμου,
- εξατομίκευση με τη διευκόλυνση των μαθητών να περιγράψουν λεκτικά έναν αλγόριθμο αλλά και διαγραμματικά,
- δυνατότητα αποκαλυπτικής & διερευνητικής μάθησης με τη δυνατότητα της μετατροπής από τη μια δομή επανάληψης στην άλλη,
- ανάπτυξη επιπρόσθετων δεξιοτήτων και εγρήγορης για τους κινδύνους που εγκυμονούν με την παρανόηση έναρξης ή λήξης ενός επαναληπτικού βρόχου.

Στο πλαίσιο των πολλαπλών πλαισίων εκφοράς ή αναπαράστασης οι μαθητές καλούνται: α) να περιγράψουν λεκτικά τον αλγόριθμο, β) να δημιουργήσουν το διάγραμμα ροής, γ) μετατρέψουν μια δομή επανάληψης και δ) να αξιολογήσουν και να αυτοαξιολογήσουν συγκρίνοντας τις λύσεις τους με τις προτεινόμενες λύσεις, αλλά και λαμβάνοντας υπόψη τις συμβουλές λύσης.

Στα φύλλα εργασίας περιλαμβάνονται

- (1) ένα έτοιμο πρόγραμμα που εκτελούν οι μαθητές και παρατηρούν τις μεταβολές των τιμών των μεταβλητών και εκφράσεων.
- (2) ένα απλό πρόβλημα προς επίλυση
- (3) ένα πιο σύνθετο πρόβλημα προς επίλυση
- (4) μερικές ερωτήσεις αυτοαξιολόγησης
- (5) διάφορα τεστ και παιχνίδια.

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Οι επαναληπτικές δομές υπήρξαν από τα πλέον διερευνημένα θέματα στη Διδακτική της Πληροφορικής, σε όλες τις βαθμίδες εκπαίδευσης, από πολλούς ερευνητές κι ερευνητικές ομάδες.

Θεωρείται ότι είναι η ίδια με την προηγούμενης ενότητας (εισαγωγή στις δομές επανάληψης).

7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ

ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Το σενάριο θα υλοποιηθεί στο εργαστήριο Πληροφορικής κάνοντας χρήση των λογισμικών «Γλώσσα» ή «Δημιουργός Διαγραμμάτων Ροής». Το περιβάλλον «Γλώσσα» ενδείκνυται καθώς έχει απλή διεπιφάνεια χρήστη και απλό συντάκτη που διευκολύνει τη συγγραφή κώδικα. Επιτρέπει τη βηματική εκτέλεση προγραμμάτων και αλγορίθμων και επισημαίνει τα λάθη που εμφανίζονται με περιγραφικό και κατανοητό τρόπο. Μπορεί να δράσει συμπληρωματικά με το μάθημα και να υποβοηθήσει τη διεξαγωγή του σεναρίου.

8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

Κατά τη μετατροπή των εντολών της δομής επανάληψης οι μαθητές συναντούν αρκετές δυσκολίες. Οι περισσότερες ανάγονται στη συνθήκη τερματισμού των εντολών. Οι μαθητές δυσκολεύονται να αντιληφθούν τότε τερματίζει ένας βρόγχος. Ειδικότερα, τα συνηθέστερα λάθη εντοπίζονται στον αρχικό έλεγχο της συνθήκης ΟΣΟ και στον έλεγχο που γίνεται στο τέλος στην εντολή ΜΕΧΡΙΣ_ΟΤΟΥ, διασφαλίζοντας τουλάχιστον μία εκτέλεση του μπλοκ εντολών στη δεύτερη περίπτωση. Ιδιαίτερες δυσκολίες αντιμετωπίζονται στη μετατροπή της ΟΣΟ και της ΜΕΧΡΙΣ_ΟΤΟΥ στην εντολή ΓΙΑ ορίζοντας λανθασμένη συνθήκη τερματισμού. Οι δυσκολίες γίνονται πολύ πιο έντονες όταν γίνεται εμφωλευμένη χρήση εντολών της δομής επανάληψης και ζητείται η μετατροπή τους.

Οι συνηθέστερες δυσκολίες που συναντούν οι μαθητές στην κατανόηση των δομών επανάληψης είναι οι εξής:

Ανεπαρκή νοητά μοντέλα: Οι μαθητές διαθέτουν ανεπαρκή μοντέλα για τις επαναληπτικές δομές. Επιπλέον, μερικές φορές τα λάθη στα νοητά μοντέλα των αρχάριων προγραμματιστών είναι ανεπαίσθητα και δύσκολο να εντοπιστούν.

Οι αναλλοίωτες σχέσεις που διέπουν μια επαναληπτική δομή (Loop invariant): Οι αρχάριοι προγραμματιστές δυσκολεύονται στον καθορισμό του τμήματος ενημέρωσης ενός βρόγχου (loop invariant) που αποτελεί σημαντικό συστατικό του. Οι μαθητές βασίζονται στα μοντέλα των βρόγχων στην αναπαράσταση μιας αλληλουχίας ενεργειών (δυναμικό μοντέλο) και όχι στην αναπαράσταση μιας αλληλουχίας σχέσεων μεταξύ των μεταβλητών (στατικό μοντέλο). Επίσης, χρησιμοποιούν διαφορετικά ονόματα σε κάθε βήμα της

επανάληψης για τη σηματοδότησης μιας μεταβλητής με συγκεκριμένη λειτουργία (functional variable) και δεν έχουν την ικανότητα να προσδιορίσουν αυθόρμητα μια συνθήκη εξόδου.

Δυσκολίες, ακόμη, εμφανίζονται και στην επιλογή της καταλληλότερης δομής:

- Οι μαθητές συνήθως χρησιμοποιούν τη στρατηγική bottom-exit, δηλαδή τη δομή ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ .. ΜΕΧΡΙΣ_ΟΤΟΥ.
- Οι μαθητές δυσκολεύονται περισσότερο με σχέδια που βασίζονται στη στρατηγική «έλεγχος τιμής μεταβλητής/ επεξεργασία μεταβλητής», παρά στη στρατηγική «επεξεργασία/ έλεγχος».
- Πολλοί μαθητές πιστεύουν ότι ένας βρόγχος «ΟΣΟ...ΕΠΑΝΑΛΑΒΕ» τερματίζεται τη στιγμή που η συνθήκη ελέγχου παύει να αληθεύει (daemon-exit) και όχι αφού η συνθήκη ελεγχθεί κατά την αμέσως επόμενη επανάληψη.
- Όσον αφορά τις δομές «ΟΣΟ..ΕΠΑΝΑΛΑΒΕ» και «ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ...ΜΕΧΡΙΣ_ΟΤΟΥ» 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-σημαιοφ (99999). Οι δυσκολίες εντοπίζονται στην αλληλεπίδραση της ανάγνωσης των δεδομένων και του βρόγχου.

9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ

Οι μαθητές εμπλέκονται ενεργά πειραματιζόμενοι με τις αλλαγές των δομών επανάληψης. Κατά την εκτέλεση των αλγορίθμων με τη χρήση της «Γλώσσας» εκτιμάται ότι θα υπάρξουν παρανοήσεις μεταξύ των δομών επανάληψης, ενώ το λογισμικό θα βοηθήσει τους μαθητές στο στάδιο της αξιολόγησης.

Επιπλέον, θα δημιουργηθεί προβληματισμός στην αναγκαιότητα διερεύνησης της μετατροπής της εντολής ΟΣΟ και της ΜΕΧΡΙΣ_ΟΤΟΥ στη ΓΙΑ. Η αντιμετώπιση αυτού του είδους διδακτικού θορύβου θα ξεπεραστεί με το φύλλο λύσεων-συμβουλών, αλλά και τη βοήθεια του διδάσκοντα.

10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Υπάρχει πολύ πλούσια διεθνής και ελληνική βιβλιογραφία – ακόμη και ελληνικές διδακτορικές διατριβές στο αντικείμενο. Δεν αναφέρονται πηγές, γιατί η ταχύτητα ανανέωσης είναι πολύ μεγάλη.

Ενδεικτικό link: <http://www.helenemartin.com/2010-05-while-loops-gone-wild/>

Υπάρχουν πολλά διεθνή Συνέδρια, με προεξάρχοντα 2 Συνέδρια της ACM, το ένα στις Η.Π.Α., ετήσιο (γκρουπ) SIGCSE και ένα στην Ευρώπη (ITICSE – το 2003 μάλιστα διοργανώθηκες τη Θεσσαλονίκη).

Η διεθνής ομάδα PPIG, ενδιαφέρεται ιδιαίτερα για την ψυχολογική διάσταση του προγραμματισμού και διοργανώνει επίσης διεθνή Συνέδρια.

Στην Ελλάδα, οι επαγγελματικές και ακαδημαϊκές ενώσεις Πληροφορικών και καθηγητών Πληροφορικής (ΕΠΥ και ΠΕΚΑΠ) διοργανώνουν τακτικά Συνέδρια ειδικά για τη διδασκαλία και τη Διδακτική της Πληροφορικής.

Πολλοί διεθνείς οργανισμοί (όπως η ACM και η ISTE <http://www.iste.org>) έχουν τμήματα αφιερωμένα ειδικά στη διδασκαλία της Πληροφορικής και ειδικότερα στις δυσκολίες που αντιμετωπίζουν κυρίως οι αρχάριοι προγραμματιστές στην επιλογή της κατάλληλης δομής επανάληψης.

11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Στους μαθητές προτείνονται μια σειρά από προβλήματα τα οποία καλούνται να επιλύσουν.

Οι διαφορές ανάμεσα στους διάφορους τύπους δομών επανάληψης, δεν αναμένεται να είναι πάντοτε άμεσα αντιληπτές από τους μαθητές. Εξάλλου το γεγονός ότι σε ορισμένες περιπτώσεις οι διαφορές είναι δυσδιάκριτες, μπορεί να οδηγήσει σε προβλήματα για την επιλογή της καταλληλότερης δομής. Για το λόγο αυτό είναι ενδεχόμενο να υπάρξουν διαφορές στην επιλογή της καταλληλότερης δομής και επομένως, με τη βοήθεια του εκπαιδευτικού, ανάπτυξη σχετικού διαλόγου.

12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ

Δεν φαίνεται να έχουμε.

13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Το μάθημα θα πραγματοποιηθεί στο σχολικό εργαστήριο πληροφορικής. Οι μαθητές προτείνεται να εργαστούν σε ομάδες των δύο ατόμων ανά ηλεκτρονικό υπολογιστή.

14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Οι δομές επανάληψης μπορούν να προσομοιωθούν με πολλά παραδείγματα από την καθημερινή ζωή. Μπορούν να δοθούν παραδείγματα διαθεματικά με άλλα μαθήματα (πχ. Φυσικής, Μαθηματικών κτλ.). Ακόμη, προσομοιώσεις θα μπορούσαν να δημιουργηθούν με τα εκπαιδευτικά περιβάλλοντα Scratch, Logo και Yenka. Πολλά παραδείγματα για τις δομές επανάληψης και τις μετατροπές μεταξύ των δομών βρίσκονται ενσωματωμένα στο περιβάλλον Yenka.

15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

Τα φύλλα εργασίας περιγράφονται με πολύ λεπτομερή τρόπο.

16. ΑΞΙΟΛΟΓΗΣΗ

Ο εκπαιδευτικός μπορεί να προτείνει τεστ αυτοαξιολόγησης (κλειστού τύπου), σταυρόλεξα, κρυπτόλεξα κλπ.

Το λογισμικό HotPotatoes (και άλλα ανάλογα, όπως και ορισμένες επιλογές του Moodle) προσφέρονται για συστηματική δημιουργία ερωτημάτων κλειστού τύπου. Κρυπτόλεξα μπορούν εύκολα να δημιουργηθούν ακόμη και σε ένα υπολογιστικό φύλλο.

17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ

Θα εξηγηθεί το θέμα των παρανοήσεων και των αυθόρμητων αντιλήψεων των μαθητών. Θα δοθούν φύλλα εργασίας τα οποία παρατίθενται παρακάτω.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1

Δραστηριότητα 1

Μετατροπή από τη Δομή Επανάληψης ΟΣΟ στη ΜΕΧΡΙΣ_ΟΤΟΥ και τη ΓΙΑ (όταν είναι εφικτό)

Δίνεται ο αλγόριθμος:

```
1 Αλγόριθμος δ1
2 άθροισμα ← 3
3 i ← 5
4 Όσο i ≤ 40 επανάλαβε
5   άθροισμα ← άθροισμα + i
6   i ← i + 10
7 Τέλος_επανάληψης
8 Εμφάνισε άθροισμα
9 Τέλος δ1
```

- 1.1. Να περιγράψετε λεκτικά τη λειτουργία του αλγόριθμου
- 1.2. Να γίνει το διάγραμμα ροής του αλγόριθμου
- 1.3. Να μετατρέψετε τη δομή επανάληψης ΟΣΟ στη Μέχρις_ΟΤΟΥ
- 1.4. Να μετατρέψετε τη δομή επανάληψης Μέχρις_ΟΤΟΥ στη ΓΙΑ (αν είναι εφικτό)

Συγκρίνετε τις λύσεις σας με τις προτεινόμενες απαντήσεις, εντοπίστε τις διαφορές με τις δικές σας απαντήσεις και αξιολογήστε τα λάθη σας.

Δραστηριότητα 2

Μετατροπή από τη Δομή Επανάληψης ΓΙΑ στην ΟΣΟ και τη ΜΕΧΡΙΣ_ΟΤΟΥ

Δίνεται ο αλγόριθμος:

```
1 Αλγόριθμος δ2
2 σύνολο_πόντων ← 0
3 Μετρητής ← 0
4 Για i από 1 μέχρι 10
5   Εμφάνισε "Δώσε όνομα παίκτη και πόντους"
6   Διάβασε παίκτης
7   Διάβασε πόντοι
8   Αν πόντοι ≥ 10 τότε
9     Μετρητής ← Μετρητής + 1
10  Τέλος_αν
11  Εμφάνισε "Ο παίκτης: ", παίκτης
12  Εμφάνισε "πέτυχε: ", πόντοι, " πόντους"
13  σύνολο_πόντων ← σύνολο_πόντων + πόντοι
14 Τέλος_επανάληψης
15 Εμφάνισε "Σύνολο πόντων: ", σύνολο_πόντων
16 Εμφάνισε "Παίκτες που πέτυχαν περισσότερους από 10 πόντους: ", Μετρητής
17 Τέλος δ2
```

- 2.1. Να περιγράψετε τη λειτουργία του αλγορίθμου
- 2.2. Να σχεδιάσετε το διάγραμμα ροής του αλγορίθμου
- 2.3. Να μετατρέψετε τη δομή επανάληψης ΓΙΑ στην ΟΣΟ
- 2.4. Να μετατρέψετε τη δομή επανάληψης ΟΣΟ στη ΜΕΧΡΙΣ_ΟΤΟΥ (αν είναι εφικτό)

Συγκρίνετε τις λύσεις σας με τις προτεινόμενες απαντήσεις, εντοπίστε τις διαφορές με τις δικές σας απαντήσεις και αξιολογήστε τα λάθη σας.

Δραστηριότητα 3

Μετατροπή από τη Δομή Επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ στην ΟΣΟ

Δίνεται ο αλγόριθμος:

```
1 Αλγόριθμος δ3
2 άθροισμα ← 0
3 Διάβασε χ
4 Αρχή_επανάληψης
5   Αν χ > 12 τότε
6     άθροισμα ← άθροισμα + χ
7     Διάβασε χ
8   Τέλος_αν
9 Μέχρις_ότου χ < 12
10 Εμφάνισε άθροισμα
11 Τέλος δ3
```

- 3.1. Να περιγράψετε λεκτικά τη λειτουργία του αλγόριθμου
- 3.2. Να σχεδιάστε το διάγραμμα ροής του αλγορίθμου
- 3.3. Να μετατρέψετε τη δομή επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ στην ΟΣΟ
- 3.4. Να μετατρέψετε τη δομή επανάληψης ΟΣΟ στη ΓΙΑ

Συγκρίνετε τις λύσεις σας με τις προτεινόμενες απαντήσεις, εντοπίστε τις διαφορές με τις δικές σας απαντήσεις και αξιολογήστε τα λάθη σας.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2

Δραστηριότητα 1

Α. Να διαπιστωθεί ποια από τα παρακάτω τμήματα αλγορίθμων είναι ισοδύναμα:

1.	2.	3.
$X \leftarrow 1$	$X \leftarrow 2$	$X \leftarrow 2$
$\Psi \leftarrow 3$	$\Psi \leftarrow 11$	$\Psi \leftarrow 11$
Όσο $\Psi > 0$ επανάλαβε	Όσο $\Psi > 8$ επανάλαβε	Όσο $\Psi > 4$ επανάλαβε
$\Psi \leftarrow \Psi - 1$	$\Psi \leftarrow \Psi - 1$	$\Psi \leftarrow \Psi - 1$
$X \leftarrow 2 + X$	$X \leftarrow 2 + X$	$X \leftarrow 3 + X$
Τέλος_επανάληψης	Τέλος_επανάληψης	Τέλος_επανάληψης
Γράψε X	Γράψε X	Γράψε X
4.	5.	6.
$X \leftarrow 2$	$X \leftarrow 3$	$X \leftarrow 2$
$\Psi \leftarrow 2$	$\Psi \leftarrow 6$	$\Psi \leftarrow -3$
Όσο $\Psi \geq 0$ επανάλαβε	Όσο $\Psi > 0$ επανάλαβε	Όσο $\Psi \leq -1$ επανάλαβε
$\Psi \leftarrow \Psi - 1$	$\Psi \leftarrow \Psi - 2$	$\Psi \leftarrow \Psi + 1$
$X \leftarrow 2 + X$	$X \leftarrow 1 + X$	$X \leftarrow 2 + X$
Τέλος_επανάληψης	Τέλος_επανάληψης	Τέλος_επανάληψης
Γράψε X	Γράψε X	Γράψε X

B. Ποια από τα παρακάτω τμήματα αλγορίθμων δίνουν το ίδιο αποτέλεσμα;

1.	2.	3.
$x \leftarrow 2$	$x \leftarrow 2$	$x \leftarrow 2$
$\psi \leftarrow -3$	Για ψ από -3 μέχρι 1	$\psi \leftarrow -3$
Όσο $\psi \leq 0$ επανάλαβε	$x \leftarrow \psi + x$	Όσο $\psi \leq 0$ επανάλαβε
$\psi \leftarrow \psi + 1$	Τέλος_επανάληψης	$\psi \leftarrow \psi + 1$
$x \leftarrow \psi + x$	Γράψε x	$x \leftarrow \psi + x$
Τέλος_επανάληψης		Τέλος_επανάληψης
Γράψε x		Γράψε x

4.	5.	6.
$x \leftarrow 2$	$x \leftarrow 2$	$x \leftarrow 2$
$\psi \leftarrow -3$	Για ψ από -3 μέχρι 0	$\psi \leftarrow -3$
Όσο $\psi \leq 0$ επανάλαβε	$x \leftarrow \psi + x$	Όσο $\psi \leq 0$ επανάλαβε
$\psi \leftarrow \psi + 1$	Τέλος_επανάληψης	$\psi \leftarrow \psi + 1$
$x \leftarrow \psi + x + 1$	Γράψε x	$x \leftarrow \psi + x - 1$
Τέλος_επανάληψης		Τέλος_επανάληψης
Γράψε x		Γράψε x

Δραστηριότητα 2

Τα παρακάτω προβλήματα να υλοποιηθούν, αν είναι εφικτό, με όλες τις δομές επανάληψης. Αν δεν είναι εφικτό, αιτιολογήστε την απάντησή σας.

- A. Να υλοποιήσετε αλγόριθμο που θα εμφανίζει τους περιττούς αριθμούς από το 1 έως το 1000.
- B. Να γίνει αλγόριθμος που θα υπολογίζει το πλήθος των όρων που μπορούν να προστεθούν στην παρακάτω παράσταση έτσι ώστε το άθροισμα να μην ξεπεράσει το 2000.

$$S = 1 + 6 + 16 + 31 + 51 + 76 + \dots$$

Δραστηριότητα 3

Τα παρακάτω προβλήματα να υλοποιηθούν, αν είναι εφικτό, με όλες τις δομές επανάληψης. Αν δεν είναι εφικτό, αιτιολογήστε την απάντησή σας.

- A. Ένας κωδικός πρόσβασης αποτελείται από 3 ψηφία XYZ, όπου γνωρίζουμε ότι το X είναι περιττός αριθμός και το Z άρτιος. Να γίνει αλγόριθμος που να μας εμφανίζει όλους τους πιθανούς κωδικούς.
- B. Να γίνει αλγόριθμος που θα διαβάζει τους βαθμούς ενός μαθητή σε κάθε μάθημα και θα υπολογίζει σε πόσα μαθήματα έχει πετύχει άριστα (>18). Ο αλγόριθμος να τερματίζεται όταν δοθεί ως είσοδος ένας αρνητικός αριθμός ή ένας αριθμός μεγαλύτερος του 20. (Να γίνει έλεγχος ορθότητας δεδομένων).

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3

Δραστηριότητα 1

Να γράψετε ισοδύναμα τμήματα αλγορίθμων χρησιμοποιώντας τη δομή Όσο... επανάλαβε:

A) Διάβασε X

Αρχή_επανάληψης

$$X \leftarrow X + 1$$

Μέχρις_ότου X = -14

B) X ← -18

Αρχή_επανάληψης

$$X \leftarrow X + 1$$

Μέχρις_ότου X = -2

Γ) X ← 4

Για κ από -10 μέχρι 22 με_βήμα 4

$$X \leftarrow X + 2$$

Τέλος_επανάληψης

Δ) X ← 3

Για κ από -5 μέχρι 10 με_βήμα 1

$$X \leftarrow X + 2$$

Τέλος_επανάληψης

Δραστηριότητα 2

Να γράψετε ισοδύναμα τμήματα αλγορίθμων χρησιμοποιώντας τη δομή Για... από... μέχρι:

A) Διάβασε X

Αρχή_επανάληψης

$$K \leftarrow K * X$$

$$X \leftarrow X + 1$$

Μέχρις_ότου X < 3

B) $X \leftarrow 9$

Αρχή_επανάληψης

$K \leftarrow K * X$

$X \leftarrow X + 1$

Μέχρις_ότου $X > 20$

Γ) $X \leftarrow -15$

$\Psi \leftarrow 13$

Όσο $\Psi \leq 10$ επανάλαβε

$K \leftarrow X * 4 - 4$

$\Psi \leftarrow \Psi + 8$

$\Lambda \leftarrow K - (\Psi + 1) / 8$

Τέλος_επανάληψης

Δ) $X \leftarrow 6$

$\Psi \leftarrow 8$

Όσο $\Psi < 0$ επανάλαβε

$X \leftarrow X + 1$

$A \leftarrow 2 * X - 1$

$\Psi \leftarrow A * \Psi + 4$

Τέλος_επανάληψης

Δραστηριότητα 3

Να γράψετε ισοδύναμα τμήματα αλγορίθμων χρησιμοποιώντας τη δομή Μέχρις_ότου:

A) $X \leftarrow 8$

Όσο $X \geq 2$ επανάλαβε

$X \leftarrow X - 2$

$\Psi \leftarrow X + 1$

Τέλος_επανάληψης

B) Διάβασε X
Όσο $X > 0$ και $X \leq 12$ επανάλαβε
 $\Psi \leftarrow X + 1$
 $X \leftarrow X - 3$
Τέλος_επανάληψης

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 - ΕΠΑΝΑΛΗΨΗ ΕΜΠΕΔΩΣΗ

Πρόσθετες ασκήσεις, δραστηριότητες

Θέματα εμπέδωσης, αξιολόγησης και αυτοαξιολόγησης των μαθητών

ΕΡΩΤΗΣΕΙΣ ΚΛΕΙΣΤΟΥ ΤΥΠΟΥ

(1) Η δομή επανάληψης ΟΣΟ δεν μετατρέπεται στην εντολή επανάληψης ΓΙΑ αν δεν γνωρίζουμε το _____ των επαναλήψεων.

ΠΛΗΘΟΣ

(2) Αν το x έχει πάντα αμετάβλητη τιμή 2, τότε η εντολή επανάληψης ΟΣΟ $x < 5$ θα προκαλέσει ατέρμονα βρόγχο.

ΣΩΣΤΟ ΛΑΘΟΣ

(3) Η δομή επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ εκτελείται τουλάχιστον μία φορά.

ΣΩΣΤΟ ΛΑΘΟΣ

(4) Η δομή επανάληψης ΟΣΟ μπορεί πάντοτε να μετατραπεί στη δομή επανάληψης ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙΣ_ΟΤΟΥ

ΣΩΣΤΟ ΛΑΘΟΣ

(5) Στη δομή επανάληψης ΕΠΑΝΑΛΑΒΕ...ΜΕΧΡΙΣ_ΟΤΟΥ οι εντολές που βρίσκονται μέσα στο βρόγχο μπορεί να μην εκτελεστούν καμία φορά αν δεν ισχύει η συνθήκη ελέγχου της δομής.

ΣΩΣΤΟ ΛΑΘΟΣ

Ολοκληρώστε τη μελέτη των φύλλων εργασίας που προορίζονται για τους μαθητές.

Ενότητα Σεναρίου 5

Ταξινόμηση με τη μέθοδο φυσαλίδας

ΤΑΞΙΝΟΜΗΣΗ ΜΟΝΟΔΙΑΣΤΑΤΟΥ ΠΙΝΑΚΑ ΜΕ ΤΗ ΜΕΘΟΔΟ ΤΗΣ ΦΥΣΑΛΙΔΑΣ

1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Ταξινόμηση μονοδιάστατου πίνακα με τη μέθοδο της φυσαλίδας.

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 4 διδακτικές ώρες.

3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Το παρόν σενάριο εντάσσεται στο μάθημα Τεχνολογικής Κατεύθυνσης «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» της Γ' Λυκείου και σύμφωνα με το ΦΕΚ 345/13-4-1999, στην 3η ενότητα «Δομές Δεδομένων και Αλγόριθμοι». Το σενάριο απευθύνεται σε μαθητές που έχουν ήδη διδαχθεί τις θεμελιώδεις έννοιες του προγραμματισμού, καθώς και τις βασικές δομές ακολουθίας, επιλογής, επανάληψης καθώς και τη δομή των μονοδιάστατων πινάκων. Οι έννοιες αυτές διδάσκονται υποχρεωτικά και εξετάζονται στις πανελλαδικές εξετάσεις της Γ' Λυκείου για την εισαγωγή στην τριτοβάθμια Εκπαίδευση.

4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Η προτεινόμενη δραστηριότητα έχει σκοπό να βοηθήσει τους μαθητές να κατανοήσουν την ταξινόμηση με τη μέθοδο της φυσαλίδας. Τα φύλλα διδασκαλίας μπορούν να χρησιμοποιηθούν για να κεντρίσουν το ενδιαφέρον των μαθητών ώστε στο τέλος του μαθήματος οι μαθητές να είναι σε θέση να εκτελούν διάφορα

παραδείγματα. Αναμένεται μετά την εκτέλεση του διδακτικού σεναρίου οι μαθητές να μπορούν:

- ✓ να κατανοούν τι είναι ταξινόμηση και ποιος ο σκοπός της
- ✓ να εφαρμόζουν τον αλγόριθμο για την ταξινόμηση των στοιχείων ενός μονοδιάστατου πίνακα
- ✓ να συνεργάζονται με σκοπό την εφαρμογή του αλγόριθμου για την ταξινόμηση των στοιχείων ενός μονοδιάστατου πίνακα
- ✓ να εκτελούν τον αλγόριθμο βήμα-βήμα και να διακρίνουν τα διαφορετικά βήματα του αλγόριθμου.

Αναμένεται ακόμη οι μαθητές να μπορούν να γράψουν προγράμματα είτε στο «Διερμηνευτή της Γλώσσας», είτε στο λογισμικό «Scratch» εφαρμόζοντας τον αλγόριθμο της ταξινόμησης της φυσαλίδας και να επαληθεύουν τα αποτελέσματα που προκύπτουν. Επίσης, οι μαθητές μετά τη χρήση του σεναρίου θα εξοικειωθούν ακόμη περισσότερο με τις δομές επιλογής, επανάληψης και τη χρήση των μονοδιάστατων πινάκων.

5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το σενάριο εστιάζει στο 2^ο κεφάλαιο του μαθήματος Τεχνολογικής Κατεύθυνσης «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» που αναφέρεται στις δομές δεδομένων. Ειδικότερα, οι μαθητές θα εξοικειωθούν με την ταξινόμηση ενός μονοδιάστατου πίνακα με τη μέθοδο της φυσαλίδας. Στο τέλος κάθε επανάληψης του αλγορίθμου θα είναι σε θέση να επαληθεύουν τα αποτελέσματα που προκύπτουν και να πιστοποιούν την ορθή εφαρμογή του.

Ο διδάσκων στην αρχή του σεναρίου θα πρέπει να κάνει μία εισαγωγή αναφορικά με την ταξινόμηση και τη χρησιμότητα της ταξινόμησης στη δομή των πινάκων. Ειδικότερα, μπορεί να αναφερθεί στα πλεονεκτήματα της ταξινόμησης και πως μας βοηθάει στη επίτευξη δύσκολων προγραμματιστικών στόχων. Για παράδειγμα, στην ανίχνευση και διαγραφή των διπλότυπων τιμών, στον υπολογισμό της διαμέσου όταν έχουμε αριθμητικές τιμές σε έναν μονοδιάστατο πίνακα και στην εφαρμογή της δυαδικής αναζήτησης. Ακολούθως, μπορεί να οριστεί ο αλγόριθμος και να αναλυθεί γραμμή-γραμμή. Μία συζήτηση αναφορικά με την αποδοτικότητα του αλγορίθμου θα μπορούσε να βοηθήσει στην κατανόηση του τρόπου λειτουργίας

ΤΟΥ.

Ο αλγόριθμος ταξινόμησης με τη μέθοδο της φυσαλίδας:

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N

ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1

ΑΝ $\text{ΠινA}[j-1] > \text{ΠινA}[j]$ ΤΟΤΕ

$T \leftarrow \text{ΠινA}[j-1]$

$\text{ΠινA}[j-1] \leftarrow \text{ΠινA}[j]$

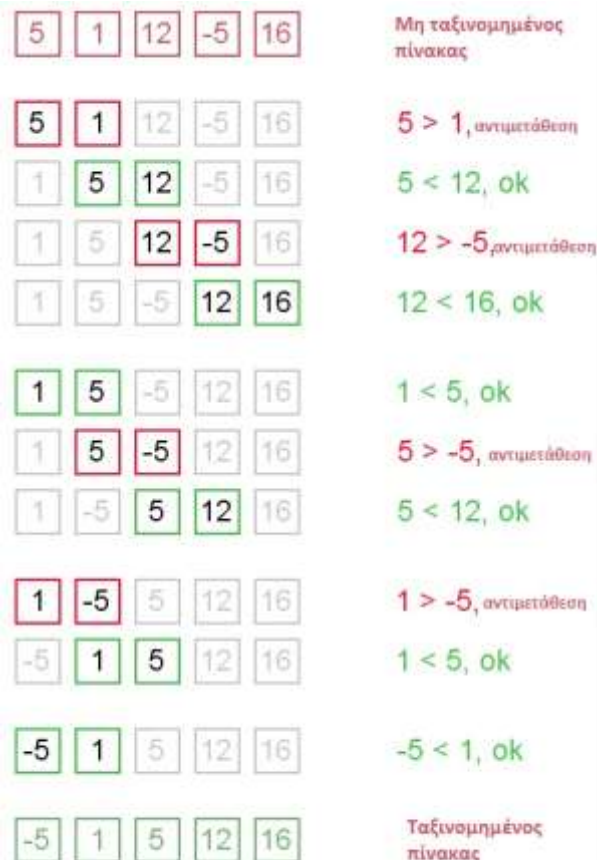
$\text{ΠινA}[j] \leftarrow T$

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Στη συνέχεια, μπορεί να δημιουργηθεί ένα παράδειγμα όπου θα εφαρμόζεται ο αλγόριθμος της φυσαλίδας. Για παράδειγμα, τον εφαρμόζουμε για τον πίνακα [5, 1, 12, -5, 6] βήμα βήμα.



Με τον ίδιο τρόπο ο διδάσκων μπορεί να δείξει και ένα ακόμη παράδειγμα, όπου θα υλοποιείται ο αλγόριθμος. Έστω, λοιπόν, πως εφαρμόζεται στον αταξινόμητο πίνακα [2, 3, 4, 5, 1] βήμα - βήμα.



Τέλος, ένα τελευταίο παράδειγμα που μπορεί να εφαρμοστεί είναι ο αταξινόμητος πίνακας [6, 1, 2, 3, 4, 5].

<table border="1"><tr><td>6</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	6	1	2	3	4	5	Μη ταξινομημένος πίνακας
6	1	2	3	4	5		
<table border="1"><tr><td>6</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	6	1	2	3	4	5	$6 > 1$, αντιμετάθεση
6	1	2	3	4	5		
<table border="1"><tr><td>1</td><td>6</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	6	2	3	4	5	$6 > 2$, αντιμετάθεση
1	6	2	3	4	5		
<table border="1"><tr><td>1</td><td>2</td><td>6</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	6	3	4	5	$6 > 3$, αντιμετάθεση
1	2	6	3	4	5		
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>6</td><td>4</td><td>5</td></tr></table>	1	2	3	6	4	5	$6 > 4$, αντιμετάθεση
1	2	3	6	4	5		
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>6</td><td>5</td></tr></table>	1	2	3	4	6	5	$6 > 5$, αντιμετάθεση
1	2	3	4	6	5		
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	1	2	3	4	5	6	$1 < 2$, ok
1	2	3	4	5	6		
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	1	2	3	4	5	6	$2 < 3$, ok
1	2	3	4	5	6		
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	1	2	3	4	5	6	$3 < 4$, ok
1	2	3	4	5	6		
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	1	2	3	4	5	6	$4 < 5$, ok
1	2	3	4	5	6		
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	1	2	3	4	5	6	Ταξινομημένος πίνακας
1	2	3	4	5	6		

Στο πλαίσιο των πολλαπλών πλαισίων εκφοράς ή αναπαράστασης οι μαθητές καλούνται: α) να περιγράψουν λεκτικά τον αλγόριθμο, β) να υλοποιήσουν ένα πρόγραμμα στο περιβάλλον “Διερμηνευτής της Γλώσσας», γ) να υλοποιήσουν ένα πρόγραμμα στο περιβάλλον Scratch και δ) να αξιολογήσουν τις λύσεις τους με γνώμονα τις προτεινόμενες λύσεις.

Στα φύλλα εργασίας περιλαμβάνονται

- (1) Η υλοποίηση του αλγορίθμου βηματικά σε συγκεκριμένα παραδείγματα.
- (2) Ένα απλό πρόβλημα προς επίλυση.
- (3) Ένα πιο σύνθετο πρόβλημα προς επίλυση.
- (4) Μερικές ερωτήσεις αξιολόγησης.
- (5) Διάφορα τεστ και παιχνίδια.

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Έρευνες έχουν γίνει και αναφέρονται στη διεθνή βιβλιογραφία σε έννοιες και δομές όπως οι μεταβλητές (Bayman & Mayer, 1983, Samurcay, 1985, Shneiderman, 1985, Du Boulay, 1986, Joni & Soloway, 1986, Pea, 1986), η δομή επανάληψης (Soloway et al., 1983, Soloway et al, 1986, Pea, 1986, Soloway, Bonar & Ehrilch, 1989), η δομή επιλογής με τον έλεγχο των συνθηκών και τις λογικές εκφράσεις (Sprohrer & Soloway, 1986, Putnam et. al., 1989, Sprohrer & Soloway, 1989, Pane & Myers, 2000).

Σε επίπεδο δομών δεδομένων και συγκεκριμένα στους πίνακες αναφέρονται αρκετά λιγότερες έρευνες όσον αφορά μαθησιακές δυσκολίες καθώς και προτάσεις αντιμετώπισης αυτών των δυσκολιών. Στις δυσκολίες αυτές αναφέρεται ότι οι μαθητές συχνά συγχέουν τον δείκτη ενός πίνακα με το αντίστοιχο στοιχείο (π.χ $i+5$ και $A[i+5]$) δυσκολεύονται να διακρίνουν το δείκτη που αντιστοιχεί σε γραμμές από αυτόν που αντιστοιχεί σε στήλες καθώς και ότι αντιμετωπίζουν αρκετές δυσκολίες στην επεξεργασία των στοιχείων ενός δισδιάστατου πίνακα κατά γραμμές ή κατά στήλες (Du Boulay, 1989).

7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Το σενάριο θα υλοποιηθεί στο εργαστήριο Πληροφορικής κάνοντας χρήση των λογισμικών «Διερμηνευτής της Γλώσσας» ή «Δημιουργός Διαγραμμάτων Ροής». Το περιβάλλον «Διερμηνευτής της Γλώσσας» ενδείκνυται καθώς έχει απλή διεπαφή χρήστη και απλό συντάκτη που διευκολύνει τη συγγραφή κώδικα. Επιτρέπει τη βηματική εκτέλεση προγραμμάτων και αλγορίθμων και επισημαίνει τα λάθη που εμφανίζονται με περιγραφικό και κατανοητό τρόπο. Μπορεί να δράσει συμπληρωματικά με το μάθημα και να υποβοηθήσει τη διεξαγωγή του σεναρίου.

8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

Οι περισσότεροι μαθητές στο συγκεκριμένο αλγόριθμο ταξινόμησης με τη μέθοδο της φυσαλίδας δυσκολεύονται να αντιληφθούν τόσο την προσπέλαση όλων των στοιχείων του πίνακα, όσο και την αντιμετάθεση των τιμών αν ισχύουν οι συνθήκες. Πιο συγκεκριμένα,

παρατηρείται πως, ενώ μετά την εφαρμογή του αλγορίθμου σε παραδείγματα μονοδιάστατων πινάκων είναι σε θέση να τον εφαρμόσουν ως μεθοδολογία, δυσκολεύονται να εφαρμόσουν γραμμή - γραμμή τον αλγόριθμο και να τυπώσουν το αποτέλεσμα των μεταβλητών, όπως αυτός ορίζεται στη θεωρία.

Οι δυσκολίες που συχνά αντιμετωπίζουν οι μαθητές, καθώς εφαρμόζουν τον αλγόριθμο εντοπίζονται στην εμφωλευμένη δομή επανάληψης ΓΙΑ j από N ΜΕΧΡΙ 1 με βήμα -1. Η δυσκολία ανάγεται στην κατανόηση τόσο της εμφώλευσης κατά την προσπέλαση των στοιχείων του πίνακα, όσο και στο αρνητικό βήμα, καθώς η προσπέλαση γίνεται από το τέλος προς την αρχή.

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N

ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ 1 ΜΕ ΒΗΜΑ -1

ΑΝ ΠινΑ[j-1] > ΠινΑ[j] ΤΟΤΕ

T ← ΠινΑ[j-1]

ΠινΑ[j-1] ← ΠινΑ[j]

ΠινΑ[j] ← T

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Ένα άλλο σημείο στο οποίο δυσκολεύονται αρκετά οι μαθητές είναι η δομή ελέγχου που έπεται των δομών επανάληψης και ουσιαστικά αφορά τη λειτουργία της αντιμετάθεσης σε περίπτωση που δεν είναι διατεταγμένα τα στοιχεία. Τέλος, γενικότερα λανθασμένες αντιλήψεις που αφορούν του πίνακες και παρουσιάζουν αρκετοί από τους μαθητές είναι η προσπέλαση των στοιχείων του πίνακα, καθώς δυσκολεύονται να προσδιορίσουν το πλήθος των στοιχείων που πρέπει να προσπελάσουν για τους υπολογισμούς. Επιπρόσθετα, ως προς τη σχεδίαση της κατάλληλης αλγοριθμικής επίλυσης φαίνεται ότι οι μαθητές αδυνατούν να αντεπεξέλθουν με επιτυχία για το λόγο ότι δεν έχουν κατανοήσει τη λογική με την οποία επιτελούνται οι διάφορες λειτουργίες σε έναν πίνακα.

9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ

Οι μαθητές εμπλέκονται ενεργά πειραματιζόμενοι με τις αλλαγές των δομών επανάληψης. Κατά την εκτέλεση των αλγορίθμων με τη χρήση του λογισμικού «Διερμηνευτή της Γλώσσας» εκτιμάται ότι θα υπάρξουν παρανοήσεις στην ταξινόμηση με τη μέθοδο της

φυσαλίδας, ενώ το λογισμικό θα βοηθήσει τους μαθητές στο στάδιο της αξιολόγησης. Το ίδιο ισχύει και για το λογισμικό Scratch.

Επιπλέον, θα δημιουργηθεί προβληματισμός στην αναγκαιότητα διερεύνησης της μεθοδολογίας και στην εφαρμογή της σε κάθε βήμα. Η αντιμετώπιση αυτού του είδους διδακτικού θορύβου θα ξεπεραστεί με το φύλλο λύσεων-συμβουλών, αλλά και τη βοήθεια του διδάσκοντα.

10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Ξένη Βιβλιογραφία. Ενδεικτικά:

- ✓ Aho, A. V., Hopcroft, J. E., and Ullman, J. D.
- ✓ The Design and Analysis of Computer Algorithms. Addison-Wesley, 1974.
- ✓ Bell, D. The principles of sorting. The Computer Journal 1 (1958), 71–77.
- ✓ Bose, R. C., and Nelson, R. J. A sorting problem. Journal of the ACM (JACM) 9, 2 (1962), 282–296.
- ✓ Demuth, H. Electronic Data Sorting. PhD thesis, Stanford University, 1956.

Ελληνική βιβλιογραφία

- ✓ Αντωνάκος, Ν., Βογιατζής, Ν., Κατωπόδης, Ι., Πατριαρχέας, Κ. (1999). «Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον». Εκδόσεις Κορυφή Α.Ε.
- ✓ Νικολαΐδης, Σ., Καθ. Πληροφορικής. (2007). Εκπαιδευτικό Λογισμικό «Γλωσσομάθεια». Ανακτήθηκε στις 19/02/12 από τη διεύθυνση: <http://www.spinnet.gr/glossomatheia/>
- ✓ Βακάλη, Α., Γιαννόπουλος, Η., Ιωαννίδης, Ν., Κοΐλιας, Χ., Μάλαμας, Κ., Μανωλόπουλος, Ι., Πολίτης, Π. (2004). «Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον». Εκδόσεις Νέων Τεχνολογιών.
- ✓ J.T. Stasko. Algorithm Animation Research at GVU. <http://www.cc.gatech.edu/gvu/softviz/algoanim/>.
- ✓ J.T. Stasko. TANGO: A Framework and System for Algorithm Animation. IEEE Computer, 23:27–39, 1990.

11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Η οργάνωση των μαθημάτων ευνοεί την αλληλεπίδραση μεταξύ των μαθητών.

12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ

Δεν φαίνεται να έχουμε.

13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Το μάθημα θα πραγματοποιηθεί στο σχολικό εργαστήριο πληροφορικής. Οι μαθητές θα εργαστούν σε ομάδες των δύο ατόμων ανά ηλεκτρονικό υπολογιστή.

14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Η ταξινόμηση μπορεί εύκολα να προσομοιωθεί από το διδάσκοντα με πολλά παραδείγματα από την καθημερινή ζωή. Μπορούν να δοθούν παραδείγματα διαθεματικά με άλλα μαθήματα (πχ. Φυσικής, Μαθηματικών κτλ.). Ακόμη, προσομοιώσεις θα μπορούσαν να δημιουργηθούν με τα εκπαιδευτικά περιβάλλοντα Scratch, Logo και Yenka. Η δεύτερη δραστηριότητα του παρόντος διδακτικού σεναρίου χρησιμοποιεί το εκπαιδευτικό περιβάλλον Scratch για επίτευξη των διδακτικών σκοπών και στόχων.

Επιπρόσθετα, ο διδάσκων μπορεί να κάνει μία πολύ σύντομη αναφορά στους υπόλοιπους αλγόριθμους ταξινόμησης και στη συνέχεια να δείξει μια πρωτότυπη ιδέα παρουσίασης που είχε το πανεπιστήμιο της Sapientia, στην Târgu Mureș της Ρουμανίας. Δημιουργήθηκαν βιντεάκια όπου μέσα από παραδοσιακούς χορούς από την ευρύτερη περιοχή εξηγώντας τη λειτουργία των πέντε κύριων αλγορίθμων ταξινόμησης. Πιο συγκεκριμένα, ο [bubble-sort](#) παρουσιάζεται με τον παραδοσιακό χορό Csángó της Ουγγαρίας, ο [insertion-sort](#) με παραδοσιακό ρουμάνικο χορό, ο [selection-sort](#) με παραδοσιακό τσιγγάνικο χορό, ο [shell-sort](#) με τον παραδοσιακό χορό Székely της Ουγγαρίας και ο [merge-sort](#) με παραδοσιακό γερμανικό χορό (transylvanian-saxon).

15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

Τα φύλλα εργασίας περιγράφονται με πολύ λεπτομερή τρόπο.

16. ΑΞΙΟΛΟΓΗΣΗ

Όπως και στις προηγούμενες παραγράφους, ο εκπαιδευτικός μπορεί να προτείνει τεστ αυτοαξιολόγησης (κλειστού τύπου), σταυρόλεξα, κρυπτόλεξα κλπ.

Το λογισμικό HotPotatoes (και άλλα ανάλογα, όπως και ορισμένες επιλογές του Moodle) προσφέρονται για συστηματική δημιουργία ερωτημάτων κλειστού τύπου.

Κρυπτόλεξα μπορούν εύκολα να δημιουργηθούν ακόμη και σε ένα υπολογιστικό φύλλο.

17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ

Θα εξηγηθεί το θέμα των παρανοήσεων, αυθόρμητων αντιλήψεων κλπ των μαθητών. Θα δοθούν φύλλα εργασίας τα οποία παρατίθενται παρακάτω.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1

Δραστηριότητα 1

Δίνεται ο αλγόριθμος ταξινόμησης:

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N

ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1

ΑΝ ΠινA[j-1]>ΠινA[j] ΤΟΤΕ

T←ΠινA[j-1]

ΠινA[j-1] ←ΠινA[j]

ΠινA[j] ←T

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

1. Να περιγράψετε λεκτικά τη λειτουργία του αλγόριθμου
2. Να εφαρμόσετε τον αλγόριθμο για τον πίνακα [15, 12, 17, -3, 2] βήμα βήμα.

Δραστηριότητα 2

Να εφαρμόσετε τον αλγόριθμο για τον πίνακα [5, 6, 7, 8, 1] βήμα βήμα.

Δραστηριότητα 3

Να εφαρμόσετε τον αλγόριθμο για τον πίνακα [16, 11, 12, 16, 27, 29] βήμα βήμα.

Συγκρίνετε τις λύσεις σας με τις προτεινόμενες απαντήσεις, εντοπίστε τις διαφορές με τις δικές σας απαντήσεις και αξιολογήστε τα λάθη σας.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2

Δραστηριότητα 1

Α. Δοκιμάστε να δημιουργήσετε το παρακάτω τμήμα αλγορίθμου στο λογισμικό Scratch:



Β. Εφαρμόστε τον παραπάνω αλγόριθμο στον ακόλουθο πίνακα και γράψτε τα αποτελέσματα βήμα-βήμα (μετά το τέλος κάθε επανάληψης).



Δραστηριότητα 2

- A. Αν ορίζαμε στη δεύτερη γραμμή του αλγορίθμου επανάλαβε ώσπου $i=j$ τότε τι θα συνέβαινε;
- B. Αν ορίζαμε στην τρίτη γραμμή του αλγορίθμου όρισε το $j=i$ τότε τι θα συνέβαινε;

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3

Δραστηριότητα 1

Χρησιμοποιώντας τον αλγόριθμο ταξινόμησης της φυσαλίδας, σχεδιάστε στο παρακάτω κουτί τα βήματα με τα οποία η μέθοδος ταξινομεί τον πιο κάτω πίνακα.

51
26
98
87
11
44
37
27

Δραστηριότητα 2

Δημιουργήστε το παραπάνω πρόγραμμα στο περιβάλλον «Διερμηνευτής της Γλώσσας» και δείτε αν είναι σωστά τα ενδιάμεσα βήματα που υπολογίσατε.

Δραστηριότητα 3

Πιο σημείο του πιο πάνω κώδικα θα αλλάζατε για να αλλάξει η σειρά ταξινόμησης?

Δοκιμάστε τον νέο κώδικα στο περιβάλλον η «Διερμηνευτής της Γλώσσας» και δείτε τα αποτελέσματα.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4 - ΕΠΑΝΑΛΗΨΗ ΕΜΠΕΔΩΣΗ

Πρόσθετες ασκήσεις, δραστηριότητες

Θέματα εμπέδωσης, αξιολόγησης και αυτοαξιολόγησης των μαθητών

ΕΡΩΤΗΣΕΙΣ ΚΛΕΙΣΤΟΥ ΤΥΠΟΥ

- Η ταξινόμηση φυσαλίδας είναι ο πιο απλός και ταυτόχρονα ο πιο γρήγορος αλγόριθμος ταξινόμησης.
ΣΩΣΤΟ **ΛΑΘΟΣ**
- Όταν ο πίνακας είναι ήδη ταξινομημένος, ο αλγόριθμος φυσαλίδας δεν εκτελεί περιττά βήματα.
ΣΩΣΤΟ **ΛΑΘΟΣ**
- Η ταξινόμηση της φυσαλίδας μπορεί να χρησιμοποιηθεί σε πίνακα με τύπο δεδομένων χαρακτήρα
ΣΩΣΤΟ ΛΑΘΟΣ
- Οι δείκτες ενός μονοδιάστατου πίνακα μπορεί να είναι και λογικές εκφράσεις
ΣΩΣΤΟ **ΛΑΘΟΣ**
- Η ταξινόμηση της φυσαλίδας ταξινομεί τα στοιχεία ενός μονοδιάστατου πίνακα μόνο σε αύξουσα σειρά
ΣΩΣΤΟ **ΛΑΘΟΣ**
- Η ταξινόμηση φυσαλίδας είναι ο μοναδικός αλγόριθμος ταξινόμησης που υπάρχει.
ΣΩΣΤΟ **ΛΑΘΟΣ**
- Στην ταξινόμηση φυσαλίδας κατ' αύξουσα σειρά συγκρίνονται οι τιμές από δύο διαδοχικές κελιά και η μικρότερη μεταφέρεται στο προηγούμενο κελί.
ΣΩΣΤΟ ΛΑΘΟΣ
- Στην ταξινόμηση φυσαλίδας χρησιμοποιούμε την αντιμετάθεση των τιμών δύο μεταβλητών.
ΣΩΣΤΟ ΛΑΘΟΣ

18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

Ολοκληρώστε τη μελέτη των φύλλων εργασίας που προορίζονται για τους μαθητές.

Δείτε τη βιβλιογραφία.

Υπάρχουν δεκάδες animations και αναπαραστάσεις των αλγορίθμων ταξινόμησης. Οι πιο γνωστές είναι η αναπαράσταση με δημοτικούς χορούς κυρίως της Ρουμανίας, Ουγγαρίας, αλλά και άλλων χωρών. Για παράδειγμα:

<http://www.youtube.com/watch?v=lyZQPjUT5B4>

<http://www.youtube.com/watch?v=h3aMVB8k8>

<http://www.youtube.com/watch?v=lkByxva0qxc>

<http://www.youtube.com/watch?v=YKIDz1J3TSw>

<http://www.youtube.com/watch?v=ty-fpXXydm8>

<http://www.youtube.com/user/AlgoRythmics>

<http://www.youtube.com/watch?v=IUqFelc84XE>

http://www.youtube.com/watch?v=MtrcEhrt_K0

<http://www.youtube.com/watch?v=MU8tn11Nv5s>

<http://www.youtube.com/watch?v=UnK5ueUgc88>

αλλά και άλλες παρουσιάσεις:

<http://www.youtube.com/watch?v=r0gV2hQYf0>

<http://www.youtube.com/watch?v=t8g-iYGHpEA>

http://www.youtube.com/watch?v=F-7kk4IY_mQ

http://www.youtube.com/channel/HCzMRkM_BeQd8

Ενότητα Σεναρίου 6

Εισαγωγή στη γλώσσα Logo

ΕΙΣΑΓΩΓΗ ΣΤΗ Logo

1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Εισαγωγή στον συναρτησιακό προγραμματισμό και τα Logo-like περιβάλλοντα: το περιβάλλον LogoMSW

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 2 έως 4 διδακτικές ώρες, ανάλογα με το επίπεδο των μαθητών.

3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Πρόκειται για διδακτικό σενάριο μιας εισαγωγής και αρχικών δραστηριοτήτων στη γλώσσα Logo και επομένως μπορεί να διδαχθεί και στην Πρωτοβάθμια (με κατάλληλες προσαρμογές) και στο Γυμνάσιο. Το αναλυτικό πρόγραμμα του Δημοτικού συνιστά περισσότερο περιβάλλοντα οπτικού προγραμματισμού (όπως το Scratch ή το Game Maker), αλλά ενδεχομένως η πολυπλοκότητα ενός περιβάλλοντος Logo να επιτρέπει δραστηριότητες χωρίς μεγάλη διδακτική επιβάρυνση. Στις Δραστηριότητες και στα Παραρτήματα περιγράφονται πολλά περιβάλλοντα και διδακτικές δραστηριότητες που είναι κατάλληλες για χρήση στην Πρωτοβάθμια Εκπαίδευση (ακόμη και στην προσχολική ηλικία).

4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Οι βασικοί στόχοι του σεναρίου που περιγράφεται παρακάτω είναι οι εξής:

- (1) εισαγωγή σε ένα προγραμματιστικό μικρόκοσμο
- (2) εισαγωγή στις γλώσσες τύπου Logo (Logo-like)

5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το σενάριο που περιγράφεται παρακάτω περιλαμβάνει 5 φύλλα εργασίας για τους μαθητές, με αυξανόμενη δυσκολία.

Τα δυο πρώτα είναι εισαγωγικά στο περιβάλλον και τη γλώσσα MSWLogo: βασικές εντολές κίνησης της χελώνας και δημιουργία και διαχείριση διαδικασιών.

Το τρίτο φύλλο εργασίας δίνει έμφαση στις παραμέτρους. Τα φύλλα εργασίας 4 και 5 είναι

πιο προηγμένα, καθώς αναφέρονται σε δυνατότητες της γλώσσας που δεν σχετίζονται κατ' ανάγκη με το γραφικό μέρος, καθώς και δραστηριότητες για μια εισαγωγή στην έννοια της αναδρομής.

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Υπάρχουν δεκάδες παραλλαγές της γλώσσας Logo, ένας πολύ μεγάλος αριθμός «κλώνων» (ελεύθερων αλλά και εμπορικών προϊόντων) και αναρίθμητα περιβάλλοντα εμπνευσμένα από τις βασικά χαρακτηριστικά της.

Στον πίνακα που ακολουθεί περιλαμβάνονται μερικές εκδόσεις της Logo και Logo-like περιβαλλόντων. Ο πίνακα είναι ενδεικτικός.

Μικρόκοσμος	Ιδιαίτερα Χαρακτηριστικά
<i>Boxer</i> (diSessa 1986)	Βασίστηκε στη Logo, αλλά δόθηκε βαρύτητα κατά τη σχεδίασή του στην αποφυγή των αδυναμιών της. Χρησιμοποιείται για τη δημιουργία προσομοιώσεων.
<i>LOGOmotion</i>	Παρέχει τη δυνατότητα ενσωμάτωσης διαδικασιών δυναμικής προσομοίωσης (animation) σε ένα πρόγραμμα Logo.
<i>MSW Logo</i>	Ίσως η πλέον διαδεδομένη έκδοση της Logo. Διατίθεται δωρεάν. http://www.softronix.com/logo.html
<i>StarLogo</i> (Resnick 1994)	Περιβάλλον που αναπτύχθηκε στο MIT και χρησιμοποιείται κυρίως για τη δημιουργία μικρόκοσμων και των προγραμματισμό προσομοιώσεων, με σκοπό τη διερεύνηση της εξέλιξής τους. Επιτρέπει την αλληλεπίδραση του περιβάλλοντος και των πρωταγωνιστών του μέσω παράλληλου προγραμματισμού.
<i>NetLogo</i>	Αποτελεί εξέλιξη της StarLogo και αναπτύχθηκε κυρίως για τη δημιουργία προσομοιώσεων. http://ccl.northwestern.edu/netlogo/
<i>MultiLogo</i>	Παραλλαγή της Logo για τη διδασκαλία των αρχών του παράλληλου προγραμματισμού.
<i>SuperLogo</i>	Παραλλαγή της Logo με ιδιαίτερες γραφικές δυνατότητες, η οποία απευθύνεται σε παιδιά μικρής ηλικίας. Ενσωματώνει έναν εκδότη σύνταξης και τεχνικές άμεσης διαχείρισης. http://www.logo.com/download/index.html
<i>Χελωνόκοσμος</i> και <i>EasyLogo</i>	Εκδόσεις Logo – παρουσιάζονται εκτενέστερα σε κατοπινή ενότητα
<i>TNG Logo</i>	Αποτελεί εξέλιξη της StarLogo. Αναπτύχθηκε κυρίως για τη δημιουργία ηλεκτρονικών παιχνιδιών στο MIT.
<i>LadyBug, Kara, Light-Bot</i>	Περιβάλλοντα που έχουν ομοιότητα με τη Logo αλλά δεν απαιτούν γραπτές εντολές

<i>Lego Mindstorms, Karel, ObjectKAREL, Robomind</i>	Ρομποτικές κατασκευές (πραγματικές ή προσομοιούμενες στην οθόνη) σε στενή σχέση με τα βασικά χαρακτηριστικά της Logo
--	--

Ορισμένες από τις παραλλαγές της Logo περιγράφονται σε μεθεπόμενες παραγράφους του παρόντος σεναρίου, ενώ άλλες αποτελούν ξεχωριστές σενάρια ή δραστηριότητες κατηγοριοποιημένες σε άλλα σενάρια. Η συμπερίληψή τους στο παρόν θα δημιουργούσε μια δυσανάλογα μεγάλη ενότητα.

Τα συνηθισμένα περιβάλλοντα της Logo δεν είναι προσανατολισμένα αποκλειστικά σε γραφικές εφαρμογές, αλλά δίνουν τη δυνατότητα και υλοποίησης αλγορίθμων που επεξεργάζονται, αριθμούς, συμβολοσειρές και δομές δεδομένων. Ωστόσο, η μεγάλη πλειονότητα των εκπαιδευτικών σεναρίων με τα περιβάλλοντα αυτά κάνουν χρήση των γραφικών δυνατοτήτων της Logo.

Τα συνήθη περιβάλλοντα Logo στηρίζονται περισσότερο στο παράδειγμα του συναρτησιακού προγραμματισμού, με την έννοια ότι κάθε πρόγραμμα αποτελεί ένα είδος συνάρτησης η οποία λαμβάνει ως όρισμα τα δεδομένα και εξάγει το αποτέλεσμα της εφαρμογής της συνάρτησης.

Αυτό εκ πρώτης όψεως μπορεί να μην είναι σημαντικό, αλλά από διδακτική άποψη σημαίνει μια τελείως διαφορετική προσέγγιση των προβλημάτων.

Έτσι δεδομένης μιας συνάρτησης $F(X,Y)$, οποία δέχεται ως όρισμα δυο αριθμούς και επιστρέφει (ως έξοδο, ως αποτέλεσμα) το μικρότερο από τους δυο, το πρόβλημα του υπολογισμού του ελαχίστου από ένα σύνολο αριθμών αντιμετωπίζεται διαφορετικά σε δυο διαφορετικά είδη προγραμματισμού:

(A) στο μοντέλο του διαδικαστικού προγραμματισμού, προσδιορίζεται κάποιος αριθμός ως ελάχιστος και στη συνέχεια ο ελάχιστος αυτός συγκρίνεται με τα υπόλοιπα στοιχεία του συνόλου, αντικαθιστάμενος κάθε φορά από κάποιο στοιχείο του συνόλου εάν είναι μικρότερο του. Η διαδικασία περιγράφεται με αναλυτικό τρόπο.

(B) στο μοντέλο του συναρτησιακού προγραμματισμού, τα δεδομένα θεωρούνται ότι αποτελούν μια λίστα (μια σειριακή δομή στην οποία είναι γνωστό το πρώτο και το τελευταίο στοιχείο και είναι προσπελάσιμο το επόμενο και το προηγούμενο στοιχείο κάθε γνωστού στοιχείου). Ο ελάχιστος τότε προσδιορίζεται ως εξής: αν η λίστα έχει ένα μοναδικό στοιχείο X , είναι ο X , αλλιώς είναι ο $F(X,Y)$ όπου X είναι το πρώτο στοιχείο της λίστας και Y ο ελάχιστος της λίστας που απομένει αν από την αρχική αφαιρέσω το X .

Το συναρτησιακό μοντέλο κάνει λοιπόν εκτεταμένη χρήση της αναδρομής.

Το μοντέλο αυτό δέχτηκε πολύ ισχυρή κριτική για 2 λόγους:

(α) είναι αναποτελεσματικό με όρους υπολογιστικού κόστους: ο υπολογιστικός χρόνος μπορεί να είναι πρακτικά πάρα πολύ μεγάλος, καθώς το σύστημα δε μπορεί να είναι βελτιστοποιημένο για κάθε πρόβλημα και επιπλέον στις αναδρομικές συναρτήσεις χρησιμοποιείται πολλή εσωτερική μνήμη.

(β) εκτός από τετριμμένες περιπτώσεις, τα σχετικά προγράμματα (οι συναρτήσεις δηλαδή) είναι εξαιρετικά πολύπλοκα και ως εκ τούτου πολύ δύσκολα ελέγχονται για την ορθότητα τους, την αξιοπιστία και την ακρίβεια τους. Οι επικριτές αυτού του είδους προγραμματισμού ισχυρίζονται ότι, κατά κάποιο τρόπο, «κρύβει» τις υπολογιστικές και αλγοριθμικές δυσκολίες από το χρήστη.

Ωστόσο, το μοντέλο το συναρτησιακού προγραμματισμού προσφέρει λύσεις σε πολλές περιπτώσεις απρόσμενα απλές και βοηθά στην κατανόηση πολλών διαδικαστικών αλγορίθμων.

Τυπικό παράδειγμα αποτελεί ο πύργος του Ανόι, του οποίου η λύση (σχετικός κώδικας) για οιονδήποτε αριθμό δίσκων είναι πολύ μικρή:

```
to hannoi :numberdisks :a :b :c
  ifelse :numberdisks = 1 [(print "ΕΝΑΣ "ΔΙΣΚΟΣ "ΑΠΟ "ΤΟ :a "ΣΤΟ
    :b)]
  [hannoi :numberdisks - 1 :a :c :b
    (print "ΕΝΑΣ "ΔΙΣΚΟΣ "ΑΠΟ "ΤΟ :a "ΣΤΟ :b)
    hannoi :numberdisks - 1 :c :b :a]
end
```

(προσοχή! στο περιβάλλον της MSW Logo πρέπει οι εντολές να είναι σε μια σειρά)

Η αναδρομή μπορεί εξίσου εύκολα, στο περιβάλλον της Logo, να εφαρμοστεί και σε γεωμετρικά σχήματα (τυπική περίπτωση τα fractals).

Βέβαια η χρήση της απαιτεί μια εξοικείωση των χρηστών (μαθητών) με ορισμένες έννοιες προγραμματισμού και αναμφισβήτητα έχει ένα μεγάλο βαθμό διανοητικής πολυπλοκότητας: η εφαρμογή της σημαίνει την «εκμετάλλευση» μερικών λύσεων, πριν καν αυτές προσδιοριστούν. Ένα ανάλογο αποτελεί στα Μαθηματικά η διδασκαλία και η χρήση της λεγόμενης επαγωγικής μεθόδου για αποδείξεις προτάσεων που εξαρτώνται από κάποιο φυσικό αριθμό N και επιχειρούμε να αποδείξουμε ότι ισχύουν για κάθε N (για παράδειγμα μαθηματικοί τύποι). Απαιτείται μεγάλη αφαιρετική ικανότητα για τη χρήση τους.

Έτσι η διδασκαλία της αναδρομής παραμένει ένα θέμα που παραμένει ανοιχτό. Πάντως η Logo, ως περιβάλλον προγραμματισμού μέσω της μελέτης σχημάτων (γεωμετρικών και άλλων), δεν απαιτεί τη χρήση αναδρομικών μεθόδων παρά μόνο σε ένα πολύ προηγμένο

επίπεδο.

Πάντως το περιβάλλον της Logo δεν είναι αυστηρά δομημένο και επιτρέπει εκχωρήσεις, τη χρήση πινάκων και άλλων στοιχείων από τον τυπικό διαδικαστικό προγραμματισμό.

Θα πρέπει τέλος να αναφερθεί ότι η Logo και η διδασκαλία της συνδέονται πολύ στενά με τα Μαθηματικά – μάλιστα ο επινοητής της Logo S. Papert αναφέρει ότι οραματίστηκε τη Logo ως ένα είδος «Μαθηματοχώρας» στην οποία οι μαθητές θα μαθαίνουν Μαθηματικά, με ένα φυσικό και διασκεδαστικό τρόπο, λύνοντας προβλήματα «εκεί», όπως ο καλύτερος τρόπος για να μάθει κανείς Αγγλικά είναι να πάει και να ζήσει στην Αγγλία, για να μάθει Γερμανικά στη Γερμανία κ.ο.κ.

Η Logo αποτελεί το προσφιλέστερο προγραμματιστικό περιβάλλον για τους Καθηγητές Μαθηματικών και στα σενάρια των Μαθηματικών που διδάσκονται στα ΚΣΕ περιλαμβάνονται και σενάρια με Logo.

Θα πρέπει τέλος να τονιστεί ότι η Logo είναι μια πολύ ευέλικτη γλώσσα: μπορεί να χρησιμοποιηθεί τόσο στην πρωτοβάθμια εκπαίδευση όσο και στην δευτεροβάθμια και επιτρέπει την επίλυση προβλημάτων μεγάλης πολυπλοκότητας. Αυτό την κάνει πολύ κατάλληλη για την εκπαίδευση, αφού επιτρέπει την επίλυση και διερεύνηση προβλημάτων πολλών ειδών και επιπέδων δυσκολίας. Βέβαια είναι αρκετά απομακρυσμένη από τα τυπικά μοντέλα διαδικαστικού προγραμματισμού – που χρησιμοποιούνται στα επαγγελματικά και ακαδημαϊκά περιβάλλοντα – αλλά είναι κατάλληλη για την εισαγωγή ορισμένων εννοιών των αλγορίθμων και του προγραμματισμού³.

7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Το διδακτικό σενάριο προβλέπεται να υλοποιηθεί στο εργαστήριο Πληροφορικής.

Το προτεινόμενο περιβάλλον MSW δεν είναι το μόνο, αλλά ένα από τα πιο πλήρη και διαδεδομένα παγκοσμίως.

Η διεπαφή του δεν είναι πλήρως βασισμένη σε εικονίδια, όπως άλλες εκδόσεις της Logo, αλλά αυτό δεν είναι κατ' ανάγκη αρνητικό: η διατύπωση των σχέσεων με κειμενικό τρόπο (αντί για διαχείριση εικονιδίων αποκλειστικά) προσφέρει μεγαλύτερη ευελιξία στην έκφραση των λύσεων, ακριβέστερα αποτελεί μέρος, τμήμα της γνώσης που επιθυμούμε να αποκτήσουν

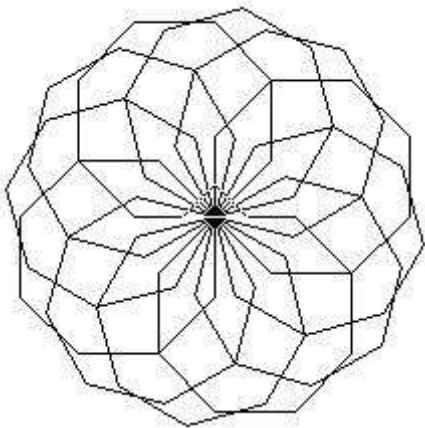
³ Αναφέρω μόνο για την καταγραφή του γεγονότος ότι στη δεκαετία του 1980, στο Πανεπιστήμιο της Grenoble (Γαλλία) ένα μάθημα του δευτέρου έτους στο Τμήμα της Πληροφορικής χρησιμοποιούσε επί πολλά έτη ως αποκλειστικό περιβάλλον μια έκδοση της Logo, πολύ όμοια με την MSWLogo. Παρόμοιες διδασκαλίες γίνονταν και σε άλλα πανεπιστήμια όπως στο Paris VI ακόμη και στο τρίτο έτος των σπουδών Πληροφορικής. Ένα τυπικό project στο μάθημα ήταν η κατασκευή στη Logo ενός mini interpreter της BASIC.

οι μαθητές.

Στα πολλά περιβάλλοντα Logo που περιγράφονται σε κατοπινές ενότητες υπάρχουν διεπαφές που είναι καθαρά γραφιστικές και επομένως έχουν τα αντίστοιχα πλεονεκτήματα και μειονεκτήματα. Οι προτεινόμενες δραστηριότητες και το σενάριο της παρούσας βασίζονται σε ένα περιβάλλον που περιλαμβάνει και εντολές κειμενικού τύπου.

Με ένα γενικό τρόπο, αν και τα περιβάλλοντα Logo δίνουν πολλές δυνατότητες για επεξεργασία των συνήθων αντικειμένων (αριθμών, δομών δεδομένων, συμβολοσειρών), ωστόσο, η μεγάλη πλειονότητα των διδασκαλιών στηρίζεται στη χρήση των γραφικών δυνατοτήτων της Logo. Στα περιβάλλοντα Logo, η σχεδίαση ακόμη και πολύπλοκων σχημάτων (ορισμένης κατηγορίας) είναι σχετικά απλή.

Για παράδειγμα, ένα σύνολο από ελάχιστες απλές εντολές επιτρέπει τη σχεδίαση σχημάτων με σύνθετη δομή:

<pre>to polla :A :B :C repeat :C [polygono :A :B rt 360 / :C] end to polygono :A :B repeat :B [fd :A rt 360 / :B] end</pre>	
---	---

Η συμβολή του συγκεκριμένου ψηφιακού περιβάλλοντος στην κατανόηση βασικών εννοιών και τεχνικών στους αλγορίθμους και στον προγραμματισμό είναι πολύ σημαντική.

Το μειονέκτημα αυτών των περιβαλλόντων είναι ότι η «μετάβαση» σε πιο κλασικά προγραμματιστικά στυλ (δηλαδή η διδασκαλία του διαδικαστικού ή του αντικειμενοστραφούς προγραμματισμού) μπορεί να αποδειχθεί πιο περίπλοκη από διδακτική άποψη, αν οι μαθητές και οι μαθήτριες έχουν συνηθίσει σε ένα συγκεκριμένο προγραμματιστικό στυλ.

Ως μειονέκτημα επίσης θα πρέπει να σημειωθεί το γεγονός ότι κατά κανόνα τα μηνύματα λάθους είναι ελάχιστα πληροφοριακά για τον αρχάριο χρήστη. Έτσι, ένα λάθος πληκτρολόγησης (για παράδειγμα η παράλειψη του συμβόλου «:» πριν από μια παράμετρο, προκαλεί ένα τελείως δυσνόητο για το χρήστη μήνυμα.

8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ / ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

Οι συνηθέστερες δυσκολίες ενός αρχάριου προγραμματιστή εντοπίζονται κυρίως στην ιδιαίτερη «γεωμετρία» της Logo.

Ο μαθητής ή η μαθήτρια, πρέπει να συνηθίσει στην ιδέα ότι η γεωμετρία της χελώνας είναι μάλλον «τοπική», δηλαδή, λαμβάνει υπόψη της τον προσανατολισμό τη χελώνας και όχι του χρήστη. Ο χρήστης οφείλει να οδηγήσει τη χελώνα, για παράδειγμα να αλλάξει κατεύθυνση κατά μια δεδομένη γωνία, ανάλογα με τον τρόπο που η χελώνα «βλέπει» το χώρο και όχι με τον τρόπο που ο χρήστης βλέπει την οθόνη.

9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΥΡΥΒΟΣ

Τα περιβάλλοντα Logo και ιδιαίτερα αυτό που χρησιμοποιείται στο σενάριο, χαρακτηρίζονται από μερικά σημαντικά στοιχεία τα οποία τους προσδίδουν ιδιαίτερα σημαντικές διδακτικές δυνατότητες.

Τα περιβάλλοντα αυτά κατά κανόνα «προσομοιώνουν» φυσικές καταστάσεις (ή καταστάσεις που έχουν προφανή χαρακτηριστικά), στις οποίες μια οντότητα (συνήθως ρομπότ, κάποιο ζώο όπως μια χελώνα, μια πασχαλίτσα κ.ά.) εκτελεί τις εντολές του προγραμματιστή-χρήστη του περιβάλλοντος. Έτσι, ο μαθητής ή η μαθήτρια που χρησιμοποιεί το περιβάλλον εξοικειώνεται πολύ γρήγορα με αυτό, ενώ ταυτόχρονα το αποτέλεσμα της εκτέλεσης ενός προγράμματος είναι αμέσως ορατό και επομένως ο έλεγχος της ορθότητας του είναι προφανής. Ένα μεγάλο μέρος λοιπόν διαπραγμάτευσης του διδακτικού συμβολαίου που σχετίζεται με το output των αποτελεσμάτων (ορθότητα και εμφάνιση των αποτελεσμάτων) προκύπτει με έναν άμεσο τρόπο, χάρη στα χαρακτηριστικά του περιβάλλοντος. Για παράδειγμα, όταν η «χελώνα» (στις τυπικές εκδόσεις της Logo) δεχθεί εντολές να μετακινηθεί σε μια μεγάλη απόσταση, μπορεί να «βγει» εκτός οθόνης – πράγμα που θεωρείται «φυσικό» - και μάλιστα να εμφανιστεί (και να συνεχίσει την κίνησή της) από την αντικρινή πλευρά της οθόνης. Αντίστοιχα, όταν σε ένα πιο τυπικό περιβάλλον προγραμματισμού όπως, για παράδειγμα, μιας BASIC, ο χρήστης δίνει εντολή για εκτύπωση ενός μηνύματος 1000 φορές - όπως στο ακόλουθο παράδειγμα σε Small Basic (<http://smallbasic.com/>):

```
for i=1 To 100  
TextWindow.WriteLine("BBBB")  
EndFor
```

το αποτέλεσμα μπορεί να είναι απροσδόκητο για το χρήστη, ο οποίος θα πρέπει να έχει

πλήρη κατανόηση του τρόπου με τον οποίο εμφανίζονται τα αποτελέσματα στην οθόνη.

Επίσης σε ένα τέτοιο περιβάλλον, η διαχείριση της οντότητας που «μετακινείται» σημαίνει ότι μπορεί, σε κάποιο βαθμό, να μην υπάρχουν συγκεκριμένα δεδομένα τα οποία θα πρέπει να εισαχθούν από το πληκτρολόγιο από το χρήστη. Η διαχείριση λοιπόν της εισόδου και εξόδου δεδομένων και πληροφοριών είναι πολύ απλοποιημένη. Η ύπαρξη τέλος μιας «οντότητας» την οποία κατευθύνει ο χρήστης, προσδίδει έναν παιγνιώδη χαρακτήρα στο περιβάλλον και κατ' επέκταση στις διδακτικές δραστηριότητες σε αυτό.

Λόγω της απλότητας του περιβάλλοντος, δεν αναμένεται ιδιαίτερος διδακτικός θόρυβος.

10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Εκτός από τη συνήθη βιβλιογραφία για την εισαγωγή στον προγραμματισμό, υπάρχει μια πλουσιότατη βιβλιογραφία και δικτυογραφία ειδικά για τις γλώσσες Logo.

Αναφέρονται για λόγους ιστορικότητας και μόνο τα βιβλία του δημιουργού της Seymour Papert, (μαθητή του J. Piaget), ερευνητή του MIT

Το *Perceptrons* (1969), το διάσημο *Mindstorms* (δεκαετία 1970, εκδομένο και στα ελληνικά με τον τίτλο «Νοητικές θύελλες») που περιέχει ολοκληρωμένη την άποψή του για τη Logo και το *The Children Machine*.

Βιβλία του Mitchel Resnick, του Andrea di Sessa, του Brian Harvey (δημιουργού της MSWLogo και συγγραφέα ενός τρίτομου έργου για μια εισαγωγή στην Επιστήμη των Υπολογιστών με τη Logo) και άλλων ερευνητών περιέχουν επίσης πολύτιμες ιδέες για τη Logo.

Από τα πολλά Διεθνή Συνέδρια για τη Logo ξεχωρίζει η σειρά των Συνεδρίων Eurologo, τα οποία τα τελευταία χρόνια μετονομάστηκαν σε *Constructionism*⁴ (όρο που επινόησε ο Seymour Papert και είναι εμπνευσμένος από τον *Constructivism* του J. Piaget).

11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Οι προτεινόμενες δραστηριότητες για τους μαθητές είναι τυπικά κονστρουκτιβιστικές: στους μαθητές προτείνονται μια σειρά από προβλήματα τα οποία καλούνται να επιλύσουν στο περιβάλλον της Logo. Οι μαθητές, μπορούν, εργαζόμενοι σε ομάδες, να διερευνήσουν τις δυνατότητες του περιβάλλοντος και να τις συνδυάσουν για την επίλυση των προβλημάτων. Η επίλυση των προβλημάτων που προτείνονται εξάλλου, καθιστά εμφανή τη λειτουργία και

⁴ Το Συνέδριο *Constructionism 2012* έγινε στην Αθήνα.

αξία των χρησιμοποιούμενων μεθόδων, εννοιών, τεχνικών κλπ. Υπενθυμίζεται ότι στο περιβάλλον της Logo η ανάδραση από το σύστημα, το σχήμα που δημιουργείται (ή το μήνυμα λάθους) είναι απολύτως ενδεικτικό τα ορθότητας της λύσης.

Η επιλογή των καταλληλότερων μεθόδων για την επίλυση των προβλημάτων μπορεί να προκύψει μέσα από κοινωνιογνωστικές συγκρούσεις, εφόσον οι μαθητές εργάζονται σε ομάδες.

12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ

Δεν φαίνεται να έχουμε.

13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Προτείνεται οι μαθητές να οργανωθούν σε ομάδες των 2-3 (το πολύ) ατόμων.

Η οργάνωση αυτή ευνοεί την αλληλεπίδραση μεταξύ των μαθητών οι οποίοι αντιμετωπίζουν την επίλυση των προτεινομένων προβλημάτων ως ένα κοινό project, ως μια κοινή προσπάθεια και όχι ως ατομικές δράσεις.

Θα πρέπει πάντως να δοθεί προσοχή στο γεγονός ότι η κατανόηση της λειτουργίας (της σημασίας δηλαδή) των εντολών της Logo είναι πολύ σημαντική για την περαιτέρω πρόδοό τους.

14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Η ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Διασυνδέσεις και επεκτάσεις των βασικών εννοιών περιγράφονται στα παραρτήματα και τις λοιπές δραστηριότητες.

15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

Τα φύλλα εργασίας δεν απαιτούν περαιτέρω ανάλυση, καθώς περιγράφουν τις εισαγωγικές εντολές και έννοιες από το περιβάλλον της Logo.

Για το πρώτο φύλλο εργασίας

Οι πρώτες εντολές και υποδείξεις έχουν ως σκοπό την εξοικείωση των μαθητών με το περιβάλλον της Logo. Οι μαθητές, συνεργαζόμενοι, θα εισαχθούν στο περιβάλλον της MSW Logo και θα εντοπίσουν τα μέρη του περιβάλλοντος στο οποίο θα εργαστούν.

Οι μαθητές, ενδεχομένως έχουν από άλλα μαθήματα μια εξοικείωση με τα προγραμματιστικά περιβάλλοντα εργασίας. Είναι επίσης ενδεχόμενο να έχουν ήδη γνωρίσει άλλα «είδη» προγραμματισμού. Ωστόσο, αυτό δεν είναι βέβαιο και έτσι ο εκπαιδευτικός θα πρέπει να

εισαγάγει τους μαθητές σε μερικά βασικά χαρακτηριστικά των περιβάλλοντων προγραμματισμού γενικά και της Logo ειδικότερα: πρέπει να εξηγήσει με απλό τρόπο ότι πρόκειται για ένα περιβάλλον, στο οποίο το σύστημα (η Logo) κατανοεί μόνο συγκεκριμένες εκφράσεις και ανταποκρίνεται με συγκεκριμένο τρόπο, πάντοτε τον ίδιο, στις εντολές του χρήστη (παιδιών).

Το γεγονός ότι πρόκειται, ενδεχομένως, για την πρώτη γλώσσα προγραμματισμού με την οποία θα έλθουν σε επαφή τα παιδιά μπορεί να σημαίνει ότι τα παιδιά είναι σχετικά μικρής ηλικίας: ο εκπαιδευτικός θα πρέπει να αντιμετωπίζει με προσοχή το γεγονός ότι ορισμένες έννοιες μπορεί να είναι δύσκολες για το επίπεδό τους.

Σε κάθε περίπτωση πάντως, ο εκπαιδευτικός πρέπει να είναι προετοιμασμένος ώστε να αντιμετωπίσει, μερικές φορές, μια στάση απαξίωσης από τα παιδιά, τα οποία βρίσκουν απογοητευτικό το γεγονός ότι πρέπει να καταβάλουν μεγάλο (διανοητικό) κόπο και προσπάθεια για να δημιουργήσουν ένα «φτωχό» αποτέλεσμα – όπως ένα απλό σχήμα, το οποίο μπορούν πολύ εύκολα να ζωγραφίσουν σε ένα άλλο περιβάλλον.

Στα πρώτα αυτά μαθήματα, δεν αναμένεται να υπάρξουν σημαντικές δυσκολίες κατανόησης των εννοιών.

Το περιβάλλον επίσης της Logo, δεν είναι δύσκολο να κατανοηθεί.

Θα πρέπει πάντως να δοθεί μια ιδιαίτερη προσοχή στα μηνύματα λάθους: πολύ συχνά (εκτός του ότι είναι στην Αγγλική) είναι εξαιρετικά δυσνόητα και δεν αντιστοιχούν στο πραγματικό πρόβλημα. Ο εκπαιδευτικός πρέπει να μεριμνήσει ιδιαίτερα για την αντιμετώπιση των μηνυμάτων. Καθώς είναι μερικές φορές πολύπλοκη η πλήρης εξήγηση τους, μπορεί απλά να εξηγήσει στους μαθητές τα σημεία τα οποία προκαλούν συνήθως μηνύματα λάθους (π.χ. εντολές με λανθασμένη γραφή κλπ).

Αντίθετα, είναι ιδιαίτερος ενδιαφέροντα, από διδακτική άποψη, τα σημεία στα οποία τα αποτελέσματα στην οθόνη είναι διαφορετικά από τα αναμενόμενα.

Να σημειωθεί ότι δεν υπάρχουν εντολές που σχετίζονται με γραφιστικά στοιχεία όπως το πάχος της γραμμής, το χρώμα της «γραφίδας», τη δυνατότητα «βαφής» μιας κλειστής περιοχής κλπ. Οι εκπαιδευτικοί μπορούν να προσθέσουν εντολές αυτού του είδους, ανάλογα με την τάξη στην οποία απευθύνονται.

Για το δεύτερο φύλλο εργασίας

Το πρόβλημα της λανθασμένης, κατά κανόνα, κατανόησης του τρόπου με τον οποίο η χελώνα σχηματίζει το τετράγωνο είναι τυπικό: οι μαθητές, ερμηνεύουν, όπως είναι αναμενόμενο, τη γωνία των 90 μοιρών ως εσωτερική γωνία του τετραγώνου, αντί για το ορθό – εξωτερική γωνία.

Είναι μια καλή ευκαιρία για τον εκπαιδευτικό να αναφερθεί στη Γεωμετρία της χελώνας και

στον τρόπο με τον οποίο λειτουργεί η χελώνα – εκτελώντας επακριβώς τις εντολές που της δίνουμε.

Σημαντικό στοιχείο είναι επίσης η κατασκευή των πολυγώνων – όπου οι μαθητές πρέπει να έχουν το κατάλληλο υπόβαθρο ώστε να υπολογίσουν τις εσωτερικές και εξωτερικές τους γωνίες.

Το «άλμα» προς τον κύκλο είναι ένα σημείο που θέλει προσοχή και θα πρέπει να τονιστεί ότι είναι ένα σχήμα που «μοιάζει» με κύκλο, αλλά είναι ένα πολύγωνο με πολλές πλευρές που έχουν μικρό μήκος (αρκεί να αυξηθεί το μήκος τους για να καταστεί εμφανές το ότι δεν πρόκειται για κύκλο, αλλά για πολύγωνο).

Σε συνδυασμό με το επόμενο φύλλο, και εφόσον οι μαθητές έχουν το κατάλληλο υπόβαθρο, η δημιουργία των πολυγώνων θα μπορούσε να παραμετροποιηθεί πλήρως έτσι ώστε μια εντολή του είδους `polygono 100 9`, να είναι αρκετή για τη δημιουργία ενός κανονικού εννεαγώνου.

Για το τρίτο φύλλο εργασίας

Οι διαδικασίες και μάλιστα με παραμέτρους αποτελεί ένα πολύ ισχυρό «εργαλείο» επίλυσης προβλημάτων είτε προγραμματιστικών, είτε γενικότερων. Αντιστοιχεί, σε κάποιο βαθμό, με την ιδέα μιας διανοητικής στρατηγικής που μας επιτρέπει να κάνουμε σημαντική οικονομία δυνάμεων: κατά την επίλυση ενός προβλήματος δε μας απασχολούν τα υποπροβλήματα των οποίων τη λύση γνωρίζουμε.

Αυτό μια καθημερινή πρακτική στην επίλυση προβλημάτων – την οποία, ωστόσο, εφαρμόζουμε χωρίς να τη συνειδητοποιούμε.

Πολύ σημαντική είναι επίσης η ονοματοδοσία των διαδικασιών: μια ονοματισμένη διαδικασία εκτελεί ένα «έργο» και αρκεί η «κλήση» της στο κατάλληλο σημείο του προγράμματος, ώστε η Logo να εκτελέσει το έργο αυτό.

Τέλος οι παράμετροι αποτελούν ένα κεφάλαιο με ιδιαίτερη σημασία καθώς επιτρέπουν τις γενικεύσεις: μια «καρέκλα» έχει μια γενική περιγραφή και το συγκεκριμένο σχήμα που βλέπω στην οθόνη του HY δεν είναι παρά ένα συγκεκριμένο στιγμιότυπο της γενικευμένης περιγραφής.

Για το τέταρτο και το πέμπτο φύλλο εργασίας

Τα φύλλα αυτά (κυρίως το πέμπτο) περιλαμβάνουν δραστηριότητες που είναι ποηγμένες, απαιτούν δηλαδή μεγάλη διανοητική προσπάθεια για τη χρήση τους (κυρίως το 5^ο φύλλο). Το παραπάνω φύλλο καλύπτει πολύ περισσότερα αντικείμενα από όσα μπορούν να διδαχθούν στο πλαίσιο μιας ωριαίας διδασκαλίας – ο εκπαιδευτικός θα πρέπει να προσαρμόσει το σενάριο στις κατά περίπτωση διδακτικές ανάγκες.

Η Logo προσφέρεται κατ' εξοχή για τη μελέτη συναρτήσεων και ειδικότερα της αναδρομής.

Βέβαια, η αναδρομή και ο συναρτησιακός προγραμματισμός παρουσιάζουν μια πολυπλοκότητα η οποία εύκολα μπορεί να γίνει πολύ σημαντική και άρα οι αντίστοιχες έννοιες και τα προβλήματα να είναι δύσκολα για τους διδασκόμενους. Ωστόσο η επιλογή των κατάλληλων προβλημάτων μπορεί να διευκολύνει την υπέρβαση των προβλημάτων αυτών.

Θα δούμε με ποιο τρόπο μπορούμε να υλοποιούμε μια «στρατηγική» επίλυσης προβλημάτων χρησιμοποιώντας τα χαρακτηριστικά της Logo.

Η στρατηγική αυτή, μπορεί, με ένα γενικό τρόπο, να περιγραφεί ως εξής:

Αν θέλω να προσδιορίσω ένα στοιχείο ενός συνόλου με μια χαρακτηριστική ιδιότητα ή να εφαρμόσω μια διαδικασία σε ένα σύνολο αριθμών, τότε το πρόβλημα μου μπορεί να λυθεί πολύ εύκολα ως εξής: αν είναι τετριμμένη περίπτωση (για παράδειγμα το σύνολο έχει ένα και μοναδικό στοιχείο) τότε η λύση είναι προφανής. Αν όμως το σύνολο δεν έχει ένα και μόνο στοιχείο, τότε το στοιχείο που ψάχνω θα είναι στα υπόλοιπα και επομένως εφαρμόζω εκ νέου (αναδρομή) τη διαδικασία που μόλις παραπάνω περιγράψαμε.

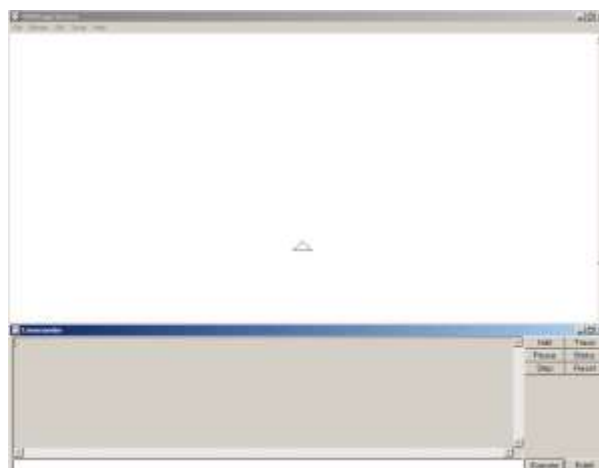
Τα σχήματα που υλοποιούνται με αναδρομή (όπως η νιφάδα του Koch) μπορεί να είναι ελκυστικά για τους μαθητές.

16. ΑΞΙΟΛΟΓΗΣΗ

Δεν προτείνονται ιδιαίτερες ασκήσεις.

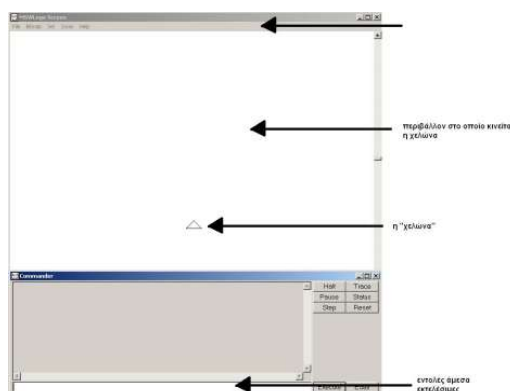
17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ

Στην πρώτη επαφή των μαθητών με τη Logo, μπορεί φυσικά να χρησιμοποιηθεί οιοδήποτε περιβάλλον Logo. Στα παρακάτω χρησιμοποιείται η MSW Logo, αλλά στη συνέχεια υπάρχει μια ενότητα που αναφέρεται στο «Χελωνόκοσμος» (και μια ακόμη στην EasyLogo). Η MSWLogo είναι άμεσα διαθέσιμη στο site από το οποίο μπορούμε να αποκτήσουμε το περιβάλλον της, αξιόπιστη, αρκετά εμπλουτισμένη (οπωσδήποτε αρκετά για το επίπεδο και το είδος διδασκαλίας για το οποίο ενδιαφερόμαστε).



Το περιβάλλον της MSWLogo

Το περιβάλλον της MSWLogo περιλαμβάνει ορισμένες διακριτές περιοχές, οι οποίες παρουσιάζονται συνοπτικά παρακάτω.



Το περιβάλλον της MSWLogo με επισημασμένες τις περιοχές που αναφέρονται στα φύλλα δραστηριότητας

Η γλώσσα Logo είναι βέβαια από τις παλιότερες γλώσσες προγραμματισμού, με εκπαιδευτικό χαρακτήρα. Αυτό δημιουργεί, ορισμένες φορές, ένα είδος κριτικής στάσης απέναντι της, όχι τόσο εξαιτίας των μειονεκτημάτων της, ως εκπαιδευτικού μέσου, αλλά γιατί θεωρείται ξεπερασμένη, απαρχαιωμένη.

Οι εκπαιδευτικοί, πολλές φορές, θεωρούν ότι η γλώσσα Logo δεν πρέπει να διδάσκεται, αφού σύγχρονες γλώσσες και περιβάλλοντα, όπως για παράδειγμα το Scratch, είναι πολύ καλύτερα, αφού είναι πιο ελκυστικά για τους μαθητές και επιπλέον ενσωματώνουν προγραμματιστικά χαρακτηριστικά πολύ πιο σύγχρονα, interface πιο ευχάριστο κλπ.

Είναι γεγονός ότι τα δυο περιβάλλοντα έχουν διαφορετική στόχευση: για παράδειγμα, τα αποτελέσματα των προγραμμάτων της Logo είναι λιγότερο εντυπωσιακά από του Scratch,

αλλά το περιβάλλον της Logo φαίνεται να είναι πολύ απλούστερο.

Σε κάθε περίπτωση, τα «διασκεδαστικά» αποτελέσματα (ζωάκια που κινούνται, βγάζουν ήχους, η δημιουργία χρωματικών εντυπωσιακών συνδυασμών κ.τ.ό.), παράγονται πολύ πιο εύκολα σε σύγχρονα περιβάλλοντα όπως το Scratch.

Ωστόσο, οι βασικές επιδιώξεις της διδασκαλίας του προγραμματισμού, για παράδειγμα η έννοια των παραμέτρων, η επίλυση προβλημάτων με τη μέθοδο «διαίρει και βασίλευε» ή ακόμη καλύτερα, η σταδιακή επίλυση σύνθετων προβλημάτων με το συνδυασμό επιμέρους στοιχείων, εξυπηρετούνται εξίσου καλά στο ένα ή το άλλο περιβάλλον.

Εξάλλου, το περιβάλλον της Logo, προσφέρεται επίσης περισσότερο για μια προσέγγιση των προβλημάτων που συνδυάζουν στοιχεία προγραμματισμού και Μαθηματικών.

Η γενίκευση της κατασκευή πολυγώνων είναι σημαντική καθώς επιτρέπει τη διασύνδεση της Logo και του προγραμματισμού της χελώνας με τη Γεωμετρία.

Το διδακτικό πρόβλημα του «περάσματος» στον κύκλο αξίζει να εξεταστεί ιδιαίτερα καθώς είναι μια προσέγγιση η οποία, ενδεχομένως, θα μπορούσε να δημιουργήσει λανθασμένες αντιλήψεις στα παιδιά.

Θα πρέπει να τονιστεί ιδιαίτερα πως ο σκοπός των δραστηριοτήτων που περιγράφονται παρακάτω, ή άλλων ανάλογων δραστηριοτήτων, οι οποίες σχετίζονται με τις παραμέτρους, δεν είναι η εκμάθηση του μηχανισμού με τον οποίο «περνιούνται» οι παράμετροι και οι τιμές τους από το πρόγραμμα που καλεί προς στην καλούμενη διαδικασία. Με άλλα λόγια, διδακτικός στόχος δεν είναι οι *μηχανισμοί* των παραμέτρων, αλλά η *έννοια της γενικευμένης περιγραφής*: ένα σχήμα έχει μερικά συγκεκριμένα χαρακτηριστικά, αλλά με τις διαδικασίες μπορώ να το αναπαραγάγω όσες φορές επιθυμώ και με τις παραμέτρους μπορώ να του αυξήσω ή να του ελαττώσω το μέγεθος, να το περιστρέψω κ.ά.

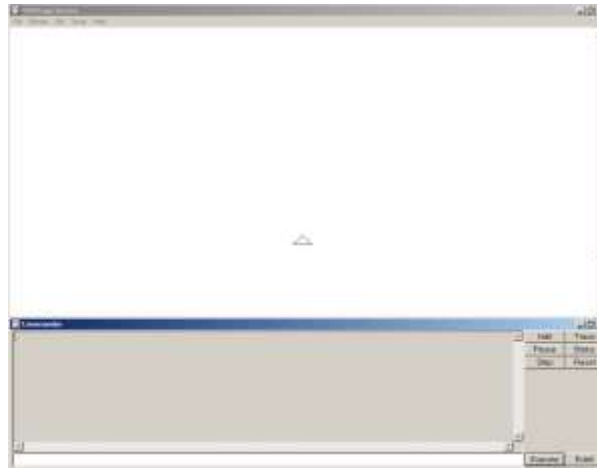
Οι μαθητές μπορούν να κατανοήσουν την αναδρομή; Το ερώτημα αυτό απασχόλησε πολύ τη διεθνή κοινότητα, κυρίως τη δεκαετία του '80. Από όσα είναι γνωστά μέχρι σήμερα, δεν υπάρχει σαφής απάντηση: υπάρχουν έρευνες οι οποίες τεκμηριώνουν την άποψη ότι «να» οι μαθητές κατανοούν την αναδρομή και μάλιστα πολύ ικανοποιητικά και έρευνες που τεκμηριώνουν την αντίθετη άποψη.

Αν και αποτελεί ένα πολύ ισχυρό εργαλείο επίλυσης προβλημάτων, δεν συμπεριλαμβάνεται συχνά και με ρητό τρόπο στα διδασκόμενα αντικείμενα.

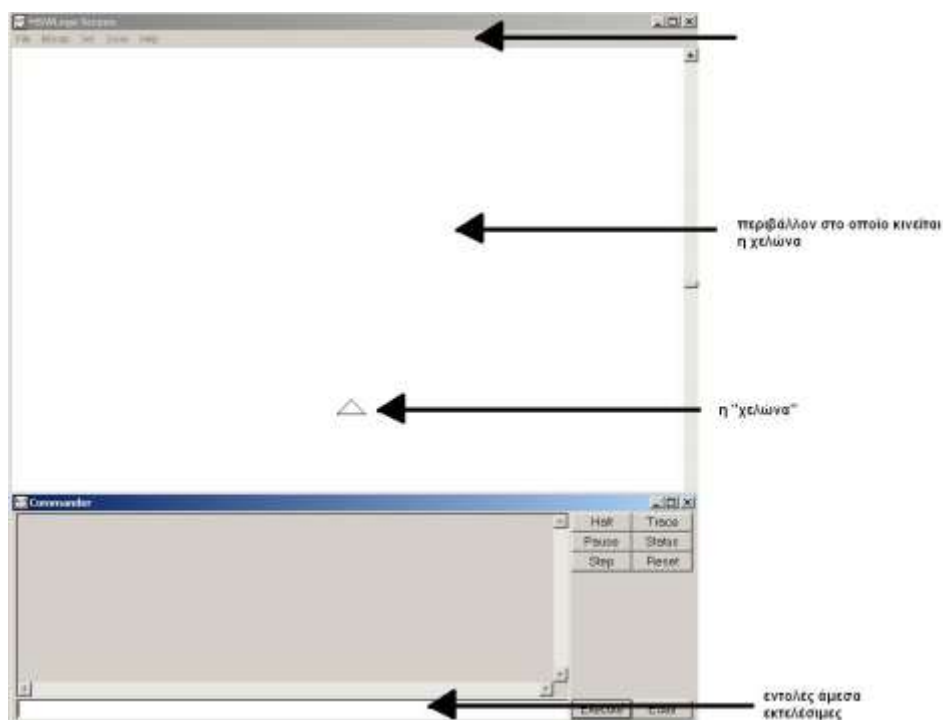
Το σχετικό σενάριο συμπεριλήφθηκε για λόγους πληρότητας και διότι τα σχήματα που δημιουργούνται με αναδρομικές διαδικασίες (fractals κλπ) είναι ιδιαίτερα δημοφιλή μεταξύ των εκπαιδευτικών (αν και δεν είναι βέβαιο ότι τα χρησιμοποιούν, με οιοδήποτε τρόπο, στη διδασκαλία τους).

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1

Γνωριμία με το περιβάλλον MSW Logo – Οι πρώτες εντολές



Άνοιξε το πρόγραμμα MSW Logo από το αντίστοιχο εικονίδιο ή από τον κατάλογο επιλογών «Έναρξη»



- Προσπάθησε να αναγνωρίσεις τα τμήματα του περιβάλλοντος που δείχνουν τα βέλη.
- Πληκτρολόγησε μια οιαδήποτε λέξη – π.χ. το όνομα σου. Εντόπισε το σημείο στο οποίο εμφανίζεται το όνομα.σου.
- Πάτησε στη συνέχεια “Enter”, δες τι θα «απαντήσει» το σύστημα. Ποιο είναι το νόημα αυτού του μηνύματος που εμφανίζεται;

- Εκτέλεσε τις παρακάτω εντολές, μία προς μία και διαπίστωσε το αποτέλεσμα τους.

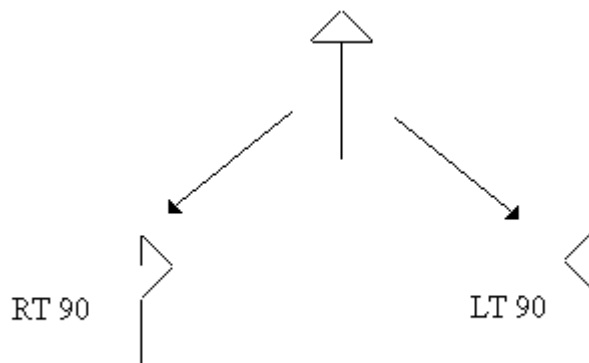
FD xx όπου το xx παριστάνει έναν αριθμό. Για παράδειγμα FD 50. Το αποτέλεσμα θα είναι να προχωρήσει η χελώνα 50 «βήματα» προς την κατεύθυνση προς την οποία είναι στραμμένη:



BK xx όπου το xx παριστάνει έναν αριθμό. Για παράδειγμα BK 50. Το αποτέλεσμα θα είναι να οπισθοχωρήσει η χελώνα κατά 50 «βήματα».



RT xx όπου το xx παριστάνει έναν αριθμό. Για παράδειγμα η εντολή RT 90 θα αναγκάσει τη χελώνα να στραφεί κατά 90 μοίρες δεξιά (κατά την έννοια της ίδιας της χελώνας), ενώ η εντολή LT 90 θα την αναγκάσει να στραφεί 90 μοίρες αριστερά:



CS «καθαρίζει» την οθόνη από γραφικά (δηλαδή σχήματα) και ξαναστέλνει τη χελώνα στην αρχή. (δηλαδή στο κέντρο της οθόνης).

CT «καθαρίζει» την οθόνη από κείμενο και ξαναστέλνει τη χελώνα στην αρχή. (δηλαδή στο κέντρο της οθόνης).

Μερικές πρόσθετες εντολές:

PU ανασηκώνεται το «μολύβι» της χελώνας κι έτσι όταν προχωράει δεν αφήνει ίχνος.

PD το τυχόν ανασηκωμένο μολύβι «ξανακατεβαίνει».

HT η χελώνα γίνεται «αόρατη»

ST η χελώνα γίνεται «ορατή»



Το παραπάνω σχήμα, είναι αποτέλεσμα των εξής εντολών:

FD 50

PU

FD 40

PD

FD 50

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2

Φτιάχνοντας σχήματα

Δραστηριότητα 1

Οι εντολές

FD 100

RT 90

τι σχήμα θα δημιουργήσουν στην οθόνη;

Οι παρακάτω εντολές:

FD 100

RT 90

FD 100

RT 90

FD 100

RT 90

FD 100

RT 90

τι σχήμα θα δημιουργήσουν στην οθόνη; Αντί για αυτές, μπορούμε να γράψουμε:

CS

repeat 4 [fd 60 rt 90]

Το αποτέλεσμα είναι το ίδιο;

Δραστηριότητα 2

Επιθυμούμε να δώσουμε μια εντολή με τη χρήση της repeat για να σχηματιστεί ένα ισόπλευρο τρίγωνο.

Προτείνεται το εξής:







CS

repeat 3 [fd 120 rt 60]

Θα έχουμε το επιθυμητό αποτέλεσμα; Δοκιμάστε το στον υπολογιστή σας.

Δραστηριότητα 3

Με την ίδια λογική μπορείτε να κατασκευάσετε διάφορα κανονικά πολύγωνα:

	Πεντάγωνο			
	Εξάγωνο			
	Οκτάγωνο			
	Εννεάγωνο			
	Δεκάγωνο			
	Κύκλος			

(ο παραπάνω πίνακας είναι από τα μαθήματα *Logo* του Harvey Brian).

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 3

Φτιάχνοντας Διαδικασίες

Δραστηριότητα 1

Η εντολή IFELSE [] [] επιτρέπει την επιλογή ανάμεσα σε 2 ενδεχόμενα.

Για παράδειγμα η εντολή

```
ifelse (5<3) [print "MEG] [print "MIK]
```

θα προκαλέσει την εκτύπωση του μηνύματος MEG, ενώ η εντολή

```
ifelse (5>3) [print "MEG] [print "MIK]
```

θα προκαλέσει την εκτύπωση του μηνύματος MIK.

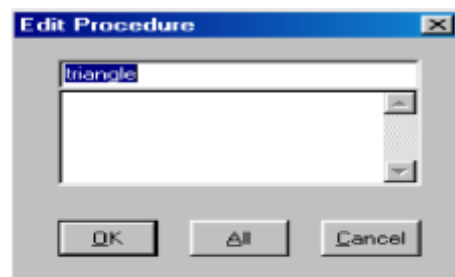
Οι διαδικασίες είναι ένα σύντομος τρόπος, για να εκτελέσουμε πολλές εντολές μαζί.

Αρκεί να δώσουμε ένα όνομα στις εντολές αυτές.

Με τη χρήση του παραθύρου Edit μπορούμε να κατασκευάσουμε διαδικασίες, όπως η διαδικασία triangle:

Από το μενού File επιλέγουμε Edit και πληκτρολογούμε το όνομα της διαδικασίας που θα κατασκευάσουμε.

Αυτομάτως εμφανίζεται το παράθυρο του Editor.

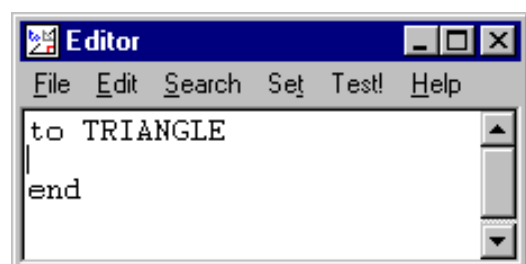


Με την εμφάνιση του παραθύρου Editor μπορούμε να γράψουμε τη διαδικασία, η οποία αρχίζει πάντοτε με τις λέξεις:

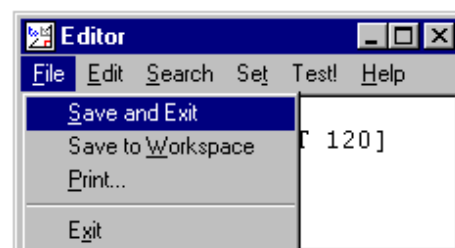
```
to name_procedure
```

όπου name_procedure είναι το όνομα της διαδικασίας (στο παράδειγμα: triangle) και τελειώνει με τη λέξη:

```
end
```



Αφού γραφεί η διαδικασία, πρέπει και να αποθηκευθεί:



Μια διαδικασία μπορεί να έχει και *ορίσματα* (*arguments*):

```
to tetragono :pleura
  repeat 4 [fd :pleura rt
  90]
end
```

Η Logo δημιουργεί μια θέση μνήμης με το όνομα pleura.

Κατά τη στιγμή της εκτέλεσης, η χελώνα θα προχωρήσει τόσα βήματα, όσα είναι γραμμένα μέσα στη θέση μνήμης pleura

Όταν λοιπόν εκτελεστεί η διαδικασία, θα πρέπει να κληθεί υποχρεωτικά με έναν αριθμό στη θέση του ορίσματος pleura: για παράδειγμα:

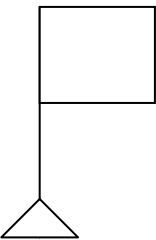
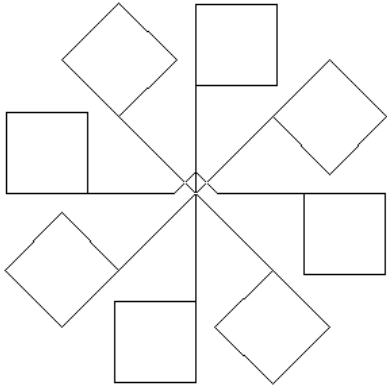
```
tetragono 100
```

Η Logo θα εκτελέσει τη διαδικασία tetragono, αντικαθιστώντας παντού το pleura με τον αριθμό 100

Δοκιμάστε το στον υπολογιστή.

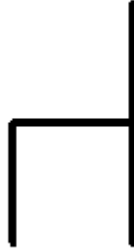
Με ποιο τρόπο μπορείτε να δώσετε εντολή στη χελώνα να σχηματίσει ένα τετράγωνο με πλευρά 120;

Η Logo επιτρέπει την επαναχρησιμοποίηση των διαδικασιών σε άλλες διαδικασίες:

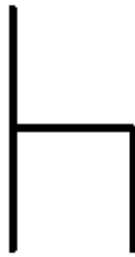
<pre>TO TETRAGONO REPEAT 4 [FD 60 RT 90] END TO SIMAIA FD 80 TETRAGONO BK 80 END</pre>	
<pre>TO TETRAGONO REPEAT 4 [FD 60 RT 90] END TO SIMAIA FD 80 TETRAGONO BK 80 END TO POLLES_SIMAIAS REPEAT 8 [SIMAIA RT 45] END</pre>	

Δραστηριότητα 2

Προτείνεται το ακόλουθο σχήμα:



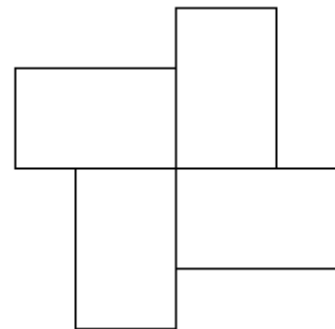
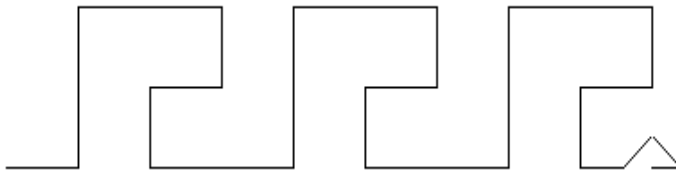
Με τρόπο θα δημιουργήσουμε μια διαδικασία **kathisma** που θα το σχεδιάζει;
Με ποιο τρόπο η διαδικασία θα μπορούσε να σχεδιάσει ένα κάθισμα με την πλάτη προς την αριστερή πλευρά;



Με ποιον τρόπο θα ήταν δυνατόν το κάθισμα αυτό να σχεδιαστεί σε μεγέθυνση ή σε σμίκρυνση; Με ποιο τρόπο θα μπορούσε να σχεδιαστεί «πλαγιασμένο»;

Δραστηριότητα 3

Να κατασκευαστούν τα παρακάτω σχήματα:



ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 4

Τα προγράμματα ως Συναρτήσεις

Δραστηριότητα 1

Η εντολή:

```
print 8 + 9
```

τι θα μας δώσει ως αποτέλεσμα;

Δοκιμάστε τις εξής εντολές:

```
make "x 8
```

```
print :x + 9
```

Ποιο είναι το αποτέλεσμα;

Τι αποτέλεσμα λοιπόν έχει η εντολή

```
make "x 8 ;
```

Δοκιμάστε τις εντολές

```
make "x 8
```

```
make "z 25
```

Ποιο θα είναι κατά τη γνώμη σας το αποτέλεσμα της εντολής

```
print :x + :z ;
```

Γράψτε τις εντολές που χρειάζονται για να προσθέσετε τρεις αριθμούς.

Οι παραπάνω εντολές σημαίνουν ότι οι αριθμητικές πράξεις μπορούν να γίνουν και με έμμεσο τρόπο, χρησιμοποιώντας μεταβλητές.

Είδαμε ότι σε μια μεταβλητή μπορούμε να αντιστοιχίσουμε έναν αριθμό, για παράδειγμα:

```
make "x 12
```

Είναι επίσης δυνατόν να αντιστοιχίσουμε σε μια μεταβλητή, μια ολόκληρη λίστα από αριθμούς:

```
make "x [1 2 3 4 6 5]
```

Τι αποτέλεσμα έχει η εντολή:

```
print :x ;
```

Οι λίστες είναι σαν σύνολα αριθμών. Δεν μπορούμε φυσικά να τις «προσθέσουμε».

Μπορούμε όμως να εφαρμόσουμε άλλες διαδικασίες.

Για παράδειγμα, γράψτε τις εντολές:

```
make "x [1 2 3 4 5]
print :x
print first :x
print butfirst :x
print bf :x
print last :x
print butlast :x
print bl :x
```

Ποια είναι τα αποτελέσματα της εκτέλεσης τους;

Ποιο θα είναι το αποτέλεσμα της εντολής:

```
print first bf :x ;
```

Με ποια εντολή η Logo θα μας εμφανίσει τη λίστα χωρίς τους δυο πρώτους αριθμούς (δηλαδή παραλείποντας το 1 και το 2);

Με ποια εντολή η Logo θα μας εμφανίσει τη λίστα χωρίς τους δυο τελευταίους αριθμούς (δηλαδή παραλείποντας το 4 και το 5);

Με ποια εντολή η Logo θα μας εμφανίσει τη λίστα χωρίς τον πρώτο και τον τελευταίο αριθμό;

Δραστηριότητα 2

Γράψτε την παρακάτω διαδικασία με δυο παραμέτρους, η οποία να δίνει ως αποτέλεσμα το άθροισμα τους:

```
to prosthesi :a :b
output :a + :b
end
```

Τι αποτέλεσμα θα δώσει η εντολή:

```
print prosthesi 10 20 ;
```

Η διαδικασία

```
to athroisma :list
```

```
ifelse (bf :list) = [] [output first :list] [output prosthesi (first :list) athroisma (bf :list)]
```

end

τι ακριβώς υλοποιεί;

Τι αποτέλεσμα θα δώσει η εντολή

print athroisma [10 20 40] ;

Μπορείτε να εξηγήσετε το αποτέλεσμα;

Τι αποτέλεσμα θα δώσει η εντολή

print athroisma [10 20 40 5 6 7 1] ;

Μπορείτε να εξηγήσετε το αποτέλεσμα;

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 5

Τα προγράμματα ως Συναρτήσεις: αναδρομή

Δραστηριότητα 1

Ο προσδιορισμός του μικρότερου από δυο αριθμούς είναι μια τετριμμένη διαδικασία

```
to min :a :b
```

```
  ifelse :a<:b [output :a] [output :b]
```

```
end
```

Μπορούμε λοιπόν να προσδιορίσουμε τον μικρότερο από μια σειρά αριθμών με την εξής απλή λογική: αν η σειρά των αριθμών έχει μόνο ένα στοιχείο, τότε ο μικρότερος είναι αυτός. Αλλιώς είναι ο μικρότερος από τον πρώτο αριθμό της σειράς και τον μικρότερο μεταξύ των υπολοίπων:

```
to elasson :list
```

```
  ifelse (bf :list) = [] [output first :list] [output min (first :list) elasson (bf :list)]
```

```
end
```

Ποιο θα είναι το αποτέλεσμα των παρακάτω εντολών

```
print elasson [5]
```

```
print elasson [5 1]
```

```
print elasson [5 4 2]
```

```
print elasson [5 12 9 2 74 56 1] ;
```

Μπορείτε να εξηγήσετε το κάθε αποτέλεσμα;

Δραστηριότητα 2

Επιθυμούμε η Logo να μας απαντά αν ένας αριθμός που της προτείνουμε βρίσκεται μέσα σε μια λίστα – ή όχι.

Αν ο αριθμός μας είναι μέσα στη λίστα, η Logo μας τυπώνει τον αριθμό που προτείνουμε. Αλλιώς μας τυπώνει «0» (υποτίθεται ότι αναζητούμε φυσικούς αριθμούς μεγαλύτερους του 1).

Η ακόλουθη διαδικασία anazitissi λειτουργεί κατ' αυτόν τον τρόπο;

```
to anazitissi :a :list
ifelse :list = [] [output 0] [ifelse :a = (first :list) [output :a] [output anazitissi :a (bf :list)]]
end
```

Μπορείτε να εξηγήσετε τη λειτουργία της;

Δραστηριότητα 3

Ο πύργος του Αννόϊ. Μπορείτε να εξηγήσετε με ποιο τρόπο η παρακάτω διαδικασία επιλύει το πρόβλημα;

```
to hanoi :numberdisks :a :b :c
ifelse :numberdisks = 1 [(print "A "disk "from :a "to :b)]
[ hanoi :numberdisks - 1 :a :c :b (print "A "disk "from :a "to :b)
  hanoi :numberdisks - 1 :c :b :a]
end
```

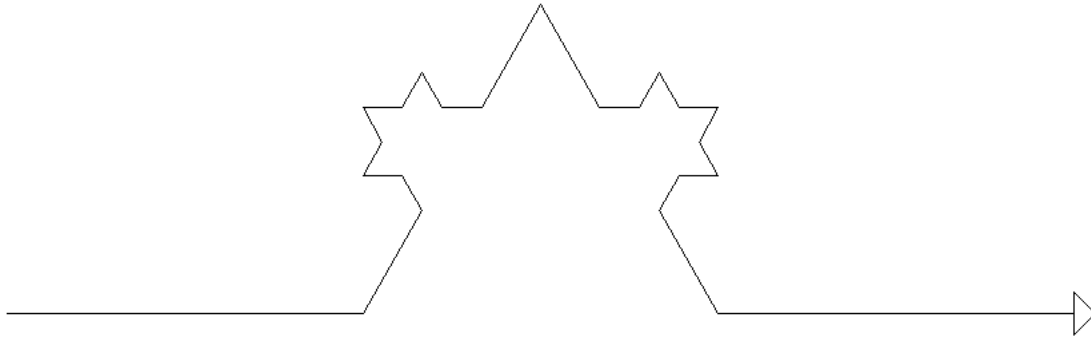
Δοκιμάστε να δείτε τον τρόπο με τον οποίο λειτουργεί με 1 δίσκο, με 2 δίσκους, με τρεις δίσκους.

Δραστηριότητα 4

Η παρακάτω διαδικασία:

```
to koch :mikos :limit
make "limit :limit - 1
ifelse :limit = 0 [fd :mikos] [fd :mikos / 3 lt 60 koch :mikos / 3 :limit rt 120 koch :mikos / 3
:limit lt 60 fd :mikos / 3]
end
```

δημιουργεί το παρακάτω σχήμα, αν εκτελεστεί η εντολή koch 800 4:



Μπορείτε να εξηγήστε πώς δημιουργείται το σχήμα;

ΟΛΑ ΤΑ ΣΧΕΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ

```
to athroisma :list
  ifelse (bf :list) = [] [output first :list] [output suma (first
    :list) athroisma (bf :list)]
end
```

```
to hannoi :numberdisks :a :b :c
  ifelse :numberdisks = 1 [(print "A "disk "from :a "to :b)] [hannoi
    :numberdisks - 1 :a :c :b (print "A "disk "from :a "to :b) hannoi
    :numberdisks - 1 :c :b :a]
end
```

```
to koch :mikos :limit
  make "limit :limit - 1
  ifelse :limit = 0 [fd :mikos] [fd :mikos / 3 lt 60 koch :mikos / 3
    :limit rt 120 koch :mikos / 3 :limit lt 60 fd :mikos / 3]
end
```

```
to mikrotero :a :b
  ifelse :a<:b [output :a] [output :b]
end
```

```
to min :list
```



```
ifelse (bf :list) = [] [output first :list] [output mikrotero (first  
:list) min (bf :list)]  
end
```

```
to search :a :list  
ifelse :list = [] [output 0] [ifelse :a = (first :list) [output :a]  
[output search :a (bf :list)]]  
end
```

```
to suma :a :b  
output :a + :b  
end
```

```
to taxinomissi :list  
ifelse (bf :list) = [] [output first :list] [output combine (min  
:list) (taxinomissi remove (min :list) :list)]  
end
```

18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

Δραστηριότητα 1

Οπτικός προγραμματισμός για μικρές ηλικίες

Η σύγχρονη τεχνολογία και η διεπαφή (interface) χρήστη δίνουν τη δυνατότητα (δημιουργίας και) διαχείρισης περιβαλλόντων προγραμματισμού (ή προ-προγραμματισμού, με την έννοια των *πρόδρομων διαδικασιών*, όπως το παράδειγμα των προ-μαθηματικών εννοιών ή των προ-μαθηματικών δραστηριοτήτων στην προσχολική εκπαίδευση και αγωγή).

Τα περιβάλλοντα αυτά είναι λοιπόν καλά προσαρμοσμένα, με τη γενική έννοια του όρου, για μια εισαγωγική διδασκαλία του προγραμματισμού σε μικρές ηλικίες (πρωτοβάθμια ακόμη και στις μικρότερες τάξεις – με κάποιες προϋποθέσεις ακόμη και σε μικρότερες ηλικίες).

Στα περιβάλλοντα αυτά ο νεαρός μαθητής ή μαθήτρια μπορεί να *καθοδηγήσει* μια «οντότητα» ή να *προγραμματίσει* ορισμένες ενέργειες και να διαπιστώσει τα αποτελέσματα των εντολών στην οθόνη.

Υπάρχουν γενικά 2 δυνατότητες: οι εντολές του εκτελούνται στιγμιαία, με σύγχρονο τρόπο, μόλις τις γράψει ή εκτελούνται διαφορισμένα, ετεροχρονισμένα, όταν αποφασίσει να τις εκτελέσει (συνολικά ή με βηματική εκτέλεση).

Το σημαντικό στοιχείο στα περιβάλλοντα αυτά είναι η *απουσία γραπτού λόγου*: με άλλα λόγια το παιδί μπορεί να προγραμματίσει (να κάνει ορισμένες βασικές ενέργειες προγραμματιστικού χαρακτήρα) χωρίς να γνωρίζει απαραίτητα γραφή και ανάγνωση.

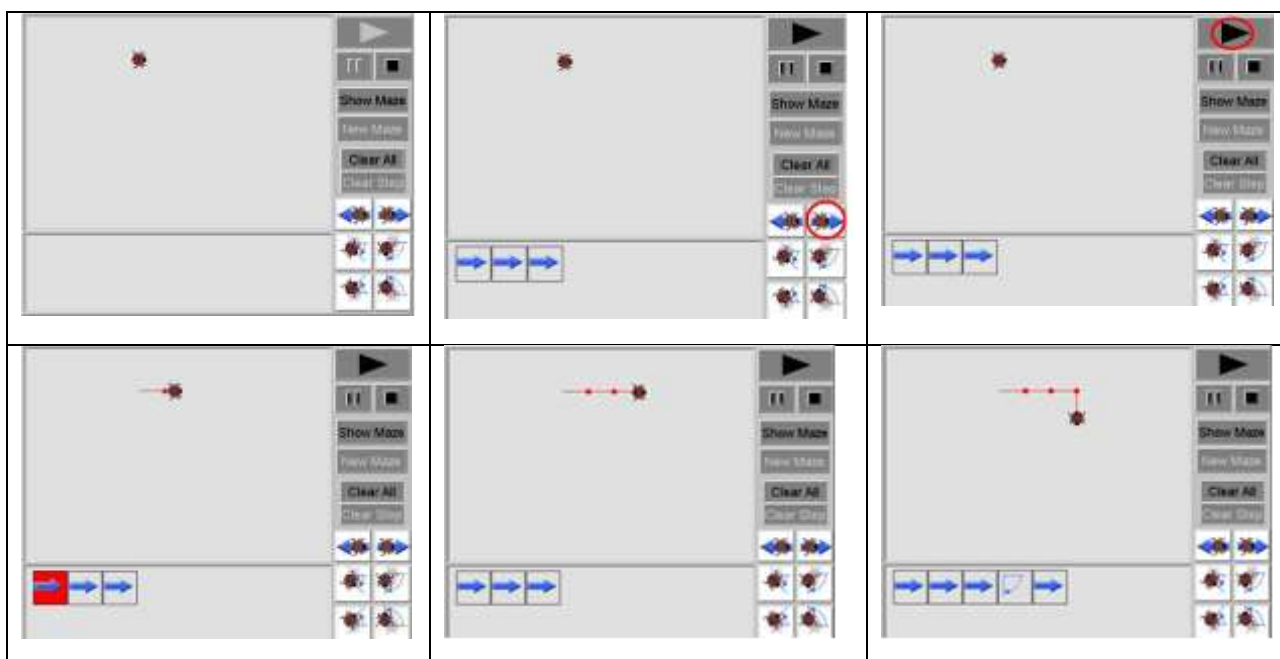
Επομένως τα περιβάλλοντα αυτά είναι ιδανικά για τις πρώτες σχολικές τάξεις ή ακόμη και στην προσχολική ηλικία.

Τα περιβάλλοντα αυτά έχουν το συνηθισμένο πλαίσιο που παρουσιάζουν τα Logo-like περιβάλλοντα: έναν «χώρο» (συνήθως ένα τμήμα της οθόνης) και μια «οντότητα» η οποία υπακούει στις εντολές του χρήστη και μετακινείται στο χώρο που προαναφέραμε.

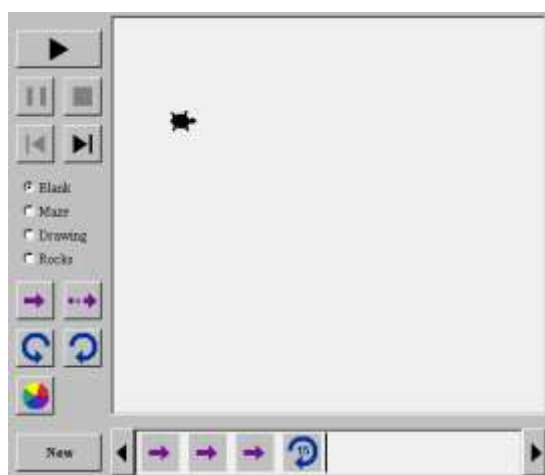
Τυπικό παράδειγμα αποτελούν η **LadyBug (πασχαλίτσα)** και η **χελωνίτσα**, ένα περιβάλλον online στην Αμερικανική Ένωση Καθηγητών Μαθηματικών:

http://nlvm.usu.edu/en/nav/frames_asid_286_g_1_t_3.html?open=activities&from=category_g_1_t_3.html

που επιτρέπουν τον προγραμματισμό αυτού του είδους:



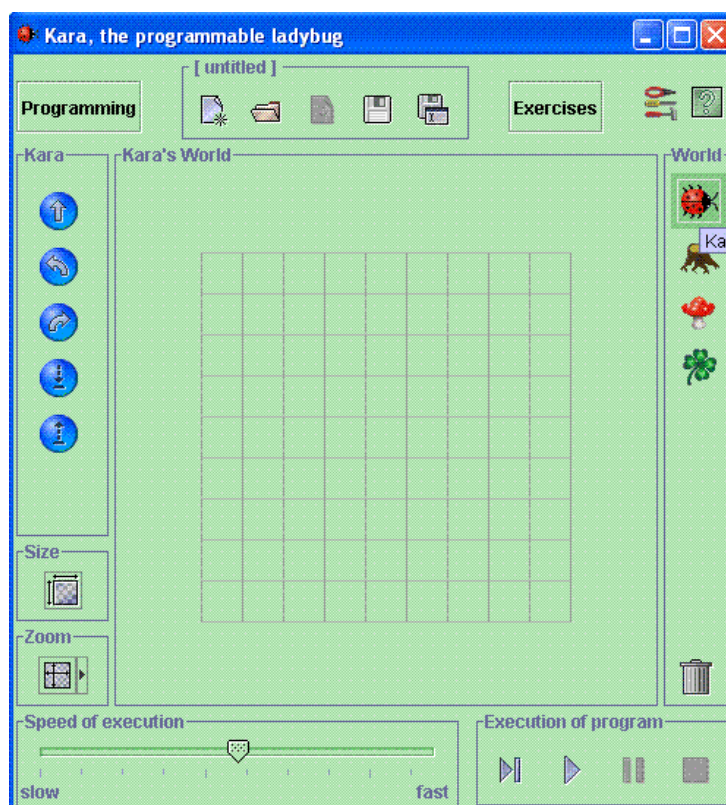
Προγραμματίζοντας την πασχαλίτσα



Προγραμματίζοντας την χελωνίτσα

Η **πασχαλίτσα Kara**, αν και σίγουρα δεν προορίζεται για την προσχολική ηλικία, βρίσκεται στο «άλλο άκρο», καθώς προγραμματίζεται μέσω μίας μηχανής πεπερασμένων καταστάσεων (finite state machine). Μπορεί κανείς να τη βρει στην εξής διεύθυνση (που αντιστοιχεί στο εκπαιδευτικό σύστημα της Ελβετίας):

<http://www.swisseduc.ch/compscience/karatojava/legokara/index.html> (τελευταία επίσκεψη, Φεβρουάριος 2013).



Η προγραμματιζόμενη πασχαλίτσα Kara

Στην πραγματικότητα η Kara αποτελεί μια ολόκληρη οικογένεια εκπαιδευτικών προγραμματιστικών περιβαλλόντων, που αντιστοιχούν σε διαφορετικά προγραμματιστικά συλ. Έτσι, υπάρχουν τα περιβάλλοντα Kara, JavaKara, KaraToJava, Turingkara, MultiKara, ακόμη και μια LegoKara:

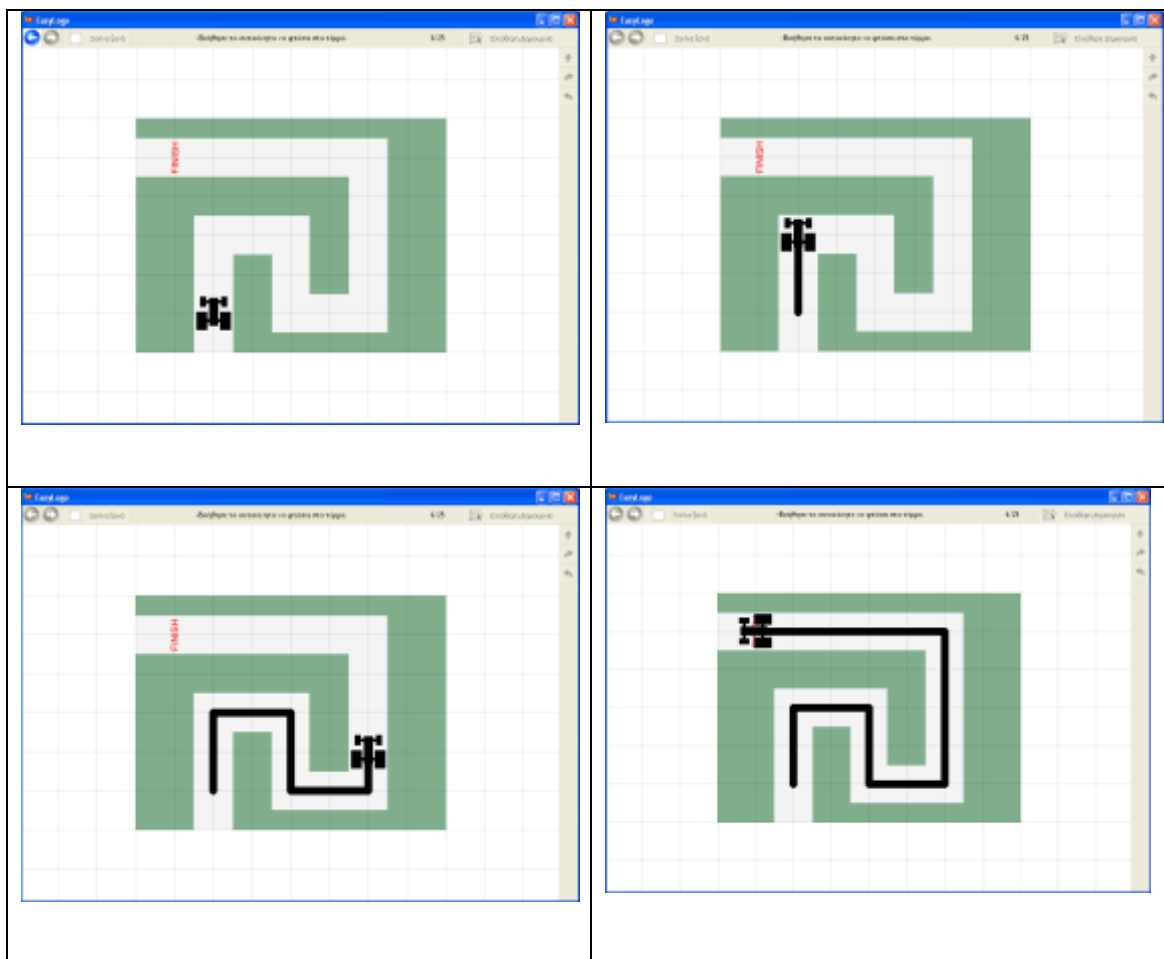


Το ρομπότ LegoKara αποτελεί τμήμα του Lego Mindstorms Set.

Στην ίδια ακριβώς κατεύθυνση βρίσκεται και το περιβάλλον EasyLogo.

Στο περιβάλλον αυτό, όπως περιγράφεται στο παρακάτω παράρτημα, δίνεται η δυνατότητα εκτέλεσης εντολών, χωρίς τη σύνταξη γραπτών εντολών («δραστηριότητες»). Έτσι, για παράδειγμα, ο χρήστης μπορεί να «καθοδηγήσει» ένα αυτοκίνητο αγώνων να φτάσει ως τον

προορισμό του:



Άμεση «καθοδήγηση» ενός αυτοκινήτου αγώνων στο περιβάλλον EasyLogo

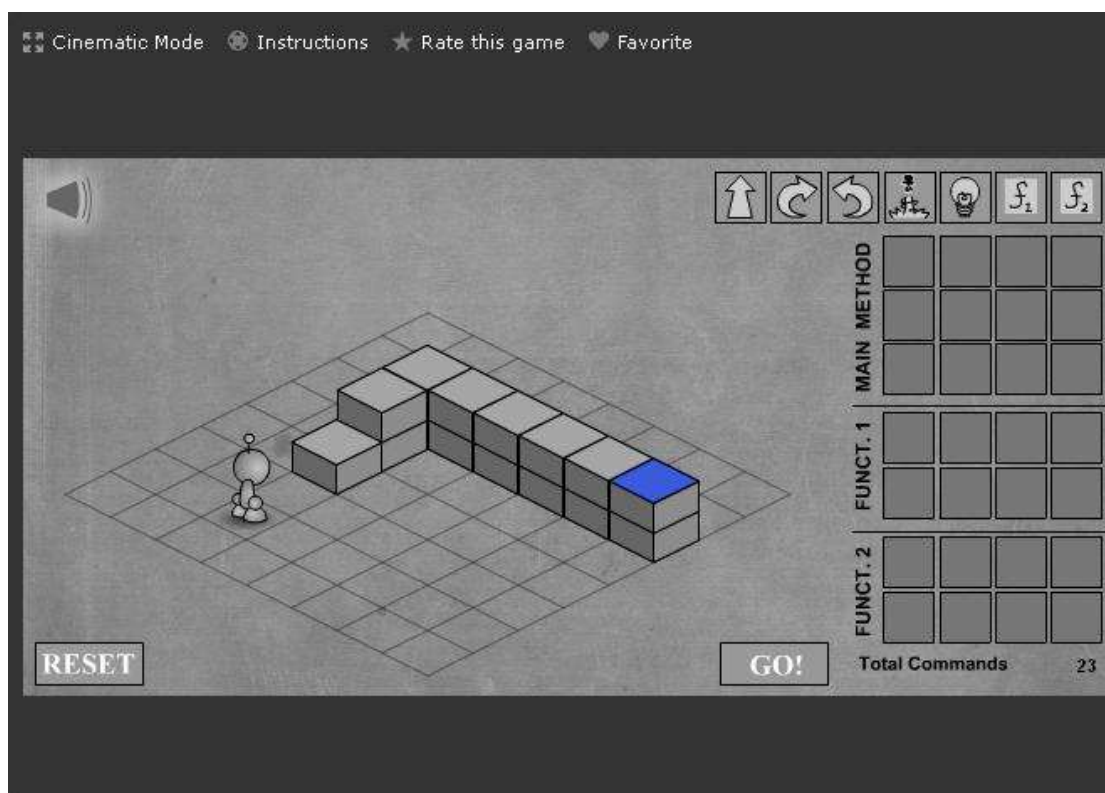
Το αναμενόμενο διδακτικό κέρδος από τη χρήση των περιβαλλόντων αυτών είναι βέβαια το γεγονός ότι το παιδί, μέσα στο πλαίσιο του κάθε περιβάλλοντος μαθαίνει ορισμένες βασικές αρχές του προγραμματισμού, όπως, για παράδειγμα, τη σειριακή εκτέλεση των εντολών – και μάλιστα με τη σειρά που δίνονται και όχι με τη σειρά που θα ήθελε ο χρήσης να εκτελούνται.

Τα περιβάλλοντα αυτά δεν είναι φυσικά τα μοναδικά στο είδος τους. Τα τελευταία χρόνια έχουν αναπτυχθεί πολλά παρόμοια περιβάλλοντα.

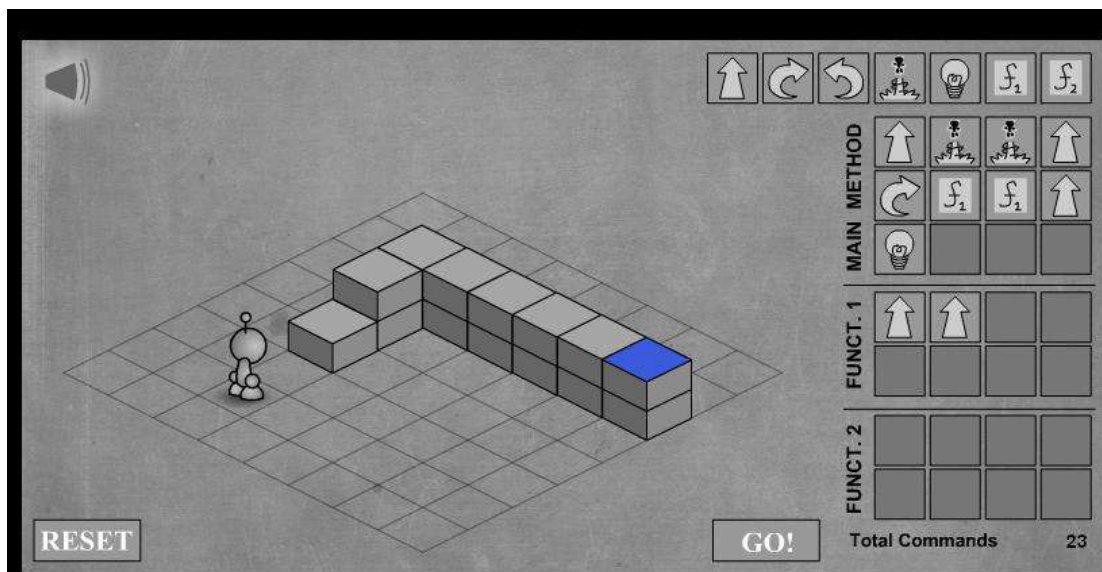
Περιβάλλοντα αυτού του είδους είναι το Treasure Hunter, το Rodozzle (καθώς και το Rodozzle Droid), όπως και το Light-Bot 2.0 (μια μετεξέλιξη του περιβάλλοντος Light-Bot) - ένα περιβάλλον μέσα στο οποίο ένα μικρό ρομπότ εκτελεί εντολές σε ένα τρισδιάστατο χώρο. Θα πρέπει να σημειωθεί πως τα παιχνίδια αυτά ανήκουν σε μια ιδιαίτερη κατηγορία παιχνιδιών στα οποία ο χρήστης πρέπει να εκτελέσει ορισμένα είδη δραστηριοτήτων, μερικές φορές κάτω από συγκεκριμένους περιορισμούς (όπως το συνολικό πλήθος των εκτελέσιμων

εντολών) – αν και σε ορισμένες περιπτώσεις ο χρήστης μπορεί να δημιουργήσει τις δικές του δραστηριότητες, οι οποίες πάντοτε εμπίπτουν στο ίδιο είδος (για παράδειγμα ένα ρομπότ να κάνει μια συγκεκριμένη διαδρομή) χωρίς δυνατότητες περαιτέρω επεκτάσεων. Θεωρητικά λοιπόν, δύσκολα θα μπορούσαν να ενταχθούν στους μικρόκοσμους (καθώς ο χρήστης δεν έχει την απαραίτητη ελευθερία δημιουργίας νέου τύπου δραστηριοτήτων). Τα παιχνίδια αυτά μπορούν να θεωρηθούν ως εκπαιδευτικά παιχνίδια που, κατά κάποιο τρόπο, εντάσσονται σε μια γενικότερη κατηγορία παιχνιδιών στρατηγικής όπως το SOKOBAN και άλλα. Μέσα σε κάποια πλαίσια έχουν τη δική τους διδακτική αξία. Είναι αξιοσημείωτο επίσης ότι πολλά από αυτά, παρά τη φαινομενική τους απλότητα, μπορούν να είναι εξαιρετικά πολύπλοκα.

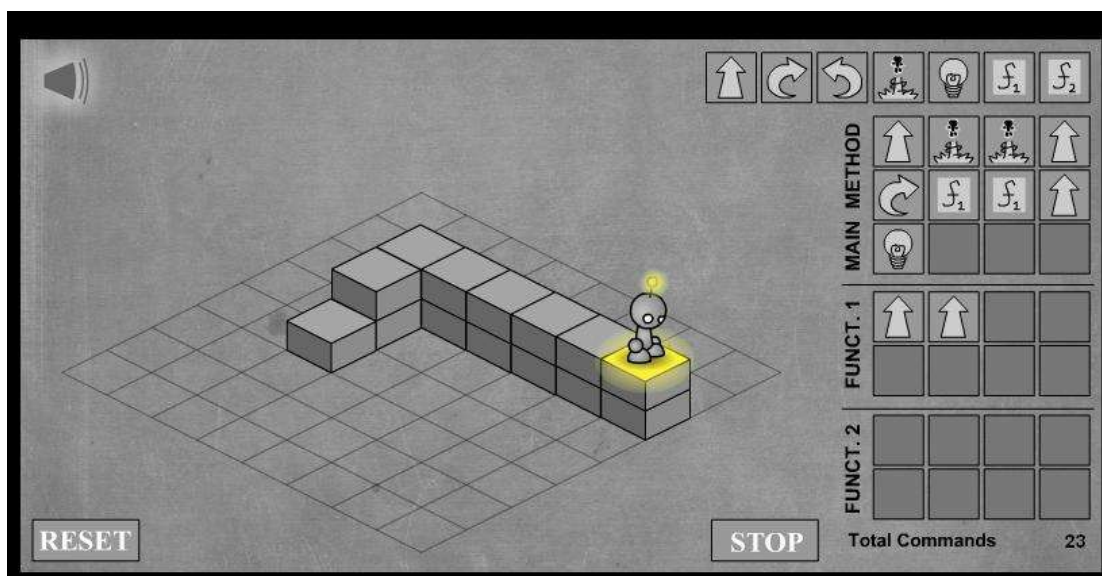
Θα πρέπει να υπογραμμιστεί ότι το Light-Bot 2.0 και τα υπόλοιπα περιβάλλοντα που αναφέρονται εδώ είναι παιχνίδια διαφόρων εταιρειών και επομένως η χρήση τους μπορεί να συνοδεύεται από ανεπιθύμητες (έμμεσες ή άμεσες) διαφημίσεις. Οι εκπαιδευτικοί θα πρέπει να είναι πολύ προσεκτικοί στη χρήση τους μέσα σε σχολικό περιβάλλον και να ελέγξουν σχολαστικά όλα αυτά τα σημεία πριν το εισαγάγουν στην τάξη.



Αρχική κατάσταση στο επίπεδο 4: το ρομπότ πρέπει να κινηθεί μέχρι το μπλε τετραγωνίδιο και να «ανάψει» τη λάμπα



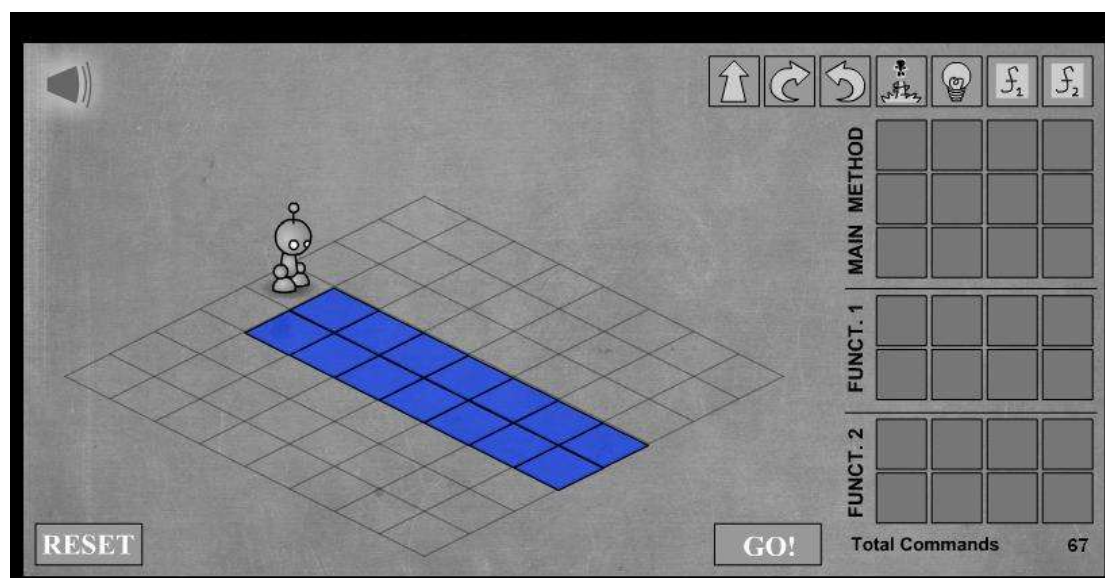
Το ρομπότ προγραμματίστηκε. Χρησιμοποιείται και μια συνάρτηση function f1 (η οποία απλώς εκτελεί 2 βήματα εμπρός)



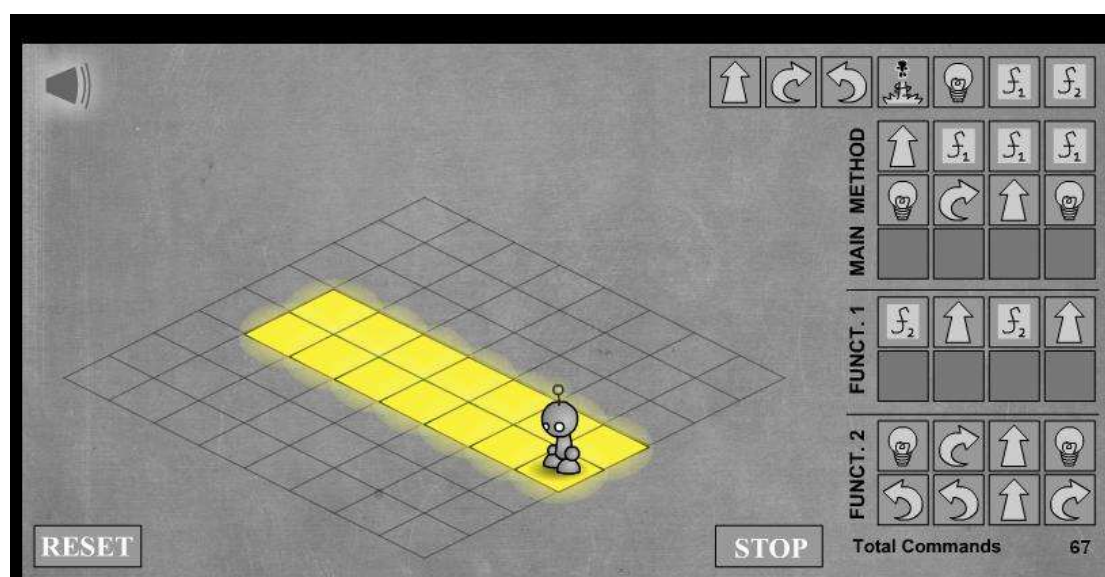
Το ρομπότ εκτέλεσε επιτυχώς την αποστολή του

Σε ορισμένες περιπτώσεις, η περιορισμένη χρήση του χώρου «προγραμματισμού» αναγκάζει τον προγραμματιστή να χρησιμοποιήσει με «οικονομία» τις εντολές (και άρα να κάνει μια «έξυπνη» χρήση των f_1 και f_2).

Έτσι στο επίπεδο 7:

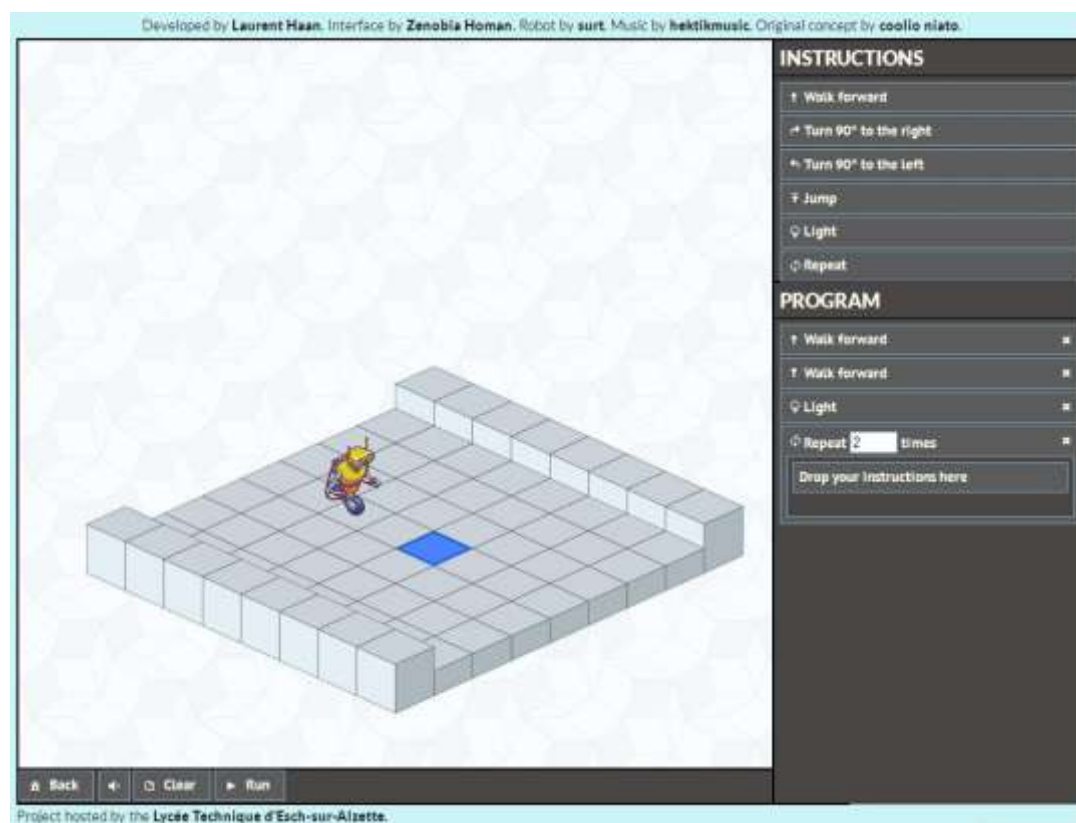


ο χρήστης πρέπει να επαναχρησιμοποιήσει τις f1 και f2:



Σε ελληνικό εκπαιδευτικό blogspot (καθηγητή Πληροφορικής <http://plioforikiatschool.blogspot.gr/2010/07/lightbot.html>) υπάρχει μια εκπαιδευτική ταινία 10 λεπτών περίπου που δείχνει τη χρήση του περιβάλλοντος Light-Bot 2.0 – και έμμεσα έναν τρόπο να δημιουργήσει κανείς «εγχειρίδια χρήσης» με «βιντεοσκόπηση» (capture) της οθόνης.

Μια άλλη έκδοση του ίδιου περιβάλλοντος (<http://www.lightbot.lu/>) υστερεί σε ορισμένα σημεία αλλά φαίνεται να αποτελεί ένα εκπαιδευτικό project:



Όπως αναφέραμε και παραπάνω, η επίλυση ενός τέτοιου «γρίφου» μπορεί να είναι δύσκολη, εξαιτίας του περιορισμένου συνολικού αριθμού διαθέσιμων εντολών. Για παράδειγμα:



Αρχική κατάσταση, μια αποτυχημένη λύση, μια λύση

Προσομοίωση: σε μια ιδιαίτερη θεώρηση των περιβαλλόντων αυτού του είδους (καθοδηγούμενες από ο χρήστη «οντότητες») θα μπορούσε, όπως έχει αναφερθεί και σε άλλα σημεία) θα ήταν δυνατόν να προσομοιώσει ο ίδιος ο εκπαιδευτικός (ή ένας μαθητής) την κίνηση ενός «ρομπότ» σε ένα περιβάλλον όπου θα ήταν εύκολο να εκτελεστούν εντολές χωρίς αμφισβήτηση (για παράδειγμα: ο εκπαιδευτικός ή ο μαθητής κινείται, στρίβει κλπ σε ένα πάτωμα με μεγάλα πλακάκια).

Για τις πολύ μικρές ηλικίες, παιχνίδια αυτού του είδους μπορούν να αποτελέσουν ένα πεδίο για μια εισαγωγή στην κατανόηση μεριών βασικών χαρακτηριστικών του προγραμματισμού (πρόδρομων καταστάσεων, προ-προγραμματισμού).

ΠΑΡΑΡΤΗΜΑ 1: ο Χελωνόκοσμος

Ο χελωνόκοσμος αποτελεί ένα τυπικό παράδειγμα περιβάλλοντος Logo-like και είναι κατασκευασμένος εξ ολοκλήρου στην Ελλάδα.

Στα παρακάτω, περιγράφονται, συνοπτικά, 2 εισαγωγικές διδακτικές δραστηριότητες που είναι βασισμένες στο Χελωνόκοσμο.

Ο «χελωνόκοσμος» (http://etl.ppp.uoa.gr/content/download/eslate_kits.htm) αποτελεί ένα περιβάλλον της Logo πλήρως ελληνικό (το περιβάλλον είναι δημιουργημένο στην Ελλάδα και όλα τα στοιχεία του, όπως μενού, μηνύματα κ.τ.λ. είναι στα ελληνικά – ακόμη και οι εντολές προς τη χελώνα είναι στα ελληνικά).

Στα επόμενα, παρατίθενται μια σειρά διδακτικών σεναρίων και μαθημάτων, ανάλογων προς τα προηγούμενα, αλλά γραμμένων στο περιβάλλον του «Χελωνόκοσμου». Τα θέματα που εξετάζονται είναι ο δομημένος προγραμματισμός («συνθέτοντας ψηφιδωτά»), ειδικότερα οι επαναληπτικές δομές («ο χελωνόκοσμος και η κερήθρα») καθώς και η αναδρομή («Φωνάζοντας το όνομά μου»).

Τα ψηφιδωτά

Το παρόν εκπαιδευτικό σενάριο σχεδιάστηκε με στόχο να αξιοποιηθεί για την εφαρμογή του τμηματικού και δομημένου προγραμματισμού κατά την επίλυση ενός προβλήματος. Η αξιοποίησή του ως τέτοιου είδους εφαρμογή αναφέρεται στην Γ' Λυκείου. Επιπρόσθετα μπορεί να αξιοποιηθεί και ως εφαρμογή των προγραμματιστικών δομών σύμφωνα με το αναλυτικό πρόγραμμα της Γ' Γυμνασίου.

Εισαγωγή

Στο σενάριο, το σχεδιαστικό πρόβλημα της σύνθεσης ενός ψηφιδωτού αναλύεται σε επιμέρους προβλήματα και επιλύεται με τη χρήση προγραμματιστικών δομών και διαδικασιών.

Διδακτικοί στόχοι

Οι μαθητές μετά το τέλος του σεναρίου θα μπορούν:

- Να αποδομούν – αναλύουν ένα προγραμματιστικό πρόβλημα σε επιμέρους μικρότερα
- Να προσδιορίζουν τη σειρά εκτέλεσης των υποπρογραμμάτων (διαδικασιών) σε ένα πρόγραμμα
- Να αναγνωρίζουν τις παραμέτρους (μεταβλητές) που μπορεί να ανταλλάσσουν

μεταξύ τους διαφορετικά υποπρογράμματα

- Να πραγματοποιούν αλλαγές σε διαδικασίες και στη ροή εκτέλεσής τους ώστε να επιλύουν διαφορετικά προβλήματα

Οργάνωση διδασκαλίας και απαιτούμενη υλικοτεχνική υποδομή

Κοινωνική ενορχήστρωση της τάξης: Ομάδες 2-3 μαθητών μπροστά σε κάθε Η/Υ

Εξοπλισμός: Εκπαιδευτικό λογισμικό «Χελωνόκοσμος» - Μικρόκοσμος «Συνθέτοντας Ψηφιδωτά»

Προαιρετικά: Βιντεοπροβολέας για την άμεση και συγχρονισμένη εισαγωγή όλης της τάξης στις δραστηριότητες καθώς και την προβολή των αποτελεσμάτων των ομάδων

Προαιρετικά: Φύλλα εργασίας για τη διατύπωση εκτιμήσεων και την αποτύπωση των αποτελεσμάτων. Εναλλακτικά μπορεί να γίνονται ολιγόλεπτες συζητήσεις στην τάξη με καταγισμό ιδεών.

Διάρκεια

Η διάρκεια υλοποίησης του σεναρίου εκτιμάται να είναι 2 με 3 διδακτικές ώρες.

Φάσεις Υλοποίησης

Φάση Α'

Αρχικά οι μαθητές ζωγραφίζουν στο χαρτί την εμφάνιση που αναμένουν ότι θα έχει ένα ψηφιδωτό αποτελούμενο από ίδια τρίγωνα. Συζητούν ποια θα είναι τα επιμέρους σχήματα που πρέπει να προγραμματιστούν, ώστε το τελικό αποτέλεσμα να είναι το ψηφιδωτό.

Δίνονται στους μαθητές οι διαδικασίες κατασκευής ενός τριγωνικού πλακιδίου και της οριζόντιας μετατόπισης της χελώνας:

```
για πλακάκι :a
επαναλαβε 3 [μ :a δ 120]
τέλος
για οριζ_μετακ :a
σπ
δ 90
μ (:a*sqrt(3))/2
α 90
σκ
τέλος
```

Οι μαθητές ξεκινούν τη διερεύνηση με τη διαδικασία «πλακάκι». Δίνουν διαφορετικές τιμές στην παράμετρο «α» και προσπαθούν να αναγνωρίσουν το ρόλο της στη διαδικασία. Πειραματίζονται δυναμικά, αξιοποιώντας τους μεταβολείς (κάνοντας κλικ στο ίχνος της χελώνας) και αναγνωρίζουν το σχήμα που δημιουργείται (ισόπλευρο τρίγωνο).

Πραγματοποιούν αντίστοιχη διερεύνηση με την «οριζ_μετακ» ώστε να αναγνωρίσουν το ρόλο της παραμέτρου. Ενώ η διαδικασία «οριζ_μετακ» φαίνεται να είναι πιο πολύπλοκη από την προηγούμενη, η διερεύνηση πραγματοποιείται εύκολα με τη χρήση του μεταβολέα και τη δυναμική αναπαράσταση των αποτελεσμάτων στον καμβά. Στη συνέχεια εκτελούν τις διαδικασίες τη μία μετά την άλλη και παρατηρούν την τελική θέση της χελώνας.

Τελικό στάδιο της παρούσης φάσης είναι η κατασκευή ενός απλού ψηφιδωτού που περιλαμβάνει πλακίδια στη σειρά. Ο αριθμός των πλακιδίων που θα περιλαμβάνεται στο απλό ψηφιδωτό μπορεί να είναι αρχικά σταθερός, αλλά στη συνέχεια οι μαθητές πρέπει να «γενικεύσουν» το υποπρόγραμμα, παραμετροποιώντας τόσο το μέγεθος, όσο και τον αριθμό των πλακιδίων. Οι αντίστοιχες παράμετροι α , β αναπαριστούνται με τους μεταβολείς, με τη βοήθεια των οποίων οι μαθητές έχουν τη δυνατότητα να δοκιμάσουν και να επαληθεύσουν την καινούργια τους διαδικασία

Ενδεικτική λύση για το «γενικευμένο» ψηφιδωτό «γραμμής»:

για γραμμή : α : β
επανάλαβε : β [πλακάκι : α οριζ_μετακ : α]
τέλος



Γραφική αναπαράσταση του ψηφιδωτού

Φάση Β'

Μετά την κατασκευή του απλού ψηφιδωτού, οι μαθητές κάνουν υποθέσεις για τις προσθήκες που απαιτούνται, προκειμένου το ψηφιδωτό να γεμίσει μεγαλύτερο μέρος της οθόνης και να εκτείνεται πλέον σε δύο διαστάσεις.

Στη συνέχεια δίνεται (έτοιμη) η διαδικασία «κατ_μετακ» και ζητείται από τους μαθητές να διερευνήσουν το αποτέλεσμά της, καθώς και το ρόλο των παραμέτρων στην τελική θέση της χελώνας. Η χρήση των μεταβολών αποτελεί πολύτιμο βοήθημα στη διερεύνηση τους.

```
για κατ_μετακ :α :β  
σπ  
α 90  
μ :β*(:α*sqrt(3))/2  
α 90  
μ :α  
δ 180  
σκ  
τέλος
```

Τελικό ζητούμενο είναι η χρήση όλων των παραπάνω διαδικασιών σε μια νέα διαδικασία που θα σχηματίζει πλέον το ολοκληρωμένο ψηφιδωτό. Οι μαθητές καλούνται να απαντήσουν στα ερωτήματα:

- Με ποια σειρά θα χρησιμοποιηθούν τα επιμέρους υποπρογράμματα;
- Πόσα πλακίδια πρέπει να τοποθετούνται; Είναι σταθερός ο αριθμός τους ή εξαρτάται από το μέγεθός τους;
- Το μέγεθος του πλακιδίου επηρεάζει με κάποιον τρόπο τη μετακίνηση της χελώνας στο επόμενο πλακίδιο;
- Η σειρά που δίνονται οι παράμετροι στα υποπρογράμματα είναι κοινή για όλα ή χρειάζεται κάποια προσαρμογή;

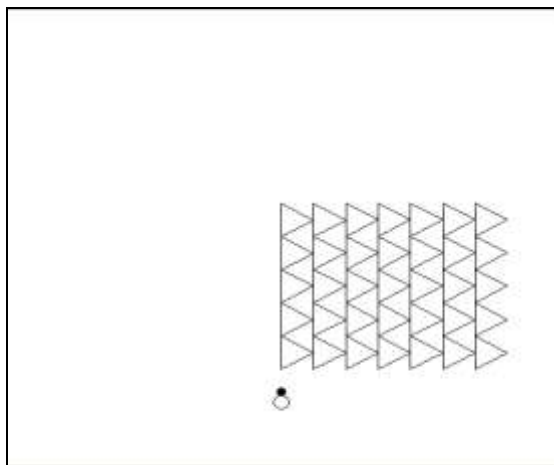
Το ψηφιδωτό στην τελική του μορφή ζητείται να είναι «γενικευμένο», να μπορεί δηλαδή να προσαρμοστεί σε διαφορετικού μεγέθους δάπεδα, αλλά και στις διακοσμητικές προτιμήσεις του «αδιοκτήτη», ο οποίος επιλέγει το μέγεθος των πλακιδίων.

Έτσι η νέα διαδικασία μπορεί να είναι:

```
για ψηφιδωτό :α :β :χ  
επανάλαβε :χ [ γραμμή :α :β κατ_μετακ :α  
:β]  
τέλος
```

Η αξιοποίηση των μεταβολέων για τη μελέτη της συμπεριφοράς της διαδικασίας «ψηφιδωτό» είναι επιβεβλημένη, αφού είναι ο μοναδικός τρόπος να μελετηθούν ταυτόχρονα οι τρεις παράμετροι της διαδικασίας. Με τη βοήθεια των μεταβολέων ο προγραμματισμός του ψηφιδωτού αποκτά μια παιγνιώδη χροιά και προσφέρει, στους μαθητές, πολλαπλά σχήματα ως αποτέλεσμα.

Ένα από τα ψηφιδωτά που μπορεί να παραχθεί από την παραπάνω διαδικασία είναι:



Επεκτάσεις

Βασική επέκταση του σεναρίου μπορεί να είναι η αλλαγή του δομικού στοιχείου του ψηφιδωτού (τριγώνου) με κάποιο άλλο γνωστό σχήμα. Αυτή η επέκταση αναδεικνύει την αξία του τμηματικού προγραμματισμού και παρέχει νέα αφορμή για δημιουργία από τους μαθητές.

Αν για παράδειγμα, το δομικό σχήμα είναι το τετράγωνο:

- Ποια θα είναι η εκτιμώμενη μορφή του ψηφιδωτού;
- Ποια διαδικασία πρέπει να μεταβληθεί ώστε να «παράγεται» το σωστό σχήμα;
- Τι αλλαγές πρέπει να γίνουν στις διαδικασίες μετατόπισης της χελώνας ώστε τα πλακίδια να παραμένουν το ένα δίπλα στο άλλο;

Άλλες επεκτάσεις που μπορούν να ζητηθούν και ως ανεξάρτητες δραστηριότητες είναι:

- A) Το ψηφιδωτό να καταλαμβάνει ολόκληρο το δάπεδο δίνοντας κατάλληλη αρχική θέση στη χελώνα
- B) Το ψηφιδωτό περιμετρικά να είναι «κλειστό», δηλαδή να περικλείεται από ευθείες γραμμές
- Γ) Να είναι δυνατή η παραμετρική επιλογή σχήματος πλακιδίου (π.χ. τετράγωνο)

Παρατηρήσεις

Το παραπάνω σενάριο δεν εστίασε στις εντολές που περιέχονται στα επιμέρους προσχεδιασμένα προγράμματα. Στόχος είναι να χρησιμοποιηθούν με προσιτό τρόπο στο «ψηφιδωτό» χωρίς να είναι απαραίτητη η γνώση π.χ. υπολογισμού του ύψους του τριγώνου. Παρόλα αυτά επεκτείνοντας τις γνωστικές περιοχές του σεναρίου είναι δυνατό να διερευνηθεί μαθηματικά πλέον η μετακίνηση της χελώνας, είτε προς το επόμενο πλακίδιο, είτε προς ένα οποιοδήποτε σημείο του δαπέδου σχετικό με σχεδιασμένο ψηφιδωτό.

Ο Χελωνόκοσμος και η Κερήθρα

Εμπλεκόμενες Γνωστικές Περιοχές

Πληροφορική: Δομή επανάληψης, δομή ακολουθίας

Εισαγωγή

Στο παρακάτω εκπαιδευτικό σενάριο προσεγγίζεται η έννοια της «Επαναληπτικής δομής» κατά την εισαγωγή των μαθητών σε αλγοριθμικές και προγραμματιστικές δομές. Ξεκινά επιλύοντας ένα πρόβλημα με «ακολουθιακό» τρόπο, αναζητά στη συνέχεια λύση μέσω επαναληπτικών δομών και τέλος αξιοποιεί τη δομή της επανάληψης στην αποτύπωση σύνθετων κατασκευών.

Διδακτικοί στόχοι

Οι μαθητές μετά το τέλος του σεναρίου θα μπορούν:

- Να αναγνωρίζουν μια ακολουθία προγραμματιστικών εντολών
- Να μετατρέπουν μια επαναλαμβανόμενη ακολουθία εντολών σε επαναληπτική δομή
- Να διακρίνουν πότε μια επαναληπτική δομή είναι αναγκαία, πότε χρήσιμη και πότε περιττή
- Να αναπαριστούν τα κρυφά βήματα που πραγματοποιούνται κατά την εκτέλεση μιας επανάληψης
- Να μπορούν να αναλύουν ένα πρόβλημα σε επιμέρους ενέργειες – δομές

Συμβατότητα με το αναλυτικό πρόγραμμα

Το παρόν εκπαιδευτικό σενάριο αναφέρεται στην έννοια της «Δομής Επανάληψης» όπως προσδιορίζεται στο αναλυτικό πρόγραμμα του μαθήματος της Πληροφορικής και μπορεί να αξιοποιηθεί στην ΣΤ' Δημοτικού, στη Γ' Γυμνασίου και στη Γ' Λυκείου.

Οργάνωση διδασκαλίας και απαιτούμενη υλικοτεχνική υποδομή

Κοινωνική ενορχήστρωση της τάξης: Ομάδες 2-3 μαθητών μπροστά σε κάθε Η/Υ

Συζήτηση και συνεργασία μέσα στην ομάδα.

Εξοπλισμός: Εκπαιδευτικό λογισμικό «Χελωνόκοσμος» - Μικρόκοσμος «Επανάληψη»

Προαιρετικά: Βιντεοπροβολέας για την άμεση και συγχρονισμένη εισαγωγή όλης της τάξης στις δραστηριότητες καθώς και την προβολή των αποτελεσμάτων των ομάδων

Προαιρετικά: Φύλλα εργασίας για τη διατύπωση εκτιμήσεων και την αποτύπωση των αποτελεσμάτων. Εναλλακτικά μπορεί να γίνονται ολιγόλεπτες συζητήσεις στην τάξη με καταγισμό ιδεών.

Διάρκεια

Η διάρκεια υλοποίησης του σεναρίου εκτιμάται να είναι 4 με 5 διδακτικές ώρες.

Αν και το σενάριο περιγράφεται σειριακά σε φάσεις, ο εκπαιδευτικός μπορεί να παραλείψει ή να επεκτείνει κάποιο κομμάτι, ανάλογα με τις προηγούμενες γνώσεις, το διαθέσιμο χρόνο και το γνωστικό επίπεδο της τάξης.

Φάσεις Σεναρίου

Φάση Α'

Κατή τη φάση αυτή οι μαθητές πειραματίζονται με απλές εντολές προς τη χελώνα («μπροστά», «δεξιά», «αριστερά») και προσπαθούν να σχηματίσουν ένα εξάγωνο. Δημιουργούν ακολουθίες εντολών τις οποίες και εκτελούν σειριακά.

Εκτιμώμενος χρόνος: 1 διδακτική ώρα

Φάση Β'

Οι μαθητές αναγνωρίζουν την πολλαπλή χρήση ίδιων εντολών και εισάγεται η νέα δομή, «η επανάληψη», με σκοπό τη βελτίωση της ακολουθίας εντολών που δημιούργησαν στην πρώτη φάση.

Εκτιμώμενος χρόνος: 1 διδακτική ώρα

Φάση Γ'

Αναδεικνύονται μέσα από συζήτηση οι περιορισμοί που υπάρχουν με τη χρήση μιας απλής δομής επανάληψης (Φάση Β') και εισάγεται η έννοια της παραμετροποίησης του εξαγώνου (του προγράμματος) με σκοπό να μπορούν να κατασκευαστούν εξάγωνα διαφορετικού μεγέθους.

Εκτιμώμενος χρόνος: 1 διδακτική ώρα

Φάση Δ'

Κατά τη φάση αυτή γίνεται από τους μαθητές διερευνητική εφαρμογή της παραμετροποιημένης δομής επανάληψης με σκοπό την κατασκευή σύνθετων σχημάτων όπως

του γνωστού σχήματος της κερήθρας ή σχημάτων τύπου ιστού αράχνης.

Εκτιμώμενος χρόνος: 1 με 2 διδακτικές ώρες

Υλοποίηση Διερεύνησης

Φάση Α'

Με τη χρήση των εντολών «μπροστά» ή «μ», «δεξιά» ή «δ», «αριστερά» ή «α», οι μαθητές προσπαθούν να οδηγήσουν τη χελώνα σε μια εξαγωνική κλειστή πορεία. Εδώ μπορούν να διατυπωθούν διαφορετικές λύσεις που έχουν διαφορετικό μήκος και διαφορετικές γωνίες στρέψης της χελώνας.

Σε κάθε περίπτωση οι μαθητές καλούνται πρώτα να συζητήσουν στην ομάδα και να εκτιμήσουν προς ποια διεύθυνση πρέπει να στρίψει η χελώνα, με ποια γωνία και πόσες κινήσεις πρέπει να κάνει, ώστε το σχήμα να κλείσει και να επανέρθει η χελώνα στο αρχικό σημείο. Τις εκτιμήσεις αυτές μπορούν να τις καταγράψουν στο χαρτί ή να τις συζητήσουν στην τάξη.

Στη συνέχεια γράφουν και εκτελούν με τη σειρά τις εντολές, παρατηρούν τα αποτελέσματα και πειραματίζονται με τις τιμές που δίνουν στις εντολές. Όταν πλέον είναι βέβαιοι για τη λύση που έδωσαν «κλείνουν» την ακολουθία σε ένα όνομα της επιλογής τους (διαδικασία) και δοκιμάζουν την κατασκευή του εξαγώνου με την κλήση του ονόματος (της διαδικασίας). Τέλος συζητούν τα πλεονεκτήματα και μειονεκτήματα της εκτέλεσης μιας ακολουθίας εντολών εντός και εκτός διαδικασίας

Επιμέρους διδακτικοί στόχοι:

Οι μαθητές μετά το τέλος της φάσης αυτής αναμένεται:

- Να μπορούν να σχηματίσουν μια ακολουθία εντολών που παρέχει συγκεκριμένο και γνωστό αποτέλεσμα (εξάγωνο)
- Να μπορούν να αναγνωρίζουν την αρχή και το τέλος μια ακολουθίας εντολών
- Να διατυπώνουν εκτιμήσεις για το αποτέλεσμα μιας ακολουθίας εντολών όταν αυτή δεν εκτελείται ολόκληρη ή όταν γίνονται αλλαγές σε ένα από τα βήματα (εντολή)
- Να μπορούν να μετατρέπουν μια ακολουθία εντολών σε μια διαδικασία

Ενδεικτική λύση για το εξάγωνο:

για εξάγωνο
μπροστά 100
δεξιά 60
μπροστά 100
δεξιά 60
μπροστά 100
δεξιά 60
μπροστά 100
δεξιά 60
μπροστά 100
δεξιά 60
μπροστά 100
δεξιά 60
μπροστά 100
δεξιά 60
τέλος

Φάση Β'

Ζητείται από τους μαθητές να αναγνωρίσουν μια ή περισσότερες εντολές που επαναλαμβάνονται στη διαδικασία που κατασκεύασαν. Η χρήση μιας καινούργιας δομής, αυτής της «Επανάληψης», δίνεται ως λύση – διευκόλυνση στους μαθητές. Πρόκειται για μια «αυτοματοποιημένη» ακολουθία εντολών που έχει μετατραπεί σε «Επανάληψη». Η λύση δίνεται «χαλασμένη» χωρίς να έχει το σωστό αριθμό επαναλήψεων και χωρίς να έχει τη σωστή γωνία στρέψης της χελώνας.

Οι μαθητές καλούνται να επέμβουν στον κώδικα της επανάληψης και να τον διορθώσουν ώστε το αποτέλεσμα να είναι το γνώριμο πλέον εξάγωνο.

Επιμέρους διδακτικοί στόχοι

Μετά το τέλος της φάσης, οι μαθητές θα μπορούν:

- Να αναγνωρίζουν τα επαναλαμβανόμενα τμήματα (εντολές) μιας ακολουθίας εντολών.
- Να μετατρέπουν μια επαναλαμβανόμενη ακολουθία σε δομή επανάληψης
- Να αναγνωρίζουν και να προσδιορίζουν το ρόλο του αριθμού των επαναλήψεων
- Να προσδιορίζουν την αρχή και το τέλος μιας επανάληψης

Ενδεικτική λύση για την επανάληψη:

```
για αυτομ_εξάγωνο  
επανάλαβε 6 [μ 100 δ 60]  
τέλος
```

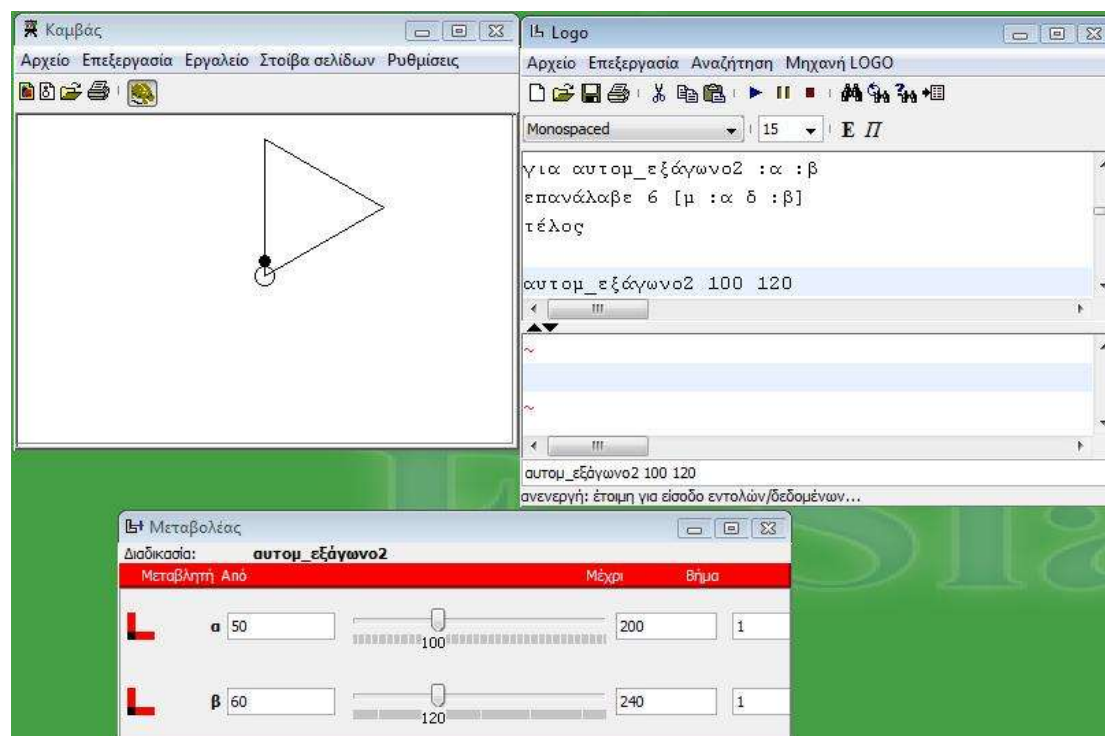
Φάση Γ'

Στη συνέχεια – στη φάση αυτή – πραγματοποιείται συζήτηση για τις δυνατότητες που έχει η χελώνα με την εκτέλεση της προηγούμενης διαδικασίας. Είναι εύκολα κατανοητό ότι η χελώνα δεν μπορεί να σχηματίσει τίποτα περισσότερο από ένα συγκεκριμένο εξάγωνο.

Δίνεται στους μαθητές η ακόλουθη διαδικασία και η αντίστοιχη εντολή κλήσης της:

```
για αυτομ_εξάγωνο2 :α :β  
επανάλαβε 6 [μ :α δ :β]  
τέλος  
  
αυτομ_εξάγωνο2 100 120
```

Οι μαθητές εκτελούν το πρόγραμμα και παρατηρούν την πορεία της χελώνας:



Παρατηρούν ότι το εξάγωνο δεν είναι πλέον εξάγωνο αλλά τρίγωνο. Παρόλο που οι εντολές

που χρησιμοποιήθηκαν φαίνεται να είναι ίδιες με αυτές του εξαγώνου το σχήμα τελικά είναι διαφορετικό.

Τώρα οι μαθητές καλούνται να αξιοποιήσουν τις δυνατότητες δυναμικού χειρισμού που παρέχει ο μικρόκοσμος για να διερευνήσουν το πρόβλημα. Χρησιμοποιούν τους μεταβολείς (κάνοντας κλικ στο ίχνος της χελώνας) και πειραματίζονται με τις τιμές τους. Εντοπίζουν τις τιμές των μεταβολέων που αντιστοιχούν στο εξαγώνο και τις καταχωρούν στην κλήση της διαδικασίας για να επαληθεύσουν το αποτέλεσμα.

Συγκρίνουν τις τιμές των μεταβολέων με τη χρήση των α , β στη διαδικασία και ανακαλύπτουν το ρόλο των παραμέτρων α , β .

Τέλος συζητούν τις δυνατότητες που τους προσφέρει αυτός ο τρόπος προγραμματισμού τόσο στην εκτέλεση της επανάληψης όσο και στο τελικό αποτέλεσμα.

Επιμέρους διδακτικοί στόχοι:

Οι μαθητές τελειώνοντας τη φάση αυτή θα μπορούν:

- Να αναγνωρίζουν το ρόλο μιας παραμέτρου στο τελικό αποτέλεσμα
- Να κάνουν εκτιμήσεις για τα σχήματα που μπορούν να δημιουργηθούν με την παραμετροποίηση της επανάληψης
- Να παραμετροποιούν μια διαδικασία
- Να εκτιμούν τα αποτελέσματα μιας διαδικασίας όταν οι παράμετροι κινούνται εκτός ορίων.

Φάση Δ'

Επιμέρους διδακτικοί στόχοι

Μετά το πέρας των θεμάτων της φάσης αυτής οι μαθητές θα μπορούν:

- Να κάνουν εκτιμήσεις για τα αποτελέσματα μιας εμφωλευμένης επανάληψης
- Να αναγνωρίζουν τη χρήση και το σκοπό μιας εμφωλευμένης επανάληψης
- Να κατανοούν τη σημασία του πλήθους των επαναλήψεων
- Να διακρίνουν τα όρια μεταξύ «εσωτερικής» και «εξωτερικής» επανάληψης

1^ο θέμα μελέτης- εφαρμογής

Στο θέμα αυτό οι μαθητές καλούνται να αξιοποιήσουν την παραμετροποιημένη επανάληψη που έφτιαξαν, για να κατασκευάσουν ένα γνωστό σχήμα που αποτελείται από ενωμένα εξάγωνα, δηλαδή μία κερήθρα.

Οι μαθητές ξεκινούν με το πρόγραμμα:

```
για κερήθρα :a :β  
επανάλαβε :β [αυτομ_εξάγωνο3 :a]  
τέλος  
  
για αυτομ_εξάγωνο3 :a  
επανάλαβε 6 [μ :a δ 60]  
τέλος
```

Εκτελούν τη διαδικασία κερήθρα (πχ κερήθρα 100 3) και πειραματίζονται χρησιμοποιώντας τους μεταβολείς.

Οι μαθητές :

α) Γνωρίζουν από την προηγούμενη φάση ότι το «αυτόμ_εξάγωνο» φτιάχνει σίγουρα ένα εξάγωνο.

β) Κάνουν εκτιμήσεις για τη συμπεριφορά της χελώνας κατά την επανάληψη της διαδικασίας «αυτομ_εξάγωνο».

γ) Παρατηρούν το αποτέλεσμα και το συγκρίνουν με τις εκτιμήσεις τους. Προσπαθούν να αιτιολογήσουν την πιθανόν μη αναμενόμενη συμπεριφορά της χελώνας (η χελώνα φαίνεται να σχηματίζει ένα και μοναδικό εξάγωνο).

δ) Πειραματίζονται χρησιμοποιώντας τους μεταβολείς και επιβεβαιώνουν ότι το σχήμα παραμένει ένα μοναδικό εξάγωνο.

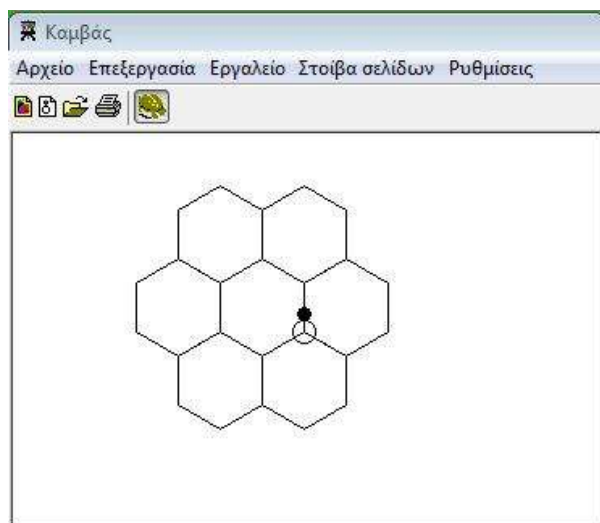
ε) Συζητούν και κάνουν εκτιμήσεις για τις αλλαγές που πρέπει να γίνουν ώστε να σχηματιστούν πολλαπλά εξάγωνα.

στ) Τελικά επεμβαίνουν στη διαδικασία «κερήθρα» προσθέτοντας μια ακόμα οδηγία μέσα στην επανάληψη, ώστε η χελώνα να αλλάζει κατεύθυνση.

Ενδεικτική λύση για την κερήθρα:

```
για κερήθρα :a :β  
επανάλαβε :β [αυτομ_εξάγωνο3 :a μ :a δ -60]  
τέλος
```

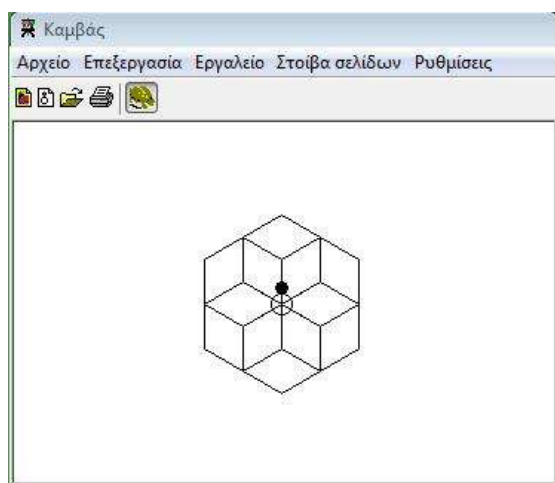
Και το αποτέλεσμα της «κερήθρας»:



Η οδηγία «μ :α δ -60» δεν αποτελεί τη μόνη λύση. Επιπρόσθετα οι μαθητές, προκειμένου να ζωγραφίσουν την κερήθρα, δοκιμάζουν διαφορετικές εντολές και τις τοποθετούν σε διαφορετικά σημεία της επανάληψης στη διαδικασία «κερήθρα». Σε κάθε περίπτωση χρησιμοποιούν τους μεταβολείς για να δοκιμάσουν δυναμικά τη νέα διαδικασία. Με αυτό τον τρόπο μπορούν να δημιουργηθούν διαφορετικά σχήματα που μπορεί να «πλησιάζουν» το επιθυμητό και να παρέχουν μια πολύ καλή βάση για πειραματισμό.

Για παράδειγμα, σε έναν πρώτο πειραματισμό μπορεί να υποθέσουν ότι αρκεί μια στροφή κάθε φορά ώστε το επόμενο εξάγωνο να σχηματίζεται δίπλα στο προηγούμενο.

Αυτό οδηγεί στο παρακάτω σχήμα:



2^ο Θέμα μελέτης – εφαρμογής

Σε αυτό το θέμα μελέτης οι μαθητές θα έρθουν αντιμέτωποι με ένα νέο και απροσδιόριστο σχήμα που μπορεί να κατασκευαστεί με τις γνώσεις που απόκτησαν και μπορεί να ισχυροποιήσει τους διδακτικούς στόχους των προηγούμενων φάσεων.

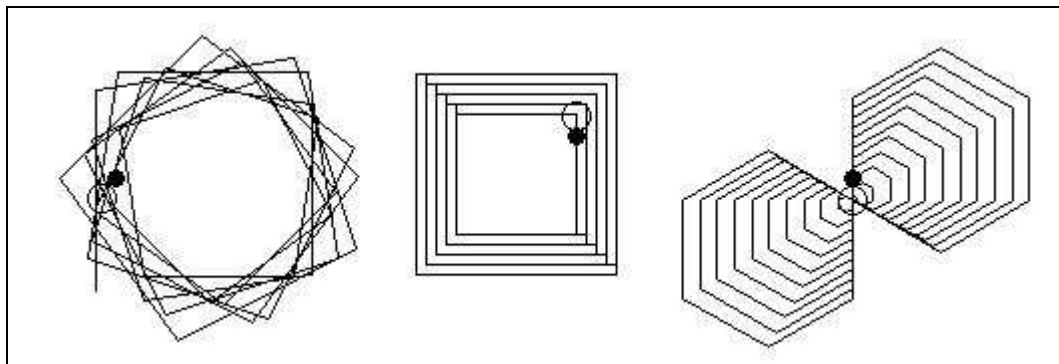
Το αποτέλεσμα εδώ, αν και μπορεί να εκτιμηθεί από το διδάσκοντα, δεν είναι προκαθορισμένο και αφήνει το μαθητή ελεύθερο στον πειραματισμό και την κατασκευή νέων σχημάτων δικής του έμπνευσης.

Ως βάση χρησιμοποιείται η παρακάτω διαδικασία:

για άγνωστο : α : β : γ
επανάλαβε : β [επανάλαβε 6 [μ : α δ : γ] φτιάξε " α : α -5]
τέλος

Οι μαθητές κάνουν εκτιμήσεις, καλούν τη διαδικασία (π.χ. «άγνωστο 100 10 60») και χρησιμοποιούν τους μεταβολείς για να προσδιορίσουν τη συμπεριφορά των μεταβλητών-παραμέτρων και των εκτελούμενων επαναλήψεων. Το νέο στοιχείο εφαρμογής εδώ, είναι η μεταβολή μιας παραμέτρου (« α ») μέσα στην ίδια την επανάληψη.

Τα παρακάτω αποτελέσματα είναι ενδεικτικά:



Φωνάζοντας το όνομά μου (εισαγωγή στην αναδρομή – recursion)

Εμπλεκόμενες Γνωστικές Περιοχές

Πληροφορική: Δομή Επανάληψης, προγραμματισμός με διαδικασίες, έννοια της Αναδρομής

Συμβατότητα με το αναλυτικό πρόγραμμα

Το παρόν εκπαιδευτικό σενάριο αναφέρεται στην έννοια της «Αναδρομής». Η έννοια αυτή δεν αποτελεί ξεχωριστή προγραμματιστική δομή αλλά σύμφωνα με το βιβλίο της Γ' Λυκείου παρουσιάζεται ως εναλλακτική λύση για προβλήματα που απαιτούν επαναληπτικές δομές. Η «Αναδρομή» ως επέκταση της ενότητας «Επανάληψης», προτείνεται να αξιοποιηθεί στη Γ' Γυμνασίου και στη Γ' Λυκείου.

Εισαγωγή

Η έννοια της «Αναδρομής» στο παρόν σενάριο, προσεγγίζεται μέσω της επίλυσης ενός πραγματικού σχεδιαστικού προβλήματος. Οι μαθητές αρχικά χρησιμοποιώντας μια προσχεδιασμένη αναδρομική διαδικασία σχηματίζουν τη γνωστή μορφή μιας πεταλούδας μέσω της οποίας διερευνούν την έννοια της αναδρομής. Τελικά προχωρούν σε πιο σύνθετα σχήματα-προβλήματα, όπου και εφαρμόζουν αναδρομική δομή για την επίλυσή τους.

Διδακτικοί στόχοι

Οι μαθητές μετά το τέλος του σεναρίου θα μπορούν:

- Να αναγνωρίζουν τη χρήση αναδρομής σε ένα πρόγραμμα
- Να προσδιορίζουν τη συνθήκη τέλους της αναδρομής
- Να συγκρίνουν μια δομή επανάληψης με μια δομή αναδρομής
- Να αναγνωρίζουν την ανάγκη χρήσης της δομής αναδρομής

Οργάνωση διδασκαλίας και απαιτούμενη υλικοτεχνική υποδομή

Κοινωνική εννοχήστρωση της τάξης: Ομάδες 2-3 μαθητών μπροστά σε κάθε Η/Υ

Εξοπλισμός: Εκπαιδευτικό λογισμικό «Χελωνόκοσμος» - Μικρόκοσμος «Αναδρομή-Επανάληψη»

Προαιρετικά: Βιντεοπροβολέας για την άμεση και συγχρονισμένη εισαγωγή όλης της τάξης στις δραστηριότητες καθώς και την προβολή των αποτελεσμάτων των ομάδων

Προαιρετικά: Φύλλα εργασίας για τη διατύπωση εκτιμήσεων και την αποτύπωση των αποτελεσμάτων. Εναλλακτικά μπορεί να γίνονται ολιγόλεπτες συζητήσεις στην τάξη με καταγισμό ιδεών.

Διάρκεια

Η διάρκεια υλοποίησης του σεναρίου εκτιμάται σε 2 με 3 διδακτικές ώρες.

Αν οι μαθητές δεν έχουν ασχοληθεί εκτενώς με τη δομή επανάληψης, τότε η διάρκεια μπορεί να επεκταθεί κατά 1 ώρα ώστε να εξεταστεί σε μεγαλύτερο βαθμό η «επανάληψη».

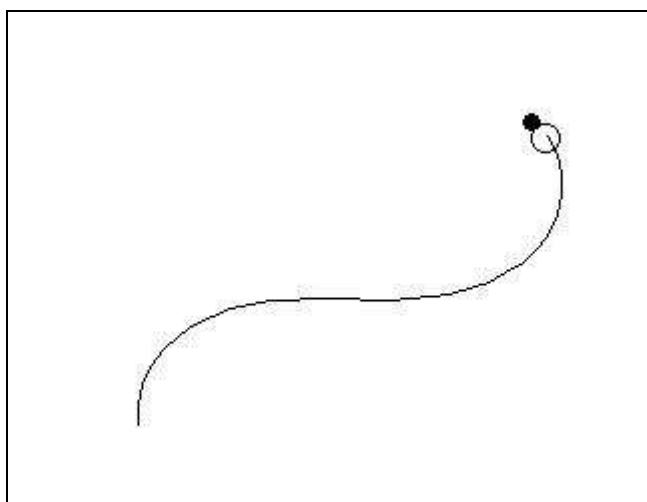
Φάσεις Υλοποίησης

Φάση Α'

Κατή τη φάση αυτή δίνεται στους μαθητές προς διερεύνηση η παρακάτω διαδικασία:

για αναδρα :a :β
αν :a < 1 [σταματησε]
μ 10
δ :β
αναδρα :a - 1 :β - 1
τέλος

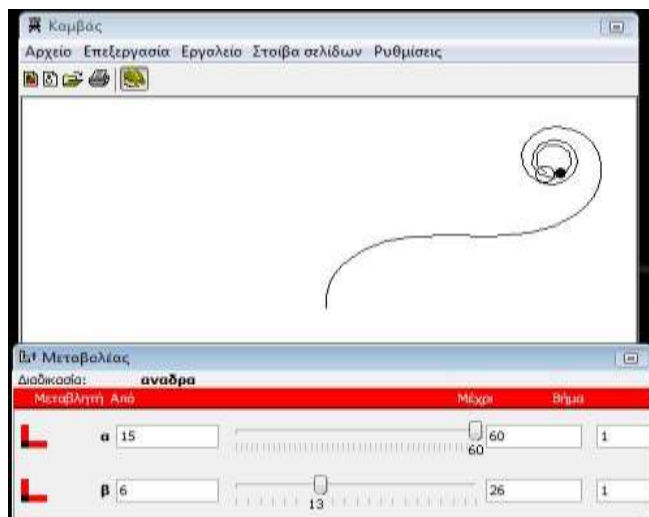
Η διαδικασία εκτελείται με ενδεικτικές τιμές για τις παραμέτρους «α» και «β» 30 και 13, οπότε και σχηματίζεται το σχήμα:



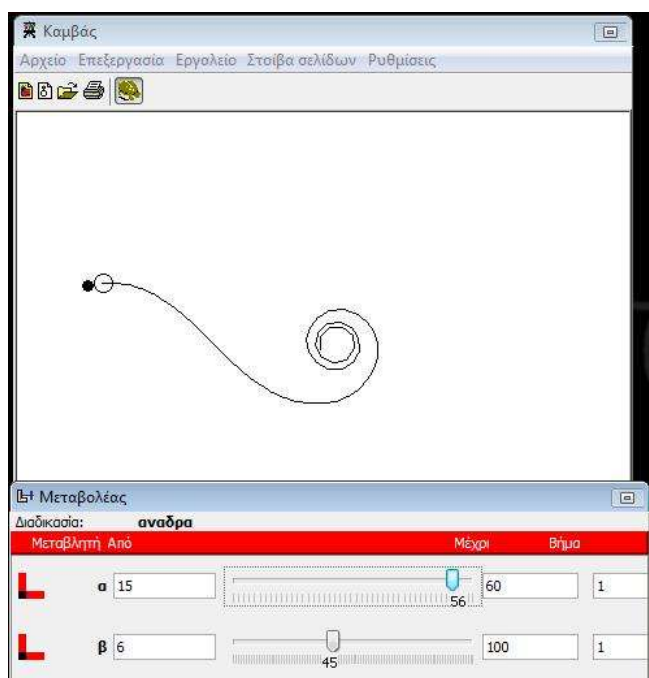
Οι μαθητές ξεκινούν τη διερεύνηση χρησιμοποιώντας τους μεταβολείς οι οποίοι αντιπροσωπεύουν τις παραμέτρους «α», «β».

Σύροντας τους ολισθητές μεταβάλλουν δυναμικά την κλίση της διαδικασίας και ταυτόχρονα παρατηρούν το αποτέλεσμα στον καμβά. Αν και ο τρόπος χρήσης της διαδικασίας είναι καινούργιος για τους μαθητές, η απλότητα του προγράμματος σε συνδυασμό με τη δυναμική αναπαράσταση που παρέχει ο χελωνόκοσμος, κάνουν εύκολο τον προσδιορισμό της ροής και των βημάτων του προγράμματος. Με τη βοήθεια των μεταβολών δημιουργείται ένα μεγάλο εύρος σχημάτων ξεκινώντας από καμπύλες γραμμές και καταλήγοντας σε σπειροειδείς μορφές.

Ενδεικτικά σχήματα που δημιουργούνται από την αναδρομή με τη χρήση των μεταβολών:



για $\alpha=60$, $\beta=13$, και



για $\alpha=56$, $\beta=45$.

Οι μαθητές καλούνται :

- Να αναγνωρίσουν το ρόλο των παραμέτρων α , β στο σχεδιαστικό αποτέλεσμα
- Να εντοπίσουν το είδος των αλλαγών που προκαλούνται σε κάθε κλήση της «αναδρα»
- Να κάνουν υποθέσεις για το τελικό σχήμα που θα προκύψει για κάποιο ζεύγος τιμών α , β
- Να εντοπίσουν τις κύριες διαφορές που εμφανίζουν τα σχήματα που δημιουργούνται από ακραίες τιμές α , β των μεταβολέων.
- Να εντοπίσουν αναλογίες-ομοιότητες με τη αντίστοιχη χρήση μιας επαναληπτικής διαδικασίας

Επιμέρους διδακτικοί στόχοι

Μετά το τέλος της διερεύνησης οι μαθητές θα έχουν τη δυνατότητα:

- Να περιγράψουν σε βήματα τη ροή εκτέλεσης των εντολών μιας αναδρομής
- Να διατυπώνουν υποθέσεις για τις αλλαγές που προκαλούνται στην πορεία της χελώνας κατά τη συνεχή επανάκληση της ίδιας διαδικασίας
- Να αναγνωρίζουν τη συνθήκη τερματισμού μιας αναδρομικής διαδικασίας

Φάση Β'

Στην προηγούμενη φάση οι μαθητές πειραματιζόμενοι με τους μεταβολείς διαπίστωσαν το μεγάλο εύρος σχημάτων που μπορούν να επιτύχουν με τη χρήση ενός πολύ μικρού και απλού προγράμματος. Σε αυτή τη φάση οι μαθητές χρησιμοποιούν την αναδρομή για να κατασκευάσουν πιο σύνθετα σχήματα ξεκινώντας από ένα γνωστό και απλό γεωμετρικό σχήμα: το εξάγωνο.

Αρχικά δίνεται προς διερεύνηση η διαδικασία κατασκευής ενός εξαγώνου.

Οι μαθητές αναγνωρίζουν τη γνωστή δομή της επανάληψης, εκτελούν την «εξαγ» και διερευνούν τη χρήση της παραμέτρου «α» με τη βοήθεια των μεταβολέων.

για εξαγ :α επανάλαβε 6 [μ :α δ 60] τέλος

Στη συνέχεια δίνεται μια νέα διαδικασία που κάνει χρήση της αναδρομής

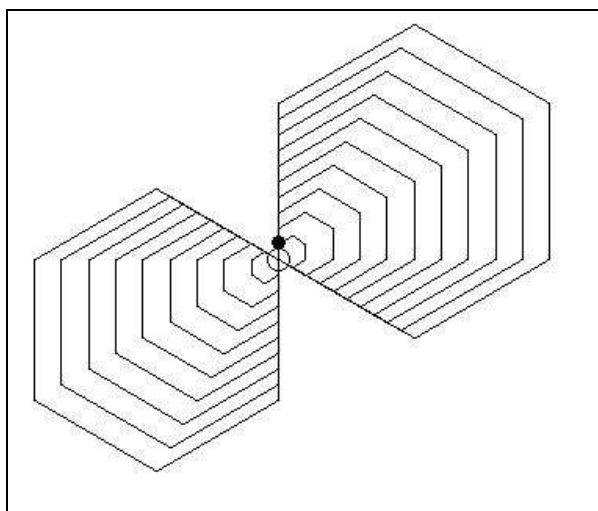
για πολυ :α :β
αν :α < 1 [σταμάτησε]
εξαγ :β
πολυ :α - 1 :β - 10
τύπωσε :β
τέλος

Οι μαθητές καλούνται :

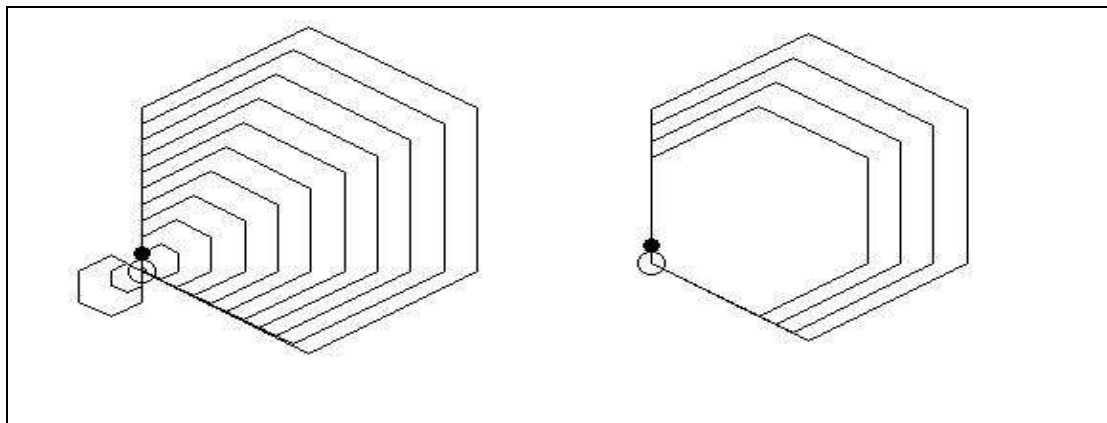
- Να αιτιολογήσουν ότι πρόκειται για αναδρομική διαδικασία
- Να κάνουν υποθέσεις για το σχήμα που θα δημιουργεί η διαδικασία
- Να ανακαλύψουν το ρόλο των παραμέτρων «α» και «β» στην κατασκευή του σχήματος

Καλώντας τη διαδικασία «πολυ» με αρχικές τιμές 20 και 100, οι μαθητές χρησιμοποιούν τους μεταβολείς για να διερευνήσουν τη νέα αναδρομική διαδικασία.

Το πρώτο σχήμα που κατασκευάζεται μοιάζει με πεταλούδα:



Με τη βοήθεια των μεταβολών κατασκευάζονται διαφορετικά σχήματα, με χαρακτηριστικά παραδείγματα τα παρακάτω:



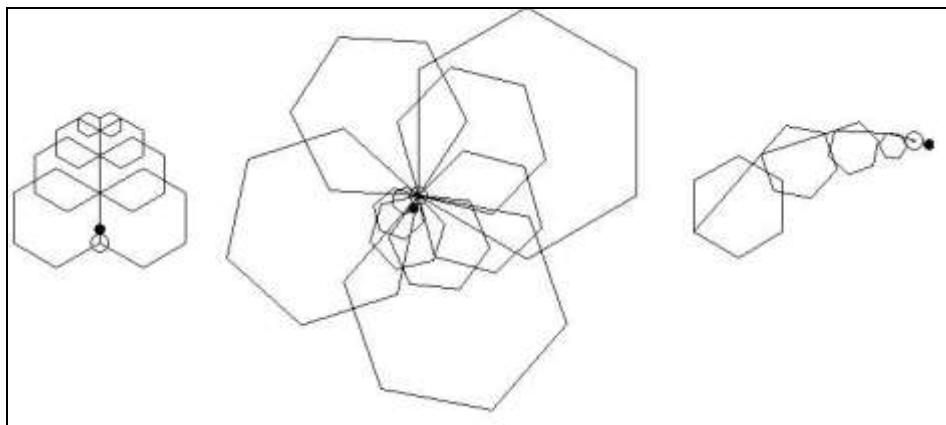
Ερωτήσεις που ανακύπτουν είναι:

- Τι ρόλο διαδραματίζει η κλήση της διαδικασίας «εξαγ»;
- Πώς «εξελίσσεται» η κατασκευή κατά τις διαδοχικές κλήσεις της διαδικασίας «πολύ»;
- Πότε το σχήμα αποκτά δύο σκέλη (όπως στην πεταλούδα);
- Ποιες είναι οι ελάχιστες τιμές α , β που σχηματίζουν το βασικό δομικό γεωμετρικό σχήμα;

Για να απαντήσουν στα παραπάνω ερωτήματα οι μαθητές εκτός από τη χρήση των μεταβολών, έχουν τη δυνατότητα να παρατηρούν και να συγκρίνουν τις ενδιάμεσες τιμές της «β» (και προαιρετικά της «α») με τη χρήση της εντολής «τύπωσε :β».

Ως τελική δραστηριότητα ζητείται από τους μαθητές να «χαλάσουν» τη διαδικασία «πολυ» με σκοπό να κατασκευάσουν με τη βοήθεια της αναδρομής το δικό τους μοναδικό σχήμα. Έτσι αλλάζοντας τις τιμές κατά την κλήση της διαδικασίας ή προσθέτοντας νέα βήματα για τη χελώνα (π.χ. μ : β) το αποτέλεσμα στον καμβά είναι εντυπωσιακό. Επεκτείνοντας τις δυνατότητες της αναδρομής, μπορούν να αλλάξουν το δομικό στοιχείο των σχημάτων βάζοντας στη θέση του το τρίγωνο ή το τετράγωνο.

Ενδεικτικά μπορούν να σχηματιστούν:



Φάση Γ΄

Στη φάση αυτή οι μαθητές αξιοποιούν την έννοια της αναδρομής όπως αυτή «ανακαλύφτηκε» στις προηγούμενες φάσεις, με σκοπό το σχηματισμό ενός δέντρου.

Πρόσθετη βοήθεια σε αυτό το σημείο μπορεί δοθεί μέσα από τη συζήτηση στην τάξη και την καταγραφή των χαρακτηριστικών που πρέπει να έχει ένα «ρεαλιστικό» δέντρο.

Στα χαρακτηριστικά αυτά περιλαμβάνονται:

- Η σύνθεση του δέντρου από πολλά ίδια κλαδιά
- Η σύνθεση ενός κλαδιού από συνεχόμενους «κορμούς» με σταθερό ή προοδευτικά μειούμενο μήκος

Η σχηματική αναπαράσταση του δέντρου σε χαρτί και η διαπραγμάτευση των παραπάνω χαρακτηριστικών μέσα στην ομάδα, μπορεί να «διευκολύνει» τους μαθητές στο σχηματισμό της κατάλληλης αναδρομικής διαδικασίας.

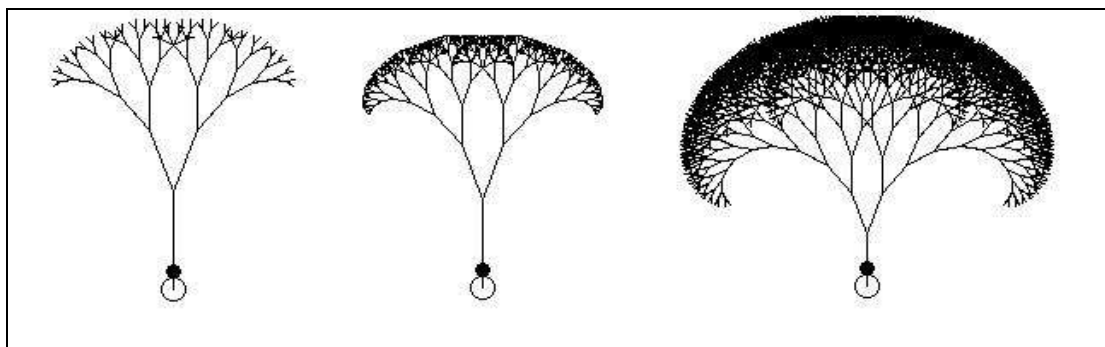
Ο κύριος διδακτικός στόχος σε αυτή τη φάση είναι η συνδυασμένη χρήση απλών εντολών και αναδρομής για την κατασκευή ενός πολύπλοκου σχήματος. Έτσι η επιτυχία σε αυτήν τη φάση δεν εξαρτάται από το σχηματισμό ενός τέλειου και μοναδικού δέντρου. Κάθε σχηματισμός που αξιοποιεί την αναδρομή και προσεγγίζει τα χαρακτηριστικά ενός δέντρου θεωρείται αποδεκτός.

Ενδεικτική λύση που θεωρεί ότι ο σχηματισμός των κλαδιών σταματά κάτω από ένα ελάχιστο όριο μήκους είναι η ακόλουθη:

για δέντρο :a
αν :a < 4 [σταμάτησε]
μ :a
α 20
δέντρο :a*2/3
δ 40
δέντρο :a*2/3
α 20
π :a
τέλος

σβγ
δέντρο 30

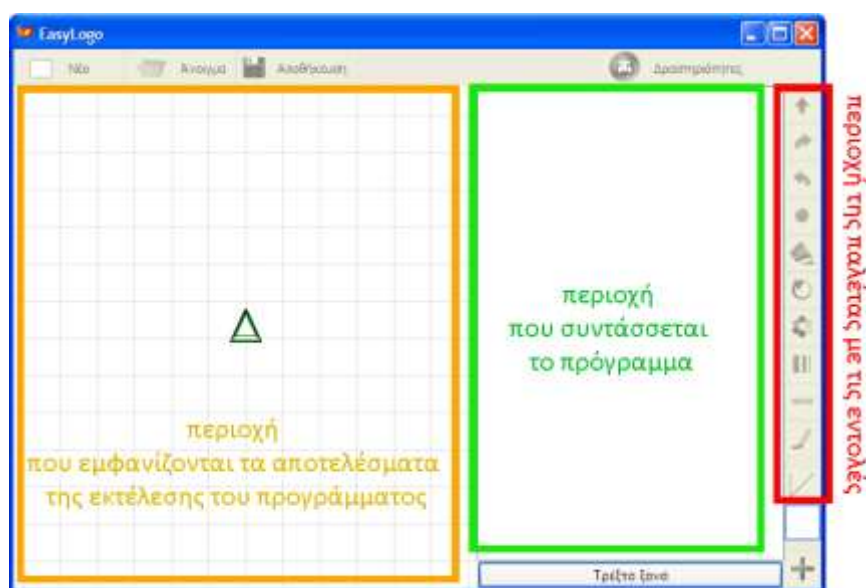
Στην παραπάνω αναδρομική διαδικασία το μήκος των κλαδιών ξεκινά από το 30 και μειώνεται συνεχώς κατά $\frac{2}{3}$ μέχρι να γίνει 4 οπότε σταματά η σχεδίαση. Μια πρόσθετη διερεύνηση που θα μπορούσε να γίνει με την ίδια διαδικασία αφορά στη διαδοχική μείωση του μήκους καθώς και στη συνθήκη τερματισμού της αναδρομής. Έτσι, με τη διαδικασία «δέντρο» μπορούμε να έχουμε τα παρακάτω ενδεικτικά αποτελέσματα:



ΠΑΡΑΡΤΗΜΑ 2: Easy Logo

Η EasyLogo, είναι μια δημιουργία του Department of Informatics Education, Comenius University of Bratislava και διατίθεται δωρεάν για εκπαιδευτικούς σκοπούς από τη διεύθυνση <http://edi.fmph.uniba.sk/~salanci/EasyLogo/index.html>.

Από εκεί μπορείτε να διαλέξετε την ελληνική έκδοση και να τη χρησιμοποιήσετε στον Η.Υ. (μπορεί να χρησιμοποιηθεί με απευθείας εκτέλεση του αρχείου EasyLogo.exe χωρίς εγκατάσταση). Το περιβάλλον της εμφανίζεται ως εξής:

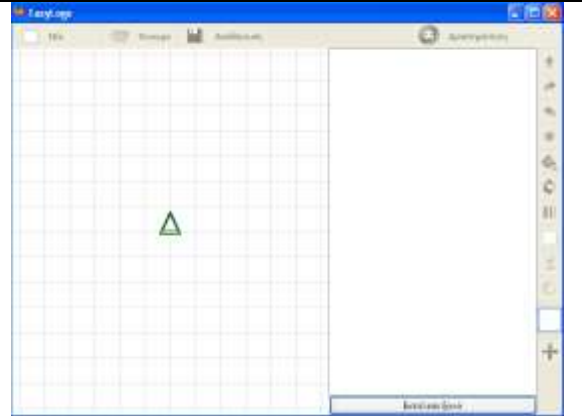
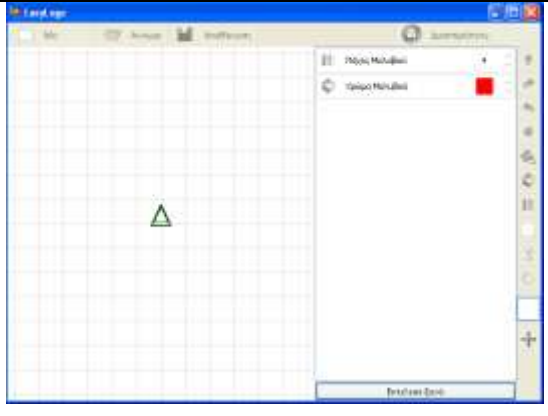
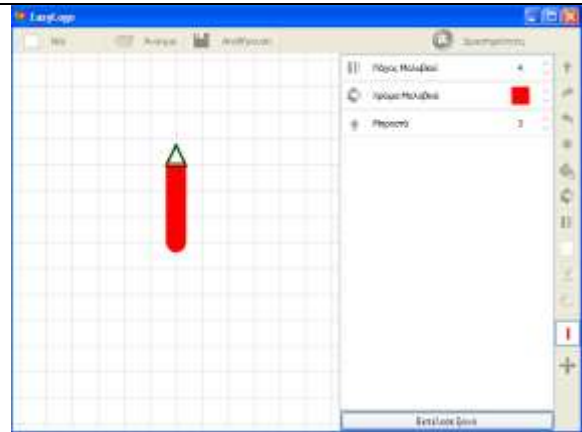
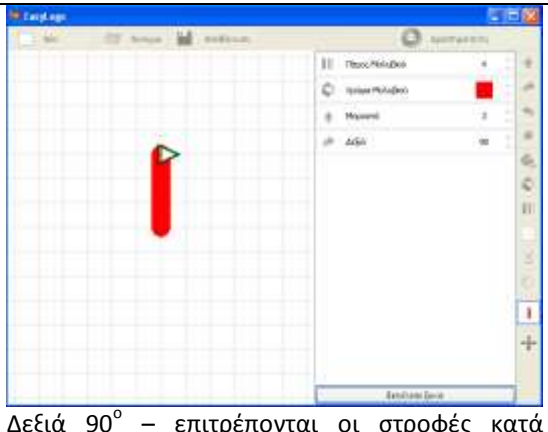
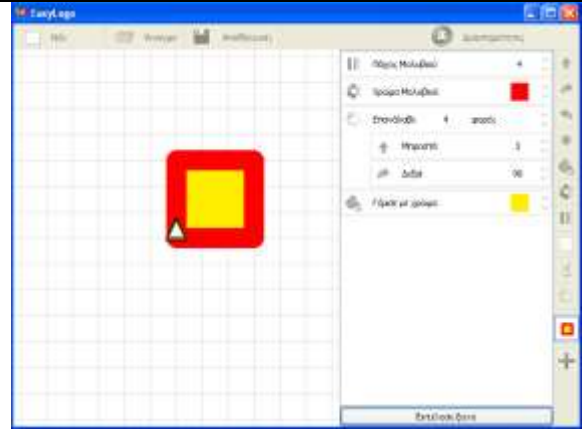
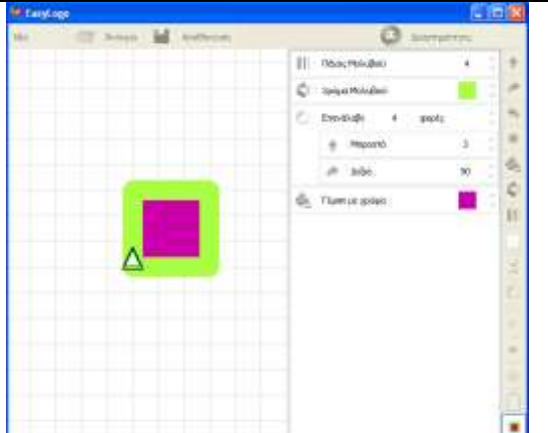


Η EasyLogo έχει μερικά διδακτικά χαρακτηριστικά που, κατά κάποιο τρόπο, την τοποθετούν «ενδιάμεσα» στα περιβάλλοντα με πλήρη απουσία γραπτών εντολών και στα περιβάλλοντα που χρησιμοποιούν τις γραπτές εντολές.

Διαθέτει εικονίδια τα οποία λειτουργούν ως ισοδύναμα εντολών κι επομένως δεν απαιτείται η σύνταξη εντολών, αφού το σύστημα δημιουργεί μόνο του το αντίστοιχο κείμενο κάθε εντολής ανάλογα με το εικονίδιο που επιλέγει ο κάθε χρήστης.

Επιπλέον, οι εντολές εκτελούνται ταυτόχρονα με τη «δημιουργία» τους και κατά την έννοια αυτή, υπάρχει άμεση συσχέτιση εντολών και αποτελέσματος.

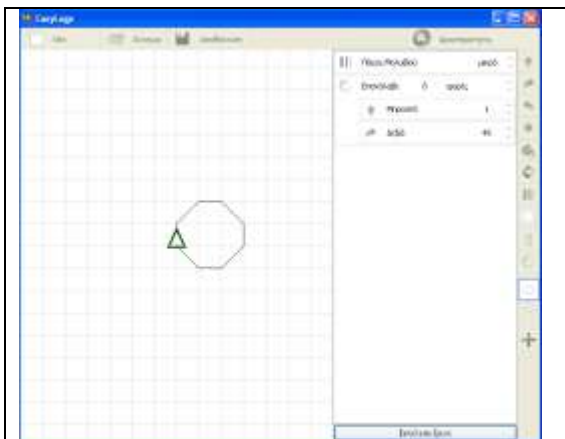
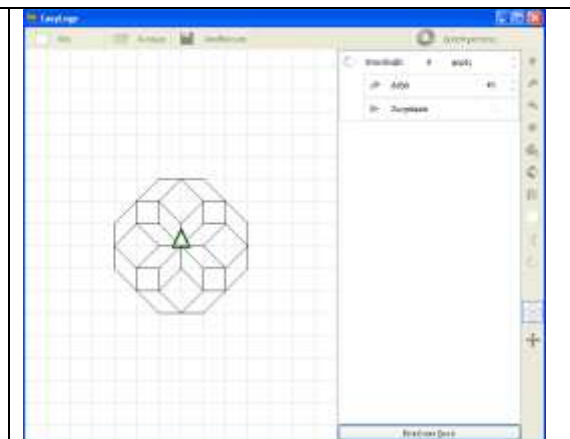
Στον παρακάτω πίνακα παρουσιάζονται τα αποτελέσματα 6 διαδοχικών εντολών:

	
<p>Καμία εντολή</p>	<p>Έχει προσδιοριστεί πάχος γραμμής 4 και χρώμα κόκκινο (με μετακίνηση των αντίστοιχων εικονιδίων)</p>
	
<p>Μπροστά 3</p>	<p>Δεξιά 90° – επιτρέπονται οι στροφές κατά πολλαπλάσια των 45°</p>
	
<p>Με την κατάλληλη επαναληπτική δομή και τον χρωματισμό της «εσωτερικής» περιοχής δημιουργείται ένα «έγχρωμο» τετράγωνο</p>	<p>Η αλλαγή στα χρώματα (στις παραμέτρους των εντολών στην περιοχή όπου συντάσσεται το πρόγραμμα) προκαλεί άμεση αλλαγή στο σχήμα</p>

Δυο πολύ σημαντικά χαρακτηριστικά του περιβάλλοντος EasyLogo είναι τα ακόλουθα:

Η EasyLogo επιτρέπει τη δημιουργία διαδικασιών. Κάθε διαδικασία που δημιουργείται αντιπροσωπεύεται στην «εργαλειοθήκη» της EasyLogo από ένα μικρό εικονίδιο που δημιουργείται επί τούτου (στην πραγματικότητα μια μικροσκοπική αναπαράσταση του σχήματος που δημιουργείται από τη «χελώνα» στην περιοχή όπου εμφανίζονται τα αποτελέσματα εκτέλεσης του προγράμματος). Ένα ειδικό σύμβολο («+») επιτρέπει τη δημιουργία νέων διαδικασιών.

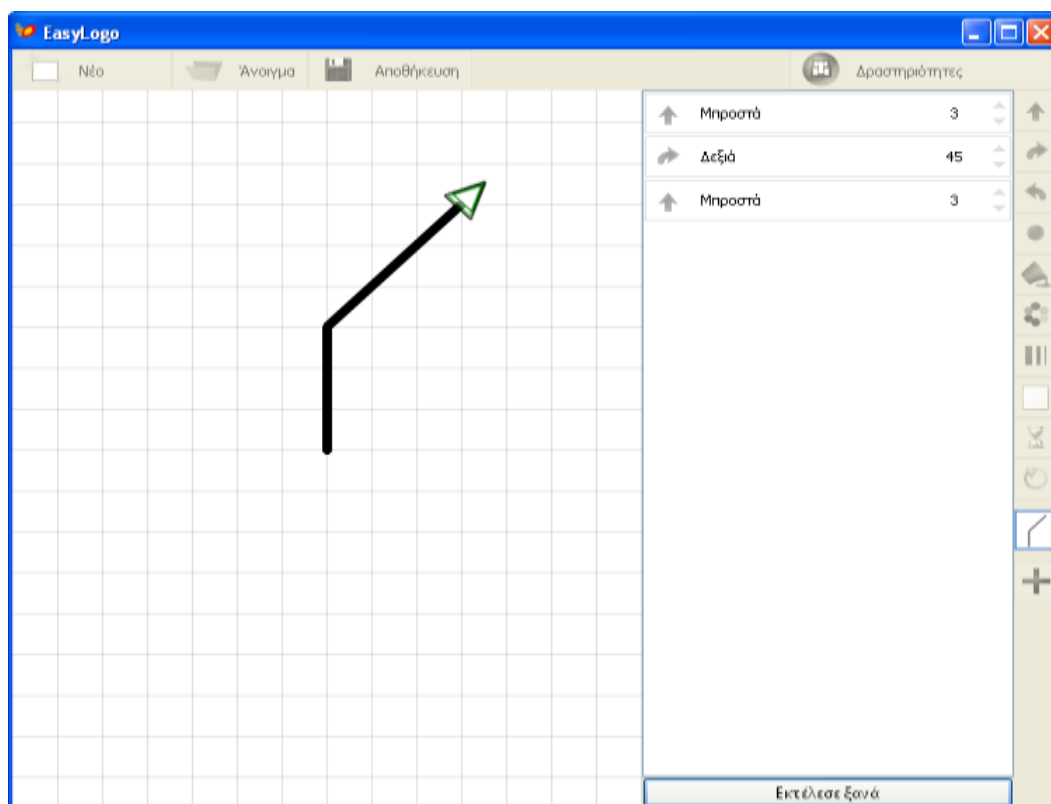
Κάθε νέα διαδικασία στην EasyLogo μπορεί να καλεί ήδη υφιστάμενες διαδικασίες απλώς μετακινώντας το σχετικό εικονίδιο στην κατάλληλη περιοχή (δηλαδή στην περιοχή όπου συντάσσεται το πρόγραμμα). Αυτός ο τρόπος κλήσης μιας διαδικασίας δια-της-μετακίνησης-ενός-εικονιδίου προφανώς εντάσσεται μέσα στη λογική του γραφιστικού/εικονικού περιβάλλοντος (και της διεπαφής). Το χαρακτηριστικό αυτό είναι σημαντικό από διδακτική σκοπιά, καθώς παρέχει τη δυνατότητα διαχείρισης (δηλαδή δημιουργίας, επεξεργασίας, κλήσης κλπ) διαδικασιών με έναν διαισθητικό τρόπο που φαίνεται να είναι άμεσα κατανοητός. Έτσι οι μαθητές μπορούν με ένα «φυσικό» τρόπο να μάθουν να δημιουργούν «αντικείμενα» (για παράδειγμα: σχήματα) συνθέτοντας δημιουργικά άλλα, απλούστερα.

	
<p>Δημιουργείται η διαδικασία «οκτάγωνο» (χωρίς όνομα – μόνο το εικονίδιο)</p>	<p>Μια νέα διαδικασία «καλεί» την προηγούμενη (οκτάγωνο) απλώς με την εντολή «Ζωγράφισε» που δημιουργείται αυτόματα όταν το εικονίδιο της διαδικασίας «οκτάγωνο» μετακινηθεί στην περιοχή όπου συντάσσεται το πρόγραμμα</p>

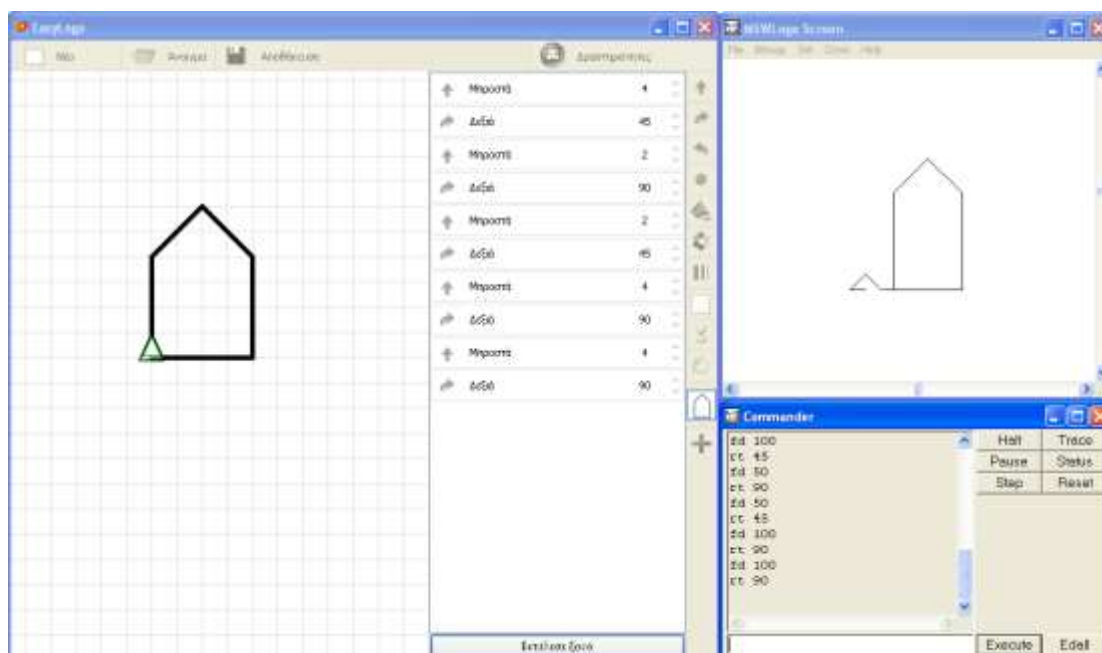
Ένα δεύτερο ιδιαίτερο χαρακτηριστικό της EasyLogo είναι η κάπως ιδιαίτερη «γεωμετρία» της. Ένα πρόγραμμα εκτελεσμένο στο περιβάλλον της EasyLogo και στο περιβάλλον της Logo, δεν δίνει πάντοτε το ίδιο αποτέλεσμα εξαιτίας του γεγονότος ότι τα δυο αυτά περιβάλλοντα δεν έχουν τους ίδιους γεωμετρικούς κανόνες.

Καθώς η «χελώνα» της EasyLogo κινείται πάντοτε σε κόμβους της υποκείμενης «σχάρας» τετραγωνιδίων, το πραγματικό μήκος του βήματος της διαφέρει, ανάλογα με τον προσανατολισμό της. Έτσι ένα βήμα σε κατεύθυνση παράλληλη προς το πλαίσιο της οθόνης

(δηλαδή «κατακόρυφα» ή «οριζόντια») την μετακινεί όσο είναι το μήκος της πλευράς του τετραγώνου. Αν όμως η χελώνα είναι στραμμένη προς άλλη κατεύθυνση, τότε το βήμα της έχει μήκος όσο η διαγώνιος του τετραγώνου.



Οι ίδιες εντολές σε δυο διαφορετικά περιβάλλοντα δεν έχουν το ίδιο αποτέλεσμα:



Θα μπορούσε άραγε το χαρακτηριστικό αυτό να έχει διδακτικές συνέπειες στον τρόπο με τον οποίο οι μαθητές και μαθήτριες αντιλαμβάνονται τη γεωμετρία και τις σχέσεις μεταξύ των μηκών;

Ενότητα Σεναρίου 7

Εννοιολογική χαρτογράφηση

Εννοιολογική Χαρτογράφηση

1. ΤΙΤΛΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Εννοιολογική χαρτογράφηση – χρήση του CmapTools για τη χαρτογράφηση εννοιών των Δικτύων Υπολογιστών

Σημείωση: Το προτεινόμενο διδακτικό σενάριο βασίζεται στο σενάριο “CmapTools: Χαρτογραφώντας έννοιες των Δικτύων Υπολογιστών της Πληροφορικής Γυμνασίου” (σελ. 367-386) που περιλαμβάνεται στο: *Επιμορφωτικό υλικό για την εκπαίδευση των επιμορφωτών στα Πανεπιστημιακά Κέντρα Επιμόρφωσης - Προτεινόμενα εκπαιδευτικά σενάρια: κλάδος ΠΕ19-20, Β' έκδοση, Πάτρα, Νοέμβριος 2011, ΙΤΥΕ “ΔΙΟΦΑΝΤΟΣ”.*

2. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το διδακτικό σενάριο προβλέπεται να διαρκέσει συνολικά 6 διδακτικές ώρες. Διατίθενται 3 διδακτικές ώρες για τη διδασκαλία των εννοιών και 3 διδακτικές ώρες για την αξιολόγηση (ομότιμη και συνεργατική).

Στην περίπτωση που οι μαθητές δεν είναι εξοικειωμένοι με την εννοιολογική χαρτογράφηση και το λογισμικό CmapTools είναι απαραίτητη η χρήση κάποιων επιπλέον εισαγωγικών δραστηριοτήτων.

Τέλος, είναι προφανές ότι μπορεί ο εκπαιδευτικός να επιλέξει να χρησιμοποιήσει μέρος των προτεινόμενων δραστηριοτήτων τόσο για τη διδασκαλία όσο και την αξιολόγηση, γεγονός που θα επηρεάσει την εκτιμώμενη διάρκεια του διδακτικού σεναρίου.

3. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ/ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Το προτεινόμενο διδακτικό σενάριο μπορεί να εκπονηθεί από μαθητές της Β' Γυμνασίου στο πλαίσιο του άξονα μαθησιακών στόχων "Χειρίζομαι και δημιουργώ με τα εργαλεία ΤΠΕ - Βασικές έννοιες ΤΠΕ", σύμφωνα με το Νέο Πρόγραμμα Σπουδών για τον Πληροφορικό Γραμματισμό στο Γυμνάσιο (Τζιμογιάννης κ.ά., 2011). Συγκεκριμένα, στόχος των δύο δραστηριοτήτων του σεναρίου που περιγράφονται στο φύλλο εργασίας είναι η διδασκαλία εννοιών των Δικτύων Υπολογιστών μέσα από τη συμπλήρωση, επέκταση, σχολιασμό,

αξιολόγηση και κατασκευή εννοιολογικών χαρτών. Επίσης, οι χάρτες των δραστηριοτήτων θα μπορούσαν να αξιοποιηθούν σε περιπτώσεις επανάληψης των συγκεκριμένων εννοιών ή ως δραστηριότητες προκαταρκτικής αξιολόγησης σε επόμενες τάξεις (Γυμνασίου ή/και Λυκείου). Στο πλαίσιο αυτό, οι εννοιολογικοί χάρτες θα μπορούσαν να κατασκευαστούν από τους μαθητές με δεδομένη μία λίστα εννοιών ή/και συνδέσμων.

4. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Ο σκοπός του συγκεκριμένου σεναρίου είναι διπλός:

- Να παρουσιάσει τα βασικά στοιχεία της τεχνικής της εννοιολογικής χαρτογράφησης στους επιμορφούμενους.
- Να παρουσιάσει ένα παράδειγμα εφαρμογής της εννοιολογικής χαρτογράφησης για την κατανόηση εννοιών των Δικτύων Υπολογιστών μέσα από ένα σύνολο σχετικών δραστηριοτήτων που αξιοποιούν το εργαλείο CmapTools.

Όσον αφορά στην παρουσίαση της τεχνικής της εννοιολογικής χαρτογράφησης παρουσιάζονται:

- τα βασικά στοιχεία ενός εννοιολογικού χάρτη, οι διάφοροι τύποι εννοιολογικών χαρτών, τα διδακτικά/μαθησιακά οφέλη της εννοιολογικής χαρτογράφησης και οι τύποι δραστηριοτήτων που μπορούν να σχεδιαστούν (*Ενότητα 6: Επιστημολογική προσέγγιση και εννοιολογική ανάλυση – Θέματα θεωρίας του διδακτικού σεναρίου*)
- οι δυνατότητες που παρέχουν τα σύγχρονα εργαλεία (λογισμικά) εννοιολογικής χαρτογράφησης (*Ενότητα 7: Χρήση Η/Υ και γενικά ψηφιακών μέσων για το διδακτικό σενάριο*)
- οι δυσκολίες που αντιμετωπίζουν οι μαθητές με την εννοιολογική χαρτογράφηση (*Ενότητα 8: Αναπαραστάσεις των μαθητών/Πρόβλεψη δυσκολιών στο διδακτικό σενάριο*).

Είναι λοιπόν προφανές ότι το υλικό των ενοτήτων 6, 7 και 8 αναφέρεται στην τεχνική της εννοιολογικής χαρτογράφησης γενικά, η οποία μπορεί να εφαρμοστεί για τη διδασκαλία οποιουδήποτε αντικειμένου και δεν αναφέρεται μόνο στο συγκεκριμένο διδακτικό σενάριο για τα Δίκτυα Υπολογιστών.

Όσον αφορά στο διδακτικό σενάριο εφαρμογής της εννοιολογικής χαρτογράφησης για τη διδασκαλία εννοιών των Δικτύων Υπολογιστών οι διδακτικοί στόχοι είναι οι μαθητές να μπορούν:

- να αναγνωρίζουν και να περιγράφουν τα βασικά δομικά στοιχεία ενός δικτύου υπολογιστών,

- να περιγράφουν τα πλεονεκτήματα της χρήσης δικτύων υπολογιστών στην καθημερινή ζωή καθώς και τα μειονεκτήματά τους,
- να διακρίνουν τα είδη των δικτύων βάσει διαφορετικών κατηγοριοποιήσεων,
- να ορίζουν άξονες και κριτήρια αξιολόγησης για την αξιολόγηση της εργασίας τους και να συντάσσουν μία φόρμα αξιολόγησης,
- να αξιολογούν μία εργασία σύμφωνα με τους άξονες και τα κριτήρια που έχουν ορίσει και να παρέχουν κατάλληλη ανατροφοδότηση για τη βελτίωση της, και
- να κρίνουν την ανατροφοδότηση που λαμβάνουν για την εργασία τους από τους συμμαθητές τους, υιοθετώντας ή απορρίπτοντας τις προτάσεις που τους παρέχονται για τη βελτίωσή της.

5. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το σενάριο αποτελείται από τις ακόλουθες 2 διδακτικο-μαθησιακές δραστηριότητες:

(1) Βασικές Έννοιες των Δικτύων Υπολογιστών - Κατηγορίες των Δικτύων Υπολογιστών.

Χρονική διάρκεια: 3 διδακτικές ώρες

Αρχικά, μέσα από την κατασκευή ενός εννοιολογικού χάρτη ή τις απαντήσεις των μαθητών σε δοσμένες ερωτήσεις, επιχειρείται η διερεύνηση της πρότερης γνώσης τους σχετικά με τις έννοιες που πρόκειται να διδαχθούν. Στη συνέχεια, οι μαθητές, ακολουθώντας τα βήματα των δύο δραστηριοτήτων του φύλλου εργασίας καλούνται να μελετήσουν τις έννοιες και το εκπαιδευτικό υλικό που παρουσιάζεται στους δύο εννοιολογικούς χάρτες αντίστοιχα (δίνονται στο Παράρτημα), να συμπληρώσουν τις έννοιες των χαρτών που απεικονίζονται με “?????”, να αναζητήσουν επιπλέον εκπαιδευτικό υλικό στο Διαδίκτυο εμπλουτίζοντας/σχολιάζοντας τους χάρτες και να τους επεκτείνουν με νέες έννοιες σύμφωνα με τις οδηγίες και τις καθοδηγητικές ερωτήσεις που έχουν στη διάθεσή τους. Μετά την επέκταση και το σχολιασμό των χαρτών πραγματοποιείται συζήτηση στην ολομέλεια της τάξης, σχετικά με τις απαντήσεις των μαθητών στα βήματα των δραστηριοτήτων. Τέλος, ανατίθεται για το σπίτι μια δραστηριότητα κατασκευής εννοιολογικού χάρτη με κεντρική έννοια "Δίκτυα Υπολογιστών" και με δοσμένη λίστα εννοιών.

(2) Αξιολογώντας τους χάρτες των συμμαθητών μου.

Χρονική διάρκεια: 3 διδακτικές ώρες

Στο πλαίσιο της συνεργατικής αξιολόγησης, οι μαθητές μαζί με τον εκπαιδευτικό καλούνται να ορίσουν τους άξονες και τα κριτήρια αξιολόγησης της εργασίας που έχει δοθεί ως εργασία για το σπίτι (κατασκευή εννοιολογικού χάρτη με δοσμένη λίστα εννοιών), να συντάξουν μία

φόρμα αξιολόγησης και να αξιολογήσουν έναν ενδεικτικό χάρτη στην ολομέλεια της τάξης (ο χάρτης μπορεί να έχει κατασκευαστεί από τον εκπαιδευτικό). Στη συνέχεια, καλούνται να αξιολογήσουν δύο εργασίες (χάρτες) των συμμαθητών τους, παρέχοντας κατάλληλη ανατροφοδότηση και να προβούν σε διορθώσεις της εργασίας τους σύμφωνα με τα σχόλια αξιολόγησης των συμμαθητών τους.

6. ΕΠΙΣΤΗΜΟΛΟΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΝΝΟΙΟΛΟΓΙΚΗ ΑΝΑΛΥΣΗ – ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Στην ενότητα αυτή παρουσιάζονται θέματα που αφορούν στην τεχνική της εννοιολογικής χαρτογράφησης, η οποία αποτελεί το βασικό διδακτικό εργαλείο του σεναρίου που αφορά στα Δίκτυα Υπολογιστών.

Η έννοια του χάρτη εννοιών/εννοιολογικού χάρτη

Ο **χάρτης εννοιών** (concept map) είναι ένα σχηματικό διάγραμμα που προσδιορίζει σχέσεις μεταξύ των βασικών εννοιών μιας περιοχής μελέτης με τη μορφή προτάσεων. Είναι μια τεχνική διδασκαλίας και μια στρατηγική μάθησης που σκοπεύει να βοηθήσει τους μαθητές να μάθουν έννοιες με ουσιαστικό τρόπο. (Βασιλοπούλου, 2001: σελ. 27).

Η διαδικασία κατασκευής ενός χάρτη εννοιών, ή αλλιώς ενός εννοιολογικού χάρτη, ονομάζεται **εννοιολογική χαρτογράφηση** (concept mapping) και αναπτύχθηκε αρχικά από ερευνητική ομάδα με επικεφαλή τον J. D. Novak στο Πανεπιστήμιο Cornell. Όπως αναφέρεται στη βιβλιογραφία (Βαρδάκα κ.α., 2005; Γουλή κ.α., 2006; Φορτούνη & Φραγκάκη, 2003) βασικό στοιχείο της εννοιολογικής χαρτογράφησης αποτελεί η θεωρία του Ausubel για τη “μάθηση με νόημα”, η οποία αναφέρεται στην προσαρμογή των νέων εννοιών στις προϋπάρχουσες γνωστικές δομές των μαθητών και στοχεύει στην ενίσχυση της εποικοδομητικής και ουσιαστικής μάθησης (meaningful learning).

Συστατικά στοιχεία ενός εννοιολογικού χάρτη

Τα βασικά συστατικά στοιχεία ενός εννοιολογικού χάρτη (EX) είναι οι κόμβοι που αναπαριστούν τις έννοιες και οι σύνδεσμοι που προσδιορίζουν πως συνδέονται μεταξύ τους οι διάφορες έννοιες του χάρτη.

Κόμβοι (έννοιες): κάθε κόμβος του χάρτη αναπαριστά μια έννοια, η οποία αναγράφεται στην ετικέτα του κόμβου.

Οι έννοιες μπορεί να αναφέρονται (Γουλή κ.α., 2006):

- σε *αντικείμενα*, τα οποία περιγράφονται συνήθως με ουσιαστικά. Οι αντίστοιχοι χάρτες χαρακτηρίζονται ως *περιγραφικοί* (descriptive concept maps).

- σε *συμβάντα/γεγονότα*, τα οποία περιγράφονται συνήθως με ρήματα. Οι αντίστοιχοι χάρτες επικεντρώνονται σε αναπαραστάσεις του τρόπου που λειτουργεί ή συμβαίνει κάτι και χαρακτηρίζονται ως *επεξηγηματικοί* (explanatory concept maps).

Φυσικά, οι έννοιες μπορεί να αναφέρονται σε ένα σύνολο αντικειμένων και γεγονότων.

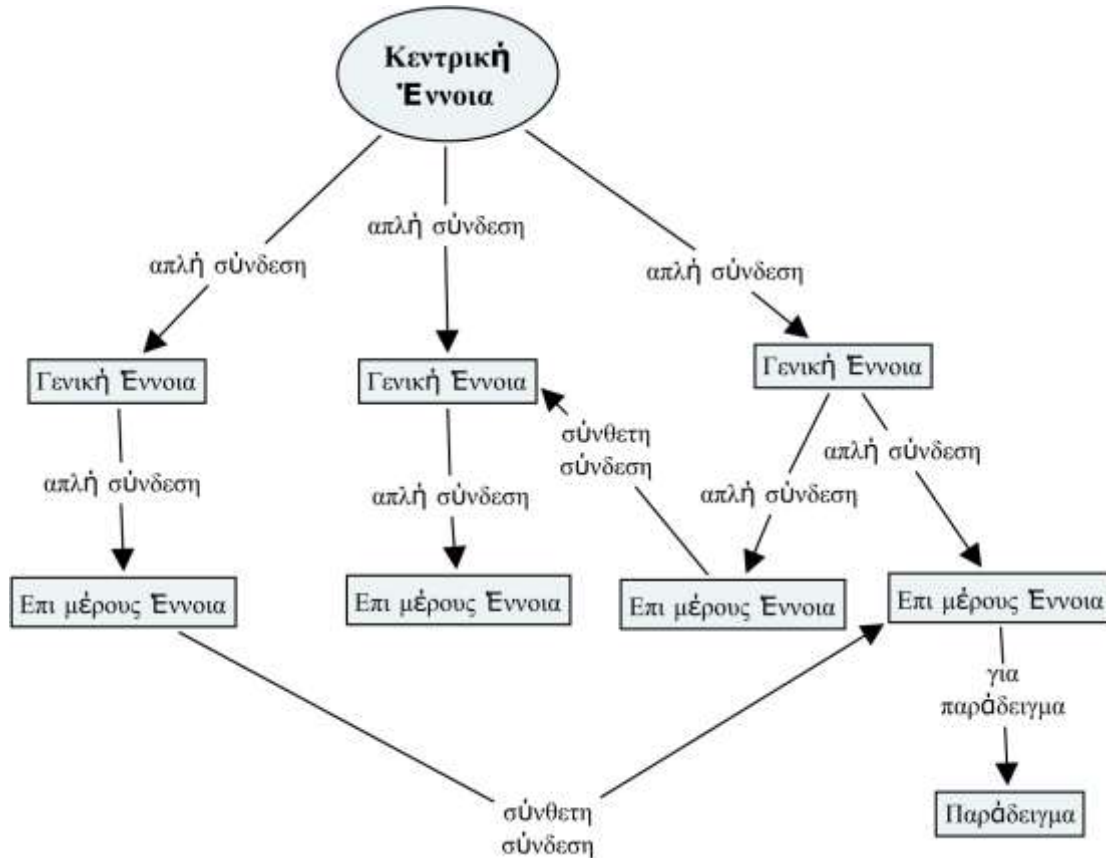
Η βασική έννοια ενός ΕΧ, η οποία αναλύεται σε ένα σύνολο εννοιών που την περιγράφουν, τοποθετείται συνήθως στην κορυφή του χάρτη και ονομάζεται *κεντρική έννοια*.

Σύνδεσμοι (συνδέσεις): οι σύνδεσμοι είναι γραμμές που συνδέουν μεταξύ τους δύο κόμβους και έχουν μία ετικέτα που με απλές και σύντομες συνδετικές λέξεις προσδιορίζει τη σχέση μεταξύ των συνδεόμενων κόμβων. Στην ουσία, οι σύνδεσμοι προσδιορίζουν τις σχέσεις μεταξύ των εννοιών του ΕΧ περιγράφοντας τον τρόπο με τον οποίο οι έννοιες αυτές συνδέονται.

Οι συνδέσεις μπορεί να είναι (Βασιλοπούλου, 2001):

- *απλές συνδέσεις* μεταξύ μια γενικότερης και μιας μερικότερης έννοιας
- *σύνθετες συνδέσεις* μεταξύ εννοιών που μπορεί βρίσκονται σε διαφορετικές περιοχές ή/και διαφορετικά επίπεδα του χάρτη, ή να είναι υπο-έννοιες διαφορετικών εννοιών.

Στην ουσία, η σύνδεση μεταξύ δύο κόμβων δημιουργεί μια **πρόταση**.



Είδη εννοιολογικών χαρτών

Οι ΕΧ διακρίνονται σε 4 βασικές κατηγορίες βάσει του σχηματοπλαισίου που χρησιμοποιούν (Αλιακίζογλου, 2009; Βαρδάκα κ.α., 2005):

Ιεραρχικοί (hierarchy concept maps): η κεντρική έννοια τοποθετείται στην κορυφή και προχωρώντας στα χαμηλότερα επίπεδα της ιεραρχίας αναλύεται στις επιμέρους έννοιες.

Αραχνοειδείς (spider concept maps): η κεντρική έννοια τοποθετείται στο κέντρο του χάρτη και γύρω από αυτή τοποθετούνται οι υπο-έννοιες.

Διαγράμματος ροής (flowchart concept maps): η κεντρική έννοια τοποθετείται στην κορυφή και ακολουθεί μια γραμμική διάταξη των υπο-εννοιών.

Συστημικοί (systems concept maps): οι έννοιες οργανώνονται με παρόμοιο τρόπο με τους χάρτες τύπου διαγράμματος ροής, αλλά επιπλέον στα άκρα τους έχουν εισροές-εκροές.

Ειδικοί τύποι ΕΧ αναφέρονται στο (Αλιακίζογλου, 2009).

Διδακτική/μαθησιακή αξία της εννοιολογικής χαρτογράφησης

Τα πλεονεκτήματα της διδακτικής αξιοποίησης τη εννοιολογικής χαρτογράφησης είναι ποικίλα τόσο για το μαθητή όσο και για το διδάσκοντα.

Τα σημαντικότερα πλεονεκτήματα για τους μαθητές, τα οποία συνοψίζονται και αναλύονται στην εργασία των (Φορτούνη & Φραγκάκη, 2003), είναι τα εξής:

- Ενεργητική – αποτελεσματική μάθηση
- Εργαλείο δημιουργικής σκέψης
- Εργαλείο ανάπτυξης και καλλιέργειας κριτικής σκέψης
- Εργαλείο μεταγνώσης
- Ενίσχυση ομαδοσυνεργατικής μάθησης
- Επίλυση προβλημάτων

Τα σημαντικότερα πλεονεκτήματα για τους διδάσκοντες είναι (Φορτούνη & Φραγκάκη, 2003):

- Μέσο οργάνωσης και παρουσίασης του μαθήματος στην τάξη.
- Διαγνωστικό εργαλείο για την ανίχνευση και αναπαράσταση των πρότερων γνώσεων των μαθητών.
- Εργαλείο αξιολόγησης της μάθησης και της εξέλιξης της γνωστικής αλλαγής μετά τη διδακτική παρέμβαση.

Οι ΕΧ έχουν χρησιμοποιηθεί σε σχετικές μελέτες από διάφορους ερευνητές, όπως επισημαίνεται στο (Γουλή κ.α., 2005; Γουλή κ.α, 2006), ως εργαλείο:

- διερεύνησης της πρότερης γνώσης των μαθητών
- διερεύνησης των αναπαραστάσεων των μαθητών για ένα συγκεκριμένο θέμα
- συνεργασίας
- εννοιολογικής αλλαγής και αξιολόγησης
- επίλυσης προβλημάτων

Τύποι δραστηριοτήτων εννοιολογικής χαρτογράφησης

- **Συμπλήρωση** ενός υπάρχοντος EX με έννοιες και συνδέσεις που λείπουν, αλλά για τις οποίες οι αντίστοιχοι κόμβοι και σύνδεσμοι είναι ήδη σχεδιασμένοι στον EX (δίνεται ένας EX με κενά)
- **Επέκταση** ενός υπάρχοντος EX με την προσθήκη εννοιών και συνδέσεων
- **Διόρθωση** ενός υπάρχοντος EX
- **Σχολιασμός** ενός EX είτε ελεύθερα είτε απαντώντας σε ερωτήσεις
- **Κατασκευή** ενός EX δίνοντας την κεντρική έννοια, ένα σύνολο εννοιών, ένα σύνολο εννοιών και συνδέσεων, μια ερώτηση ή ένα σχετικό κείμενο
- Συνδυασμός των παραπάνω

7. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Το διδακτικό σενάριο προβλέπεται να υλοποιηθεί στο εργαστήριο Πληροφορικής, στο οποίο θα είναι εγκατεστημένο το λογισμικό εννοιολογικής χαρτογράφησης CmapTools.

Είναι προφανές ότι το διδακτικό σενάριο μπορεί να υλοποιηθεί σχεδιάζοντας τους νοητικούς χάρτες με χαρτί και μολύβι. Αν και η χρήση ενός λογισμικού εννοιολογικής χαρτογράφησης μπορεί να απαιτεί (αρχικά τουλάχιστον και μέχρι οι μαθητές να εξοικειωθούν με τις λειτουργίες του) περισσότερο χρόνο, τα πλεονεκτήματα που προκύπτουν από τη χρήση του είναι πολλά:

- εύκολη διόρθωση ενός εννοιολογικού χάρτη, όπως: αναδιοργάνωση κόμβων, ετικετών και συνδέσεων ή δυναμική μετακίνηση κόμβων με ταυτόχρονη ενημέρωση των συνδέσεων
- μεγάλη επιφάνεια σχεδίασης του χάρτη
- δυνατότητα προσθήκης ποικίλων μορφών πληροφοριών σε ένα χάρτη, όπως εικόνες, κείμενα, σχόλια, σύνδεσμοι κ.τ.λ.

- δυνατότητα εξαγωγής ενός χάρτη σε μια ποικιλία διαφορετικών μορφών, όπως: εικόνα, γραφικά SVG, ιστοσελίδα, σύνολο προτάσεων (txt) κ.τ.λ.
- δυνατότητα διαμοιρασμού εννοιολογικών χαρτών και διεξαγωγής συνεργατικών δραστηριοτήτων.

Ένας μεγάλος αριθμός εργαλείων εννοιολογικής χαρτογράφησης είναι πλέον διαθέσιμος. Ενδεικτικά αναφέρονται τα εξής λογισμικά, τα οποία διατίθενται ελεύθερα:

- CmapTools: <http://cmap.ihmc.us/>
- Synergo: http://hci.ece.upatras.gr/index.php?option=com_content&task=view&id=103&Itemid=103
- COMPASS: <http://hermes.di.uoa.gr/compass/>

Για τη δημιουργία των εννοιολογικών χαρτών του συγκεκριμένου διδακτικού σεναρίου χρησιμοποιήθηκε το λογισμικό CmapTools.

8. ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ/ ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ

Στη συγκεκριμένη ενότητα παρουσιάζονται ενδεχόμενες δυσκολίες των μαθητών που αφορούν στην τεχνική της εννοιολογικής χαρτογράφησης και παρατηρούνται ανεξάρτητα από το γνωστικό αντικείμενο για το οποίο χρησιμοποιείται η τεχνική αυτή.

Οι μαθητές αντιμετωπίζουν δυσκολίες στην κατασκευή ΕΧ, οι οποίες εντοπίζονται κυρίως στα εξής (Βασιλοπούλου, 2001; Γουλή κ.α., 2006):

- Προσδιορισμό των εννοιών που θα συμπεριληφθούν στον χάρτη.
- Διατύπωση απλών, σύντομων και σαφών προτασιακών συνδέσεων μεταξύ των εννοιών.

Είναι απαραίτητη λοιπόν μια προκαταρκτική φάση, πριν οι μαθητές να ξεκινήσουν να σχεδιάζουν ΕΧ (Βασιλοπούλου, 2001). Αρχικά, πρέπει να εξηγηθεί στους μαθητές σε τι ακριβώς αναφέρεται ο όρος “έννοια” και ποια είναι η σημασία της, καθώς επίσης και να αποσαφηνιστεί ο τρόπος με τον οποίο εντοπίζονται οι έννοιες σε ένα κείμενο (συνήθως εμφανίζονται ως ουσιαστικά ή ουσιαστικά με προσδιοριστικά επίθετα). Στη συνέχεια, οι μαθητές πρέπει να εξασκηθούν στον εντοπισμό εννοιών, ζητώντας τους – για παράδειγμα – να εντοπίσουν τις έννοιες από μια λίστα λέξεων που τους δίνεται. Οι μαθητές πρέπει επίσης να εξασκηθούν μέσα στην τάξη στον καθορισμό απλών, σύντομων και σαφών συνδέσεων μεταξύ γνωστών εννοιών, ή και πιο δύσκολων.

Οι Γουλή κ.α. (2006) προτείνουν για την περίοδο εξοικείωσης των μαθητών με τις έννοιες

και τις προτασιακές συνδέσεις μεταξύ τους, δραστηριότητες που περιλαμβάνουν τη συμπλήρωση ή δημιουργία προτάσεων και τη συμπλήρωση-επέκταση-κατασκευή ενός χάρτη από μια λίστα εννοιών/συνδέσεων που δίνεται στους μαθητές. Χρήσιμες συμβουλές/πρακτικές για την αποδοτική χρήση των εννοιολογικών χαρτών παρέχονται επίσης στην εργασία (Τζελέπη, 2012).

Στην περίπτωση που οι μαθητές αξιοποιούν κάποιο λογισμικό για τη δημιουργία EX, όπως το CmapTools, θα πρέπει το σενάριο να εμπλουτιστεί με επιπλέον δραστηριότητες εξοικείωσης/διερεύνησης των λειτουργιών του λογισμικού.

Όσον αφορά στο συγκεκριμένο διδακτικό σενάριο που παρουσιάζεται για τα Δίκτυα Υπολογιστών και στην περίπτωση που οι μαθητές δεν έχουν εξοικειωθεί προηγουμένως με την εννοιολογική χαρτογράφηση και το λογισμικό CmapTools, θα πρέπει το σενάριο να εμπλουτιστεί με ανάλογες δραστηριότητες και οι οδηγίες στο φύλλο εργασίας να είναι περισσότερο καθοδηγητικές. Επίσης, θεωρείται δεδομένο ότι οι μαθητές έχουν αναπτύξει βασικές δεξιότητες στην αναζήτηση πληροφοριών και στη χρήση του Διαδικτύου.

9. ΔΙΔΑΚΤΙΚΟ ΣΥΜΒΟΛΑΙΟ – ΔΙΔΑΚΤΙΚΗ ΜΕΤΑΤΟΠΙΣΗ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ - ΔΙΔΑΚΤΙΚΟΣ ΘΟΡΥΒΟΣ

Στην περίπτωση που δεν υπάρχει προηγούμενη εμπειρία των μαθητών με την εννοιολογική χαρτογράφηση και το λογισμικό CmapTools, μπορεί να προκύψουν οι δυσκολίες που αναφέρονται στην ενότητα 8, με συνέπεια την αδυναμία επίτευξης των στόχων του διδακτικού σεναρίου στην εκτιμώμενη διάρκεια.

10. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Novak, J. D. & A. J. Cañas, The Theory Underlying Concept Maps and How to Construct and Use Them, Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition, 2008, available at: <http://cmap.ihmc.us/Publications/ResearchPapers/TheoryUnderlyingConceptMaps.pdf>

IHMC CmapTools, available at: <http://cmap.ihmc.us/>

My Ten Favorite Educational Maps, available at: <http://edudemic.com/2012/10/education-mind-maps/>

Αλιακίζογλου, Ε. (2009). Concept mapping: Αξιολόγηση μεθόδων και εργαλείων – Εφαρμογή σε ενδεικτικά παραδείγματα. *Μεταπτυχιακό Πρόγραμμα Γεωπληροφορικής, Σχολή Αγρονόμων και Τοπογράφων Μηχανικών, ΕΜΠ.*

Βαρδάκα, Μ., Βαρδάκας, Ε., Αλιμήσης, Δ. (2005). Η βασισμένη στον Η/Υ δημιουργία εννοιολογικών χαρτών και η διδακτική αξιοποίησή τους. *Πρακτικά 3ου Πανελληνίου Συνεδρίου “Αξιοποίηση των Τεχνολογιών Πληροφορίας και Επικοινωνίας στην Εκπαίδευση”, Σύρος 13-15 Μαΐου 2005, 562-568.*

Βασιλοπούλου, Μ. (2001). *Ο χάρτης εννοιών ως εργαλείο μάθησης, Εφαρμογές στη Διδακτική της Βιολογίας και την Περιβαλλοντική Εκπαίδευση*, Αθήνα.

Γουλή, Ε., Γόγουλου, Α., Παπανικολάου, Κ. και Γρηγοριάδου, Μ. (2005). Αξιοποιώντας τον εννοιολογικό χάρτη ως εργαλείο διδασκαλίας και αξιολόγησης στο μάθημα Πληροφορικής Γυμνασίου. Στο Α. Τζιμογιάννης (επιμ.), *Πρακτικά 3ου Πανελληνίου Συνεδρίου "Διδακτική της Πληροφορικής"*, Κόρινθος.,

Γουλή, Ε., Γόγουλου, Α., & Γρηγοριάδου, Μ. (2006). Ο Εννοιολογικός Χάρτης στην Εκπαιδευτική Διαδικασία του μαθήματος της Πληροφορικής: Μια Πιλοτική Διερεύνηση. *Θέματα στην Εκπαίδευση, Ειδικό Αφιέρωμα: Σύγχρονη έρευνα στη Διδακτική της Πληροφορικής*, 7:3, 351-377. Εκδόσεις Ελληνικά Γράμματα.

Τζελέπη, Σ. (2012). Χρήση Εννοιολογικών Χαρτών για τη Διδασκαλία της Θεματικής Ενότητας Δίκτυα Η/Υ της Β' Γυμνασίου. *Πρακτικά Ημερίδας Πληροφορικής του ΠΑΚΕ Κεντρικής Μακεδονίας "Η πληροφορική στην εποχή του Νέου Σχολείου"*, Θεσσαλονίκη 27 Μαρτίου 2012.

Τζιμογιάννης, Α. κ.ά. (2011). Πρόγραμμα Σπουδών για τον Πληροφορικό Γραμματισμό στο Γυμνάσιο. 2η Έκδοση, Μάρτιος 2011.

Φορτούνη, Τ. & Φραγκάκη, Μ. (2003). Εννοιολογική χαρτογράφηση: μια διδακτική παρέμβαση. *Πρακτικά 2ου Πανελληνίου Συνεδρίου "Αξιοποίηση των Τεχνολογιών Πληροφορίας και Επικοινωνίας στην Εκπαίδευση"*, Σύρος 9-11 Μαΐου 2003, 411-424.

11. ΥΠΟΚΕΙΜΕΝΗ ΘΕΩΡΙΑ ΜΑΘΗΣΗΣ

Ενεργητική συμμετοχή, εννοιολογική χαρτογράφηση, συνεργατική αξιολόγηση, ομότιμη αξιολόγηση.

12. ΕΠΙΣΗΜΑΝΣΗ ΜΙΚΡΟΜΕΤΑΒΟΛΩΝ

Δεν φαίνεται να έχουμε.

13. ΟΡΓΑΝΩΣΗ ΤΗΣ ΤΑΞΗΣ – ΕΦΙΚΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ

Στους υπολογιστές του εργαστηρίου Πληροφορικής είναι εγκατεστημένο το λογισμικό SmartTools και οι μαθητές έχουν στη διάθεσή τους, για την εκπόνηση των δραστηριοτήτων του σεναρίου, το βιβλίο της Πληροφορικής Γυμνασίου καθώς και τους χάρτες με το εκπαιδευτικό υλικό που παρουσιάζονται στο Παράρτημα. Θα ήταν επιθυμητό, ανάλογα με το πλήθος των μαθητών και τον αριθμό των διαθέσιμων υπολογιστών του εργαστηρίου, η αξιοποίηση ενός υπολογιστή ανά μαθητή με πρόσβαση στο Διαδίκτυο.

14. ΕΠΕΚΤΑΣΕΙΣ/ΔΙΑΣΥΝΔΕΣΕΙΣ ΤΩΝ ΕΝΝΟΙΩΝ Ή ΤΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Οι προτεινόμενες δραστηριότητες μπορούν να εμπλουτιστούν με την προσθήκη περισσότερων εννοιών που σχετίζονται με τα δίκτυα υπολογιστών, αλλά για τη σύνδεση με άλλες έννοιες ή και την παρουσίαση νέων εννοιών.

15. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

Ανάλυση φύλλου εργασίας 1

Στο πλαίσιο της συγκεκριμένης διδακτικο-μαθησιακής δραστηριότητας, οι μαθητές θα διδαχθούν έννοιες των Δικτύων Υπολογιστών με την αξιοποίηση του εννοιολογικού χάρτη ως εργαλείου διδασκαλίας.

Προτεινόμενη Χρονική διάρκεια: 3 διδακτικές ώρες.

Σε περίπτωση δίωρου μαθήματος Πληροφορικής, προτείνεται η Β και Γ επιμέρους δραστηριότητα να πραγματοποιηθούν σε ένα μάθημα και η Α να έχει προηγηθεί σε προηγούμενο μάθημα. Σε περίπτωση μαθήματος μίας διδακτικής ώρας προτείνεται η Β επιμέρους δραστηριότητα να πραγματοποιηθεί σε δύο μαθήματα (ένας εννοιολογικός χάρτης σε κάθε μάθημα).

Στόχοι: Οι μαθητές θα μπορούν να:

- αναγνωρίζουν και να περιγράφουν τα βασικά δομικά στοιχεία ενός δικτύου υπολογιστών,
- περιγράφουν τα πλεονεκτήματα της χρήσης δικτύων υπολογιστών στην καθημερινή ζωή καθώς και τα μειονεκτήματά τους,
- διακρίνουν τα είδη των δικτύων βάσει διαφορετικών κατηγοριοποιήσεων, και
- ορίζουν τις έννοιες ταχύτητα/ρυθμός μετάδοσης και πρωτόκολλα επικοινωνίας.

Απαιτούμενη υλικοτεχνική υποδομή

- Εργαστήριο υπολογιστών με πρόσβαση στο Διαδίκτυο και εγκατεστημένο το λογισμικό SmartTools.
- Οι εννοιολογικοί χάρτες με το εκπαιδευτικό υλικό (παρουσιάζονται στο Παράρτημα).
- Το φύλλο εργασίας των μαθητών

Επιμέρους Δραστηριότητες

Α. Διερεύνηση της πρότερης γνώσης των μαθητών (διάρκεια 30 λεπτά)

Γνωστοποίηση στους μαθητές από τον εκπαιδευτικό του θέματος και των στόχων του μαθήματος και διερεύνηση της πρότερης γνώσης τους σχετικά με τις υπό διδασκαλία έννοιες.

Για τη διερεύνηση της πρότερης γνώσης θα μπορούσε να δοθεί ως δραστηριότητα στους

μαθητές η κατασκευή ενός εννοιολογικού χάρτη με δεδομένο σύνολο εννοιών (περιορίζοντας το γνωστικό πεδίο και παρέχοντας στους μαθητές καθοδήγηση για τις έννοιες που θα πρέπει να απεικονίσουν) ή η ελεύθερη κατασκευή του χάρτη με έννοιες που θεωρούν οι μαθητές απαραίτητες ότι περιγράφουν την κεντρική έννοια "Δίκτυα Υπολογιστών".

Δραστηριότητα Διερεύνησης Πρότερης Γνώσης

Να κατασκευάσετε (με "χαρτί - μολύβι" ή με τη βοήθεια του λογισμικού CmapTools) έναν εννοιολογικό χάρτη με κεντρική έννοια την έννοια «Δίκτυα Υπολογιστών» χρησιμοποιώντας όσες από τις έννοιες γνωρίζετε από την ακόλουθη λίστα εννοιών και ορίζοντας κατάλληλους συνδέσμους.

Διαθέσιμη Λίστα Εννοιών

Κόμβοι	Ζεύξεις Επικοινωνίας	Υπολογιστές
Δρομολογητής	Ταχύτητα/Ρυθμός Μετάδοσης	Αρτηρία
Πρωτόκολλο	Ενσύρματα Δίκτυα	Ασύρματα Δίκτυα
Δακτύλιος	Δίκτυο Ευρείας Περιοχής	Γραμμές χαλκού
Οπτική Ίνα	Μητροπολιτικό δίκτυο	Αστέρας
Τοπικό δίκτυο	Γεωγραφική διασπορά	Διαδίκτυο
Ασφάλεια	Περιφερειακές Συσκευές	Ιοί
Δορυφόροι	Ανταλλαγή δεδομένων	Διαμοιρασμός πόρων

Στην περίπτωση που οι μαθητές δεν είναι εξοικειωμένοι με την τεχνική της εννοιολογικής χαρτογράφησης, θα μπορούσε να τους ζητηθεί να απαντήσουν γραπτά σε ανοικτές ερωτήσεις της μορφής: α) Ποια είναι τα βασικά δομικά στοιχεία ενός δικτύου υπολογιστών; β) Να αναφέρετε 2 λόγους που χρησιμοποιούμε τα δίκτυα υπολογιστών καθώς και 2 προβλήματα που μπορεί να δημιουργηθούν από τη χρήση τους. γ) Ποια είδη δικτύων γνωρίζετε με βάση τη γεωγραφική τους διασπορά; δ) Τι καλείται πρωτόκολλο επικοινωνίας; ε) Ποια μέσα σύνδεσης/ μετάδοσης μπορεί να χρησιμοποιηθούν σε ένα δίκτυο υπολογιστών; κ.λπ.

B. Χαρτογραφώντας έννοιες των Δικτύων Υπολογιστών (διάρκεια 90 λεπτά)

Οι μαθητές καλούνται να εκπονήσουν τις δύο δραστηριότητες του φύλλου εργασίας χρησιμοποιώντας το εργαλείο CmapTools (<http://cmap.ihmc.us>) και ακολουθώντας τα βήματα του φύλλου εργασίας. Η πρώτη δραστηριότητα αφορά την επέκταση, συμπλήρωση και σχολιασμό του εννοιολογικού χάρτη "Δίκτυα Υπολογιστών - Βασικές Έννοιες" και η δεύτερη δραστηριότητα αφορά την επέκταση, συμπλήρωση και σχολιασμό του

εννοιολογικού χάρτη "Κατηγοριοποίηση Δικτύων Υπολογιστών". Οι δύο χάρτες παρουσιάζονται στο Παράρτημα, Εικόνα 1 και Εικόνα 2 αντίστοιχα, και θα πρέπει χρησιμοποιώντας το εργαλείο SmartTools να κατασκευαστούν από τον εκπαιδευτικό. Επίσης, σε κάποιες από τις έννοιες των χαρτών έχει προστεθεί εκπαιδευτικό υλικό (εικόνες ή/και κείμενα) που επεξηγεί τις συγκεκριμένες έννοιες. Το εκπαιδευτικό υλικό που έχει δημιουργηθεί παρουσιάζεται στις Εικόνες του Παραρτήματος.

Οι εργασίες των βημάτων του φύλλου εργασίας θα πρέπει να πραγματοποιηθούν από τους μαθητές πάνω στους δύο χάρτες, αξιοποιώντας τις δυνατότητες του λογισμικού SmartTools. Οι νέοι χάρτες των μαθητών που θα προκύψουν από τις απαντήσεις τους στα βήματα του φύλλου εργασίας αποθηκεύονται σε κατάλογο που υποδεικνύει ο εκπαιδευτικός, μετά από σχετική καθοδήγηση του (αν απαιτείται).

Γ. Ανακεφαλαίωση & Ατομική Πρακτική - Εξάσκηση (διάρκεια 15 λεπτά)

Πραγματοποιείται από τους μαθητές ανακεφαλαίωση/σύνοψη των εννοιών και των σχέσεών τους που απεικονίζονται στους χάρτες που δημιουργήθηκαν στην επιμέρους δραστηριότητα Β και επεξηγείται η εργασία που έχουν να εκπονήσουν στο σπίτι οι μαθητές, η οποία αφορά στην κατασκευή ενός εννοιολογικού χάρτη (με "χαρτί - μολύβι" ή με τη χρήση του λογισμικού SmartTools) με κεντρική έννοια την έννοια "Δίκτυα Υπολογιστών" (παρατίθεται στη συνέχεια ως Φύλλο Αξιολόγησης).

Ανάλυση φύλλου εργασίας 2

Στο πλαίσιο της συγκεκριμένης διδακτικο-μαθησιακής δραστηριότητας, οι μαθητές, σε επίπεδο ολομέλειας της τάξης και σε συνεργασία με τον εκπαιδευτικό, θα ορίσουν τους άξονες και τα κριτήρια αξιολόγησης των εργασιών τους και θα αξιολογήσουν έναν ενδεικτικό χάρτη (μπορεί να έχει κατασκευαστεί από τον εκπαιδευτικό). Στη συνέχεια, με βάση τη φόρμα αξιολόγησης που δημιούργησαν, θα αξιολογήσουν δύο χάρτες των συμμαθητών τους, παρέχοντας προτάσεις βελτίωσης – ο εκπαιδευτικός θα πρέπει να έχει προετοιμάσει για αξιολόγηση τους χάρτες που έλαβε από τους μαθητές. Τέλος, θα κληθούν να διορθώσουν τη δική τους εργασία με βάση τα σχόλια και την αξιολόγηση που έλαβαν από τους συμμαθητές τους.

Προτεινόμενη Χρονική διάρκεια: 3 διδακτικές ώρες.

Στόχοι: Οι μαθητές θα μπορούν να:

- ορίζουν άξονες και κριτήρια αξιολόγησης για την αξιολόγηση της εργασίας τους,
- συντάσσουν μία φόρμα αξιολόγησης,

- αξιολογούν μία εργασία σύμφωνα με τους άξονες και τα κριτήρια που έχουν ορίσει,
- εντοπίζουν προβλήματα στην εργασία και να παρέχουν κατάλληλη ανατροφοδότηση για τη βελτίωση της, και
- να κρίνουν την ανατροφοδότηση που λαμβάνουν για την εργασία τους από τους συμμαθητές τους, υιοθετώντας ή απορρίπτοντας τις προτάσεις που τους παρέχονται για τη βελτίωσή της.

Απαιτούμενη υλικοτεχνική υποδομή

- Εργαστήριο υπολογιστών με εγκατεστημένο το λογισμικό επεξεργασίας κειμένου και το CmapTools.
- Χάρτες των μαθητών που έχουν κατασκευαστεί σύμφωνα με το φύλλο αξιολόγησης και ο ενδεικτικός χάρτης που θα αξιολογηθεί στην ολομέλεια της τάξης.

Επιμέρους Δραστηριότητες

A. Κατασκευή των χαρτών (διάρκεια 15 λεπτά)

Οι μαθητές θα κατασκευάσουν με τη χρήση του εργαλείου CmapTools το χάρτη που τους είχε ανατεθεί ως εργασία για το σπίτι σε περίπτωση που τον έχουν κατασκευάσει με "χαρτί - μολύβι" (επιμέρους δραστηριότητα Γ της 1η διδακτικο-μαθησιακής δραστηριότητας) και θα τον αποστείλουν ηλεκτρονικά στον εκπαιδευτικό. Οι μαθητές που έχουν ήδη αποστείλει ηλεκτρονικά το χάρτη, μπορούν να συζητήσουν μεταξύ τους για τον τρόπο που θα μπορούσαν να αξιολογήσουν τις εργασίες τους. Σε περίπτωση που όλοι οι μαθητές έχουν αποστείλει ηλεκτρονικά την εργασία στον εκπαιδευτικό, η συγκεκριμένη δραστηριότητα μπορεί να παραλειφθεί.

B. Συζήτηση - Καθορισμός Αξόνων και Κριτηρίων Αξιολόγησης - Δημιουργία Φόρμας Αξιολόγησης - Αξιολόγηση Ενδεικτικού Χάρτη (διάρκεια 60 λεπτά)

Στο πλαίσιο της συνεργατικής αξιολόγησης, οι μαθητές μαζί με τον εκπαιδευτικό καλούνται να ορίσουν τους άξονες και τα κριτήρια αξιολόγησης προκειμένου να δημιουργήσουν τη φόρμα αξιολόγησης. Ο εκπαιδευτικός είναι υπεύθυνος για το συντονισμό της συζήτησης και την καταγραφή των προτάσεων των μαθητών στον πίνακα. Ενδεικτικοί άξονες αξιολόγησης θα μπορούσε να είναι το *Περιεχόμενο* και η *Παρουσίαση*. Ως προς τον άξονα *Περιεχόμενο*, θα μπορούσαν να οριστούν κριτήρια όπως *Πληρότητα Εννοιών*, *Πληρότητα*

Συνδέσμων/Προτάσεων, Ορθότητα Εννοιών, Ορθότητα Σχηματιζόμενων Προτάσεων. Ως προς τον άξονα Παρουσίαση, θα μπορούσαν να οριστούν κριτήρια όπως Ορθογραφικά/Συντακτικά λάθη, Αναγνωσιμότητα Χάρτη. Τα κριτήρια επίσης θα μπορούσαν να διατυπωθούν και ως ερωτήματα. Στο πλαίσιο αυτό, και ορίζοντας κλίμακα αξιολόγησης καθώς και βαρύτητα σε κάθε κριτήριο (στην περίπτωση που συμφωνηθεί να αξιολογήσουν τις εργασίες τους και ποσοτικά), κάποιοι από τους μαθητές αναλαμβάνουν να συντάξουν ηλεκτρονικά τη φόρμα αξιολόγησης που θα χρησιμοποιήσουν. Μια ενδεικτική φόρμα αξιολόγησης παρουσιάζεται στον Πίνακα 1. Στη συνέχεια, οι μαθητές μαζί με τον εκπαιδευτικό συμπληρώνουν τη φόρμα αξιολόγησης για έναν ενδεικτικό εννοιολογικό χάρτη (κρίνεται σκόπιμο στο χάρτη να παρουσιάζονται προβλήματα ορθότητας και πληρότητας τόσο στις έννοιες που απεικονίζονται όσο και στις προτάσεις/συνδέσμους), προκειμένου να κατανοήσουν οι μαθητές πώς θα πρέπει να αξιολογήσουν τις εργασίες των συμμαθητών τους, πώς θα πρέπει να συμπληρώσουν τη φόρμα αξιολόγησης, ποιες προτάσεις βελτίωσης μπορούν να κάνουν και πώς γράφουμε/εκφραζόμαστε όταν αξιολογούμε.

Γ. Αξιολόγηση Εργασιών μαθητών (διάρκεια 40 λεπτά)

Σε κάθε μαθητή δίνονται από τον εκπαιδευτικό δύο εργασίες που θα πρέπει να αξιολογηθούν, σύμφωνα με τη φόρμα αξιολόγησης που δημιουργήθηκε στην επιμέρους δραστηριότητα Β. Οι μαθητές θα ανοίξουν τους χάρτες που τους έχουν δοθεί με το εργαλείο SmartTools προκειμένου να τους αξιολογήσουν και θα συμπληρώσουν ηλεκτρονικά τη φόρμα αξιολόγησης για κάθε εργασία, αποθηκεύοντας τη με όνομα *Αξιολόγηση-Εργασία-Επίθετο_Μαθητή-Επίθετο_Αξιολογητή.doc* (όπου στο Επίθετο_Μαθητή συμπληρώνεται το επίθετο του μαθητή που αφορά η εργασία και στο Επίθετο_Αξιολογητή συμπληρώνεται το επίθετο του μαθητή που κάνει την αξιολόγηση, π.χ. *Αξιολόγηση-Εργασία-Παπαδόπουλος-Γεωργίου.doc*). Κρίνεται σημαντικό να τονίσει ο εκπαιδευτικός ότι στόχος της αξιολόγησης είναι να βοηθήσουν τους συμμαθητές τους να βελτιώσουν την εργασία τους και να τους παρέχουν κατάλληλες συμβουλές και ανατροφοδότηση προς την κατεύθυνση αυτή.

Δ. Βελτίωση Εργασιών (διάρκεια 20 λεπτά)

Στο πλαίσιο της συγκεκριμένης δραστηριότητας, οι μαθητές βρίσκουν από το φάκελο που τους έχει υποδείξει ο εκπαιδευτικός τις φόρμες αξιολόγησης που αφορούν τη δική τους εργασία. Μελετούν τα σχόλια και τις προτάσεις των συμμαθητών τους και διορθώνουν την εργασία τους βασισμένοι στην ανατροφοδότηση που έλαβαν καθώς και στην εμπειρία που απέκτησαν ως αξιολογητές των εργασιών των συμμαθητών τους. Τέλος, αποστέλλουν ηλεκτρονικά στον εκπαιδευτικό τη βελτιωμένη εργασία τους με σχετική ένδειξη στο όνομα του αρχείου. Αν υπάρχει χρόνος, κρίνεται σκόπιμο να δοθούν στους μαθητές οι χάρτες που

κατασκεύασαν στη δραστηριότητα διερεύνησης της πρότερης γνώσης προκειμένου να δουν, να προβληματιστούν και να συζητήσουν τη διαφορά των χαρτών τους στις δύο χρονικές περιόδους.

Πίνακας 1. Ενδεικτική Φόρμα Αξιολόγησης

	Κλίμακα Αξιολόγησης		
Άξονες / Κριτήρια Αξιολόγησης	Καθόλου Ικανοποιητική	Μέτρια	Ικανοποιητική
Περιεχόμενο			
Πληρότητα εννοιών	Πολλές έννοιες (περισσότερες από 15) που δίνονται στη λίστα εννοιών δεν απεικονίζονται στο χάρτη.	Κάποιες έννοιες (λιγότερες από 15) που δίνονται στη λίστα εννοιών δεν απεικονίζονται στο χάρτη.	Όλες οι έννοιες που δίνονται στη λίστα εννοιών απεικονίζονται στο χάρτη (μπορεί να έχουν προστεθεί επιπλέον έννοιες).
Πληρότητα Συνδέσμων / Προτάσεων	Υπάρχουν πολλές συνδέσεις εννοιών στο χάρτη που δεν απεικονίζονται.	Μπορούν να προστεθούν επιπλέον συνδέσεις μεταξύ των εννοιών. Επίσης, παρουσιάζεται ένας μικρός αριθμός σύνθετων συνδέσεων.	Υπάρχουν απλοί σύνδεσμοι μεταξύ των εννοιών του χάρτη. Επίσης, έχει οριστεί και ένας ικανοποιητικός αριθμός σύνθετων συνδέσεων.
Ορθότητα Εννοιών	Απεικονίζονται στο χάρτη έννοιες που δεν είναι ορθές ή/και υπάρχουν έννοιες στο χάρτη που δεν είναι τοποθετημένες στο σωστό επίπεδο (δηλαδή δεν αναλύουν/επεξηγούν την έννοια με την οποία συνδέονται).	Υπάρχει ένας μικρός αριθμός εννοιών που δεν είναι ορθές ή/και δεν είναι τοποθετημένες στο σωστό επίπεδο.	Όλες οι έννοιες που απεικονίζονται στο χάρτη είναι ορθές και περιγράφουν /αναλύουν την κεντρική έννοια.
Ορθότητα Σχηματιζόμενων Προτάσεων	Ένας μεγάλος αριθμός προτάσεων που απεικονίζονται στο χάρτη δεν είναι ορθές. Δηλαδή, απεικονίζονται προτάσεις μεταξύ εννοιών που δεν θα έπρεπε να συνδέονται ή/και οι σύνδεσμοι που χρησιμοποιούνται για τη δημιουργία των προτάσεων δεν είναι κατάλληλοι ή είναι λανθασμένοι.	Υπάρχει ένας μικρός αριθμός προτάσεων που δεν είναι ορθές.	Σχεδόν όλες οι προτάσεις που απεικονίζονται στο χάρτη είναι ορθές.

Παρουσίαση			
Ορθογραφικά / Συντακτικά λάθη	Υπάρχουν αρκετά ορθογραφικά και συντακτικά λάθη.	Υπάρχουν λίγα ορθογραφικά και συντακτικά λάθη.	Δεν υπάρχουν ορθογραφικά και συντακτικά λάθη.
Αναγνωσιμότητα Χάρτη	Ο χάρτης είναι δύσκολο να διαβαστεί λόγω της ταξινόμησης των εννοιών και των συνδέσμων στην επιφάνεια εργασίας. Δεν χρησιμοποιούνται εργαλεία της παλέτας στυλ.	Παρουσιάζονται μερικά προβλήματα στην αναγνωσιμότητα του χάρτη και χρειάζεται να χρησιμοποιηθούν επιπλέον εργαλεία της παλέτας στυλ.	Ο χάρτης είναι ευανάγνωστος και επιμελημένος, χρησιμοποιώντας αρκετά εργαλεία της παλέτας στυλ (π.χ. χρώματα, σχήματα συνδέσεων κ.λπ.)
Σχόλια για την εργασία			
Προτάσεις βελτίωσης της εργασίας			

16. ΑΞΙΟΛΟΓΗΣΗ

Για την αξιολόγηση των μαθητών μπορεί να ετοιμαστεί το σχετικό φύλλο εργασίας που αναλύθηκε παραπάνω και παρουσιάζεται στη συνέχεια.

Μέσα από την εργασία των μαθητών που είχαν για το σπίτι (κατασκευή εννοιολογικού χάρτη για την έννοια "Δίκτυα Υπολογιστών"), τις φόρμες αξιολόγησης που συμπλήρωσαν για την αξιολόγηση των εργασιών των συμμαθητών τους και τη βελτιωμένη έκδοση της εργασίας τους μετά από την αξιολόγηση και την ανατροφοδότηση που έλαβαν από τους συμμαθητές τους είναι δυνατόν ο εκπαιδευτικός να αξιολογήσει το βαθμό επίτευξης των στόχων του σεναρίου.

17. ΤΟ ΕΠΙΜΟΡΦΩΤΙΚΟ ΣΕΝΑΡΙΟ

Θα δοθούν 2 φύλλα εργασίας, τα οποία παρατίθενται παρακάτω.

Αναλυτική περιγραφή και ανάλυση των φύλλων εργασίας δίνεται στην ενότητα 15.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 1 – Βασικές Έννοιες & Κατηγορίες Δικτύων Υπολογιστών

Δραστηριότητα 1η

Σας δίνεται ο εννοιολογικός χάρτης "Δίκτυα Υπολογιστών - Βασικές Έννοιες" που αφορά στην κεντρική έννοια "Δίκτυα Υπολογιστών". Να ανοίξετε το λογισμικό SmartTools από την επιφάνεια εργασίας και στη συνέχεια από τις Προβολές, να επιλέξετε να ανοίξετε το συγκεκριμένο χάρτη.

Βήμα 1ο: Μελέτη του χάρτη "Δίκτυα Υπολογιστών - Βασικές Έννοιες" και του διαθέσιμου εκπαιδευτικού υλικού.

Στην έννοια "Δίκτυα Υπολογιστών" παρουσιάζονται δύο εικονίδια:

- το 1ο εικονίδιο αντιστοιχεί σε δύο συνδέσμους που οδηγούν στα κείμενα "Δίκτυα Υπολογιστών" και "Παραδείγματα Δικτύων" και
- το 2ο εικονίδιο αφορά στον εννοιολογικό χάρτη "Κατηγοριοποίηση Δικτύων Υπολογιστών" της δεύτερης δραστηριότητας.

Ομοίως τα εικονίδια στις έννοιες "Κόμβοι" και "Ζεύξεις Επικοινωνίας" οδηγούν σε αντίστοιχα κείμενα για τις συγκεκριμένες έννοιες. Επιλέγοντας το εικονίδιο του σχολιασμού που εμφανίζεται στην έννοια "Σταθμοί Εργασίας" παρουσιάζεται ένα κείμενο που επεξηγεί την έννοια. Τέλος, περνώντας το ποντίκι πάνω από τις έννοιες "Δίκτυα Υπολογιστών" και "Ζεύξεις Επικοινωνίας" εμφανίζονται κείμενα (έχουν προστεθεί μέσω του εργαλείου "Προσθήκη Πληροφοριών") που επεξηγούν τις έννοιες.



Εργασία: Να μελετήσετε τις έννοιες που απεικονίζονται στο χάρτη "Δίκτυα Υπολογιστών - Βασικές Έννοιες" καθώς και το εκπαιδευτικό υλικό που είναι διαθέσιμο για τις έννοιες. Δεν χρειάζεται στο 1ο Βήμα να μεταβείτε στον εννοιολογικό χάρτη "Κατηγοριοποίηση Δικτύων Υπολογιστών" μέσω του αντίστοιχου συνδέσμου στην έννοια "Δίκτυα Υπολογιστών".

Βήμα 2ο: Συγγραφή/Προσθήκη εκπαιδευτικού υλικού.



Εργασία: Στην έννοια "Δίκτυα Υπολογιστών" έχει οριστεί ένας σύνδεσμος (2ο εικονίδιο) με όνομα "Παραδείγματα Δικτύων", που οδηγεί σε αντίστοιχο word κείμενο. Ενεργοποιώντας το σύνδεσμο, να μεταβείτε στο word κείμενο και να αναφέρετε σε μια παράγραφο παραδείγματα δικτύων από την καθημερινή σας ζωή. Στη συνέχεια, εστιάζομενοι στην περίπτωση των δικτύων υπολογιστών, να αναφέρετε 5 πλεονεκτήματα και μειονεκτήματα της χρήσης τους. Να αποθηκεύσετε το

κείμενο στο χάρτη.

Βήμα 3ο: Συμπλήρωση των εννοιών στο χάρτη που απεικονίζονται ως "?????".



Εργασία: Μελετώντας το υλικό που σας παρέχεται πάνω στις έννοιες, να συμπληρώσετε στο χάρτη σας τις έννοιες που απεικονίζονται με "?????".

Βήμα 4ο: Επέκταση του χάρτη με νέες έννοιες.



Εργασία: Στο χάρτη απεικονίζονται έννοιες που αφορούν στις δυνατές χρήσεις των Δικτύων Υπολογιστών (π.χ. ανταλλαγή δεδομένων/πληροφοριών, διασκέδαση, διαμοιρασμός πόρων κ.λπ.). Να επεκτείνετε το χάρτη σας προσθέτοντας έννοιες που αφορούν παραδείγματα για κάθε μία από τις δυνατές χρήσεις και να ορίσετε κατάλληλους συνδέσμους. Για παράδειγμα:

- Τι δεδομένα/πληροφορίες μπορούμε να ανταλλάξουμε;
- Πώς χρησιμοποιούμε τα δίκτυα υπολογιστών για διασκέδαση;
- Σε ποιες υπηρεσίες/εφαρμογές μπορούμε να έχουμε πρόσβαση;
- Πώς μπορεί να επιτευχθεί η συνεργασία ανθρώπων από απόσταση;
- Ποιους πόρους μπορούμε να διαμοιράσουμε; κ.λπ.

Βήμα 5ο: Επέκταση του χάρτη με νέες έννοιες.



Εργασία: Μελετώντας το υλικό που παρέχεται για τις έννοιες "Ζεύξεις Επικοινωνίας" και "Εξυπηρετητές", να επεκτείνετε το χάρτη σας προσθέτοντας έννοιες και ορίζοντας κατάλληλους συνδέσμους που επεξηγούν/αναλύουν τις συγκεκριμένες έννοιες (π.χ. δυνατά είδη ζεύξεων επικοινωνίας, είδη εξυπηρετητών και λειτουργία τους κ.λπ.).

Βήμα 6ο: Αναζήτηση πληροφοριών, συγγραφή/προσθήκη εκπαιδευτικού υλικού και επέκταση του χάρτη με νέες έννοιες.



Εργασία: Για να είναι δυνατή η επικοινωνία των κόμβων χρειάζονται τα πρωτόκολλα. Την έννοια αυτή θα πρέπει να τη συμπληρώσατε στο χάρτη σας στο Βήμα 3. Να αναζητήσετε πληροφορίες στο Διαδίκτυο για τη συγκεκριμένη έννοια (π.χ. στην ελληνική ΒΙΚΙΠΑΙΔΕΙΑ), να τις προσθέσετε στο χάρτη σας χρησιμοποιώντας το εργαλείο *Σχολιασμού της έννοιας* (είναι διαθέσιμο κάνοντας δεξί κλικ πάνω στη έννοια και επιλέγοντας το "Σχολιασμός") και να επεκτείνετε το χάρτη σας αναφέροντας παραδείγματα γνωστών πρωτοκόλλων επικοινωνίας.

Βήμα 7ο: Αναζήτηση πληροφοριών, συγγραφή/προσθήκη εκπαιδευτικού υλικού και επέκταση του χάρτη με νέες έννοιες.



Εργασία: Στο Βήμα 3 θα πρέπει να συμπληρώσατε στο χάρτη σας τις έννοιες "Συγκεντρωτές" (για τη σύνδεση υπολογιστών) και "Δρομολογητές" (ως ένα παράδειγμα κόμβων). Για τις συγκεκριμένες έννοιες, να αναζητήσετε πληροφορίες στο Διαδίκτυο καθώς και εικόνες και να τις προσθέσετε χρησιμοποιώντας τα εργαλεία "*Προσθήκη & Επεξεργασία συνδέσμων σε πηγές*" και "*Προσθήκη διεύθυνσης ιστού*". Να επεκτείνετε το χάρτη σας προσθέτοντας έννοιες που αφορούν στις αγγλικές ονομασίες των συγκεκριμένων εννοιών.

Βήμα 8ο: Αναζήτηση πληροφοριών, συγγραφή/προσθήκη εκπαιδευτικού υλικού και επέκταση του χάρτη με νέες έννοιες.



Εργασία: Να αναζητήσετε πληροφορίες στο Διαδίκτυο για την έννοια "Ταχύτητα/Ρυθμός Μετάδοσης" (π.χ. ορισμός της έννοιας, τυπικές τιμές ρυθμού μετάδοσης, παράγοντες από τους οποίους εξαρτάται ο ρυθμός μετάδοσης, κ.λπ.) και να τις προσθέσετε στο χάρτη σας χρησιμοποιώντας το εργαλείο *Σχολιασμού της έννοιας*. Στη συνέχεια, να επεκτείνετε το χάρτη προσθέτοντας κατάλληλες έννοιες και συνδέσμους προκειμένου να εξηγήσετε/αναλύσετε την έννοια "Ταχύτητα/Ρυθμός Μετάδοσης".

Δραστηριότητα 2η

Βήμα 9ο: Μελέτη του χάρτη "Κατηγοριοποίηση Δικτύων Υπολογιστών" και του διαθέσιμου εκπαιδευτικού υλικού.

Να επιλέξετε το 3ο εικονίδιο που εμφανίζεται στην έννοια "Δίκτυα Υπολογιστών" του εννοιολογικού χάρτη "Δίκτυα Υπολογιστών - Βασικές Έννοιες". Ο χάρτης που εμφανίζεται αφορά στις δυνατές κατηγοριοποιήσεις των δικτύων υπολογιστών. Στις έννοιες "Τηλεπικοινωνιακός Φορέας Εξυπηρέτησης", "Αστέρας", "Μέσο Σύνδεσης/ Μετάδοσης" και "Καλώδια" έχει οριστεί εκπαιδευτικό υλικό για την επεξήγηση των εννοιών.



Εργασία: Να μελετήσετε τις έννοιες που απεικονίζονται στο χάρτη "Κατηγοριοποίηση Δικτύων Υπολογιστών" καθώς και το εκπαιδευτικό υλικό που είναι διαθέσιμο για τις έννοιες.

Βήμα 10ο: Επέκταση και Συμπλήρωση του χάρτη για τις έννοιες "Τηλεπικοινωνιακός Φορέας Εξυπηρέτησης" και "Μέσο Σύνδεσης/Μετάδοσης".



Εργασία: Μελετώντας το υλικό που σας παρέχεται για τις συγκεκριμένες έννοιες, να επεκτείνετε το χάρτη σας προσθέτοντας έννοιες και ορίζοντας κατάλληλους συνδέσμους που επεξηγούν/αναλύουν τις έννοιες καθώς και να συμπληρώσετε τις έννοιες που απεικονίζονται με "?????" (π.χ. διάκριση των δικτύων με βάση τον τηλεπικοινωνιακό φορέα εξυπηρέτησης, συνηθισμένα μέσα σύνδεσης/μετάδοσης, διάκριση των δικτύων με βάση το μέσο σύνδεσης/μετάδοσης κ.λπ.).

Βήμα 11ο: Αναζήτηση πληροφοριών για τη διάκριση των δικτύων ως προς τη γεωγραφική τους διασπορά, συμπλήρωση και επέκταση του χάρτη.



Εργασία: Να αναζητήσετε πληροφορίες στο Διαδίκτυο ή στο σχολικό βιβλίο για τη διάκριση των δικτύων υπολογιστών με βάση τη γεωγραφική τους διασπορά. Στη συνέχεια, να συμπληρώσετε τις έννοιες που απεικονίζονται ως "?????" και αφορούν στα είδη των δικτύων υπολογιστών ως προς τη γεωγραφική τους διασπορά και να αναλύσετε τις έννοιες που θα συμπληρώσετε δίνοντας παραδείγματα τέτοιων δικτύων καθώς και τυχόν χαρακτηριστικά τους.

Βήμα 12ο: Αναζήτηση πληροφοριών και συγγραφή/προσθήκη υλικού.



Εργασία: Ως προς τον τρόπο σύνδεσης/μετάδοσης, τα δίκτυα υπολογιστών μπορούν να διακριθούν σε αστέρα, αρτηρία, δακτύλιο. Για την έννοια "Αστέρας" δίνεται επεξηγηματικό εκπαιδευτικό υλικό (ορισμός,

πλεονεκτήματα, μειονεκτήματα κ.λπ.) καθώς και μία εικόνα. Ανάλογα, για τις έννοιες "Αρτηρία" και "Δακτύλιος", να αναζητήσετε πληροφορίες και εικόνες στο Διαδίκτυο και να τις προσθέσετε ως πηγές στις αντίστοιχες έννοιες.

Βήμα 13ο: Αναζήτηση πληροφοριών και συγγραφή/προσθήκη εκπαιδευτικού υλικού.



Εργασία: Να αναζητήσετε πληροφορίες στο Διαδίκτυο ή στο σχολικό βιβλίο για τα δίκτυα δεδομένων, τα δίκτυα πολυμέσων και τα κοινωνικά δίκτυα. Να προσθέσετε τις πληροφορίες που θα βρείτε ως σχολιασμό στις συγκεκριμένες έννοιες και να συμπληρώσετε στο χάρτη σας ένα παράδειγμα κοινωνικού δικτύου (η έννοια απεικονίζεται με "????").

Βήμα 14ο: Επέκταση του χάρτη για την έννοια "Καλώδια" και συγγραφή/προσθήκη εκπαιδευτικού υλικού.



Εργασία: Μελετώντας το υλικό που παρέχεται για την έννοια "Καλώδια", να επεκτείνετε το χάρτη σας προσθέτοντας έννοιες και ορίζοντας κατάλληλους συνδέσμους που επεξηγούν/αναλύουν τη συγκεκριμένη έννοια. Επίσης, για κάθε τύπο καλωδίου που θα προσθέσετε, να αναζητήσετε εικόνες από το Διαδίκτυο και να τις προσθέσετε ως πηγές στις συγκεκριμένες έννοιες.

Βήμα 15ο: Επέκταση του χάρτη για την έννοια "Ασύρματα δίκτυα".



Εργασία: Ανάλογα με το μέσο σύνδεσης/μετάδοσης, τα δίκτυα μπορούν να διακριθούν σε ενσύρματα και σε ασύρματα δίκτυα (έννοιες που πρέπει να έχετε συμπληρώσει στο Βήμα 11). Να αναζητήσετε πληροφορίες στο Διαδίκτυο ή στο σχολικό βιβλίο για τα ασύρματα δίκτυα και να επεκτείνετε το χάρτη σας προσθέτοντας έννοιες και ορίζοντας κατάλληλους συνδέσμους που επεξηγούν/αναλύουν τη συγκεκριμένη έννοια.

Βήμα 16ο: Προσθήκη σύνθετων συνδέσεων.



- Εργασία:** Μελετώντας το χάρτη που έχετε κατασκευάσει μέχρι στιγμής, να προσθέσετε επιπλέον συνδέσεις μεταξύ των εννοιών που απεικονίζονται. Για παράδειγμα:
- Υπάρχει σύνδεση μεταξύ των εννοιών "Τοπικά Δίκτυα" και "Ενσύρματα δίκτυα";
 - Υπάρχει σύνδεση μεταξύ των εννοιών "Οπτική ίνα" και "Καλώδια"; κ.λπ.

ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ 2 – Αξιολόγηση

Να κατασκευάσετε έναν εννοιολογικό χάρτη με κεντρική έννοια την έννοια «**Δίκτυα Υπολογιστών**» χρησιμοποιώντας έννοιες από την ακόλουθη λίστα εννοιών και ορίζοντας κατάλληλους συνδέσμους.

Επίσης, μπορείτε να προσθέσετε μέχρι 5 επιπλέον έννοιες της αρεσκείας σας που δεν υπάρχουν στη λίστα και περιγράφουν την κεντρική έννοια.

Διαθέσιμη Λίστα Εννοιών

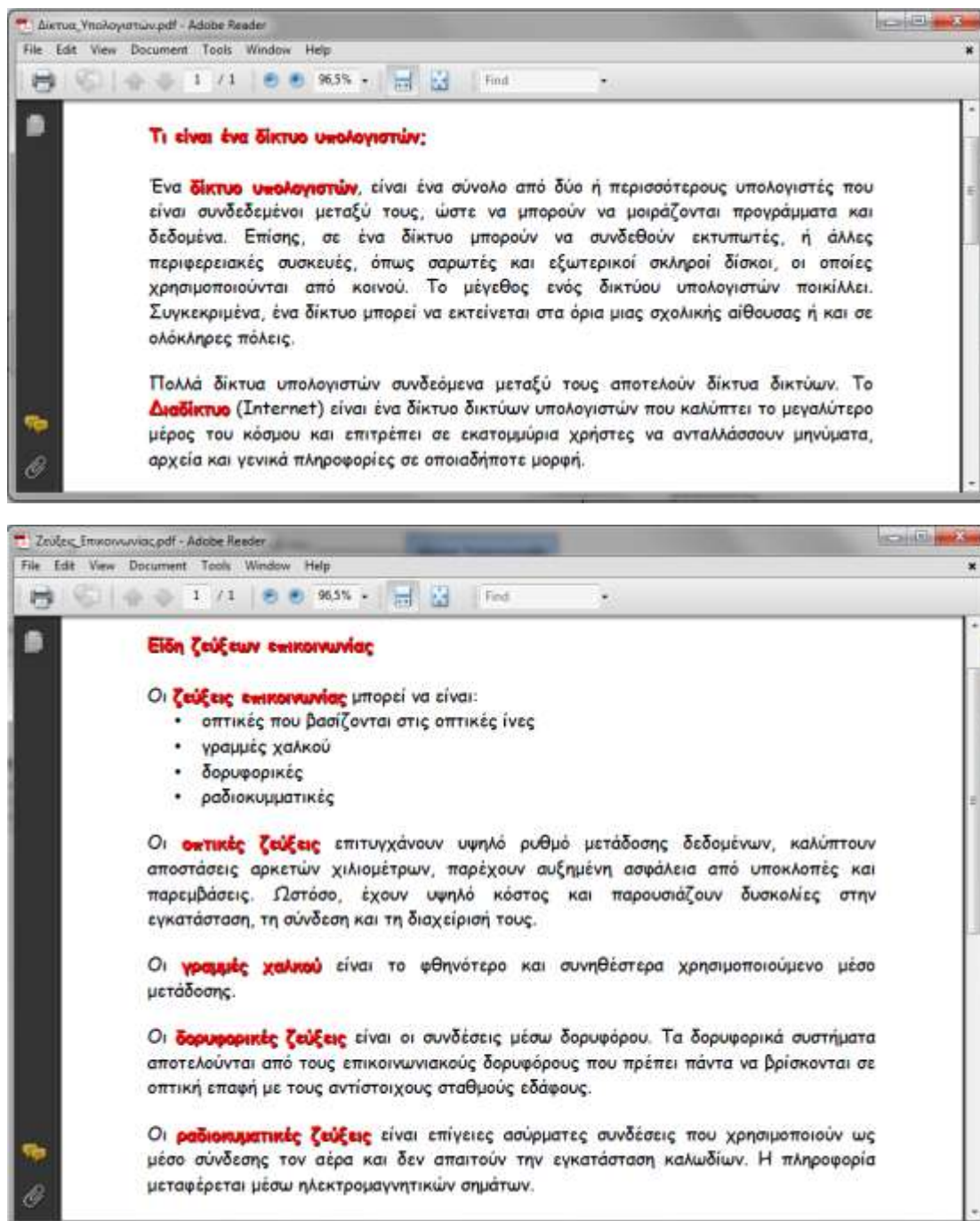
Δίκτυα Υπολογιστών	Ζεύξεις Επικοινωνίας	Υπολογιστές
Δρομολογητής	Ταχύτητα/Ρυθμός Μετάδοσης	Τραπεζικό δίκτυο
Πρωτόκολλο	Ενσύρματα Δίκτυα	Ασύρματα Δίκτυα
Σχολικό Εργαστήριο	Δίκτυο Ευρείας Περιοχής	Γραμμές χαλκού
Οπτική Ίνα	Μητροπολιτικό δίκτυο	Ραδιοκύματα
Τοπικό δίκτυο	Γεωγραφική διασπορά	Διαδίκτυο
Ασφάλεια	Περιφερειακές Συσκευές	Ιοί
Δορυφόροι	Ανταλλαγή δεδομένων	Διαμοιρασμός πόρων
Κόμβοι	Συγκεντρωτής	TCP/IP
Ethernet	bps	Καλώδια
Λογισμικό	αρχεία	

Ο χάρτης θα πρέπει να κατασκευαστεί χρησιμοποιώντας το λογισμικό CmapTools, το οποίο μπορείτε να κατεβάσετε από τη διεύθυνση <http://cmap.ihmc.us> και να εγκαταστήσετε στο υπολογιστή σας. Σε περίπτωση που δεν διαθέτετε υπολογιστή, ο χάρτης μπορεί να κατασκευαστεί με "χαρτί - μολύβι" και θα τον κατασκευάσετε ηλεκτρονικά στο επόμενο μάθημα στο εργαστήριο.

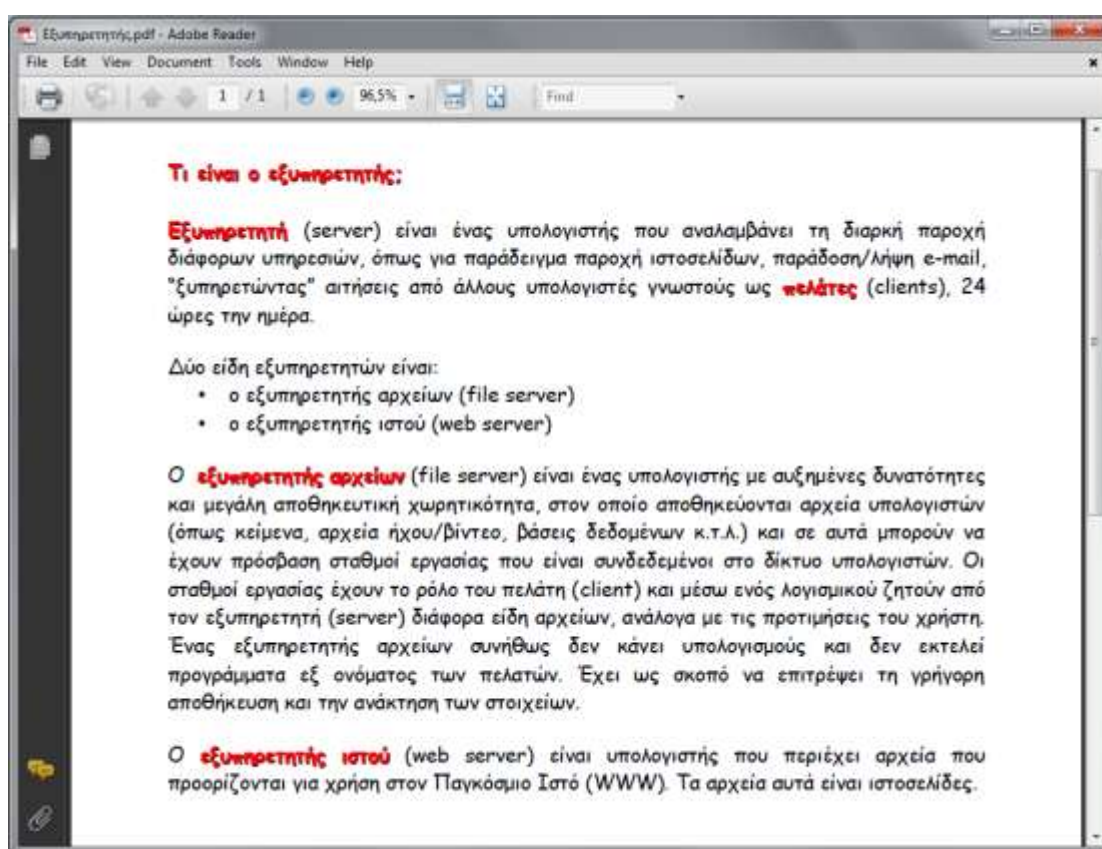
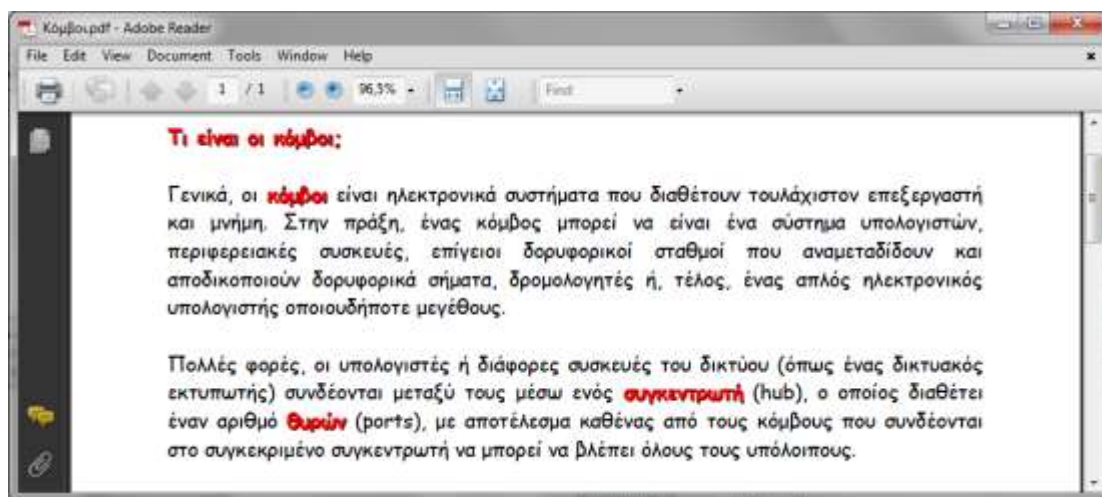
Θα πρέπει να αποθηκεύσετε το χάρτη σας ως **Εργασία-Επίθετο.cmap**, όπου στο Επίθετο θα δηλώσετε το επίθετό σας (π.χ. *Εργασία-Παπαδόπουλος.cmap*). Οι ηλεκτρονικοί σας χάρτες, μετά την αποθήκευσή τους, θα πρέπει να αποσταλούν με e-mail στον καθηγητή Πληροφορικής (*xx@ddd.gr*) μέχρι το επόμενο μάθημα στις ../..

18. ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ – ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

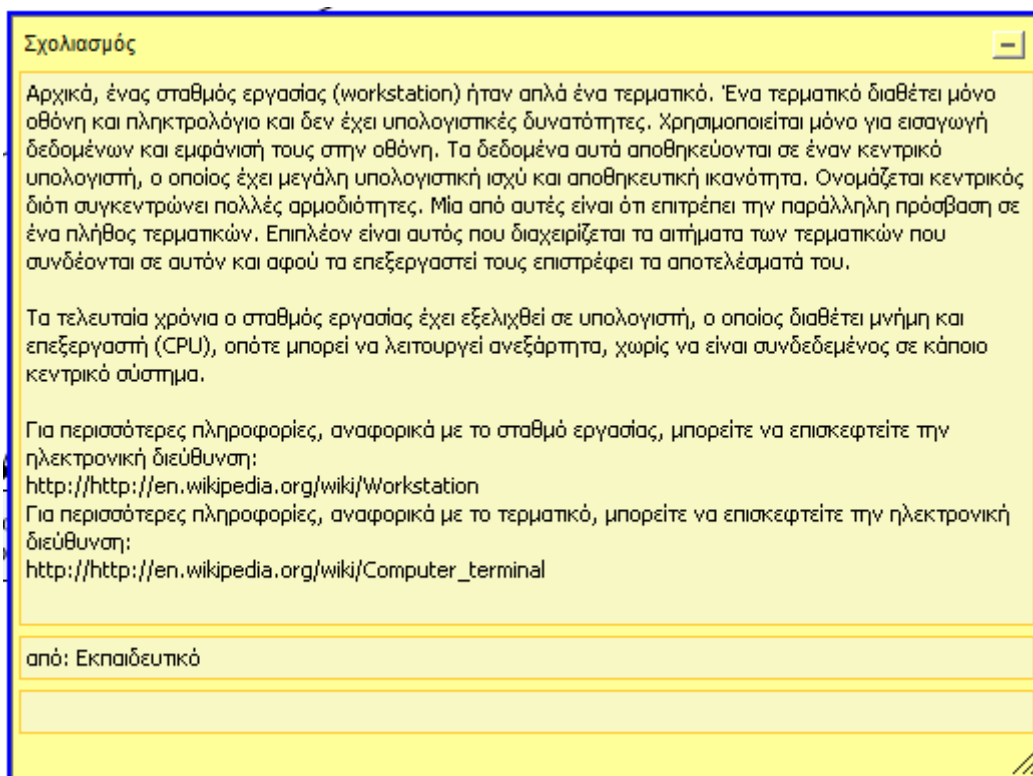
- Μελετήστε τα φύλλα εργασίας του διδακτικού σεναρίου, σχολιάστε και τροποποιήστε κατάλληλα – εφόσον το κρίνετε απαραίτητο – τις σχετικές δραστηριότητες.
- Αναζητήστε στις ιστοσελίδες των εργαλείων εννοιολογικής χαρτογράφησης πληροφορίες για τη δυνατότητα υποστήριξης συνεργατικών δραστηριοτήτων. Προτείνετε κάποια συνεργατική δραστηριότητα που θα μπορούσε να ενταχθεί στο διδακτικό σενάριο για τα δίκτυα υπολογιστών που παρουσιάστηκε.
 - CmapTools: <http://cmap.ihmc.us/>
 - Synergo: http://hci.ece.upatras.gr/index.php?option=com_content&task=view&id=103&Itemid=103
 - COMPASS: <http://hermes.di.uoa.gr/compass/>
- Θεωρώντας ότι οι μαθητές σας δεν γνωρίζουν την τεχνική της εννοιολογικής χαρτογράφησης ετοιμάστε ένα φύλλο εργασίας για την εξοικείωση με την τεχνική της εννοιολογικής χαρτογράφησης (και το εργαλείο CmapTools).
- Δημιουργήστε ένα διδακτικό σενάριο που θα περιλαμβάνει δραστηριότητες εννοιολογικής χαρτογράφησης για τη διδασκαλία κάποιου θέματος πληροφορικής της επιλογής σας.



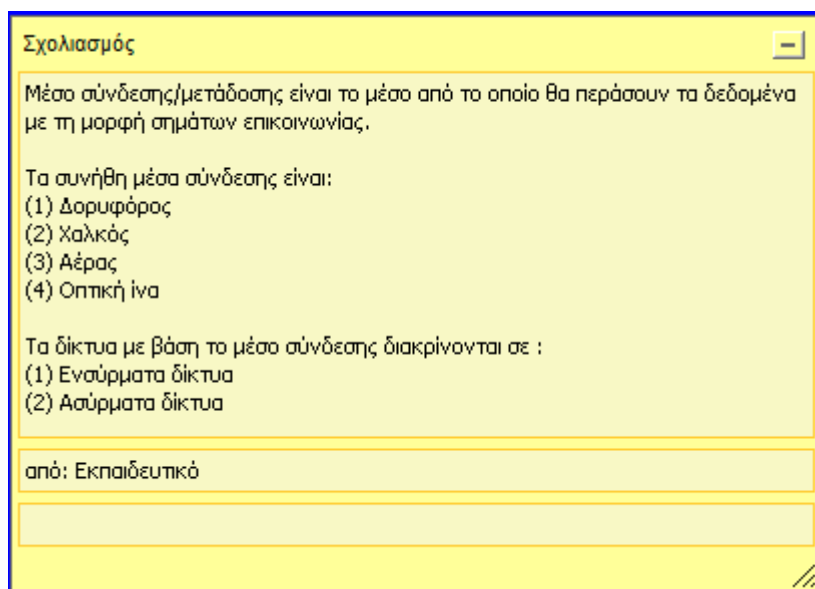
Εικόνα 3. Το εκπαιδευτικό υλικό του χάρτη “Δίκτυα Υπολογιστών – Βασικές Έννοιες”.



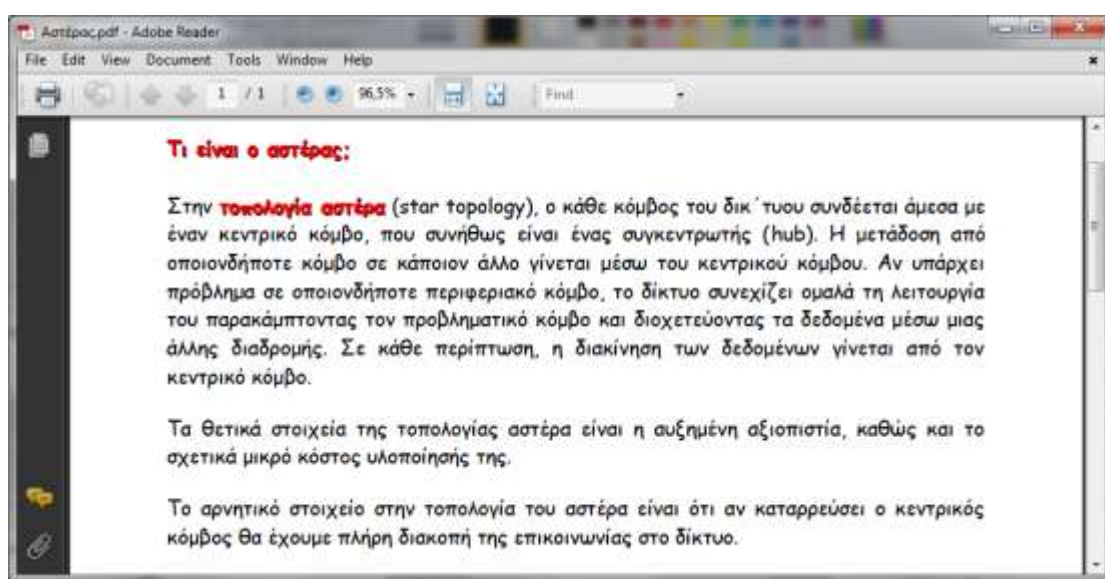
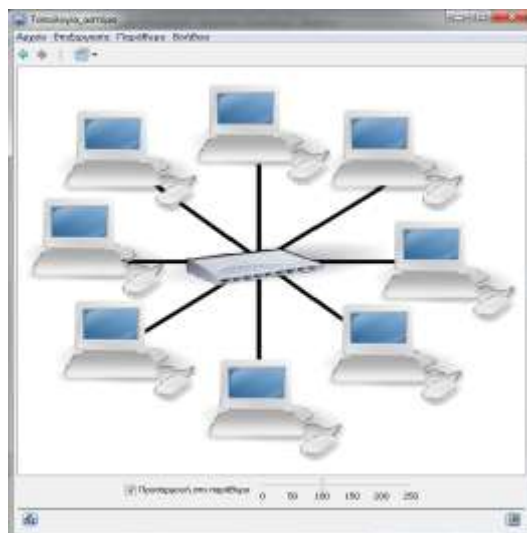
Εικόνα 3 (...συνέχεια). Το εκπαιδευτικό υλικό του χάρτη "Δίκτυα Υπολογιστών – Βασικές Έννοιες".



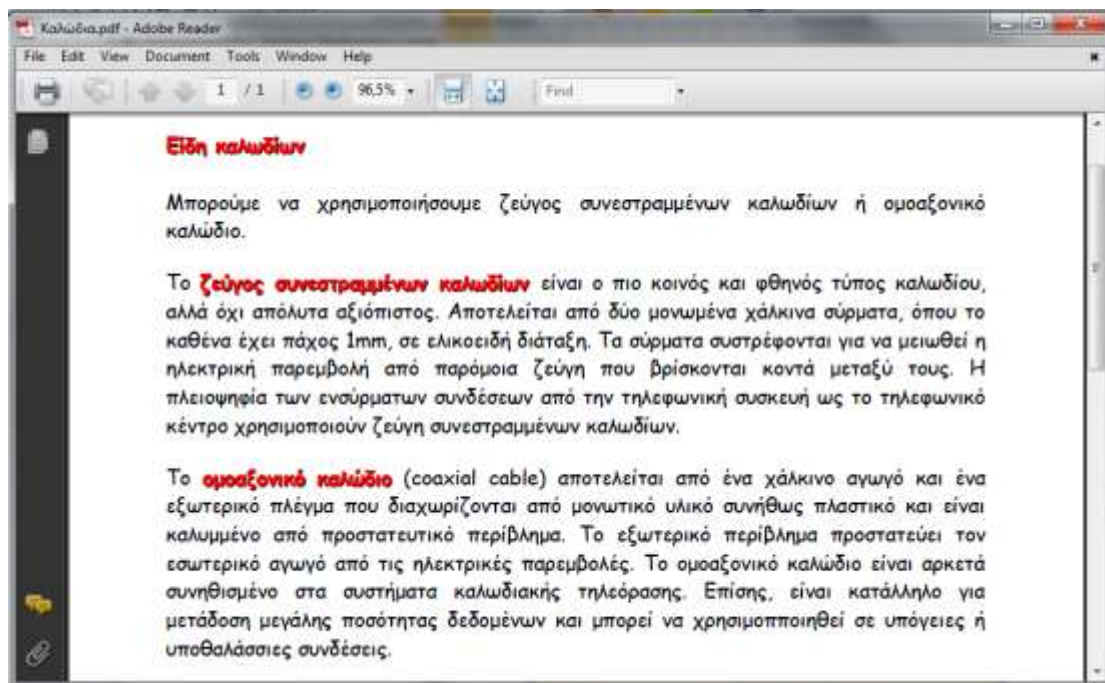
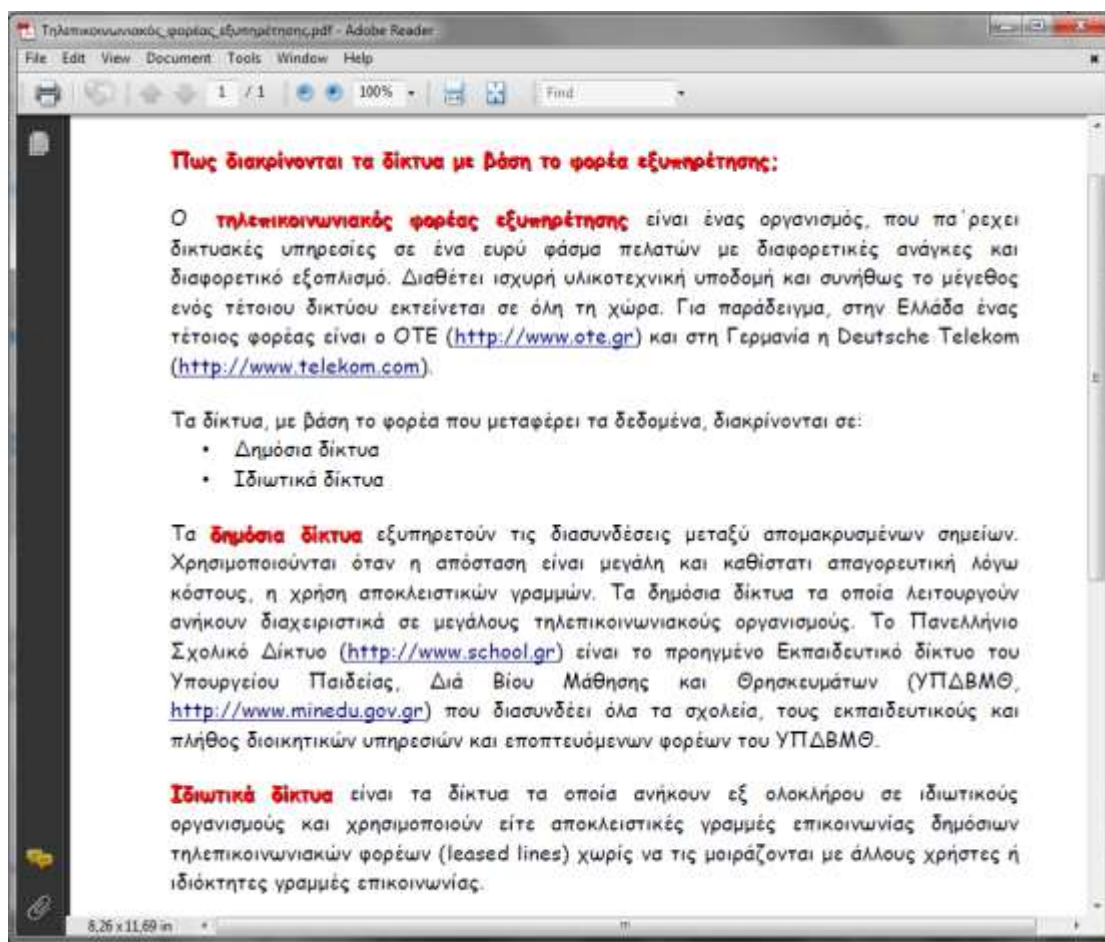
Εικόνα 3 (...συνέχεια). Το εκπαιδευτικό υλικό του χάρτη “Δίκτυα Υπολογιστών – Βασικές Έννοιες”.



Εικόνα 4. Το εκπαιδευτικό υλικό του χάρτη “Κατηγοριοποίηση Δικτύων Υπολογιστών”.



Εικόνα 4 (...συνέχεια). Το εκπαιδευτικό υλικό του χάρτη “Κατηγοριοποίηση Δικτύων Υπολογιστών”.



Εικόνα 4 (... συνέχεια). Το εκπαιδευτικό υλικό του χάρτη “Κατηγοριοποίηση Δικτύων Υπολογιστών”.