



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ
ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ

**Ολοκληρωμένη προσέγγιση παιχνιδοποίησης της διδασκαλίας της
γλώσσας προγραμματισμού Python**

Βελαώρα Χρυσούλα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων

Κακαρούντας Αθανάσιος

Λαμία, 2019



UNIVERSITY OF THESSALY
SCHOOL OF SCIENCE
INFORMATICS AND COMPUTATIONAL BIOMEDICINE

**Integrated gamification approach of the Python programming
language teaching**

Velaora Chrysoula

Master thesis

Supervisor

Kakarountas Athanasios

Lamia , 2019



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ
ΚΑΤΕΥΘΥΝΣΗ
«ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΩΝ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ»

**Ολοκληρωμένη προσέγγιση παιχνιδοποίησης της διδασκαλίας της
γλώσσας προγραμματισμού Python**

Βελαώρα Χρυσούλα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων

Κακαρούντας Αθανάσιος

Λαμία, 2019

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο «Ολοκληρωμένη προσέγγιση παιγνιδοποίησης της διδασκαλίας της γλώσσας προγραμματισμού Python» αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η ΔΗΛΩΝ/-ΟΥΣΑ

Ημερομηνία

Υπογραφή

Ολοκληρωμένη προσέγγιση παιχνιδοποίησης της διδασκαλίας της γλώσσας προγραμματισμού Python

Βελαώρα Χρυσούλα

Τριμελής Επιτροπή:

Ιωάννης Αναγνωστόπουλος, Αναπληρωτής Καθηγητής

Βασίλειος Δρακόπουλος, Επίκουρος Καθηγητής

Κακαρούντας Αθανάσιος, Επίκουρος Καθηγητής

Επιστημονικός Σύμβουλος:

Κακαρούντας Αθανάσιος, Επίκουρος Καθηγητής

Περίληψη

Η παρούσα διπλωματική εργασία επιχειρεί μέσα από σύντομη βιβλιογραφική ανασκόπηση να σκιαγραφήσει κυρίως α) τον τρόπο λειτουργίας των γλωσσών προγραμματισμού ανάλογα με την κατηγορία στην οποία ανήκουν, β) τα βασικά χαρακτηριστικά της γλώσσας προγραμματισμού Python στα οποία στηρίζεται, σύμφωνα με ερευνητές, η αξιοποίησή της στην εκπαίδευση, όπως για παράδειγμα ότι είναι: απλή, εύκολη στην εκμάθηση, ελεύθερη και ανοικτού κώδικα, υψηλού επιπέδου, φορητή, διερχόμενη με άμεση ανατροφοδότηση προς τον εκπαιδευόμενο, με πλούσιες εκτεταμένες βιβλιοθήκες και άλλα και γ) την εμφανή συνεισφορά της βιωματικής μάθησης, της παιχνιδοποίησης και των αναδυόμενης τεχνολογίας στη διδασκαλία της. Επίσης, στόχος της εργασίας είναι ο σχεδιασμός τεσσάρων διδακτικών σεναρίων που αποτελούν μία νέα διδακτική πρόταση για την εμπέδωση συγκεκριμένων διδακτικών εννοιών της, ενσωματώνοντας στοιχεία δυναμικής παιχνιδιών, αναδυόμενη τεχνολογία και προσεγγίζοντας το θέμα που πραγματεύονται, με βιωματικό τρόπο σε ένα αυθεντικό πλαίσιο μάθησης επιλύοντας προβλήματα της καθημερινής ζωής. Η κύρια διδακτική προσέγγιση των διδακτικών σεναρίων είναι αυτή του κοινωνικού εποικοδομισμού. Τέλος, στα πλαίσια της εργασίας διενεργήθηκε μια έρευνα με σκοπό την αξιολόγηση της επίδρασης του σχεδιασμού των διδακτικών σεναρίων στα κίνητρα των εκπαιδευομένων. Το δείγμα της έρευνας αποτέλεσαν 15 προπτυχιακοί φοιτητές του τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας. Συγκεκριμένα στο 1ο διδακτικό σενάριο συμμετείχαν 9 φοιτητές, στο 2ο διδακτικό σενάριο 6 φοιτητές, στο 3ο διδακτικό σενάριο 10 φοιτητές και στο 4ο διδακτικό σενάριο 5 φοιτητές οι οποίοι μετά το πέρας του κάθε διδακτικού σεναρίου συμπλήρωσαν ένα ερωτηματολόγιο ανίχνευσης κινήτρων που βασίστηκε στο ερωτηματολόγιο CIS (Course Interest Survey) το οποίο σχεδιάστηκε από τον Keller το 1987 με ερωτήσεις 5-βάθμιας κλίμακας τύπου Likert αναφορικά με τις μεταβλητές κίνητρα, προσοχή-σχετικότητα, εμπιστοσύνη-ικανοποίηση που αποτελούν τις συνιστώσες του μοντέλου κινήτρων ARCS του Keller. Τα αποτελέσματα έδειξαν ότι ο σχεδιασμός των διδακτικών σεναρίων συνδυάζοντας τη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας ενισχύει τα κίνητρα των εκπαιδευομένων και οδηγούν στην επίτευξη των μαθησιακών στόχων.

Λέξεις κλειδιά

Python, σενάριο, βιωματική μάθηση, Pyboard, παιχνιδοποίηση.

Abstract

This diploma thesis attempts, through a brief bibliographical review, to outline a) the way that the programming languages work according to their category; b) the basic features of the Python programming language on which, according to researchers, it is based its use in education such as, for example, it is simple, easy to learn, free and open source, high-level, portable, interpreted with direct feedback to the learner, rich with extensive libraries and others; and (c) the apparent contribution of experiential learning, gamification and modern technologies to its teaching. The aim of the thesis is to design four teaching scenarios that constitute a new teaching proposal for the consolidation of specific teaching modules, incorporating elements of dynamic games, modern technology and approaching the subject that they are dealing with, in an experiential way in an authentic learning framework by solving problems of everyday life. The main teaching approach of the teaching scenarios is that of social constructivism. Finally, a study was conducted to assess the impact of the design of teaching scenarios on learner motivation. The sample of the survey was 15 undergraduate students of the Department of Informatics with Applications in Biomedicine of the University of Thessaly. In particular, 9 students participated in the first didactic scenario, 6 in the second didactic scenario, 10 in the 3rd didactic scenario and 5 in the fourth didactic scenario. After completing each teaching scenario, they completed an incentive-based questionnaire based on the CIS questionnaire survey, which was designed by Keller in 1987 with Likert's 5-step scale questions on the variables motives, attention-relativity, confidence-satisfaction that are the components of the Keller's ARCS motivation model. The results showed that designing learning scenarios by combining experiential learning, gamification and the use of modern technologies strengthens the motivation of learners and leads to the achievement of learning goals.

Key Words

Python, scenario, experiential learning, Pyboard, gamification.

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όλους τους ανθρώπους που με βοήθησαν στην ολοκλήρωση της παρούσας διπλωματικής εργασίας. Αρχικά, ευχαριστώ θερμά τον επίκουρο καθηγητή και επιβλέποντα της διπλωματικής μου εργασίας, κ. Κακαρούντα Αθανάσιο, για την καθοδήγησή του κατά την εκπόνησή της, όλους τους διδάσκοντές μου για τις γνώσεις που μου μετέδωσαν κατά τη διάρκεια της φοίτησής μου στο μεταπτυχιακό πρόγραμμα, την επιτροπή εξέτασης της εργασίας μου και τους φοιτητές που συμμετείχαν στα διδακτικά σενάρια. Τέλος, οφείλω ιδιαίτερες ευχαριστίες στα αγαπημένα πρόσωπα της οικογένειάς μου, στον πατέρα μου Πέτρο και στην μητέρα μου Βασιλική για την υποστήριξή τους, κυρίως στον σύζυγό μου και καθηγητή φυσικής Δημήτρη για την κατανόηση, την στήριξη και την συνεισφορά του στην εργασία μου παρέχοντας τις γνώσεις του για τη συνδεσμολογία των διαφόρων εξαρτημάτων που απαιτήθηκαν στη δοκιμαστική πλακέτα και τους γιους μου Γιώργο και Πέτρο για την υπομονή τους και αφιερώνω σε όλους με πολλή αγάπη τη διπλωματική μου εργασία.

Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή	13
1.1 Θεωρητική θεμελίωση	13
1.2 Στόχος της διπλωματικής εργασίας.....	15
1.3 Ερευνητικά ερωτήματα	15
1.4 Οργάνωση της διπλωματικής εργασίας	16
Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση.....	18
2.1 Ορισμός της Γλώσσας προγραμματισμού.....	18
2.2 Κατηγορίες των Γλωσσών προγραμματισμού	19
2.2.1 Προστακτικός Προγραμματισμός.....	19
2.2.2 Συναρτησιακός προγραμματισμός.....	21
2.2.3 Λογικός Προγραμματισμός	22
2.3 Γλώσσα προγραμματισμού Python.....	23
2.3.1 Σύντομη ιστορία της Python.....	23
2.3.2 Χαρακτηριστικά της Python	24
2.3.3 Python και Εκπαίδευση.....	26
2.3.4 Κατάταξη της Python σε σχέση με άλλες γλώσσες.....	30
2.4 Παραδοσιακή διδασκαλία και Βιωματική μάθηση.....	33
2.4.1 Βιωματική μάθηση (Παιχνιδοποίηση).....	33
2.4.2 Παραδοσιακή διδασκαλία και παιχνιδοποίηση στη διδασκαλία των γλωσσών προγραμματισμού.....	35
Κεφάλαιο 3: Διδακτικά σενάρια.....	38
3.1 Όρισμός διδακτικού σεναρίου.....	38
3.2 Διδακτικά σενάρια στη γλώσσα προγραμματισμού Python	38
3.2.1 Σκοπός και στόχος των διδακτικών σεναρίων.....	38

3.2.2	Οργάνωση της τάξης και χρήση Η/Υ και ψηφιακών μέσων για το διδακτικό σενάριο	38
3.2.3	Ένταξη διδακτικών σεναρίων στο πρόγραμμα σπουδών	39
3.2.4	Υποκείμενη θεωρία μάθησης	39
3.2.5	Περιγραφή διδακτικών σεναρίων	40
3.2.6	Διδακτικό συμβόλαιο – Διδακτικός θόρυβος	44
3.2.7	Χρήση εξωτερικών πηγών	44
3.3	Διδακτικό σενάριο 1: Εισαγωγικές έννοιες του προγραμματισμού και δομή ακολουθίας	45
3.3.1	Τίτλος διδακτικού σεναρίου	45
3.3.2	Εκτιμώμενη διάρκεια διδακτικού σεναρίου	45
3.3.3	Προαπαιτούμενες γνώσεις διδακτικού σεναρίου	45
3.3.4	Σκοποί και στόχοι του διδακτικού σεναρίου	45
3.3.5	Περιγραφή του διδακτικού σεναρίου	46
3.3.6	Αναπαραστάσεις εκπαιδευομένων / Πρόβλεψη δυσκολιών στο διδακτικό σενάριο	54
3.3.7	Επεκτάσεις / Διασυνδέσεις των εννοιών ή των δραστηριοτήτων	56
3.3.8	Φύλλα εργασίας 1, 2, 3 και 4	56
3.4	Διδακτικό σενάριο 2: Δομή επιλογής	67
3.4.1	Τίτλος διδακτικού σεναρίου	67
3.4.2	Εκτιμώμενη διάρκεια διδακτικού σεναρίου	67
3.4.3	Προαπαιτούμενες γνώσεις διδακτικού σεναρίου	67
3.4.4	Σκοποί και στόχοι του διδακτικού σεναρίου	67
3.4.5	Περιγραφή του διδακτικού σεναρίου	68
3.4.6	Αναπαραστάσεις εκπαιδευομένων / Πρόβλεψη δυσκολιών στο διδακτικό σενάριο	77
3.4.7	Επεκτάσεις / Διασυνδέσεις των εννοιών ή των δραστηριοτήτων	78

3.4.8	Φύλλα εργασίας 5, 6 ,7,8 και 9.	78
3.5	Διδακτικό σενάριο 3: Δομή επανάληψης.....	91
3.5.1	Τίτλος διδακτικού σεναρίου	91
3.5.2	Εκτιμώμενη διάρκεια διδακτικού σεναρίου.....	91
3.5.3	Προαπαιτούμενες γνώσεις διδακτικού σεναρίου.....	91
3.5.4	Σκοποί και στόχοι του διδακτικού σεναρίου	91
3.5.5	Περιγραφή του διδακτικού σεναρίου.....	92
3.5.6	Αναπαραστάσεις εκπαιδευομένων / Πρόβλεψη δυσκολιών στο διδακτικό σενάριο 100	
3.5.7	Επεκτάσεις / Διασυνδέσεις των εννοιών ή των δραστηριοτήτων.....	102
3.5.8	Φύλλα εργασίας 10, 11, 12 και 13.	102
3.6	Διδακτικό σενάριο 4: Λίστες.....	113
3.6.1	Τίτλος διδακτικού σεναρίου	113
3.6.2	Εκτιμώμενη διάρκεια διδακτικού σεναρίου.....	113
3.6.3	Προαπαιτούμενες γνώσεις διδακτικού σεναρίου.....	113
3.6.4	Σκοποί και στόχοι του διδακτικού σεναρίου	113
3.6.5	Περιγραφή του διδακτικού σεναρίου.....	113
3.6.6	Αναπαραστάσεις εκπαιδευομένων / Πρόβλεψη δυσκολιών στο διδακτικό σενάριο 117	
3.6.7	Επεκτάσεις / Διασυνδέσεις των εννοιών ή των δραστηριοτήτων.....	117
3.6.8	Φύλλα εργασίας 14,15 και 16.	117
Κεφάλαιο 4: Αξιολόγηση διδακτικών σεναρίων		124
4.1	Στόχος της έρευνας	124
4.2	Ερευνητικές μεταβλητές	124
4.3	Δείγμα της έρευνας	126
4.4	Μέσα συλλογής δεδομένων	126

4.5	Μέθοδος ανάλυσης δεδομένων	127
4.6	Ερευνητικά ερωτήματα	128
4.7	Ερευνητικές υποθέσεις.....	128
4.8	Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης.....	129
4.8.1	Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης 1ου Διδακτικού Σεναρίου	130
4.8.2	Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης 2ου Διδακτικού Σεναρίου	133
4.8.3	Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης 3ου Διδακτικού Σεναρίου	137
4.8.4	Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης 4ου Διδακτικού Σεναρίου	141
	Κεφάλαιο 5: Συμπεράσματα	146
	Βιβλιογραφία	147
	Ξενόγλωσση.....	147
	Ελληνόγλωσση.....	151
	Παράρτημα Α.....	153
	Παράρτημα Β.....	162
	Ευρετήριο.....	164

Κεφάλαιο 1: Εισαγωγή

1.1 Θεωρητική θεμελίωση

Οι ηλεκτρονικοί υπολογιστές σήμερα διαδραματίζουν πολύ σημαντικό ρόλο στον ψηφιακό κόσμο και στην καθημερινότητα των περισσότερων ανθρώπων. Αποτελούνται από το υλικό όπως πληκτρολόγια και οθόνες και το λογισμικό δηλαδή τα λειτουργικά συστήματα όπως τα Windows και τα προγράμματα εφαρμογών, όπως το Office. Το λογισμικό καθοδηγεί το υλικό για να παρέχει χρήσιμες λειτουργίες στους χρήστες, ενώ το υλικό παρέχει θεμελιώδη υποστήριξη στο λογισμικό. Ο προγραμματισμός είναι ο σχεδιασμός λογισμικού χρησιμοποιώντας ένα σύνολο βασικών οδηγιών και οι γλώσσες προγραμματισμού αναφέρονται σε οδηγίες υλικού μέσω βασικών δηλώσεων οι οποίες είναι πιο εύκολες στην κατανόηση και τη χρήση (Zhang, 2015).

Από τη δεκαετία του 1960 και μέχρι σήμερα, έχουν αναπτυχθεί διάφορα είδη προγραμματισμού τα οποία έχουν υποστηρίξει πολλές γλώσσες προγραμματισμού. Τα βασικά προγραμματιστικά υποδείγματα είναι ο Προστακτικός προγραμματισμός (Imperative programming) και ο Δηλωτικός προγραμματισμός (Declarative programming) στα οποία ανήκουν διάφορες υποκατηγορίες προγραμματισμού (Αράπογλου Α. , et al., 2016).

Τον Δεκέμβριο του 1989, ο ολλανδός προγραμματιστής Guido van Rossum αποφάσισε να συντάξει έναν διερμηνευτή για μια νέα γλώσσα προγραμματισμού, την «Python», (Tollervey, 2015) ως εικονική μηχανή έτσι ώστε να μπορεί να κάνει τη γλώσσα και τα προγράμματά της φορητά σε όλες τις πλατφόρμες. Η Python χρησιμοποιείται στον προγραμματισμό του διαδικτύου, στη δημιουργία σεναρίων για διακομιστές, στα γραφικά υπολογιστών, στην οπτικοποίηση, στα μαθηματικά, στην εκπαίδευση, στη μάθηση της μηχανικής, την τεχνητή νοημοσύνη, τα μεγάλα δεδομένα, τη ρομποτική και πολλούς άλλους τομείς (Lee K. , 2017). Τα βασικά χαρακτηριστικά της γλώσσας προγραμματισμού Python είναι ότι χαρακτηρίζεται ως: απλή, εύκολη στην εκμάθηση, ελεύθερη και ανοικτού κώδικα, υψηλού επιπέδου, φορητή, διερμηνευόμενη με άμεση ανατροφοδότηση προς τον εκπαιδευόμενο και με πλούσιες εκτεταμένες βιβλιοθήκες (Swaroop, 2008).

Ο χρόνος εκτέλεσης των προγραμμάτων της Python μπορεί να μην είναι πάντα τόσο γρήγορος όσο είναι στις μεταγλωττιζόμενες γλώσσες όπως η C και η C++ (Αγγελιδάκης, 2015). Ένα εργαλείο βελτιστοποίησης εκτέλεσης αποτελεί το σύστημα Cython που είναι μια υβριδική γλώσσα η οποία συνδυάζει τον κώδικα Python με τη δυνατότητα κλήσης

λειτουργιών C και τη χρήση δηλώσεων τύπου C για μεταβλητές, παραμέτρους και χαρακτηριστικά κλάσης (Lutz, 2013).

Η αξιοποίησή της γλώσσας προγραμματισμού Python στην εκπαίδευση στηρίζεται στα παραπάνω χαρακτηριστικά της αλλά και στο γεγονός ότι η κοινότητα της Python είναι μια ομάδα ανθρώπων που εκτιμούν το ανοιχτό πνεύμα, ασχολούνται ενεργά, η αποστολή τους είναι να προάγουν, να προστατεύουν και να προωθούν τη γλώσσα και να αφιερώνουν τον χρόνο τους για εκπαιδευτική υποστήριξη (Tollervey, 2015). Σύμφωνα με πρόσφατες έρευνες η Python βρίσκεται στις πρώτες θέσεις στην κατάταξη των κορυφαίων γλωσσών προγραμματισμού (Putano, 2017), (Cass, 2018).

Η διδασκαλία μιας γλώσσας προγραμματισμού είναι μια μεγάλη πρόκληση από μόνη της. Η επιτυχία της εξαρτάται σε μεγάλο βαθμό από πολλούς παράγοντες, όπως την εμπειρία προγραμματισμού του εκπαιδευτή, τις διδακτικές ικανότητές του και το είδος της παιδαγωγικής που χρησιμοποιεί στη τάξη (Kumar & Khurana, 2012). Ο εκπαιδευτικός οφείλει να λαμβάνει υπόψη τη συμπεριφορά και τα ενδιαφέροντα των εκπαιδευομένων, να παρέχει συνεχή ανατροφοδότηση, να αλληλεπιδρά με αυτούς, να χρησιμοποιεί αναδυόμενη τεχνολογία, να χρησιμοποιεί τις τελευταίες παιδαγωγικές μεθόδους, να προκαλεί το ενδιαφέρον τους και να συμβάλει στην αύξηση της συμμετοχής στην τάξη (Beeland, 2002). Τα κοινά παιδαγωγικά εργαλεία που διατίθενται από τον εκπαιδευτή είναι πίνακας, παρουσιάσεις PowerPoint, προσομοιώσεις και περιεχόμενο πολυμέσων όπως ηχητικά / βίντεο, κινούμενα σχέδια κλπ (Kumar & Khurana, 2012).

Η μάθηση είναι ένα πολυδιάστατο φαινόμενο και επηρεάζεται από πολλούς παράγοντες. Ένας από τους βασικότερους παράγοντες είναι τα κίνητρα (motivation). Η παιχνιδοποίηση (gamification) σε περιβάλλοντα μάθησης έχει αποδειχθεί ότι είναι μια επιτυχημένη επιλογή με σκοπό την αύξηση των κινήτρων των εκπαιδευομένων για συμμετοχή και μάθηση (Iosup & Epema, 2014), (Morrison & DiSalvo, 2014). Η μάθηση με τη χρήση παιχνιδιού είναι δύσκολη, προκλητική και αγχωτική, αλλά αν είναι καλά σχεδιασμένα τα παιχνίδια μπορούν να βοηθήσουν τους εκπαιδευόμενους να αποκτήσουν τις δεξιότητες, τις γνώσεις και τις ικανότητες που πρέπει σε σύντομο χρονικό διάστημα και αποτελεσματική ανάκληση (Karr, 2012).

Η παιχνιδοποίηση αποτελεί τον σχεδιασμό της αλληλεπίδρασης με ένα παιχνίδι που μπορεί πραγματικά να βοηθήσει τους εκπαιδευόμενους να μάθουν δύσκολες έννοιες του προγραμματισμού με αρκετά εύκολο και διασκεδαστικό τρόπο (Carroll & Thomas, 1988).

Τα τελευταία χρόνια στην Ελλάδα η εισαγωγή της γλώσσας προγραμματισμού Python στα εισαγωγικά μαθήματα προγραμματισμού στην Τριτοβάθμια Εκπαίδευση και στην Επαγγελματική Εκπαίδευση γεννά την ανάγκη για έναν κατάλληλο διδακτικό μετασχηματισμό της γλώσσας.

1.2 Στόχος της διπλωματικής εργασίας

Στόχος της παρούσας διπλωματικής αποτελεί ο σχεδιασμός τεσσάρων διδακτικών σεναρίων που αποτελούν μία νέα διδακτική πρόταση για την εμπέδωση συγκεκριμένων διδακτικών εννοιών της γλώσσας προγραμματισμού Python, ενσωματώνοντας στοιχεία δυναμικής παιχνιδιών, αναδυόμενη τεχνολογία και προσεγγίζοντας το θέμα που πραγματεύονται, με βιωματικό τρόπο σε ένα αυθεντικό πλαίσιο μάθησης επιλύοντας προβλήματα της καθημερινής ζωής. Η κύρια διδακτική προσέγγιση των διδακτικών σεναρίων είναι αυτή του κοινωνικού εποικοδομισμού.

Πιο συγκεκριμένα επιχειρήθηκε:

- ✓ ο σχεδιασμός ενός παιχνιδιού με χρωματιστές κάρτες το οποίο βασίζεται στον παράγοντα τύχη, το οποίο τροποποιείται σταδιακά από τη μια ενότητα ενός σεναρίου στην άλλη και από το ένα σενάριο στο άλλο όπου κάθε φορά προκύπτει ανάγκη χρήσης διαφορετικών εντολών με σκοπό την επίλυσή του,
- ✓ η παρουσίαση του ίδιου παιχνιδιού με τη χρήση της πλακέτας κυκλωμάτων MicroPython Pyboard v1.1,
- ✓ η χρήση της βιωματικής μάθησης με σκοπό την οικοδόμηση νέων γνώσεων πάνω στις ήδη υπάρχουσες,
- ✓ η διδασκαλία του γνωστικού αντικείμενου στο εργαστήριο πληροφορικής
- ✓ η ανάπτυξη της ψηφιακής εκδοχής που προσομοιώνει τη λειτουργία του παιχνιδιού στη γλώσσα προγραμματισμού Python και MicroPython,
- ✓ η αξιολόγηση των διδακτικών σεναρίων για το αν παρέχουν δυνατότητες για την ανάπτυξη κινήτρων για το αντικείμενο του προγραμματισμού.

1.3 Ερευνητικά ερωτήματα

Τα ερευνητικά ερωτήματα είναι τα παρακάτω:

1ο ερευνητικό ερώτημα

Ο σχεδιασμός των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας μπορεί να ενισχύσει τα **κίνητρα** των

εκπαιδευομένων με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python;

2ο ερευνητικό ερώτημα

Υπάρχει στατιστικά σημαντική επίδραση στην **προσοχή** και την **συσχέτιση** των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python;

3ο ερευνητικό ερώτημα

Υπάρχει στατιστικά σημαντική επίδραση στην **εμπιστοσύνη** και στην **ικανοποίηση** των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python;

1.4 Οργάνωση της διπλωματικής εργασίας

Στο πρώτο κεφάλαιο γίνεται η θεωρητική θεμελίωση της διπλωματικής εργασίας, παρουσιάζεται ο στόχος, τα ερευνητικά ερωτήματα και μια σύντομη παρουσίαση της δομής των κεφαλαίων.

Στο δεύτερο κεφάλαιο, μέσα από βιβλιογραφική ανασκόπηση σκιαγραφείται α) η χρησιμότητα των υπολογιστών στην καθημερινότητα των περισσότερων ανθρώπων, η ανάγκη δημιουργίας των γλωσσών προγραμματισμού και ο τρόπος λειτουργίας τους ανάλογα με το προγραμματιστικό υπόδειγμα στο οποίο ανήκουν, β) τα βασικά χαρακτηριστικά της γλώσσας προγραμματισμού Python στα οποία στηρίζεται, σύμφωνα με ερευνητές, η αξιοποίησή της στην εκπαίδευση, όπως για παράδειγμα ότι είναι: απλή, εύκολη στην εκμάθηση, ελεύθερη και ανοικτού κώδικα, υψηλού επιπέδου, φορητή, διερμηνευόμενη με άμεση ανατροφοδότηση προς τον εκπαιδευόμενο, με πλούσιες εκτεταμένες βιβλιοθήκες και άλλα και γ) η εμφανής συνεισφορά της βιωματικής μάθησης, της παιχνιδοποίησης και της αναδυόμενης τεχνολογίας στη διδασκαλία της.

Στο τρίτο κεφάλαιο, περιγράφονται τα τεσσάρων διδακτικά σενάρια που αποτελούν μία νέα διδακτική πρόταση για την εμπέδωση συγκεκριμένων διδακτικών ενοτήτων της, ενσωματώνοντας στοιχεία δυναμικής παιχνιδιών, αναδυόμενη τεχνολογία και

προσεγγίζοντας το θέμα που πραγματεύονται, με βιωματικό τρόπο σε ένα αυθεντικό πλαίσιο μάθησης επιλύοντας προβλήματα της καθημερινής ζωής. Η κύρια διδακτική προσέγγιση των διδακτικών σεναρίων είναι αυτή του κοινωνικού εποικοδομισμού.

Στο τέταρτο κεφάλαιο γίνεται η στατιστική ανάλυση των δεδομένων και αναφέρονται τα αποτελέσματα της αξιολόγησης του σχεδιασμού των διδακτικών σεναρίων από τους προπτυχιακούς φοιτητές του τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας ως προς την επίδρασή τους στην ενίσχυση των εσωτερικών κινήτρων των εκπαιδευομένων και συγκεκριμένα την προσοχή, σχετικότητα, εμπιστοσύνη και ικανοποίηση που αποτελούν τις συνιστώσες του μοντέλου κινήτρων ARCS του Keller.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα συμπεράσματα της έρευνας.

Στο τέλος, παρουσιάζονται η βιβλιογραφία, τα παραρτήματα και το ευρετήριο.

Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση

2.1 Ορισμός της Γλώσσας προγραμματισμού

Οι ηλεκτρονικοί υπολογιστές σήμερα διαδραματίζουν πολύ σημαντικό ρόλο στον ψηφιακό κόσμο και στην καθημερινότητα των περισσότερων ανθρώπων. Πιο συγκεκριμένα, οι υπολογιστές χρησιμοποιούνται σε διάφορα ψηφιακά συστήματα, όπως είναι οι έξυπνες τηλεοράσεις, τα τηλέφωνα, τα αυτοκίνητα, τα μετρό, τα αεροπλάνα, τα ρομπότ, τα εργοστάσια και άλλα.

Οι περισσότεροι άνθρωποι χρησιμοποιούν τους υπολογιστές καθημερινά στην εργασία τους μέσω αυτών των ψηφιακών συστημάτων και αποτελούν ένα απαραίτητο εργαλείο για αυτούς. Για παράδειγμα, οι διαχειριστές αμοιβαίων κεφαλαίων χρησιμοποιούν υπολογιστές για τις συναλλαγές τους, οι αρχιτέκτονες για την κατασκευή και επαλήθευση του σχεδιασμού ενός κτιρίου, οι γλωσσολόγοι για να μελετήσουν και να σχολιάσουν ένα κείμενο, οι επιστήμονες της φυσικής για την προσομοίωση μεγάλων πειραμάτων στη φυσική και τη χημεία και τα σούπερ μάρκετ για να προγραμματίσουν τον μικρότερο χρόνο παράδοσης μιας παραγγελίας στο σπίτι με τη συντομότερη διαδρομή. Επομένως, η χρήση υπολογιστών είναι χρήσιμη για όλες τις βιομηχανίες και για όλους τους κλάδους.

Οι υπολογιστές αποτελούνται από το υλικό και το λογισμικό. Το υλικό αναφέρεται σε φυσικά συστατικά, όπως τσιπ, ηλεκτρολογία και οθόνες, ενώ το λογισμικό αναφέρεται στα λειτουργικά συστήματα όπως τα Windows και τα προγράμματα εφαρμογών, όπως το Office και το Firefox. Το λογισμικό καθοδηγεί το υλικό για να παρέχει χρήσιμες λειτουργίες στους χρήστες, όπως επεξεργασία κειμένου, περιήγηση στο Internet και παιχνίδια. Από την άλλη πλευρά, το υλικό παρέχει θεμελιώδη υποστήριξη στο λογισμικό, προσφέροντας ένα σύνολο βασικών οδηγιών τις οποίες συνδυάζει το λογισμικό για την επίτευξη σύνθετης λειτουργικότητας.

Ο προγραμματισμός είναι ο σχεδιασμός λογισμικού χρησιμοποιώντας ένα σύνολο βασικών οδηγιών. Οι γλώσσες προγραμματισμού αναφέρονται σε οδηγίες υλικού μέσω βασικών δηλώσεων, οι οποίες είναι πιο εύκολες στην κατανόηση και τη χρήση. Χρησιμοποιώντας μια γλώσσα προγραμματισμού, μπορεί κανείς να γράψει σύνθετο λογισμικό, επιτυγχάνοντας προσαρμοσμένες λειτουργίες με συνδυασμό βασικών δηλώσεων. Αυτός είναι ο βασικός τρόπος χρήσης ενός υπολογιστή και είναι απαραίτητος όταν το υπάρχον λογισμικό δεν ικανοποιεί κάποιες απαιτήσεις, όπως η ανάγκη υπολογισμού νέων εξισώσεων ή προσομοίωση συγκεκριμένων πειραμάτων της μηχανικής (Zhang, 2015).

2.2 Κατηγορίες των Γλωσσών προγραμματισμού

Από τη δεκαετία του 1960 και μέχρι σήμερα, έχουν αναπτυχθεί διάφορα είδη προγραμματισμού τα οποία έχουν υποστηρίξει πολλές γλώσσες προγραμματισμού. Οι κατευθύνσεις στις οποίες κατηγοριοποιούνται ονομάζονται προγραμματιστικά υποδείγματα (programming paradigms). Τα βασικά προγραμματιστικά υποδείγματα αναφέρονται παρακάτω (Αράπογλου Α. , και συν., 2016):

- **Ο Προστακτικός προγραμματισμός (Imperative programming).** Στο γενικό αυτό υπόδειγμα ανήκουν διάφορες υποκατηγορίες προγραμματισμού, όπως είναι ο Δομημένος προγραμματισμός (Structured programming), ο μη Δομημένος προγραμματισμός (Unstructured programming), ο Διαδικαστικός προγραμματισμός (Procedural programming) και ο Αντικειμενοστραφής προγραμματισμός (Object-oriented programming).
- **Ο Δηλωτικός προγραμματισμός (Declarative programming).** Στο γενικό αυτό υπόδειγμα ανήκουν διάφορες υποκατηγορίες προγραμματισμού, όπως είναι ο Συναρτησιακός προγραμματισμός (Functional programming) και ο Λογικός προγραμματισμός (Logic programming).

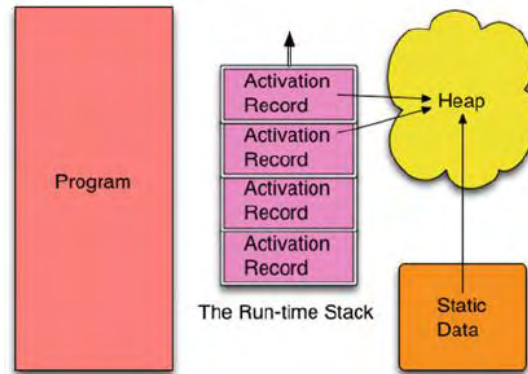
Παρακάτω περιγράφονται αναλυτικά τα βασικά προγραμματιστικά υποδείγματα σύμφωνα με τον (Lee K. , 2017):

2.2.1 Προστακτικός Προγραμματισμός

Δεδομένου ότι το μέγεθος των προγραμμάτων έχει αυξηθεί, κρίθηκε απαραίτητο να τροποποιηθεί η δομή τους. Αρχικά, μια συνάρτηση ονομαζόταν υπο-ρουτίνα. Συναρτήσεις, διαδικασίες και υπο-ρουτίνες εισήχθησαν από γλώσσες όπως η Algol 60, έτσι ώστε τα προγράμματα να μπορούν να διασπαστούν σε απλούστερα υποπρογράμματα, παρέχοντας τη δυνατότητα στους προγραμματιστές να οργανώνουν πιο εύκολα τον κώδικα τους. Όροι όπως η από πάνω προς τα κάτω ή από κάτω προς τα πάνω σχεδίαση χρησιμοποιήθηκαν για να περιγράψουν αυτή τη διαδικασία διάσπασης των προγραμμάτων σε απλούστερα υποπρογράμματα η οποία ονομάζεται δομημένος προγραμματισμός. Οι περισσότερες σύγχρονες γλώσσες παρέχουν τα μέσα για τη διάσπαση των προγραμμάτων σε απλούστερα υποπρογράμματα. Συχνά αναφέρουμε αυτήν την δομημένη προσέγγιση ως μοντέλο προστακτικού προγραμματισμού (Imperative programming).

Για την υλοποίηση και κλήση συναρτήσεων στην αρχιτεκτονική von Neumann, ήταν αναγκαία η οργάνωση των δεδομένων ενός προγράμματος. Στο μοντέλο προστακτικού

προγραμματισμού, η μνήμη χωρίζεται σε περιοχές που κατέχουν το πρόγραμμα και τα δεδομένα. Η περιοχή δεδομένων χωρίζεται περαιτέρω στη στατική ή την καθολική περιοχή δεδομένων, τη στοίβα χρόνου εκτέλεσης και τον σωρό όπως απεικονίζεται στην Εικόνα 1 .



Εικόνα 1. Προστακτικό μοντέλο¹

Στα τέλη της δεκαετίας του '70 και της δεκαετίας του '80 ο Niklaus Wirth και ο Bjarne Stroustrup ενδιαφέρθηκαν να αναπτύξουν γλώσσες προγραμματισμού που υποστηρίζουν ένα επιπλέον επίπεδο οργάνωσης που ονομάζεται αντικειμενοστραφής προγραμματισμός, συχνά συντομευμένο ως OOP (object-oriented programming). Ο αντικειμενοστραφής προγραμματισμός εξακολουθεί να χρησιμοποιεί το μοντέλο προστακτικού προγραμματισμού. Η προσθήκη ενός μέσου για να περιγράψει κλάσεις αντικειμένων δίνει στους προγραμματιστές τη δυνατότητα να οργανώσουν με έναν άλλο τρόπο τον κώδικα τους σε συναρτήσεις που σχετίζονται με συγκεκριμένο τύπο αντικειμένου.

Όταν ένα πρόγραμμα εκτελείται, χρησιμοποιεί ένα ειδικό καταχωρητή, που ονομάζεται δείκτης στοίβας (SP), που δείχνει την εγγραφή ενεργοποίησης η οποία βρίσκεται στην κορυφή της στοίβας χρόνου εκτέλεσης. Η στοίβα χρόνου εκτέλεσης περιέχει μία εγγραφή ενεργοποίησης για κάθε κλήση συνάρτησης ή διαδικασίας που αυτή τη στιγμή δεν έχει ολοκληρωθεί στο πρόγραμμα. Η εγγραφή ενεργοποίησης στην κορυφή της στοίβας αντιστοιχεί στην τρέχουσα κλήση συνάρτησης. Κατά την κλήση μιας συνάρτησης, ωθείται μια εγγραφή ενεργοποίησης στη στοίβα χρόνου εκτέλεσης. Μετά την εκτέλεση της συνάρτησης, απωθείται η εγγραφή ενεργοποίησης μειώνοντας τον δείκτη στοίβας για να δείξει την προηγούμενη εγγραφή ενεργοποίησης.

Μια εγγραφή ενεργοποίησης περιέχει πληροφορίες σχετικά με την αντίστοιχη συνάρτησή της, όπως οι τοπικές μεταβλητές της συνάρτησης, η τιμή του μετρητή προγράμματος πριν

¹ <https://link.springer.com/content/pdf/10.1007%2F978-3-319-70790-7.pdf>

από την κλήση της συνάρτησης, ο οποίος ονομάζεται συχνά διεύθυνση επιστροφής και άλλες πληροφορίες όπως οι παράμετροι που μεταβιβάζονται στη συνάρτηση.

Τα στατικά ή καθολικά δεδομένα αναφέρονται σε δεδομένα και συναρτήσεις που ισχύουν σε όλο το πρόγραμμα. Η περιοχή αποθήκευσης των καθολικών δεδομένων εξαρτάται από την εφαρμογή του μεταγλωττιστή ή του διερμηνευτή. Σε ορισμένες περιπτώσεις μπορεί να είναι μέρος του κώδικα του προγράμματος. Σε κάθε περίπτωση, σε αυτή την περιοχή αποθηκεύονται οι σταθερές, οι καθολικές μεταβλητές και, ενδεχομένως, οι ενσωματωμένες συναρτήσεις.

Ο σωρός είναι μια περιοχή για δυναμική παραχώρηση μνήμης. Η λέξη δυναμική σημαίνει ότι συμβαίνει όταν το πρόγραμμα εκτελείται. Όλα τα δεδομένα που δημιουργούνται κατά την εκτέλεση βρίσκονται στο σωρό. Τα δεδομένα στο σωρό δεν έχουν ονόματα σχετικά με τις τιμές που αποθηκεύονται εκεί. Οι μεταβλητές που ονομάζονται δείκτες ή αναφορές δείχνουν τα δεδομένα στο σωρό τα οποία μπορεί να περιέχουν δείκτες που δείχνουν σε άλλα δεδομένα, τα οποία είναι επίσης συνήθως στο σωρό.

Όπως και η αρχιτεκτονική von Neumann, ο πρωταρχικός στόχος του προστακτικού προγραμματισμού είναι η λήψη δεδομένων ως είσοδο, η μετατροπή τους μέσω ενημερώσεων στη μνήμη και, στη συνέχεια, η παραγωγή της εξόδου. Το προστακτικό μοντέλο χρησιμοποιείται από πολλές σύγχρονες γλώσσες. Ορισμένες παραλλαγές αυτού του μοντέλου χρησιμοποιούνται από γλώσσες όπως η Algol 60, C ++, C, Java, VB.net, Python και πολλές άλλες γλώσσες.

2.2.2 Συναρτησιακός προγραμματισμός

Το συναρτησιακό μοντέλο εστιάζει στην κλήση συναρτήσεων. Οι συναρτήσεις και οι παράμετροι είναι τα κύρια μέσα για τον μετασχηματισμό των δεδομένων.

Τα δεδομένα γενικά δεν αλλάζουν στο συναρτησιακό μοντέλο, αλλά είναι νέες τιμές που προκύπτουν από παλαιές τιμές. Ένα καθαρό συναρτησιακό μοντέλο δεν επιτρέπει καμία ενημέρωση στις υπάρχουσες τιμές. Ωστόσο, οι περισσότερες συναρτησιακές γλώσσες επιτρέπουν περιορισμένες ενημερώσεις στη μνήμη στο προστακτικό μοντέλο.

Η εννοιολογική άποψη που παρουσιάζεται στην εικόνα 1 είναι παρόμοια με την άποψη του συναρτησιακού κόσμου. Ωστόσο, η διαφορά μεταξύ προγράμματος και δεδομένων εξαλείφεται. Μια συνάρτηση είναι δεδομένα όπως οποιοδήποτε άλλο στοιχείο δεδομένων.

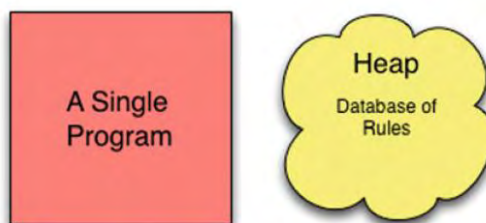
Η περιοχή στατικών δεδομένων εξακολουθεί να υπάρχει, αλλά δεν παίζει πολύ σημαντικό ρόλο στο συναρτησιακό μοντέλο. Η στοίβα χρόνου εκτέλεσης γίνεται πιο σημαντική, επειδή οι περισσότερες εργασίες ολοκληρώνονται καλώντας συναρτήσεις. Οι συναρτησιακές γλώσσες είναι πολύ πιο προσεκτικές για το πώς επιτρέπουν στους προγραμματιστές να έχουν πρόσβαση στον σωρό και, ως εκ τούτου, χωρίς να ασχολούνται ιδιαίτερα με τον σωρό όταν προγραμματίζουν σε μια συναρτησιακή γλώσσα. Τα δεδομένα είναι σίγουρα δυναμικά, αλλά μόλις δημιουργηθούν τα δεδομένα στο σωρό δεν τροποποιούνται σε ένα καθαρό συναρτησιακό μοντέλο. Τα μη καθαρά μοντέλα ενδέχεται να επιτρέψουν κάποια τροποποίηση της αποθήκευσης, λόγω της επιρροής των προστακτικών γλωσσών που εισέρχονται στο συναρτησιακό μοντέλο.

Η Lisp, Scheme, Scala, Clojure, Elixir, Haskell, Caml και Standard ML είναι παραδείγματα συναρτησιακών γλωσσών. Μερικές συναρτησιακές γλώσσες μπορεί να μην υποστηρίζουν ενημερώσεις των μεταβλητών π.χ. η γλώσσα προγραμματισμού Scheme σε αντίθεση με τις περισσότερες συναρτησιακές γλώσσες προγραμματισμού που τις υποστηρίζουν, όπως η Standard ML και η Lisp. Η Scala είναι μια πρόσφατη συναρτησιακή γλώσσα που υποστηρίζει επίσης αντικειμενοστραφή προγραμματισμό.

2.2.3 Λογικός Προγραμματισμός

Το λογικό μοντέλο, που απεικονίζεται στην Εικόνα 2, είναι αρκετά διαφορετικό από το προστακτικό ή συναρτησιακό μοντέλο, γιατί ο προγραμματιστής δεν γράφει στην πραγματικότητα ένα πρόγραμμα. Συγκεκριμένα, ο προγραμματιστής παρέχει μια βάση γεγονότων ή κανόνων. Από αυτή τη βάση δεδομένων, ένα μόνο πρόγραμμα προσπαθεί να απαντήσει σε ερωτήσεις με απάντηση ένα «Ναι» ή ένα «Όχι». Στην περίπτωση της Prolog, το πρόγραμμα λειτουργεί με ένα προβλέψιμο τρόπο όπου ο προγραμματιστής παρέχει τα γεγονότα με μια σειρά που καθορίζει τον τρόπο με τον οποίο θα εκτελεστεί το πρόγραμμα. Η πραγματική εφαρμογή αυτής της εννοιολογικής άποψης επιτυγχάνεται με μια εικονική μηχανή.

Υπάρχει ακόμα η έννοια ενός σωρού στην Prolog. Κάποιος μπορεί να προσθέσει νέους κανόνες και να ανακαλέσει κανόνες. Για να προστεθούν δυναμικά κανόνες ή να ανακληθούν πρέπει να υπάρχει ένας σωρός. Στην πραγματικότητα, υπάρχει και η έννοια της στοίβας χρόνου εκτέλεσης. Ωστόσο, η στοίβα χρόνου εκτέλεσης και ο σωρός είναι τόσο κρυμμένοι όπου αμφισβητείται για το εάν θα πρέπει να εμφανίζονται στο εννοιολογικό μοντέλο.



Εικόνα 2. Λογικό μοντέλο²

2.3 Γλώσσα προγραμματισμού Python

2.3.1 Σύντομη ιστορία της Python

Τον Δεκέμβριο του 1989, ένας ολλανδός προγραμματιστής που ονομαζόταν Guido van Rossum αναζητούσε ένα «χόμπι» για να απασχοληθεί κατά τη διάρκεια των Διακοπών των Χριστουγέννων. Αποφάσισε, λοιπόν, να συντάξει έναν διερμηνευτή για μια νέα γλώσσα προγραμματισμού που είχε σκεφτεί την οποία ονόμασε «Python» από μία βρετανική κωμωδία που ονομαζόταν «Monty Python's Flying Circus» (Tollervey, 2015). Η ιστορία αυτής της γλώσσας είναι διαθέσιμη στο διαδίκτυο³.

Ο κ. Van Rossum σχεδίασε τον διερμηνευτή της Python ως εικονική μηχανή. Η εικονική μηχανή της Python είναι εσωτερική λεπτομέρεια υλοποίησης του διερμηνευτή της Python. Χρησιμοποιώντας μια εικονική μηχανή κατά την εφαρμογή μιας γλώσσας προγραμματισμού μπορεί να κάνει τη γλώσσα και τα προγράμματά της φορητά σε όλες τις πλατφόρμες. Η Python εκτελείται σε πολλές διαφορετικές πλατφόρμες όπως τα Mac OS X, Linux και Microsoft Windows της Apple. Οι εικονικές μηχανές μπορούν επίσης να παρέχουν υπηρεσίες που διευκολύνουν την υλοποίηση της γλώσσας.

Οι προγραμματιστές παγκοσμίως έχουν αγκαλιάσει την Python, έχουν αναπτύξει πολλές βιβλιοθήκες για αυτή και έχουν γράψει πολλά προγράμματα. Η Python έχει κερδίσει δημοτικότητα μεταξύ προγραμματιστών λόγω της φορητότητάς της και της δυνατότητας να παρέχει βιβλιοθήκες σε άλλους. Ο Guido van Rossum δηλώνει στην ιστορία του για την Python: «Ένα μεγάλο σύνθετο σύστημα πρέπει να έχει πολλαπλά επίπεδα επεκτασιμότητας. Αυτό μεγιστοποιεί τις ευκαιρίες για τους χρήστες, εξελιγμένα ή όχι, για να βοηθηθούν». Η επεκτασιμότητα αναφέρεται στην ικανότητα προσδιορισμού των βιβλιοθηκών των κλάσεων για την επίλυση προβλημάτων από πολλούς διαφορετικούς τομείς. Η Python χρησιμοποιείται στον προγραμματισμό του διαδικτύου, στη δημιουργία σεναρίων για διακομιστές, στα

² <https://link.springer.com/content/pdf/10.1007%2F978-3-319-70790-7.pdf>

³ <http://python-history.blogspot.com>

γραφικά υπολογιστών, στην οπτικοποίηση, στα μαθηματικά, στην εκπαίδευση, και πολλούς άλλους τομείς.

Ο κ. Van Rossum συνεχίζει, λέγοντας «Η φιλοσοφία του σχεδιασμού που χρησιμοποιήθηκε κατά τη δημιουργία της Python είναι πιθανώς ένας από τους κύριους λόγους για την απόλυτη επιτυχία της. Παρατηρήθηκε ότι η Python λειτούργησε «αρκετά καλά» για τους σκοπούς της. Καθώς η βάση των χρηστών μεγάλωνε, οι προτάσεις για βελτίωση ενσωματώθηκαν σταδιακά στη γλώσσα». Η ανάπτυξη της βάσης των χρηστών υπήρξε το κλειδί για την επιτυχία της Python. Ο αριθμός των προγραμματιστών που γνωρίζουν την Python έχει αυξηθεί και δείχνει το ενδιαφέρον για τη βελτίωση της γλώσσας. Η Python έχει τώρα δύο σημαντικές εκδόσεις Python 2 και Python 3. Η Python 3 δεν είναι συμβατή με την Python 2. Αυτή η διακοπή της συμβατότητας έδωσε στους προγραμματιστές της Python την ευκαιρία να κάνουν βελτιώσεις στη γλώσσα (Lee K. , 2017).

2.3.2 Χαρακτηριστικά της Python

Η Python είναι μια εύκολη στην εκμάθηση και ισχυρή γλώσσα προγραμματισμού. Έχει αποδοτικές δομές δεδομένων υψηλού επιπέδου και μια απλή αλλά αποτελεσματική προσέγγιση στον αντικειμενοστρεφή προγραμματισμό. Η κομψή σύνταξη της Python και οι δυναμικοί τύποι της, μαζί με τη λειτουργία της ως διερμηνευόμενη (αντί μεταγλωττιζόμενης) γλώσσα, την καθιστούν την ιδανική γλώσσα για δημιουργία σεναρίων και για ταχεία ανάπτυξη εφαρμογών σε πολλούς τομείς και στις περισσότερες πλατφόρμες. Αναλυτικά τα περισσότερα από τα χαρακτηριστικά της είναι τα εξής (Swaroop, 2008):

- **Απλή:** Είναι μια απλή και μινιμαλιστική γλώσσα που η ομοιότητά της με ψευδοκώδικα είναι ένα από τα πιο ισχυρά σημεία της.
- **Εύκολη στην εκμάθηση:** Έχει μια ασυνήθιστα απλή σύνταξη.
- **Ελεύθερη και Ανοικτού Κώδικα:** Είναι ένα παράδειγμα ΕΛΛΑΚ (Ελεύθερο Λογισμικό και Λογισμικό Ανοικτού Κώδικα) που σημαίνει ότι υπάρχει δυνατότητα διανομής αντιγράφων αυτού του λογισμικού, διαβάσματος του πηγαίου κώδικά του, δημιουργίας αλλαγών σ' αυτό και χρησιμοποίησης κομματιών του σε νέα ελεύθερα προγράμματα. Το ΕΛΛΑΚ βασίζεται στην ιδέα μιας κοινότητας που μοιράζεται τη γνώση.
- **Γλώσσα υψηλού επιπέδου:** Κατά τη συγγραφή προγραμμάτων δεν απαιτούνται γνώσεις για τις χαμηλού επιπέδου λεπτομέρειες όπως η διαχείριση της μνήμης που χρησιμοποιείται από τα προγράμματα.

- **Φορητή:** Έχει υλοποιηθεί, λόγω του ανοικτού της κώδικα, σε πολλές πλατφόρμες που σημαίνει ότι όλα τα προγράμματα μπορούν να εκτελεστούν σε οποιαδήποτε από αυτές τις πλατφόρμες χωρίς να χρειάζονται καθόλου αλλαγές.
- **Διερμηνευόμενη:** Ένα πρόγραμμα που γράφεται σε μια μεταγλωττιζόμενη γλώσσα όπως η C ή η C++ μετατρέπεται από την πηγαία γλώσσα, για παράδειγμα τη C ή τη C++ σε μια γλώσσα του υπολογιστή (δυναμικός κώδικας δηλαδή 0 και 1) χρησιμοποιώντας ένα μεταγλωττιστή. Ο συνδέτης αντιγράφει το πρόγραμμα στη μνήμη και αρχίζει να το εκτελεί. Η Python, αντίθετα, δε χρειάζεται μεταγλώττιση σε δυαδικό αρχείο. Απλά εκτελεί το πρόγραμμα απ' ευθείας από τον πηγαίο κώδικα. Εσωτερικά, η Python μετατρέπει τον πηγαίο κώδικα σε μια ενδιάμεση μορφή που ονομάζεται bytecode και μετά το μεταφράζει στη γλώσσα του υπολογιστή και το εκτελεί. Επομένως, η χρήση της Python είναι πολύ πιο εύκολη αφού δε χρειάζεται η μεταγλώττιση του προγράμματος, η σύνδεση με τις κατάλληλες βιβλιοθήκες, κ.λπ. και τα προγράμματα της Python είναι εξαιρετικά φορητά.
- **Αντικειμενοστρεφής:** Υποστηρίζει τόσο τον διαδικασιοστρεφή προγραμματισμό (procedure-oriented) όσο και τον αντικειμενοστρεφή προγραμματισμό (object-oriented). Στο διαδικασιοστρεφή προγραμματισμό, το πρόγραμμα δομείται πάνω σε διαδικασίες ή συναρτήσεις οι οποίες είναι επαναχρησιμοποιήσιμα κομμάτια από προγράμματα. Στις αντικειμενοστρεφείς γλώσσες, τα προγράμματα δομούνται πάνω σε αντικείμενα τα οποία συνδυάζουν δεδομένα και λειτουργικότητα.
- **Επεκτάσιμη:** Ένα τμήμα κώδικα εκτελείται πολύ γρήγορα αν προγραμματιστεί σε C ή C++ και μετά χρησιμοποιηθεί από την Python.
- **Ενσωματώσιμη:** Ενσωματώνεται μέσα στα προγράμματα σε C/C++ με σκοπό τη δυνατότητα «scripting» για τους χρήστες.
- **Εκτεταμένες βιβλιοθήκες:** Η Πρότυπη βιβλιοθήκη της Python είναι πραγματικά τεράστια και μπορεί να βοηθήσει σε θέματα σχετικά με κανονικές εκφράσεις, δημιουργία τεκμηρίωσης, δοκιμές μονάδων, νημάτωση, βάσεις δεδομένων, περιηγητές ιστού, CGI, FTP, email, XML, XML-RPC, HTML, αρχεία WAV, κρυπτογράφηση, γραφικές διεπαφές χρήστη (GUI -graphical user interfaces), Tk, και άλλα.
- **Χρόνος εκτέλεσης:** Ο χρόνος εκτέλεσης των προγραμμάτων της Python μπορεί να μην είναι πάντα τόσο γρήγορος όσο είναι στις μεταγλωττιζόμενες (compiled) γλώσσες όπως η C και η C++. Αυτό οφείλεται στο ότι ένα πρόγραμμα σε Python δεν μεταγλωττίζεται σε δυαδικό κώδικα μηχανής που εκτελείται άμεσα από τον επεξεργαστή του υπολογιστή.

Το μειονέκτημα αυτό αντισταθμίζεται από το ότι πολλές φορές είναι σημαντικότερη η εξοικονόμηση χρόνου που έχουμε κατά την ανάπτυξη μιας εφαρμογής σε Python (Αγγελιδάκης, 2015).

Ένα εργαλείο βελτιστοποίησης εκτέλεσης αποτελεί το σύστημα Cython⁴ (βασισμένο στην εργασία του έργου Pyrex) που είναι μια υβριδική γλώσσα η οποία συνδυάζει τον κώδικα Python με τη δυνατότητα κλήσης λειτουργιών C και τη χρήση δηλώσεων τύπου C για μεταβλητές, παραμέτρους και χαρακτηριστικά κλάσης. Ο κώδικας Cython μπορεί να συνταχθεί στον κώδικα C που χρησιμοποιεί την Python / C API, ο οποίος στη συνέχεια μπορεί να συνταχθεί πλήρως. Αν και όχι πλήρως συμβατό με το πρότυπο Python, η Cython μπορεί να είναι χρήσιμη τόσο για την χρήση εξωτερικών βιβλιοθηκών της C και για την κωδικοποίηση των αποτελεσματικών επεκτάσεων C για την Python (Lutz, 2013).

2.3.3 Python και Εκπαίδευση

Γενικά, η Python θεωρείται ως εξαιρετική επιλογή ως πρώτη γλώσσα προγραμματισμού για λόγους όπως (Tollervey, 2015):

- **Αναγνωσιμότητα**

Κάθε εντολή στον κώδικα είναι στη δική του γραμμή και είναι προφανές ποιες γραμμές κώδικα σχετίζονται μεταξύ τους μόνο με την εμφάνισή τους, δηλαδή με τον τρόπο που ο κώδικας είναι στοιχισμένος. Οι περισσότερες γλώσσες προγραμματισμού χρησιμοποιούν σύμβολα και όχι εσοχές για την ένδειξη του πεδίου εφαρμογής. Για παράδειγμα, πολλές γλώσσες όπως η Java, JavaScript και C χρησιμοποιούν άγκιστρα και ερωτηματικά για το σκοπό αυτό. Η έλλειψη συμβόλων όπως '{', '}' και ';' στον κώδικα καθιστούν πολύ πιο εύκολη την ανάγνωση του κώδικα σε Python. Σε σύγκριση με τις άλλες γλώσσες, η σύνταξη της Python είναι απλή και εύκολη στην κατανόησή της, γιατί δεν απαιτεί μεγάλη προσπάθεια για την ανάγνωση και τη συγγραφή του κώδικα. Επομένως, οτιδήποτε μειώνει την προσπάθεια που απαιτείται στον προγραμματισμό είναι θετικό σε ένα εκπαιδευτικό πλαίσιο.

- **Απλότητα**

Οι βασικές έννοιες και οι γνώσεις που απαιτούνται για τη συγγραφή της Python είναι απλές, εύκολες στην εκμάθηση, στη χρήση και την απομνημόνευση. Επιπλέον, η Python είναι μια

⁴ <http://cython.org>

γλώσσα προγραμματισμού που θα αντιδράσει αν ο προγραμματιστής κάνει κάποιο λάθος και ονομάζει διάφορες έννοιες χρησιμοποιώντας κοινή κατανοητή αγγλική ορολογία.

▪ **Επεκτασιμότητα**

Οι προγραμματιστές πρέπει συχνά να επαναχρησιμοποιούν τις υπάρχουσες βιβλιοθήκες για την επίτευξη ενός κοινού έργου. Μια βιβλιοθήκη είναι σαν ένα βιβλίο συνταγών με οδηγίες για την εκτέλεση ορισμένων σχετικών εργασιών. Αυτό σημαίνει ότι οι προγραμματιστές δεν χρειάζεται να ξεκινούν από το μηδέν ούτε να ανακαλύπτουν ξανά τον τροχό κάθε φορά που αντιμετωπίζουν ένα κοινό πρόβλημα. Ενώ οι περισσότερες γλώσσες προγραμματισμού έχουν μηχανισμούς για να γράφουν και να επαναχρησιμοποιούν βιβλιοθήκες κώδικα, η Python έχει μια μεγάλη και εκτεταμένη τυποποιημένη βιβλιοθήκη.

Πολλές γλώσσες έχουν εκτεταμένες βιβλιοθήκες κώδικα και πλαίσια εφαρμογής, αλλά η ισχύς της Python είναι ευρεία. Τα SciPy και NumPy χρησιμοποιούνται από επιστήμονες και μαθηματικούς, το NLTK χρησιμοποιείται από τους γλωσσολόγους για την ανάλυση του κειμένου, το Pandas χρησιμοποιείται εκτενώς από τους στατιστικολόγους και το OpenStack χρησιμοποιείται για την οργάνωση και τον έλεγχο υπολογιστικών πόρων που βασίζονται σε cloud. Η διδασκαλία μιας γλώσσας που έχει τόσο μεγάλη χρήση σε πραγματικό κόσμο έχει ένα προφανές όφελος: οι μαθητές αποκτούν μια δεξιότητα σε μια γλώσσα προγραμματισμού που έχει πραγματική οικονομική αξία. Μια άλλη πτυχή του να είναι ανοιχτά επεκτάσιμη είναι ότι η Python είναι ένα έργο ανοιχτού κώδικα. Ο καθένας μπορεί να συμμετάσχει στην ανάπτυξη της γλώσσας υποβάλλοντας αναφορές σφαλμάτων στους κύριους προγραμματιστές (με επικεφαλή τον ίδιο τον Guido van Rossum). Υπάρχει επίσης μια κατανοητή και απλή διαδικασία με την οποία προτείνονται και υλοποιούνται νέα χαρακτηριστικά της γλώσσας: οι Προτάσεις Ενίσχυσης Python (PEPs)⁵. Με αυτό τον τρόπο, η γλώσσα αναπτύσσεται με πλήρη εικόνα της κοινότητας που την χρησιμοποιεί με την ευκαιρία της κοινότητας να ενημερώνει και να καθοδηγεί τη μελλοντική της ανάπτυξη.

▪ **Διαλειτουργικότητα**

Η Python λειτουργεί σε Microsoft Windows, Mac OS X, Linux και πολλά άλλα λειτουργικά συστήματα και συσκευές. Αυτό είναι σημαντικό σε ένα εκπαιδευτικό πλαίσιο, επειδή η Python λειτουργεί στους υπολογιστές που χρησιμοποιούνται στα σχολεία. Οι μαθητές μπορούν επίσης να την χρησιμοποιήσουν στους υπολογιστές που έχουν στο σπίτι τους, ανεξάρτητα από τη μάρκα ή το μοντέλο που έχουν. Οι διαχειριστές του σχολικού

⁵ <https://www.python.org/dev/peps/pep-0001/>

εργαστηρίου πολλές φορές δεν επιτρέπουν στους εκπαιδευτικούς να εγκαταστήσουν τίποτα εκτός από το Microsoft Office σε σχολικούς υπολογιστές. Η Python ως υπηρεσία που παρέχεται μέσω ιστοτόπου είναι μια εξαιρετική λύση στο πρόβλημα αυτό. Οι χρήστες απλώς με έναν φυλλομετρητή επισκέπτονται έναν ιστότοπο και χρησιμοποιούν την Python χωρίς να εγκαταστήσουν κανένα πρόσθετο λογισμικό.

Η Python λειτουργεί επίσης σε μικροσυσκευές, όπως το Raspberry Pi. Εκτελείται ακόμη και σε μικροελεγκτές, χαμηλής ισχύος, που έχουν σχεδιαστεί για να εκτελούνται μέσα σε ενσωματωμένες συσκευές, τηλεχειριστήρια και παιχνίδια. Το έργο MicroPython δημιούργησε μια βελτιστοποιημένη έκδοση της Python 3, για τέτοιες συσκευές, και παρέχει μια μικρή πλακέτα κυκλωμάτων όπως απεικονίζεται στην Εικόνα 3 η οποία εκτελεί μια λιτή έκδοση της Python, τη γλώσσα MicroPython.



Εικόνα 3. MicroPython πλακέτα⁶

Αυτό το πολύ απλό λειτουργικό σύστημα που βασίζεται στην Python μπορεί να χρησιμοποιηθεί ως βάση για όλα τα ενδιαφέροντα και διασκεδαστικά ηλεκτρονικά έργα. Στην πλακέτα μπορούν να συνδεθούν οθόνη LCD, ηχείο, μικρόφωνο και κινητήρες. Η δημιουργία ενός απλού ρομπότ με μια τέτοια συσκευή είναι σχετικά εύκολη. Πιο πρόσφατα, το BBC ανακοίνωσε το έργο MicroBit, μια προγραμματιζόμενη συσκευή για παιδιά όπως απεικονίζεται στην Εικόνα 4. Ακριβώς όπως το Raspberry Pi και οι MicroPython πλακέτες, έχει την ικανότητα να αλληλεπιδρά με άλλες συσκευές μέσω συνδέσεων I / O.



Εικόνα 4. MicroBit προγραμματιζόμενη συσκευή για παιδιά⁷

⁶ <https://www.cs.rutgers.edu/~biglars/python/python-in-education.pdf>

Η Python είναι μία από τις τρεις υποστηριζόμενες γλώσσες για τον προγραμματισμό του MicroBit. Το σημαντικό εκπαιδευτικό πλεονέκτημα είναι ότι με την εκμάθηση της Python, ένας μαθητής μπορεί να εξερευνήσει διάφορες πλατφόρμες χρησιμοποιώντας εργαλεία και κώδικα που είναι οικεία σε αυτόν.

▪ **Ανθρωπότητα**

Η κοινότητα της Python είναι μια ομάδα ανθρώπων που εκτιμούν το ανοιχτό πνεύμα, ασχολούνται ενεργά, η αποστολή τους είναι να προάγουν, να προστατεύουν και να προωθούν τη γλώσσα και να αφιερώνουν τον χρόνο τους για εκπαιδευτική υποστήριξη. Αυτά είναι όλα τα χαρακτηριστικά που διευκολύνουν τόσο τους εκπαιδευτικούς όσο και τους μαθητές να εμπλακούν με τον πολιτισμό της Python.

Όπως ανέφερε ο Eben Upton από το Ίδρυμα Raspberry Pi, η κοινότητα Python είναι «ακριβώς αυτό που θέλετε. Η εκπαίδευση είναι ένα βασικό κομμάτι της κοινότητας». Οι προγραμματιστές της Python (που ονομάζονται Pythonists, Pythonistas ή / και Pythonauts) είναι επίσης μια καλά οργανωμένη ομάδα και έχουν δημιουργήσει το Python Software Foundation (PSF) ως σημείο συγκέντρωσης της κοινότητας. Σημαίνει, επίσης, ότι υπάρχει νομική οντότητα με την οποία οι κυβερνήσεις, οι εταιρείες και άλλα ιδρύματα μπορούν να αλληλεπιδρούν επισήμως.

Επιπλέον λόγοι που αναφέρονται σε διάλεξη του (Δημητριάδης, 2016) αποτελούν σύμφωνα με τον (Guo P. , 2008):

- Όμορφη, καθαρή σύνταξη, και συνοπτική σε σωστό βαθμό ώστε ούτε να προκαλεί οπτική υπερφόρτωση ούτε σύγχυση ή ελλιπή κατανόηση.
- Διερμηνευόμενη, άρα η εκτέλεση εκκινεί άμεσα και παρέχεται άμεση ανάδραση στον μαθητή (καθώς κάποια σφάλματα εμφανίζονται αργότερα στο χρόνο εκτέλεσης), σε αντίθεση με μεταγλωττιζόμενες (compiled) γλώσσες στατικού τύπου όπου η εκτέλεση δεν εκκινεί αν δεν διορθωθούν όλα τα συντακτικά σφάλματα (κάτι που καθυστερεί και αποθαρρύνει τον μαθητή).
- Αλληλεπιδραστική γραμμή εντολών (command prompt), ώστε να παρέχεται άμεση ανατροφοδότηση κατά την εκτέλεση απλών εντολών κώδικα.
- Δεν υπάρχει απαίτηση για δήλωση μεταβλητών κάτι που απλοποιεί τη συγγραφή και εκτέλεση του κώδικα για τον αρχάριο.

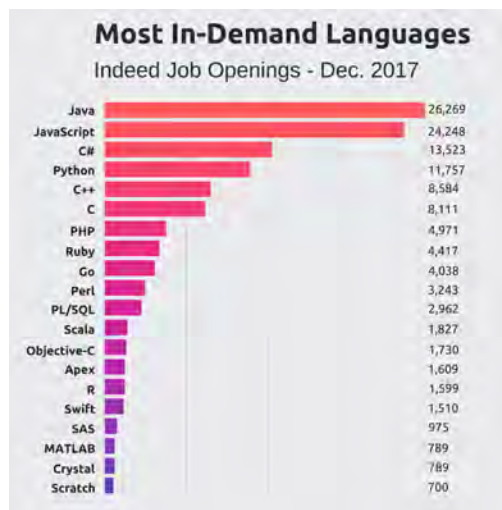
⁷ <https://www.cs.rutgers.edu/~biglars/python/python-in-education.pdf>

2.3.4 Κατάταξη της Python σε σχέση με άλλες γλώσσες.

Οι πιο έγκυρες μετρήσεις για την εξέταση των κορυφαίων γλωσσών προγραμματισμού είναι οι εξής (Putano, 2017):

- **Indeed.com**

Η Indeed είναι η μεγαλύτερη μηχανή αναζήτησης εργασίας στον κόσμο που αντιπροσωπεύει μια καλή μέτρηση των θέσεων εργασίας που σχετίζονται με τις πιο δημοφιλείς γλώσσες προγραμματισμού. Συγκεκριμένα, εξετάστηκε ο αριθμός των θέσεων εργασίας για τις 50 κορυφαίες γλώσσες προγραμματισμού σύμφωνα με τον Δείκτη Tiobe.



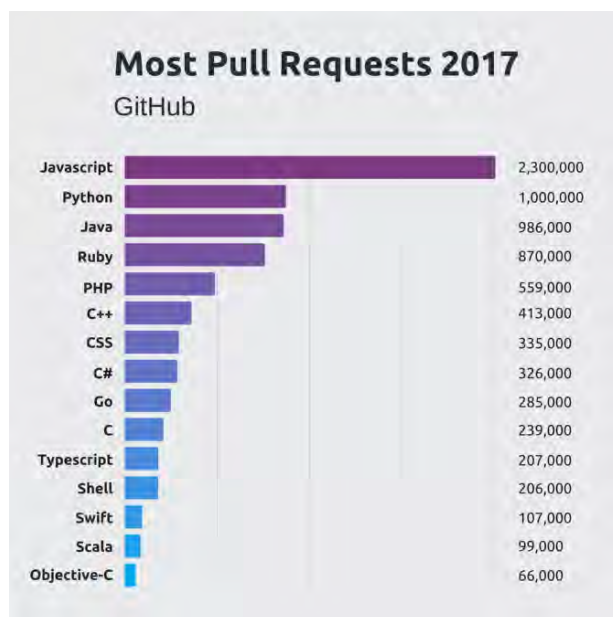
Εικόνα 5. Ευκαιρίες εργασίας με αντικείμενο τις δημοφιλείς γλώσσες προγραμματισμού⁸

Το παραπάνω γράφημα, όπως απεικονίζεται στην εικόνα 5, καταγράφει τις ευκαιρίες εργασίες που σχετίζονται με τις 50 πιο δημοφιλείς γλώσσες προγραμματισμού από το ευρετήριο Tiobe όπου αναζητήθηκε το «("Language name") Developer». Είναι λογικό οι πιο δημοφιλείς γλώσσες να έχουν περισσότερες ευκαιρίες εργασίας με την Python να βρίσκεται η Python βρίσκεται στην τέταρτη θέση. Ωστόσο, είναι ενδιαφέρον να βλέπουμε λιγότερο δημοφιλείς γλώσσες, όπως η Apex (Salesforce μητρική γλώσσα), στις 20 πρώτες γλώσσες το οποίο μπορεί να δείχνει μια ανοδική τάση στη δημοτικότητα αυτών των γλωσσών.

- **GitHub**

Το GitHub είναι ένα από τα μεγαλύτερα αποθετήρια κώδικα που χρησιμοποιούνται σήμερα. Κάθε χρόνο δημιουργείται μια έκθεση «Year in Review», μοιράζοντας στατιστικά στοιχεία σχετικά με την δική τους κοινότητα προγραμματιστών. Η μέτρησή τους που ονομάζεται «Pull Requests Opened» είναι ένας άλλος δείκτης δημοτικότητας των γλωσσών.

⁸ <https://stackify.com/popular-programming-languages-2018/>



Εικόνα 6. Αιτήματα έλξης των γλωσσών προγραμματισμού⁹

Το Github, όπως απεικονίζεται στην εικόνα 6, δίνει τα κορυφαία 15 αιτήματα έλξης από την κοινότητά τους. Τα αιτήματα έλξης είναι ένας δείκτης του ποσού του κώδικα που γράφεται. Σύμφωνα με την OSS-Watch, «Το αίτημα έλξης είναι μια μέθοδος υποβολής συνεισφορών σε ένα ανοιχτό αναπτυξιακό έργο.» Το Javascript έχει ένα τεράστιο αποτύπωμα στο GitHub, με περισσότερο από το διπλάσιο αριθμό αιτήσεων έλξης από τη δεύτερη κορυφαία γλώσσα. Το 2017, η Python ξεπέρασε την Java ως τη δεύτερη πιο δημοφιλή γλώσσα με αιτήματα έλξης.

Η δημοτικότητα της Python έχει αυξηθεί σταθερά τα τελευταία 15 χρόνια, καταλήγοντας τελικά στην κορυφή του δείκτη Tiobe πριν από μερικά χρόνια. Αυτό συμβαίνει επειδή η Python είναι μια σημαντική γλώσσα σε ορισμένες από τις τεχνολογίες σήμερα. Η μάθηση της μηχανικής, η τεχνητή νοημοσύνη, τα μεγάλα δεδομένα και η ρομποτική βασίζονται σε μεγάλο βαθμό στην Python (η ρομποτική βασίζεται επίσης στην C για τη χρήση της στον προγραμματισμό συστημάτων). Η Cyber Security, μία από τις κορυφαίες προκλήσεις λογισμικού της εποχής μας, καθοδηγείται επίσης από την Python. Είναι εκπληκτικό το πόσο απλή είναι η μάθηση της Python. Είναι πλέον η πιο δημοφιλής εισαγωγική γλώσσα που διδάσκεται στα πανεπιστήμια και συχνά χρησιμοποιείται από έμπειρους προγραμματιστές ως δεύτερη ή τρίτη γλώσσα.

- **IEEE Spectrum** (Cass, 2018)

⁹ <https://stackify.com/popular-programming-languages-2018/>

Η πέμπτη ετήσια διαλογική κατάταξη των κορυφαίων γλωσσών προγραμματισμού ως αποτέλεσμα συνδυασμού μετρήσεων από πολλές πηγές σε 47 γλώσσες.

Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥️ 📱	100.0
2. C++	📱 🖥️ 📱	99.7
3. Java	🌐 📱 🖥️	97.5
4. C	📱 🖥️ 📱	96.7
5. C#	🌐 📱 🖥️	89.4
6. PHP	🌐	84.9
7. R	🖥️	82.9
8. JavaScript	🌐 📱	82.6
9. Go	🌐 🖥️	76.4
10. Assembly	📱	74.1

Εικόνα 7. Κατάταξη των κορυφαίων γλωσσών προγραμματισμού (IEEE Spectrum)¹⁰

Η Python έχει διατηρηθεί στην πρώτη θέση, όπως απεικονίζεται στην εικόνα 7. Τώρα η C++ είναι η γλώσσα που ανέβηκε στους 99,7 βαθμούς, ενώ ο C μετακινείται στην τέταρτη θέση με 96,7 βαθμούς.

Η Python είναι στην κορυφή, γιατί εμφανίζεται ως ενσωματωμένη γλώσσα. Προηγουμένως, η γραφή για ενσωματωμένες εφαρμογές είχε μεγάλη κλίση προς τις μεταγλωττισμένες γλώσσες, για να αποφευχθεί η επιβάρυνση της εκτίμησης κώδικα σε κίνηση σε μηχανές με περιορισμένη ισχύ επεξεργασίας και μνήμη. Αλλά ενώ ο νόμος του Moore μπορεί να ξεθωριάζει, δεν είναι ακόμα νεκρός. Πολλοί σύγχρονοι μικροελεγκτές έχουν πλέον αρκετή ισχύ για να φιλοξενήσουν έναν διερμηνέα της Python. Μια ωραία πτυχή της χρήσης της Python με αυτόν τον τρόπο είναι ότι είναι πολύ βολική σε ορισμένες εφαρμογές η χρήση του συνημμένου υλικού μέσω της διάδρασης ή η δυναμική ενεργοποίηση της λειτουργίας των scripts.

Ένας άλλος λόγος για την αυξανόμενη δημοτικότητα της Python μπορεί να είναι η μικρή πτώση της R που έφτασε στο No. 5 το 2016, έπεσε στο No. 6 πέρυσι και τώρα βρίσκεται στην έβδομη θέση. Η R είναι μια γλώσσα εξειδικευμένη για τη διαχείριση στατιστικών στοιχείων και μεγάλων δεδομένων. Δεδομένου ότι το ενδιαφέρον για μεγάλα σύνολα δεδομένων έχει στραφεί όλο και περισσότερο στις εφαρμογές τους στη μηχανική μάθηση, στην ύπαρξη υψηλής ποιότητας βιβλιοθηκών της Python τόσο για στατιστικές όσο και για μάθηση της μηχανικής ίσως καθιστά την ευέλικτη Python πιο ελκυστική από ότι την πιο εξειδικευμένη R.

¹⁰ <https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>

2.4 Παραδοσιακή διδασκαλία και Βιωματική μάθηση.

2.4.1 Βιωματική μάθηση (Παιχνιδοποίηση)

Το περιβάλλον των εκπαιδευομένων σήμερα (Maia Filev Rodrigo & Graeml Reis Felipe , 2015) είναι διαφορετικό σε σχέση με κάποιες δεκαετίες πριν και οι εκπαιδευτικοί απαιτείται να εφαρμόσουν διαφορετικές μεθόδους για τη βελτιστοποίηση της απόδοσής τους. Τα παιχνίδια αποτελούν μέρος της ζωής των εκπαιδευομένων σήμερα και σίγουρα μπορούν να συνεισφέρουν στην απόδοσή τους αυξάνοντας το ενδιαφέρον τους, τη διασκέδασή τους και τα κίνητρα για τη μάθηση και προωθώντας τη συμμετοχή τους στη δική τους διαδικασία μάθησης.

Οι (Gioia & Brass, 1985), στο άρθρο «Teaching the TV Generation», υπογραμμίζουν ότι η διδασκαλία και η μάθηση βελτιώνονται όταν το στυλ διδασκαλίας είναι ανάλογο με το στυλ μάθησης. Το 1986, οι συγγραφείς πρότειναν ότι οι εκπαιδευόμενοι που μεγαλώνουν σε ένα περιβάλλον που κυριαρχείται από οπτικές εικόνες δεν θα πρέπει πλέον να διδάσκονται μόνο με λεκτικό τρόπο. Οι (Proserpio & Gioia, 2007), στο άρθρο «Teaching the Virtual Generation», σημειώνουν ότι η διδασκαλία δεν απευθύνεται σε μια λεκτική, ούτε σε μια οπτική γενιά, αλλά σε μια εικονική. Η ιδέα που υποστηρίζουν είναι ότι η βέλτιστη διδασκαλία και μάθηση συμβαίνουν όταν το στυλ διδασκαλίας είναι ευθυγραμμισμένο με το στυλ μάθησης.

Οι (Brown & Vaughan, 2009) υποστηρίζουν ότι το παιχνίδι προφανώς δεν έχει κάποιο νόημα, δεν φαίνεται να έχει καμία πρακτική αξία, αλλά έχει έμφυτη έλξη, είναι διασκεδαστικό και παρέχει ψυχολογική διέγερση. Αν και το παιχνίδι περιλαμβάνει αναμονή, προσδοκία, ερώτηση, περιέργεια, άγχος και αβεβαιότητα, οδηγεί σε έκπληξη, απροσδόκητες καταστάσεις, ανακάλυψη, νέες αισθήσεις και προοπτικές αλλαγής, δημιουργώντας ευχαρίστηση και καλή αίσθηση (Brown & Vaughan, 2009). Το παιχνίδι δίνει ελευθερία για αυτοσχεδιασμό, παρέχει μια ελευθερία από τον πιεσμένο χρόνο, σταματά την ανησυχία, οδηγεί στη συνειδητοποίηση ενός διαφορετικού εαυτού (Brown & Vaughan, 2009).

Ο (Kapp, 2012) σημειώνει ότι οι άνθρωποι παίζουν παιχνίδια επειδή είναι διασκεδαστικά και έχουν τη δυνατότητα να ενημερώνουν και εκπαιδεύουν. Όταν το παιχνίδι είναι δομημένο από κανόνες, στόχους, προκλήσεις και προωθεί την αλληλεπίδραση των συμμετεχόντων ορίζεται ως «παιχνίδι». Ένα παράδειγμα σύμφωνα με τον (Kapp, 2012), απεικονίζει τις δραστηριότητες όπου δίνονται βαθμοί, ανταμοιβές και εμβλήματα για καθημερινά πράγματα που γίνονται στο χώρο εργασίας ή στο σπίτι. Ωστόσο, (Kapp, 2012) επισημαίνει ότι η

πραγματική δύναμη της σκέψης που βασίζεται στο παιχνίδι είναι η αφοσίωση, η αφήγηση, η οπτικοποίηση των χαρακτήρων και η επίλυση του προβλήματος.

Ο (Karr, 2012) δεν προσπαθεί να παραπλανήσει, υποστηρίζει ότι η μάθηση με τη χρήση παιχνιδιού είναι δύσκολη, προκλητική και αγχωτική, αλλά αν είναι καλά σχεδιασμένα τα παιχνίδια μπορούν να βοηθήσουν τους εκπαιδευόμενους να αποκτήσουν τις δεξιότητες, τις γνώσεις και τις ικανότητες που πρέπει σε σύντομο χρονικό διάστημα και αποτελεσματική ανάκληση. Επιπλέον, η μάθηση με τη χρήση παιχνιδιού επιταχύνει την καμπύλη εμπειρίας της μάθησης, επιτρέπει την επίτευξη πολύπλοκων θεμάτων και παρέχει μια συστηματική όψη της όλης διαδικασίας (Karr, 2012).

Αρκετές μελέτες αποδεικνύουν ότι δεν υπάρχει πλέον ένα στερεότυπο παίκτη. Οποιοσδήποτε μπορεί να είναι παίκτης παιχνιδιών ανεξαρτήτων ηλικίας, όχι μόνο παιδιά ή έφηβοι (Kark, 2011) και (Association, 2014). Άλλες μελέτες αποδεικνύουν ότι οι εκπαιδευόμενοι που εμπλέκονται σε βιωματική μάθηση έχουν την τάση να επιτυγχάνουν αποτελεσματική μάθηση (Kolb, 2014) και (Gentry, 1990).

Οι (Camarero, Rodríguez, & San José. , 2010), υποστηρίζουν ότι οι εκπαιδευόμενοι μπορούν να προσεγγίσουν μια εργασία με ενεργό πειραματισμό (υλοποίηση) ή από αντανακλαστική παρατήρηση (παρακολούθηση). Για τον (Hawtreay, 2007), η πειραματική μάθηση συμβαίνει κάθε φορά που ο εκπαιδευόμενος αλλάζει το ρόλο του παθητικού ακροατή σε εκείνο του ενεργού ερωτώμενου.

Επομένως, η παιχνιδοποίηση (Sprint & Cook, 2015) περιλαμβάνει τη χρήση στοιχείων σχεδίασης παιχνιδιών σε ένα πλαίσιο που δεν αφορά κάποιο παιχνίδι με σκοπό να συμμετάσχουν οι παίκτες σε μία διαδραστική πρόκληση (Deterding, Dixon, Khaled, & Nacke, 2011). Ένας εξειδικευμένος ορισμός της παιχνιδοποίησης στη μάθηση από τον (Karr, 2012) έχει ως εξής: «Η παιχνιδοποίηση χρησιμοποιεί μηχανική και αισθητική βασισμένη στο παιχνίδι και τη σκέψη των παιχνιδιών με σκοπό την αφοσίωση των ανθρώπων, την παρακίνηση για δράση, την προώθηση της μάθησης και την επίλυση προβλημάτων». Στην τάξη, η παιχνιδοποίηση χρησιμοποιεί ένα εκπαιδευτικό παιχνίδι με σκοπό την εκπαίδευση και την ποσοτικοποίηση της προόδου του μαθητή. Ένα εκπαιδευτικό παιχνίδι είναι ένα σύστημα που προωθεί τη μάθηση ενώ εξακολουθούν να εμπλέκονται οι παίκτες (σε αυτό το πλαίσιο, οι εκπαιδευόμενοι) σε μια πρόκληση που διέπεται από κανόνες, ανατροφοδότηση και ένα μαθησιακό αποτέλεσμα. Η παιχνιδοποίηση σε περιβάλλοντα μάθησης αποδεικνύεται ότι είναι μια επιτυχημένη επιλογή με σκοπό την αύξηση των

κινήτρων των μαθητών για συμμετοχή και μάθηση (Iosup & Erema, 2014), (Morrison & DiSalvo, 2014).

2.4.2 Παραδοσιακή διδασκαλία και παιχνιδοποίηση στη διδασκαλία των γλωσσών προγραμματισμού

Σήμερα, η βιομηχανία πληροφορικής (Kumar & Khurana, 2012) παρέχει μια δυναμική και ισχυρή καριέρα σε προγραμματιστές, αλλά η πλειοψηφία των αποφοίτων της πληροφορικής δεν έχουν τις απαραίτητες ικανότητες για να ανταποκριθούν στις προσδοκίες της βιομηχανίας. Έχει παρατηρηθεί ότι τέτοιοι φοιτητές δεν επικεντρώνονται πολύ στον προγραμματισμό υπολογιστών κατά τη διάρκεια των σπουδών τους. Ο κύριος λόγος της έλλειψης ενδιαφέροντος στον προγραμματισμό είναι η έλλειψη κινήτρων και η δέσμευση των φοιτητών να μάθουν τις έννοιες προγραμματισμού. Σύμφωνα με τους (Kumar & Khurana, 2012) μια λύση είναι η εισαγωγή τη παιχνιδοποίησης στη διδασκαλία του προγραμματισμού. Δεδομένου ότι τα παιχνίδια μπορούν να χρησιμοποιηθούν για να δημιουργήσουν ένα περιβάλλον με κίνητρα στην τάξη, η μέθοδος αυτή θα βελτιώνει συνολικά τη διαδικασία διδασκαλίας και μάθησης. Η συμμετοχή των φοιτητών και τα κίνητρα για την εκμάθηση του προγραμματισμού μετρήθηκαν χρησιμοποιώντας μια έρευνα βασισμένη σε ένα ερωτηματολόγιο στο οποίο συμμετείχαν 207 φοιτητές ενός μεταπτυχιακού προγράμματος πληροφορικής (University of Punjab). Το αποτέλεσμα της έρευνας και του ερωτηματολογίου έδειξε μια ισχυρή προτίμηση για τη χρήση της παιχνιδοποίησης στην οικοδόμηση των γνώσεων του προγραμματισμού.

Η συμμετοχή των φοιτητών στα μαθήματα είναι πάντα ένα σοβαρό θέμα στα ακαδημαϊκά ιδρύματα που μπορεί να οφείλεται στον καταναγκασμό των φοιτητών, στο επίπεδο δυσκολίας του μαθήματος ή λόγω της σοβαρότητας των φοιτητών στην καριέρα τους. Οι επιμελείς φοιτητές σημαίνουν περισσότερη επιτυχία όσον αφορά την κατανόηση της γνώσης και καλό μέλλον στην καριέρα τους (Beeland, 2002). Η διδασκαλία ενός μαθήματος προγραμματισμού είναι μια μεγάλη πρόκληση από μόνη της. Η επιτυχία του εξαρτάται σε μεγάλο βαθμό από πολλούς παράγοντες, όπως την εμπειρία προγραμματισμού του εκπαιδευτή, τις διδακτικές ικανότητές του και το είδος της παιδαγωγικής που χρησιμοποιεί στις τάξεις. Τα κοινά παιδαγωγικά εργαλεία που διατίθενται με τον εκπαιδευτή είναι πίνακας, παρουσιάσεις PowerPoint, προσομοιώσεις και περιεχόμενο πολυμέσων όπως ηχητικά / βίντεο, κινούμενα σχέδια κλπ.

Οι (Lee & Hammer, 2011) προσδιόρισαν τρεις βασικούς τομείς στους οποίους η παιχνιδοποίηση μπορεί να χρησιμεύσει ως μια σημαντική πτυχή στην καριέρα του

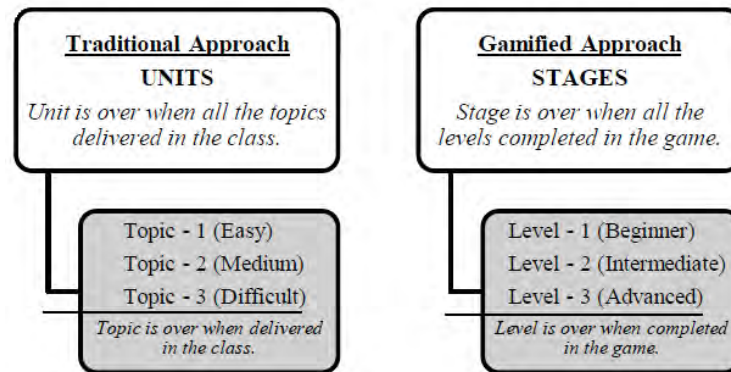
εκπαιδευόμενου που μπορεί να διαδραματίσει σημαντικό κινητήριο ρόλο εκπληρώνοντας τις ψυχικές, ευαίσθητες και δημόσιες ανάγκες του. Το κίνητρο δεν είναι τίποτε άλλο εκτός από την καλύτερη αντιμετώπιση στο θέμα της συμμετοχής του χρήστη σε οποιαδήποτε δραστηριότητα. Έχει ειπωθεί από τον (Boekaerts, 2002) στη δημοσίευσή του «Motivation to Learn» ότι ο εκπαιδευτικός διαδραματίζει πάντα σημαντικό ρόλο στην καριέρα του εκπαιδευόμενου μέσω των κινήτρων και της αλληλεπίδρασης.

Εάν ο εκπαιδευτικός είναι καλός σε κάτι τότε μπορεί πραγματικά να βοηθήσει τους εκπαιδευόμενους να κάνουν περισσότερα από αυτόν. Έτσι ένας καλός εκπαιδευτικός πρέπει να εξετάσει τη συμπεριφορά των φοιτητών στην τάξη, όπως το εάν αισθάνονται ενθαρρυμένοι ή αποθαρρυμένοι, τι τους αρέσει ή τι δεν τους αρέσει και περισσότερο κυρίως τι βρίσκουν ενδιαφέρον και προφανώς βαρετό στην τάξη. Η ανατροφοδότηση είναι πολύ σημαντική όταν ο εκπαιδευτικός αλληλεπιδρά με τους εκπαιδευόμενους της τάξης. Καθώς η τεχνολογία αναδύεται πολύ γρήγορα αυτές τις μέρες, είναι ευθύνη των εκπαιδευτών να χρησιμοποιούν τις τελευταίες παιδαγωγικές μεθόδους που προκαλούν το ενδιαφέρον των μαθητών στην τάξη (Beeland, 2002) και αυτό θα συμβάλει στην αύξηση της συμμετοχής στην τάξη.

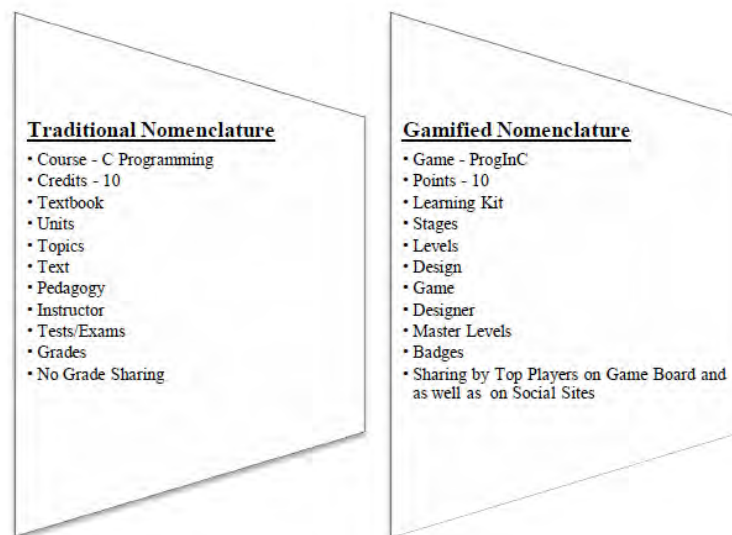
Η ολοκλήρωση μιας εργασίας με τη διασκεδαστική μάθηση είναι σημαντικά καλύτερη. Το αποτελεσματικό παιδαγωγικό εργαλείο (Cisco, 2008) πρέπει να περιλαμβάνει μια ποικιλία μέσων, στάδια αλληλεπίδρασης, πτυχές μάθησης και, κυρίως, διδασκαλία που γίνεται από τον εκπαιδευτικό για σύνθετα περιβάλλοντα για να γίνει η μάθηση καλύτερη από κάθε άποψη.

Η εμπειρία και η μάθηση (Gee, 2008) είναι οι δύο παράγοντες που κινούνται παράλληλα. Οι άνθρωποι που θέλουν να λύσουν μια μελλοντική πτυχή του προβλήματος πρέπει να κάνουν μέγιστη χρήση αυτού του μηχανισμού. Αυτή η έννοια εφαρμόζεται πλήρως στα παιχνίδια. Ένα παιχνίδι μπορεί να θεωρηθεί ως αποτελεσματικό και γεμάτο μάθηση εάν περιέχει ένα είδος εμπειριών που ένας παίκτης μπορεί να αποκτήσει κατά τη διάρκεια της συμμετοχής του στο παιχνίδι. Κάποιος μπορεί να ορίσει την παιχνιδοποίηση ως έναν σχεδιασμό αλληλεπίδρασης με ένα παιχνίδι (Carroll & Thomas, 1988) που μπορεί πραγματικά να βοηθήσει τους εκπαιδευόμενους να μάθουν δύσκολες έννοιες του προγραμματισμού με αρκετά εύκολο και διασκεδαστικό τρόπο. Η Microsoft (McDonald, Musson, & Smith, 2008) είναι ένα μια δημοφιλής εταιρεία που έχει διερευνήσει πως η χρήση τυχερών παιχνιδιών μπορεί να μετασχηματίσει την κατανόηση από το ένα περιβάλλον στο άλλο. Στις εικόνες 8

και 9 φαίνεται η αντιστοιχία της παραδοσιακής προσέγγισης και της παιχνιδοποίησης και της ονοματολογίας που χρησιμοποιείται.



Εικόνα 8. Παραδοσιακή προσέγγιση και παιχνιδοποίηση¹¹



Εικόνα 9. Ισοδυναμία ονοματολογίας των δύο προσεγγίσεων¹²

¹¹ http://www.academia.edu/7826506/Gamification_in_Education_Learn_Programming_with_Fun

¹² http://www.academia.edu/7826506/Gamification_in_Education_Learn_Programming_with_Fun

Κεφάλαιο 3: Διδακτικά σενάρια

3.1 Όρισμός διδακτικού σεναρίου

Το διδακτικό σενάριο είναι η περιγραφή μιας διδασκαλίας με εστιασμένο γνωστικό αντικείμενο, συγκεκριμένους εκπαιδευτικούς στόχους, διδακτικές αρχές και πρακτικές στο οποίο περιλαμβάνονται και στοιχεία όπως οι ρόλοι του εκπαιδευτικού και των εκπαιδευομένων, η αλληλεπίδρασή τους με τα χρησιμοποιούμενα μέσα και υλικό, οι αντιλήψεις των εκπαιδευομένων, τα ενδεχόμενα διδακτικά εμπόδια και γενικότερα όλα εκείνα τα στοιχεία που θεωρούνται σημαντικά στη σύγχρονη θεωρία. Ένα διδακτικό σενάριο μπορεί να διαρκέσει περισσότερες από μία διδακτικές ώρες και υλοποιείται συνήθως μέσα από εκπαιδευτικές δραστηριότητες αυξανόμενου βαθμού δυσκολίας (ΙΤΥΕ, 2013).

3.2 Διδακτικά σενάρια στη γλώσσα προγραμματισμού Python

3.2.1 Σκοπός και στόχος των διδακτικών σεναρίων

Σκοπός της συγκεκριμένης διπλωματικής εργασίας είναι η σύνταξη τεσσάρων διδακτικών σεναρίων που αφορούν στην εκμάθηση βασικών εννοιών και δομών του προγραμματισμού και συγκεκριμένα στη γλώσσα προγραμματισμού Python.

Στόχος του κάθε διδακτικού σεναρίου είναι να αποτελέσει μία νέα διδακτική πρόταση για την εμπέδωση μιας διδακτικής ενότητας της γλώσσας προγραμματισμού Python και να προσεγγίσει το θέμα που πραγματεύεται, με βιωματικό τρόπο σε ένα αυθεντικό πλαίσιο μάθησης επιλύοντας ένα πρόβλημα της καθημερινής ζωής, για παράδειγμα παίζοντας ένα παιχνίδι, το οποίο τροποποιείται σταδιακά από τη μια ενότητα ενός σεναρίου στην άλλη και από το ένα σενάριο στο άλλο όπου κάθε φορά προκύπτει ανάγκη χρήσης διαφορετικών εντολών με σκοπό την επίλυσή του.

3.2.2 Οργάνωση της τάξης και χρήση Η/Υ και ψηφιακών μέσων για το διδακτικό σενάριο

Τα διδακτικά σενάρια προτείνεται να λάβουν χώρα σε εργαστήριο πληροφορικής όπου οι εκπαιδευόμενοι θα εργαστούν σε ομάδες των δύο ατόμων ανά ηλεκτρονικό υπολογιστή. Για τη διδασκαλία των διδακτικών σεναρίων απαιτούνται:

- Εργαστήριο πληροφορικής, στο οποίο θα πρέπει να έχει εγκατασταθεί η έκδοση της γλώσσας προγραμματισμού Python 3.6¹³ ή σύνδεση στο διαδίκτυο και ένας φυλλομετρητής έτσι ώστε να χρησιμοποιηθεί διαδικτυακά¹⁴.

¹³ <https://www.python.org/downloads/>

- Βιντεοπροβολέας, ώστε να μπορεί ο εκπαιδευτικός να προβάλλει το περιβάλλον της Python.
- Πλακέτα MicroPython Pyboard v1.1, μια συμπαγής και ισχυρή πλακέτα ανάπτυξης ηλεκτρονικών συσκευών που εκτελεί τη γλώσσα MicroPython, Usb καλώδιο, διάφορα καλώδια (Jumper wires), πλακέτα δοκιμών (Breadboard), Leds διαφόρων χρωμάτων, RGB leds, αντιστάσεις (Resistors) και πιεζόμενα κουμπιά (Momentary push buttons).
- Σύνδεση στο διαδίκτυο και ένας φυλλομετρητής έτσι ώστε να χρησιμοποιηθεί διαδικτυακά η γλώσσα MicroPython¹⁵ αν κριθεί απαραίτητο από τον εκπαιδευτικό ή να χρησιμοποιηθεί διαδικτυακά η πλακέτα¹⁶ για την εκτέλεση κάποιων scripts που παρέχονται από την ιστοσελίδα της MicroPython.

3.2.3 Ένταξη διδακτικών σεναρίων στο πρόγραμμα σπουδών

Τα διδακτικά σενάρια απευθύνονται κυρίως σε προπτυχιακούς και μεταπτυχιακούς φοιτητές που φοιτούν σε τμήματα των οποίων το πρόγραμμα σπουδών περιλαμβάνει τη συγκεκριμένη γλώσσα προγραμματισμού όπου συνήθως προτείνεται το περιβάλλον προγραμματισμού Python IDLE της έκδοσης Python 3.6. Ανάλογα με την έκδοση που χρησιμοποιείται απαιτείται προσοχή σε κάποιες διαφοροποιήσεις της.

3.2.4 Υποκείμενη θεωρία μάθησης

Η κύρια διδακτική προσέγγιση των διδακτικών σεναρίων είναι ο κοινωνικός εποικοδομισμός. Χρησιμοποιώντας κατάλληλα απλά παραδείγματα επιδιώκεται οι εκπαιδευόμενοι να αντιληφθούν τις διάφορες έννοιες του προγραμματισμού και να είναι ικανοί να αξιολογούν και οργανώνουν τις έννοιες αυτές σε νοητικές δομές, συστηματοποιώντας και ολοκληρώνοντας τις πρότερες εμπειρίες τους. Σύντομες υλοποιήσεις σε μια γλώσσα προγραμματισμού που γίνονται στο εργαστήριο, παρέχουν στους εκπαιδευόμενους τη δυνατότητα δόμησης και ελέγχου της εγκυρότητας του εκάστοτε οικοδομήματος για μια έννοια και αποτελούν ταυτόχρονα σημείο κοινής αναφοράς, τρόπο επιβεβαίωσης της ορθής αντίληψης που αναπτύχθηκε και ευκαιρία για περαιτέρω εμβάθυνση και εισαγωγή νέων εννοιών.

Τα σενάρια, προτείνεται να πραγματοποιούνται ακολουθώντας μεθόδους αναζήτησης και ανακάλυψης για την οικοδόμηση της γνώσης. Όσον αφορά τις εκπαιδευτικές τεχνικές

¹⁴ <https://www.tutorialspoint.com/codingground.htm>

¹⁵ <https://micropython.org/unicorn/>

¹⁶ <http://micropython.org/live/>

προτείνεται να ακολουθούνται οι πλέον συμμετοχικές ενεργητικές εξ αυτών, με τη συχνή χρήση αυθεντικών παραδειγμάτων από τον πραγματικό κόσμο. Ο εκπαιδευτικός λειτουργεί ως διευκολυντής για την οικοδόμηση της γνώσης από τον εκπαιδευόμενο, κινούμενος με τεχνικές «προοδευτικά μειούμενης στήριξης – scaffolding» (Αράπογλου Α. , Βραχνός, Κανίδης, Μακρυγιάννης, Μπελεσιώτης, & Τζήμας, 2015).

Οι εκπαιδευόμενοι εργαζόμενοι σε ομάδες, όπως υποστηρίζουν οι θεωρίες του Vygotsky, θα έχουν τη δυνατότητα να αλληλεπιδράσουν σε μεγάλο βαθμό μεταξύ τους, να αντιμετωπίσουν κοινωνιογνωστικές συγκρούσεις και έτσι να αναγκαστούν να διατυπώσουν τις γνώμες τους, να εκφράσουν απόψεις, να επιχειρηματολογήσουν, να διαφωνήσουν και να εμπλουτίσουν έτσι και να εκλεπτύνουν τις αρχικές τους αντιλήψεις (Κοτίνη & Τζαλέπη, 2013).

3.2.5 Περιγραφή διδακτικών σεναρίων

Τα βήματα που προτείνεται να ακολουθήσει ο εκπαιδευτικός στην αρχή των διδακτικών σεναρίων περιγράφονται παρακάτω:

Βήμα 1: περιγράφει αρχικά ένα απλό παιχνίδι με χρωματιστές κάρτες με σκοπό την εκμάθηση του προγραμματισμού με τη χρήση της βιωματικής μάθησης και οι εκπαιδευόμενοι καλούνται να παίξουν το παιχνίδι το οποίο περιγράφεται αναλυτικά.

Περιγραφή παιχνιδιού¹⁷

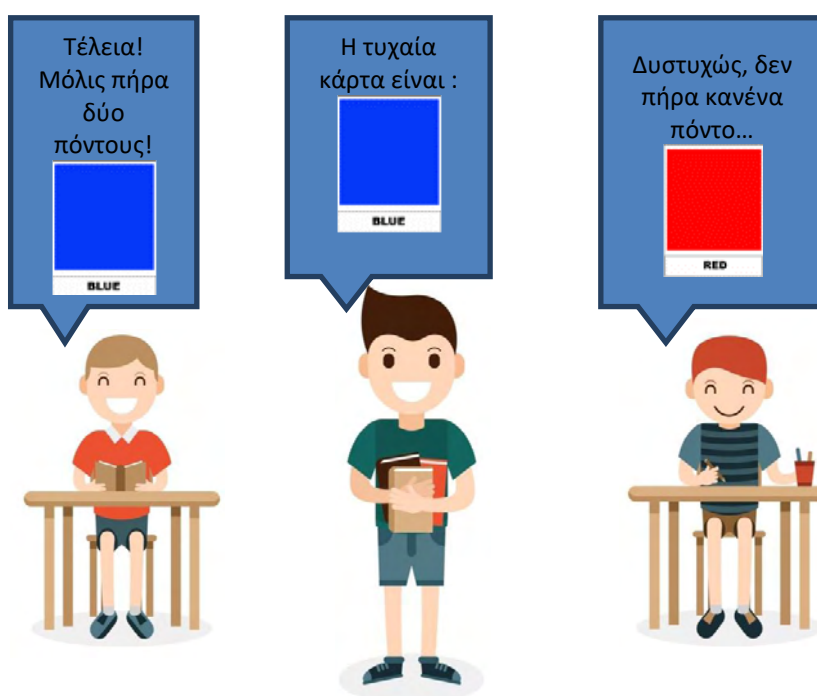
Το παιχνίδι περιλαμβάνει χρωματιστές κάρτες, κόκκινες και μπλε, και παίζεται με δύο βασικούς παίκτες και έναν βοηθητικό παίκτη. Οι χρωματιστές κάρτες τοποθετούνται στο τραπέζι χωρίς να φαίνεται ο χρωματισμός τους και τα βήματα του παιχνιδιού που ακολουθούν είναι τα παρακάτω:

1. Ο πρώτος παίκτης, έπειτα ο δεύτερος παίκτης και τέλος ο βοηθητικός παίκτης επιλέγουν με τη σειρά από μια τυχαία κάρτα και αποκαλύπτουν τον χρωματισμό τους όπως φαίνεται στην εικόνα 10.
2. Ο πρώτος και ο δεύτερος παίκτης συγκρίνουν το χρώμα της κάρτας τους με το χρώμα της κάρτας του βοηθητικού παίκτη και σε περίπτωση που κάποιος παίκτης έχει το ίδιο χρώμα παίρνει δύο πόντους διαφορετικά δεν παίρνει κανέναν πόντο όπως φαίνεται στην εικόνα 11.

¹⁷ Οδηγίες του παιχνιδιού βρίσκονται στο παράρτημα Α.



Εικόνα 10. Παιχνίδι χρωματιστών καρτών – επιλογές ¹⁸



Εικόνα 11. Παιχνίδι χρωματιστών καρτών – αποτέλεσμα ¹⁹

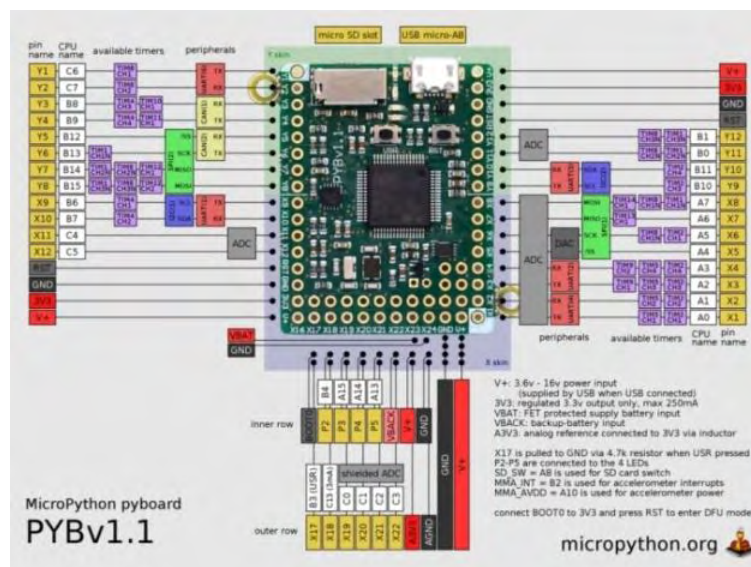
Μετά την περιγραφή του συγκεκριμένου παιχνιδιού ο εκπαιδευτικός προτείνεται να προβληματίσει τους εκπαιδευόμενους για την εξέλιξη του παιχνιδιού, για παράδειγμα ότι θα

¹⁸ <https://www.freepik.com/>

¹⁹ <https://www.freepik.com/>

μπορούσε να αυξηθεί ο αριθμός των χρωμάτων των καρτών ή ο αριθμός των παικτών ή ο αριθμός των γύρων, να μεταβάλλεται ο αριθμός των πόντων ανάλογα με τον χρωματισμό των καρτών ή να παίρνει ο κάθε παίκτης περισσότερες κάρτες κτλ. με σκοπό να τους διεγείρει το ενδιαφέρον.

Βήμα 2: παρουσιάζει βασικές έννοιες και πολύ συνοπτικά τα βασικά εξαρτήματα και τη συνδεσμολογία της πλακέτας, όπως φαίνεται στην εικόνα 12, ώστε να εισαχθούν οι εκπαιδευόμενοι στον κόσμο των κυκλωμάτων και του προγραμματισμού τους και να αντιληφθούν την εκτέλεση του ίδιου παιχνιδιού με τη χρήση της πλακέτας.



Εικόνα 12. Αναλυτική περιγραφή της πλακέτας Pyboard²⁰

Περιγραφή παιχνιδιού²¹

Το παιχνίδι περιλαμβάνει 6 χρωματιστά led λαμπάκια, 3 κόκκινα και 3 μπλε, 3 push buttons τα οποία τοποθετούνται σε μια πλακέτα δοκιμών και παίζεται με δύο βασικούς παίκτες και έναν βοηθητικό παίκτη. Ο κάθε παίκτης έχει ένα κόκκινο και ένα μπλε led λαμπάκι και ένα push button. Τα βήματα του παιχνιδιού που ακολουθούν είναι τα παρακάτω:

3. Ο πρώτος παίκτης, έπειτα ο δεύτερος παίκτης και τέλος ο βοηθητικός παίκτης πιέζουν με τη σειρά το δικό τους push button το οποίο παράγει από μια τυχαία τιμή 0 ή 1 και ανάλογα με την τιμή ανάβει το αντίστοιχο led λαμπάκι. Η τιμή 0 ανήκει στο κόκκινο λαμπάκι και η τιμή 1 στο μπλε λαμπάκι.

²⁰ <https://micropython.org/>

²¹ Το φωτογραφικό υλικό του παιχνιδιού βρίσκεται στο παράρτημα Α.

4. Ο πρώτος και ο δεύτερος παίκτης συγκρίνουν το χρώμα από το δικό τους led λαμπάκι με το χρώμα του βοηθητικού παίκτη και σε περίπτωση που κάποιος έχει αναμμένο το ίδιο χρώμα παίρνει δύο πόντους και τα δύο led λαμπάκια του ανάβουν, διαφορετικά δεν παίρνει κανέναν πόντο και το led λαμπάκι του σβήνει. Σε περίπτωση ισοβαθμίας τα led λαμπάκια σβήνουν.

Μετά την περιγραφή του συγκεκριμένου παιχνιδιού ο εκπαιδευτικός προτείνεται να προβληματίσει πάλι τους εκπαιδευόμενους για την εξέλιξη του παιχνιδιού, για παράδειγμα για το τι αλλαγές θα έπρεπε να γίνουν στην πλακέτα δοκιμών έτσι ώστε να μπορέσουν να υλοποιηθούν οι αλλαγές που περιγράφηκαν παραπάνω.

Τα βήματα που προτείνεται να ακολουθήσει ο εκπαιδευτικός κατά τη διάρκεια των διδακτικών σεναρίων περιγράφονται παρακάτω:

Βήμα 1: εισάγει την κάθε αλγοριθμική δομή ή δομή δεδομένων εξηγώντας την αναγκαιότητα για τον συγκεκριμένο τύπο δομής με ένα παράδειγμα. Η βιωματική μάθηση μπορεί να βοηθήσει τους εκπαιδευόμενους να δομήσουν έννοιες σχετικές με τον προγραμματισμό υπολογιστών. Καλό θα είναι, μέσα από πειραματισμούς, προσομοιώσεις και στοχευμένες ερωτήσεις, οι εκπαιδευόμενοι να μπορούν να οικοδομούν νέες γνώσεις πάνω στις υπάρχουσες και να αποσαφηνίζουν τυχόν μπερδεμένες έννοιες.

Βήμα 2: εξηγεί τη σύνταξη της δομής και αναλύει τη λειτουργία της. Συγκεκριμένα, προσομοιώνει τη λειτουργία ενός μικρού προγράμματος που γράφει ο ίδιος στον πίνακα και εξηγεί τις διαφορετικές τιμές των μεταβλητών και των εκφράσεων. Θέτει κάποιες ερωτήσεις στους εκπαιδευόμενους για να βεβαιωθεί ότι έχουν κατανοήσει τη σύνταξη της δομής και τη λειτουργία της.

Βήμα 3: χρησιμοποιεί φύλλα εργασιών τα οποία σύμφωνα με τον (Κόμης Β. , 2015) λαμβάνουν υπόψη α) την ψυχολογική και γνωστική προετοιμασία, β) τη διδασκαλία του γνωστικού αντικειμένου, γ) την εμπέδωση, δ) την αξιολόγηση και ε) την μεταγνωστική διαδικασία και περιλαμβάνουν δραστηριότητες και ασκήσεις εμπέδωσης από τους συγγραφείς (Καρκαμάνης, 2015), (Τσιωτάκης, 2008) και (Λαζαρίνης, 2015) καθώς και μια δραστηριότητα όπου ο εκπαιδευόμενος καλείται να αναπτύξει την ψηφιακή εκδοχή που προσομοιώνει τη λειτουργία του παιχνιδιού στη γλώσσα προγραμματισμού Python, την εκτελεί και παρατηρεί τις μεταβολές των τιμών των μεταβλητών και των εκφράσεων.

Επίσης, τα διδακτικά σενάρια περιλαμβάνουν παραρτήματα, τα οποία μπορούν να συμβουλευτούν ο εκπαιδευτικός και οι εκπαιδευόμενοι, με οδηγίες του παιχνιδιού και

αντίστοιχο φωτογραφικό υλικό καθώς διαθέσιμα είναι και ο αντίστοιχος κώδικας στην MicroPython και ενδεικτικές απαντήσεις των δραστηριοτήτων των φύλλων εργασίας σε Python. Η φιλοσοφία που ακολουθείται είναι να δοθεί περισσότερο βάρος στον προγραμματισμό και τον κώδικα που γράφουν οι εκπαιδευόμενοι στην Python και λιγότερο στον σχεδιασμό των κυκλωμάτων και τον προγραμματισμό στην MicroPython.

3.2.6 Διδακτικό συμβόλαιο – Διδακτικός θόρυβος

Κατά τη διάρκεια των σεναρίων εκτιμάται ότι δεν θα δημιουργηθούν σημαντικά προβλήματα σε σχέση με την ταχύτητα λειτουργίας του ηλεκτρονικού υπολογιστή και του λογισμικού τα οποία θα μπορούσαν να επηρεάσουν τη ροή του μαθήματος (διδακτικός θόρυβος). Επίσης, το διδακτικό συμβόλαιο δεν θα ανατραπεί, γιατί τα φύλλα εργασίας είναι απλά και οδηγούν τον εκπαιδευόμενο βήμα – βήμα στην ομαλή εξοικείωση του με το λογισμικό.

3.2.7 Χρήση εξωτερικών πηγών

Παρακάτω προτείνονται κάποιες εξωτερικές πηγές που θα μπορούσαν να χρησιμοποιηθούν συμπληρωματικά στα διδακτικά σεναρία.

- Σχολικό Βιβλίο Γ' ΕΠΑ.Λ - Προγραμματισμός Υπολογιστών.²²
- Υποστηρικτικό Υλικό - Εισαγωγή στον προγραμματισμό με την Python.²³
- Εκπαιδευτικό Υλικό - *A Byte Of Python* (el).²⁴
- Lutz M. (2013), *Learning Python (5th Edition)*, O'Reilly Media²⁵
- Tollervey N. (2018), *Programming with MicroPython*, Shroff / Oreilly Reprints.²⁶
- Bell C. (2017), *MicroPython for the Internet of Things: A Beginner's Guide to Programming with Python on Microcontrollers*, Apress.²⁷
- Damien G., Sokolovsky P. and contributors (2018), *MicroPython Documentation*.²⁸

²²http://www.iep.edu.gr/images/IEP/EPISTIMONIKI_YPIRESIA/Epist_Monades/B_Kyklos/Tee/2017/GEpal/G_Epal_Progr_Ypologiston_Simeioseis_Mathiti_2h_ekdosi_2017.pdf

²³http://aggelid.mysch.gr/pythonbook/INTRODUCTION_TO_COMPUTER_PROGRAMMING_WITH_PYTHON.pdf

²⁴https://ia801303.us.archive.org/32/items/AByteOfPythonEl/a_byte_of_python-el.pdf

²⁵<http://www.dsf.unica.it/~fiore/LearningPython.pdf>

²⁶https://books.google.gr/books/about/Programming_with_MicroPython.html?id=VCM3DwAAQBAJ&redir_esc=y

²⁷https://books.google.gr/books/about/MicroPython_for_the_Internet_of_Things.html?id=70NADwAAQBAJ&redir_esc=y

²⁸<https://media.readthedocs.org/pdf/micropython-pfalcon/stable/micropython-pfalcon.pdf>

3.3 Διδακτικό σενάριο 1: Εισαγωγικές έννοιες του προγραμματισμού και δομή ακολουθίας

3.3.1 Τίτλος διδακτικού σεναρίου

Διδακτικό σενάριο στις εισαγωγικές έννοιες του προγραμματισμού και στη διδασκαλία της δομής ακολουθίας στη γλώσσα προγραμματισμού Python.

3.3.2 Εκτιμώμενη διάρκεια διδακτικού σεναρίου

Το διδακτικό σενάριο προβλέπεται να διαρκέσει συνολικά 3 διδακτικές ώρες.

3.3.3 Προαπαιτούμενες γνώσεις διδακτικού σεναρίου

Το διδακτικό σενάριο προορίζεται για εκπαιδευόμενους χωρίς κάποιες στοιχειώδεις γνώσεις προγραμματισμού.

3.3.4 Σκοποί και στόχοι του διδακτικού σεναρίου

Ο βασικός σκοπός του διδακτικού σεναρίου είναι η εισαγωγή των εκπαιδευομένων στη δομή ακολουθίας η οποία περιγράφεται σε τρεις ενότητες ως εξής:

1η Ενότητα: Βασικές έννοιες προγραμματισμού

2η Ενότητα: Δομή ακολουθίας

3η Ενότητα: Ενσωματωμένες Συναρτήσεις

Το διδακτικό σενάριο έχει ως στόχο μέσω πειραματισμού και προσομοιώσεων τη δημιουργία απλών προγραμμάτων. Πιο συγκεκριμένα, στόχος είναι οι εκπαιδευόμενοι να:

- κατανοήσουν την έννοια του αλγορίθμου
- διακρίνουν την αναγκαιότητα χρήσης της δομής ακολουθίας
- κατανοήσουν βασικές έννοιες του προγραμματισμού όπως μεταβλητή, σταθερά, τύποι δεδομένων, τελεστές, αριθμητική και λογική έκφραση, εντολή ανάθεσης τιμής, εντολές εισόδου και εξόδου και σχόλια
- συντάσσουν σωστά τις εντολές
- μπορούν να αντιληφθούν συντακτικά και λογικά λάθη που οδηγούν στην αδυναμία εκτέλεσης των προγραμμάτων
- είναι ικανοί να προσδιορίζουν την εντολή που πρέπει να χρησιμοποιήσουν ανάλογα με την ενέργεια που θέλουν να εκτελέσουν
- μπορούν να αναπαριστούν ένα πρόβλημα που εμπεριέχει δομή ακολουθίας κωδικοποιημένο σε Python βλέποντας τα αποτελέσματα άμεσα στον υπολογιστή
- εξοικειωθούν με τη χρήση ΤΠΕ

- μπορούν να δουλεύουν ομαδοσυνεργατικά.

3.3.5 Περιγραφή του διδακτικού σεναρίου

1η Ενότητα – Βασικές έννοιες προγραμματισμού

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να εξηγήσει τον ορισμό του αλγορίθμου, τα κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος και τους τρόπους αναπαράστασης ενός αλγορίθμου χρησιμοποιώντας ένα παράδειγμα, όπως τη διαδικασία παρασκευής ενός ελληνικού καφέ.

▪ Ορισμός αλγορίθμου

Αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος (Βακάλη, και συν., 2017).

Για παράδειγμα, οι ενέργειες της παρασκευής του ελληνικού καφέ είναι (Ντζιός & Κοψίνης, 2008):

- ✓ Δώσε την ποσότητα του καφέ σε κουταλάκια του γλυκού.
- ✓ Δώσε την ποσότητα της ζάχαρης σε κουταλάκια του γλυκού.
- ✓ Βάλε τη ζάχαρη στο μπρίκι.
- ✓ Βάλε τον καφέ στο μπρίκι.
- ✓ Βάλε ένα φλιτζάνι νερό στο μπρίκι.
- ✓ Άναψε το μάτι της κουζίνας.
- ✓ Βάλε το μπρίκι στο μάτι της κουζίνας.
- ✓ Ανακάτεψε μέχρι να φουσκώσει.
- ✓ Κλείσε το μάτι της κουζίνας.
- ✓ Βάλε τον καφέ στο φλιτζάνι.

▪ Κριτήρια αλγορίθμου

Τα κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος είναι: η είσοδος, η έξοδος, η καθοριστικότητα, η περατότητα και η αποτελεσματικότητα (Βακάλη, και συν., 2017).

Για παράδειγμα, στην παρασκευή του ελληνικού καφέ πρέπει να ικανοποιείται (Ντζιός & Κοψίνης, 2008):

- ✓ το κριτήριο της εισόδου, δηλαδή να είναι διαθέσιμα τα υλικά όπως ο καφές, η ζάχαρη, το νερό, το κουτάλι, το μπρίκι και το μάτι της κουζίνας,
- ✓ το κριτήριο της εξόδου, δηλαδή να παρασκευαστεί ο ελληνικός καφές που είναι έτοιμος να τον πει κάποιος,
- ✓ το κριτήριο της καθοριστικότητας, δηλαδή να είναι καθορισμένη η δοσολογία των υλικών,
- ✓ το κριτήριο της περατότητας, δηλαδή να έχουμε ανάψει το μάτι της κουζίνας έτσι ώστε να μην περιμένουμε χωρίς λόγο και
- ✓ το κριτήριο της αποτελεσματικότητας, δηλαδή να μπορούμε να εκτελέσουμε όλες τις απαραίτητες ενέργειες.

▪ Τρόποι αναπαράστασης αλγορίθμου

Οι τρόποι αναπαράστασης αλγορίθμου είναι: με ελεύθερο κείμενο, με διαγραμματικές τεχνικές, με φυσική γλώσσα κατά βήματα και με κωδικοποίηση (Βακάλη, και συν., 2017). Στο παρόν σενάριο θα χρησιμοποιηθεί η κωδικοποίηση, δηλαδή σύνταξη προγραμμάτων σε γλώσσα προγραμματισμού Python.

Βήμα 2:

Έπειτα, ο εκπαιδευτικός προτείνεται να εξηγήσει βασικές έννοιες του προγραμματισμού όπως περιγράφονται στον πίνακα 1:

Πίνακας 1: Βασικές έννοιες προγραμματισμού

Μεταβλητή	Για να χρησιμοποιήσουμε μια μεταβλητή απαιτείται να της δώσουμε ένα όνομα και στη συνέχεια να της εκχωρήσουμε κάποια τιμή. Όλα τα δεδομένα σε ένα πρόγραμμα Python αναπαρίστανται με αντικείμενα ή με σχέσεις μεταξύ των αντικειμένων, με κάθε αντικείμενο να έχει μια ταυτότητα (identity), έναν τύπο και μία τιμή που καταλαμβάνει δική της θέση στη μνήμη. Για παράδειγμα, το 12 είναι ένα αντικείμενο με τιμή 12, τύπου int (ακέραιος) που όταν μια μεταβλητή πάρει την τιμή 12 ουσιαστικά θα δείχνει στη θέση μνήμης που βρίσκεται το 12. Αν στη συνέχεια στην ίδια μεταβλητή εκχωρήσουμε μια άλλη τιμή τότε θα δείξει σε άλλη θέση στη μνήμη. Οι μεταβλητές στην Python λειτουργούν σαν ετικέτες με κάποιο όνομα όπως απεικονίζεται στην εικόνα 13, που χρησιμεύουν για να αναφερόμαστε σε κάποια αντικείμενα. Ο τύπος της μεταβλητής είναι ο εκάστοτε τύπος του
------------------	--

	<p>αντικειμένου στην οποία αναφέρεται. Για τη χρησιμοποίηση μιας μεταβλητής δεν απαιτείται η δήλωσή της, ενώ μπορεί να εκχωρήσουμε διαφορετικούς τύπους τιμών σε μια μεταβλητή κατά τη διάρκεια ενός προγράμματος, όπως ακέραιες τιμές, κινητής υποδιαστολής και συμβολοσειρές. Η Python παρακολουθεί όλες τις τιμές και τις διαγράφει όταν πάνσουν να υπάρχουν μεταβλητές που να αναφέρονται σε αυτές. Η διαδικασία αυτή ονομάζεται συλλογή σκουπιδιών (garbage collection) (Αράπογλου Α. , και συν., 2016).</p>
Σταθερά	<p>Προκαθορισμένη τιμή που παραμένει αμετάβλητη σε όλη τη διάρκεια της εκτέλεσης ενός αλγορίθμου (Βακάλη, και συν., 2017).</p>
Κανόνες ονοματολογίας μεταβλητών	<p>Δεν επιτρέπεται να ξεκινά το όνομα μιας μεταβλητής με αριθμό και δεν πρέπει να είναι όμοιο με κάποιο όνομα ενσωματωμένης συνάρτησης ή εντολής. Συνήθως είναι ένα όνομα σχετικό με το είδος της μεταβλητής με λατινικούς χαρακτήρες, που μπορεί να συνοδεύεται από κάποιον αριθμό και την κάτω_παύλα, όπως για παράδειγμα: <code>onoma1</code>, <code>onoma_2</code> (Αράπογλου Α. , και συν., 2016).</p>
Τύποι δεδομένων	<p>Οι τύποι δεδομένων προσδιορίζουν τον τρόπο αναπαράστασης των δεδομένων εσωτερικά στον υπολογιστή, καθώς και το είδος της επεξεργασίας τους από αυτόν. Στην Python είναι ο αριθμητικός, ο λογικός (boolean) και οι συμβολοσειρές ή αλφαριθμητικά (strings).</p> <p>Ο αριθμητικός τύπος είναι κυρίως τριών τύπων:</p> <ul style="list-style-type: none"> ▪ ακέραιοι (Integer) π.χ 19. ▪ αριθμοί κινητής υποδιαστολής (floating point) π.χ οι 256.14 και 28.2E-5, όπου το σύμβολο E δηλώνει δύναμη του 10. ▪ μιγαδικοί αριθμοί (complex numbers). <p>Ο λογικός τύπος (boolean) που δέχεται μόνο δύο τιμές, την τιμή True (Αληθής) και την τιμή False (Ψευδής).</p> <p>Οι συμβολοσειρές είναι μια ακολουθία από χαρακτήρες που μπορεί να αποτελείται από περισσότερες από μία λέξεις και μπορούμε να ορίσουμε μια συμβολοσειρά με μονά εισαγωγικά ή με διπλά π.χ 'Καλημέρα' (Αράπογλου Α. , και συν., 2016).</p>
Τελεστές	<p>Σύμβολα που χρησιμοποιούνται στις διάφορες πράξεις. Διακρίνονται σε:</p>

Αριθμητικοί: ******(δύναμη), *****(πολλαπλασιασμός), **/** (διαίρεση), **%**
(υπόλοιπο ακεραίας διαίρεσης), **//** (πηλίκο ακεραίας διαίρεσης), **+**
(πρόσθεση), **-** (αφαίρεση)

Σχεσιακοί: **>** (μεγαλύτερο), **<** (μικρότερο), **!=** (διάφορο), **==** (ίσο), **<=**
(μικρότερο ή ίσο), **>=** (μεγαλύτερο ή ίσο)

Λογικοί: **not** (OXI-άρνηση), **and** (KAI-σύζευξη), **or** (H-διάζευξη)
(Swaroop, 2008)

Ιεραρχία των τελεστών

Η ιεραρχία των τελεστών, από τη μεγαλύτερη προς τη μικρότερη, είναι η παρακάτω:

***, /, %, //**

+, -

>, <, !=, ==, <=, >=

not (OXI-άρνηση),

and (KAI-σύζευξη),

or (H-διάζευξη)

Παρατήρηση: Αν θέλουμε να αλλάξουμε την ιεραρχία των πράξεων, μπορούμε να χρησιμοποιήσουμε παρενθέσεις (Swaroop, 2008).

Εκφράσεις

Προτάσεις που δημιουργούνται από τελεστές, σταθερές και μεταβλητές. Υπάρχουν αριθμητικές, απλές λογικές και σύνθετες λογικές εκφράσεις (Βακάλη, et al., 2017).

Εντολή ανάθεσης τιμής

Στο αριστερό μέρος δίνουμε το όνομα της μεταβλητής, στη συνέχεια χρησιμοποιούμε τον τελεστή εκχώρησης “=” και στο δεξιό μέρος βάζουμε την τιμή, μια άλλη μεταβλητή ή μια έκφραση που έχει ως αποτέλεσμα μια τιμή (Αράπογλου Α. , et al., 2016).

Εντολή εισόδου

input(): χρησιμοποιείται για την εισαγωγή τιμής σε μια μεταβλητή από το πληκτρολόγιο. Το πρόγραμμα αναμένει από το χρήστη να πληκτρολογήσει μια τιμή. Η τιμή που εισάγεται αποδίδεται σε μια μεταβλητή.

Σύνταξη:

Όνομα μεταβλητής= **input**(‘μήνυμα’), π.χ **age=input**(‘Give your age’)

(Αράπογλου Α. , et al., 2016)

Εντολή εξόδου

print: χρησιμοποιείται για την εμφάνιση τιμών στην οθόνη του υπολογιστή.

Σύνταξη

print(όνομα_μεταβλητής), π.χ. print(name)

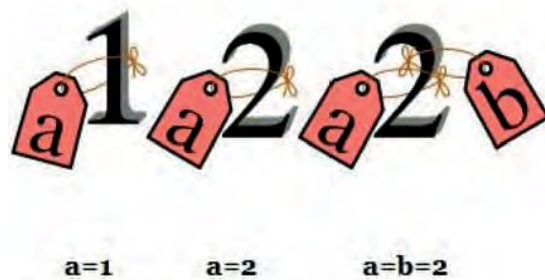
print(αριθμός), π.χ. print(1045.34)

print('συμβολοσειρά'), π.χ. print('θαλασσινός αέρας')

print(όνομα_μεταβλητής (τελεστής) αριθμός) (Αράπογλου Α. , et al., 2016)

Εισαγωγή σχολίων

Εισάγονται θέτοντας μπροστά από αυτά το σύμβολο # (Αράπογλου Α. , et al., 2016).



Εικόνα 13. Έννοια της μεταβλητής²⁹

Πιο συγκεκριμένα προτείνεται η εξοικείωση των παραπάνω εννοιών μέσα από δραστηριότητες με τη χρήση του διερμηνευτή. Οι δραστηριότητες μπορεί να έχουν διερευνητικό χαρακτήρα καλώντας τους εκπαιδευόμενους αρχικά να απαντήσουν τι αναμένουν να εμφανιστεί στην οθόνη μετά την εκτέλεση των προτεινόμενων εντολών και να επαληθεύσουν την υπόθεση τους πειραματιζόμενοι με το προγραμματιστικό περιβάλλον. Για τη διερεύνηση του εκάστοτε τύπου της μεταβλητής προτείνεται να χρησιμοποιηθεί η μέθοδος type() και της διεύθυνσης της μεταβλητής στη μνήμη προτείνεται να χρησιμοποιηθεί η εντολή id(x) σε απλές δραστηριότητες.

Επιπρόσθετα, σύμφωνα με τον (Guo, 2013) η αξιοποίηση τεχνικών και εργαλείων για την οπτικοποίηση των διαδικασιών (οπτικοποιημένη βήμα προς βήμα εκτέλεση εντολών με το Python Tutor OnLine³⁰ μπορεί συμβάλει στην κατανόηση της λειτουργίας της μεταβλητής.

²⁹ <http://www.etpe.gr/custom/pdf/etpe2389.pdf>

³⁰ <http://www.Pythontutor.com/>

Το προγραμματιστικό περιβάλλον δίνει τη δυνατότητα σχηματικής απεικόνισης των ονομάτων των μεταβλητών σε σχέση με τις τρέχουσες τιμές τους κατά τη στιγμή εκτέλεσης του προγράμματος (Αράπογλου Α. , 2016).

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 1.

2η Ενότητα – Δομή ακολουθίας

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή ακολουθίας με ένα παράδειγμα της καθημερινότητας (Πίνακας 2) το οποίο είναι:

Πίνακας 2: Παράδειγμα καθημερινότητας για τη δομή ακολουθίας

Αλγόριθμος προετοιμασίας γεύματος (Βακάλη, et al., 2017)

- συγκέντρωση των υλικών,
- προετοιμασία των μαγειρικών σκευών,
- παρασκευή του φαγητού,
- ετοιμασία της σαλάτας,
- στρώσιμο του τραπεζιού,
- γεύμα,
- καθαριότητα του τραπεζιού και
- πλύσιμο των πιάτων και των μαγειρικών σκευών.

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή ακολουθίας με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 3) το οποίο είναι³¹:

Πίνακας 3: Παράδειγμα με τη χρήση της πλακέτας για τη δομή ακολουθίας

Ερμηνεία των εντολών	MicroPython
Εισάγεται η βιβλιοθήκη pyb	<code>import pyb</code>
Ανάβει το κόκκινο led	<code>pyb.LED(1).on()</code>
Ανάβει το πράσινο led	<code>pyb.LED(2).on()</code>
Ανάβει το κίτρινο led	<code>pyb.LED(3).on()</code>
Ανάβει το μπλε led	<code>pyb.LED(4).on()</code>
Σβήνει το κόκκινο led	<code>pyb.LED(1).off()</code>

³¹ Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνα 23).

Σβήνει το πράσινο led	<code>pyb.LED(2).off()</code>
Σβήνει το κίτρινο led	<code>pyb.LED(3).off()</code>
Σβήνει το μπλε led	<code>pyb.LED(4).off()</code>

Βήμα 2:

Επίσης, ο εκπαιδευτικός προτείνεται να αναφέρει ότι οι τρεις αλγοριθμικές δομές που θα χρησιμοποιηθούν είναι: η Δομή ακολουθίας, η Δομή επιλογής και η Δομή επανάληψης (Βακάλη, et al., 2017) και να εξηγήσει ότι η δομή ακολουθίας εφαρμόζεται στις περιπτώσεις όπου μια σειρά από εντολές εκτελούνται η μία μετά την άλλη με τη σειρά. Η δομή ακολουθίας χρησιμοποιείται πρακτικά για την επίλυση απλών προβλημάτων, όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών (Αράπογλου Α. , et al., 2016).

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 4) το οποίο είναι:

Πίνακας 4:Πρόγραμμα σε Python για τη δομή ακολουθίας

Ερμηνεία των εντολών	Python
Εμφανίζεται το μήνυμα «Give a number» στην οθόνη, εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής του στη μεταβλητή num1	<code>num1=int(input('Give a number'))</code>
Εμφανίζεται το μήνυμα «Give a number» στην οθόνη, εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής του στη μεταβλητή num2	<code>num2=int(input('Give a number'))</code>
Υπολογίζεται ο μέσος όρος των δύο παραπάνω αριθμών και γίνεται ανάθεση της τιμής του αποτελέσματος στην μεταβλητή average	<code>average=(num1+num2)/2</code>
Εμφανίζεται στην οθόνη το μήνυμα «The average is» και δίπλα η τιμή της μεταβλητής average	<code>print('The average is:', average)</code>

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 2.

3η Ενότητα – Ενσωματωμένες συναρτήσεις

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τις συναρτήσεις και τη βιβλιοθήκη με ένα παράδειγμα της καθημερινότητας (Πίνακας 5) το οποίο είναι:

Πίνακας 5: Παράδειγμα καθημερινότητας για τις συναρτήσεις

Εκτέλεση μιας **συνταγής** μέσα από ένα **βιβλίο συνταγών**

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τις συναρτήσεις και τη βιβλιοθήκη με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 6) το οποίο είναι³²:

Πίνακας 6: Παράδειγμα με τη χρήση της πλακέτας για τις συναρτήσεις

Ερμηνεία των εντολών	MicroPython
Εισάγεται η βιβλιοθήκη pyb	<code>import pyb</code>
Παράγεται μια τυχαία ακέραια τιμή από το 0 έως το 3 και γίνεται ανάθεσή της στην μεταβλητή x	<code>x=(pyb.rng()%4)</code> ³³
Ανάβει το led λαμπάκι με αριθμό μια τιμή από τις παραπάνω αυξημένη κατά ένα δηλαδή 1, 2, 3 ή 4	<code>pyb.LED(x+1).on()</code>
Περιμένει 500ms	<code>pyb.delay(500)</code>
Σβήνει το παραπάνω led	<code>pyb.LED(x+1).off()</code>

Βήμα 2:

Επίσης, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι η Python περιέχει πλούσιες βιβλιοθήκες (μονάδες λογισμικού) από έτοιμο κώδικα προγραμματισμού, αλλά και άλλες επιπρόσθετες εξωτερικές βιβλιοθήκες (αρθρώματα ή modules) με έτοιμο κώδικα που μπορούν οι εκπαιδευόμενοι να τις χρησιμοποιούν στα προγράμματά τους ακολουθώντας ένα σύνολο απλών κανόνων και εντολών ή να δημιουργήσουν δικές τους ώστε να τις χρησιμοποιούν κατάλληλα σε διαφορετικά προγράμματά τους, αποφεύγοντας να γράφουν, κάθε φορά, κώδικα από την αρχή.

Οι ενσωματωμένες συναρτήσεις για τη μετατροπή τιμών δεδομένων από έναν τύπο σε έναν άλλο είναι οι: `float()` μετατρέπει ακέραιους και συμβολοσειρές σε δεκαδικούς αριθμούς, `int()` δέχεται οποιαδήποτε αριθμητική τιμή και τη μετατρέπει σε ακέραιο κόβοντας τα δεκαδικά ψηφία, αν υπάρχουν, `str()` δέχεται οποιαδήποτε τιμή και την μετατρέπει σε συμβολοσειρά, `abs()` επιστρέφει την απόλυτη τιμή ενός αριθμού, `pow(a,b)` επιστρέφει τη δύναμη του α υψωμένη στο β, `divmod(x,y)` επιστρέφει το ακέραιο ηλίκο και το ακέραιο υπόλοιπο της

³² Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνα 24).

³³ Με τη χρήση της συνάρτησης `rng()` παράγεται μια τυχαία τιμή 30 bits (Damien & Sokolovsky, 2018)

διαίρεσης x/y . Οι εξωτερικές βιβλιοθήκες για να χρησιμοποιηθούν θα πρέπει να εισαχθούν στο πρόγραμμά με την εντολή `import`. Για παράδειγμα, για τη χρήση μαθηματικών συναρτήσεων σε ένα πρόγραμμα, θα πρέπει μία από τις αρχικές εντολές του προγράμματός να είναι: `import math` και για την πρόσβαση σε μια από τις συναρτήσεις, θα πρέπει να δηλώσουμε το όνομα της μονάδας και το όνομα της συνάρτησης, χωρισμένα με μια τελεία, μορφή που ονομάζεται συμβολισμός με τελεία (dot notation) (Αράπογλου Α. , et al., 2016).

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 7) το οποίο είναι:

Πίνακας 7: Πρόγραμμα σε Python για τις συναρτήσεις

Ερμηνεία των εντολών	Python
Εισάγεται η βιβλιοθήκη <code>math</code>	<code>import math</code>
Εμφανίζεται το μήνυμα «Give a number» στην οθόνη και εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής του στη μεταβλητή <code>x</code>	<code>x=int(input('Give a number'))</code>
Υπολογίζεται η τετραγωνική ρίζα της μεταβλητής <code>x</code> και γίνεται ανάθεση της τιμής του αποτελέσματος στη μεταβλητή <code>riza</code>	<code>riza = math.sqrt(x)</code>
Εμφανίζεται στην οθόνη η τιμή της μεταβλητής <code>riza</code>	<code>print (riza)</code>

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 3.

3.3.6 Αναπαραστάσεις εκπαιδευομένων / Πρόβλεψη δυσκολιών στο διδακτικό σενάριο

Τα περισσότερα λάθη των εκπαιδευομένων οφείλονται σε λανθασμένες αντιλήψεις που έχουν για την έννοια της μεταβλητής (Pane & Myers, 1996). Η ομάδα της (Samurcay, 1989) διεξήγαγε μια εμπειρική μελέτη από μια ομάδα σπουδαστών που φοιτούσαν στο 5ο έτος της Β/θμιας Εκπ/σης στη Γαλλία και είχαν διδαχθεί 15 ώρες προγραμματισμό σε Pascal στο Laboratoire de Psychologie du Travail EPHE/CNRS στο Παρίσι με σκοπό να διαπιστωθούν οι ικανότητες διαχείρισης της έννοιας της μεταβλητής. Η ομάδα της (Samurcay, 1989) διακρίνει τα παρακάτω:

- Ο τελεστής απόδοσης τιμής έχει τέσσερις τύπους χρήσης όταν χρησιμοποιείται με μεταβλητές: ανάθεση μιας σταθερής τιμής π.χ. `lesson = 'Πληροφορική'`, ανάθεση μιας

τιμής ως αποτέλεσμα υπολογισμών π.χ. $b=x * 4$, αντιγραφή π.χ. $l=m$ και συσσώρευση π.χ. $sum = sum + number$. Ο εκπαιδευόμενος μπορεί να χρησιμοποιήσει τις γνώσεις του από τα μαθηματικά που σχετίζονται με την έννοια της μεταβλητής και της ισότητας εκτός από την περίπτωση της συσσώρευσης όπου η μεταβλητή δεν είναι πλέον μια διεύθυνση με μια τιμή αλλά είναι και πρέπει να αντιμετωπίζεται ως μια συνάρτηση της εκτέλεσης του προγράμματος ή μια αλληλουχία τιμών.

- Υπάρχουν δύο είδη μεταβλητών βάσει της λειτουργικής τους σημασίας (functional meaning) στο μοντέλο του σχεδίου δράσης του εκπαιδευόμενου: εξωτερικές μεταβλητές, των οποίων οι τιμές ελέγχονται από τους χρήστες του προγράμματος και εσωτερικές μεταβλητές, των οποίων οι τιμές ελέγχονται από τον προγραμματιστή και είναι νοητικά πιο δύσκολος ο χειρισμός τους, γιατί απαιτεί τη συνεχή ανάπτυξη αναπαραστάσεων της τόσο ευμετάβλητης κατάστασης του υπολογιστικού συστήματος.

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

- Μεταβλητές με «πεπλεγμένο» υπολογισμό τιμών. Όταν οι μεταβολές μιας μεταβλητής εξαρτώνται από τις μεταβολές μια άλλης μεταβλητής.
- Δυσκολία κατανόησης του γεγονότος ότι οι πράξεις επιτρέπονται μεταξύ μεταβλητών του ίδιου ή συγγενούς τύπου δεδομένων και του γεγονότος ότι οι επιτρεπτές είναι διαφορετικές ανάλογα με την κατηγορία μεταβλητών.

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

- Αρκετοί εκπαιδευόμενοι δυσκολεύονται να κατανοήσουν πως γίνεται η απόδοση τιμής σε μια μεταβλητή με μια εντολή `Input()`. Συγκεκριμένα, πιστεύουν ότι με μια εντολή `Input()` μπορούν να εισαχθούν περισσότερες από μία τιμές σε μια μεταβλητή, ότι μια εντολή `Input()` στην οποία το όνομα της μεταβλητής έχει κάποια σημασία στη φυσική γλώσσα, η σημασία του ονόματος επηρεάζει την επιλογή της τιμής της και ότι η σειρά δήλωσης των μεταβλητών είναι αυτή που καθορίζει τη σειρά εισαγωγής τιμών.
- Συγχέονται δύο μεταβλητές. Για παράδειγμα στην αλληλουχία
 $P = \text{int}(\text{input}())$
 $Q = Q + 1$ η δεύτερη εντολή εκτελείται σαν να ήταν $Q=P + 1$
- Αντιστροφή της εντολής απόδοσης τιμής. Η εντολή $A = B$ θεωρείται ότι ανταλλάσσει τις τιμές των δύο μεταβλητών.

- Χρησιμοποιείται πάντα η αρχική τιμή της μεταβλητής ακόμη και αν αυτή έχει μεταβληθεί.
- Οι τιμές των μεταβλητών εμφανίζονται όταν η μεταβλητή συναντάται στο αριστερό μέρος μιας έκφρασης.
- Η τιμή της μεταβλητής που βρίσκεται στο αριστερό μέρος εμφανίζεται όποτε αλλάζει η τιμή της (Ξυνόγαλος, 2005).

Συγκεκριμένα, η γλώσσα Python παρέχει μια ποικιλία διαφορετικών συμβολικών εκφράσεων για την ανάθεση τιμής σε μία μεταβλητή που μπορεί να αποτελέσει τελικά εμπόδιο στην κατανόηση της έννοιας. Για παράδειγμα, η γλώσσα Python παρέχει μεταξύ άλλων δυνατότητες όπως (Αράπογλου Α. , 2016):

- Ανάθεση τιμής σε περισσότερες από μία μεταβλητές: `metavliti1 = metavliti2 = metavliti3 = 15`.
- Ανάθεση τιμών με τη σειρά: `x, y=10, 34`. Στο παράδειγμα το `x` παίρνει τη τιμή 10 ενώ το `y` την τιμή 34. Η δήλωση `x, y, z = 10, 27, "Ελευθερία"` εκχωρεί στο `x` το 10 στο `y` το 27 και στο `z` τη συμβολοσειρά Ελευθερία.
- Αντιμετάθεση τιμών: Χαρακτηριστικό παράδειγμα αποτελεί η δήλωση για την αντιμετάθεση τιμών μεταξύ δύο μεταβλητών: `x, y = y, x`.

Επιπρόσθετα, η (Mészárosová, 2015) υποστηρίζει ότι οι εκπαιδευτικοί του υπό εξέταση δείγματος της έρευνας της, θεωρούν ως πιθανό διδακτικό εμπόδιο για την εμβάθυνση στην έννοια της μεταβλητής:

- την έλλειψη τμήματος δηλώσεων των τύπων των μεταβλητών σε συνδυασμό με τη δυνατότητα αλλαγής του τύπου των τιμών μιας μεταβλητής κατά τη διάρκεια εκτέλεσης ενός προγράμματος
- η εμφάνιση της τιμής 0 μετά τον υπολογισμό της διαίρεσης $x = 1/2$ σε σχέση με την εμφάνιση της τιμής 0.5 της δήλωσης `x = 1/2.0` στην έκδοση 2.7.x της Python ή η χρήση ακεραίων ως λογικοί σε λογικές εκφράσεις (`x=1 if x: print "Ok"`, ανάλογα και με την έκδοση της Python που χρησιμοποιείται).

3.3.7 Επεκτάσεις / Διασυνδέσεις των εννοιών ή των δραστηριοτήτων

Μετά την ολοκλήρωση του παρόντος σεναρίου οι εκπαιδευόμενοι θα είναι προετοιμασμένοι για να εργαστούν με επόμενα σενάρια στα οποία περιλαμβάνεται η δομή επιλογής.

3.3.8 Φύλλα εργασίας 1, 2, 3 και 4.

Απάντηση:.....

c. >>> x=20

>>> id(x)

Απάντηση:.....

Τι παρατηρείς; Η μεταβλητή x δείχνει στη θέση μνήμης που βρίσκεται το 'book' ή το 20;

Απάντηση:.....

d. >>> y=x

>>> id(y)

Απάντηση:.....

Τι παρατηρείς; Η μεταβλητή x δείχνει στη θέση μνήμης που δείχνει και η μεταβλητή y;

Απάντηση:.....

e. >>> x=y=z=30

>>> id(x)

Απάντηση:.....

>>> id(y)

Απάντηση:.....

>>> id(z)

Απάντηση:.....

Τι παρατηρείς; Οι μεταβλητές x, y, z δείχνουν στην ίδια θέση μνήμης;

Απάντηση:.....

f. >>> x,y=50,100

Ποια τιμή έχει η μεταβλητή x και ποια τιμή έχει η μεταβλητή y;

Απάντηση:.....

g. >>> x,y=y,x

Ποια τιμή έχει η μεταβλητή x και ποια τιμή έχει η μεταβλητή y;

Απάντηση:.....

4. Να γράψεις ποια από τα παρακάτω ονόματα είναι αποδεκτά ως ονόματα μεταβλητών. Σε κάθε μία από αυτές ανέθεσε μία σταθερή τιμή π.χ. x=9, εκτέλεσε τις εντολές ανάθεσης στο Python Shell και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

a. True

c. first name

e. 123abc

b. age

d. day1

f. George

Απάντηση:.....

b. Το x να είναι πολ/σιο του y .

Απάντηση:.....

c. Το x να είναι περιττός αριθμός και μικρότερος του 100.

Απάντηση:.....

d. Το x να λαμβάνει όλες τις τιμές εκτός από το 1.

Απάντηση:.....

e. Το x να είναι μεγαλύτερο του 10 ή μικρότερο του y .

Απάντηση:.....

f. Το x να είναι τετραπήφιος και άρτιος.

Απάντηση:.....

Δραστηριότητα 3

Να γράψεις τις παρακάτω εντολές εκχώρησης. Εκτέλεσέ τις στο Python Shell και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου αν $a=10$ και $b=20$.

1. Ανέθεσε στο x τον αριθμό 10.

Απάντηση:.....

2. Αύξησε το x κατά 1.

Απάντηση:.....

3. Ανέθεσε στο x τη διαφορά του a από το b .

Απάντηση:.....

4. Ανέθεσε στο x το τετράγωνό του.

Απάντηση:.....

5. Ανέθεσε στο x τον μέσο όρο των a και b .

Απάντηση:.....

Δραστηριότητα 4

1. Να γράψεις τι παράγεται ως έξοδος από τις παρακάτω εντολές. Εκτέλεσέ τις στο Python Shell και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

a. `>>> x=5`
`>>> y=10`
`>>> print('x=',x,'y=',y)`

Απάντηση:.....

b. `>>> a=10`
`>>> b=20`
`>>> c=a+b`
`>>> print("The sum is", c)`

Απάντηση:.....

2. Να εκτελέσεις στο Python Shell την εντολή `name = input('Give your name:')` και πληκτρολόγησε το όνομά σου. Πληκτρολόγησε τη μεταβλητή `name` για να επιβεβαιώσεις ότι η μεταβλητή `name` δείχνει στο όνομά σου.

```
>>> name = input('Give your name:')
```

Απάντηση:.....

Φύλλο εργασίας 2

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
x = int(input('Give a number'))
```

```
y = x % 5
```

```
x = x // 5 - y ** 2
```

```
z = (x + y) / 2
```

```
y = z - y
```

```
print(x,y,z)
```

1. Να προσδιορίσεις ποια είναι η εντολή εισόδου και ποια η εντολή εξόδου.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παραπάνω πρόγραμμα αν δοθεί ως είσοδος η τιμή 84 και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής). Εκτέλεσε το πρόγραμμα στο αρχείο f2d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα μέσα από το οποίο ένας υπάλληλος μπορεί να μάθει τον μηνιαίο μισθό του. Ο μηνιαίος μισθός του προκύπτει από τον βασικό του μισθό που είναι 830€, ένα επίδομα 25€ που λαμβάνει για κάθε τρία χρόνια εργασίας και το ποσό των υπερωριών που λαμβάνει 8€ ανά ώρα. Επομένως, το πρόγραμμα θα εισάγει τα χρόνια εργασίας του και τις υπερωρίες που έχει εργαστεί και θα υπολογίζει και θα εμφανίζει τον τελικό του μισθό που προκύπτει από το άθροισμα των παραπάνω.
2. Να τροποποιήσεις το πρόγραμμά σου έτσι ώστε από τον τελικό του μισθό να αφαιρούνται οι κρατήσεις οι οποίες είναι το 10% του τελικού του μισθού.
3. Αποθήκευσε το πρόγραμμα με όνομα myf2d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f2d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε έναν σύνδεσμο φιλάθλων μιας ομάδας βόλεϊ όπου διεξήχθησαν εκλογές για την ανάδειξη του προέδρου της. Υπήρχαν τρεις υποψήφιοι όπου ο καθένας έλαβε έναν αριθμό ψήφων. Επομένως, το πρόγραμμα θα εισάγει το επίθετο, το όνομα και τον αριθμό ψήφων που έλαβε ο κάθε ένας υποψήφιος, θα υπολογίζει το ποσοστό % που έλαβε ο κάθε ένας υποψήφιος επί του συνόλου των ψήφων και θα εμφανίζει το επίθετο, το όνομα του κάθε υποψηφίου και το ποσοστό που έλαβε.
2. Αποθήκευσε το πρόγραμμα με όνομα myf2d3.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f2d3.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα χρησιμοποιώντας δομή ακολουθίας που να δέχεται ως είσοδο τα ονόματα των δύο παικτών και να εμφανίζει τα ονόματά τους.
2. Αποθήκευσε το πρόγραμμα με όνομα myf2d4.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f2d4.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 3

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
x = 3
y = pow(x,3) - 4
z = y // x
print(y,z,x)
x = (x + z) % y
y = (y + z) // x
z = x * y - z ** 2
print(y,z,x)
```

1. Να προσδιορίσεις τη συνάρτηση που έχει το παραπάνω πρόγραμμα .

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παραπάνω πρόγραμμα και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής). Εκτέλεσε το πρόγραμμα στο αρχείο f3d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που να εισάγει τις συντεταγμένες δύο σημείων ενός καρτεσιανού συστήματος και να υπολογίζει και να εμφανίζει την απόσταση μεταξύ των δύο σημείων από τον τύπο: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ με τη βοήθεια της βιβλιοθήκης math και της συνάρτησης sqrt(x).
2. Αποθήκευσε το πρόγραμμα με όνομα myf3d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Παρατήρηση: Οι συντεταγμένες προσδιορίζονται με αριθμούς που έχουν ένα δεκαδικό ψηφίο.

Το πρόγραμμα στο αρχείο f3d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που να εισάγει δύο ακέραιους αριθμούς και υπολογίζει και να εμφανίζει την τιμή της μαθηματικής παράστασης $f(x)$ που δίνεται παρακάτω με τη βοήθεια της βιβλιοθήκης `math` και των συναρτήσεων `cos(x)`, `sin(x)` και `sqrt(x)`.

$$a = 2x^2 + 3x + |x|$$

$$b = \eta\mu(y) + \sigma\upsilon\nu(y)$$

$$f(x) = \frac{\sqrt{a^2 - b^2}}{(a+b)^2} - ab$$

Παρατήρηση: Θεωρείστε ότι η τιμή που θα δοθεί στο y είναι σε μοίρες.

2. Αποθήκευσε το πρόγραμμα με όνομα `myf3d3.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f3d3.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Να τροποποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 2 έτσι ώστε να παράγει τυχαία μια τιμή (με τη χρήση της βιβλιοθήκης `random` και της συνάρτησης `randint`) 0 ή 1 για κόκκινη και μπλε κάρτα αντίστοιχα για τον πρώτο παίκτη, μια για τον δεύτερο παίκτη και μια για τον βοηθητικό παίκτη και να εμφανίζει τα ονόματα και τους αριθμούς των δύο παικτών καθώς και τον αριθμό του βοηθητικού παίκτη μετά την εμφάνιση σχετικού μηνύματος.
2. Αποθήκευσε το πρόγραμμα με όνομα `myf3d4.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f3d4.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 4 - Τεστ αυτοαξιολόγησης

Για κάθε μία από τις παρακάτω 10 προτάσεις να επιλέξεις λέξη Σωστό, αν είναι Σωστή, ή τη λέξη Λάθος, αν είναι λανθασμένη.

1. Μια σταθερά μπορεί να αλλάξει τιμή κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου.

Σωστό Λάθος

2. Κατά την εκτέλεση ενός προγράμματος μπορεί να αλλάξει η τιμή και ο τύπος μιας μεταβλητής.

Σωστό Λάθος

3. Η τιμή "False" αποτελεί αλφαριθμητική τιμή, ενώ η τιμή False αποτελεί λογική τιμή.

Σωστό Λάθος

4. Σε μια έκφραση εκτελούνται πρώτα οι σχεσιακοί τελεστές και στη συνέχεια οι αριθμητικοί.

Σωστό Λάθος

5. Ο αριθμητικός τελεστής // υπολογίζει το υπόλοιπο της διαίρεσης δυο ακεραίων αριθμών.

Σωστό Λάθος

6. Η λογική έκφραση $x \text{ or } \text{not}(x)$ είναι πάντα αληθής για κάθε τιμή της λογικής μεταβλητής x .

Σωστό Λάθος

7. Δεξιά και αριστερά μιας εντολής ανάθεσης μπορεί να βρίσκεται η ίδια μεταβλητή.

Σωστό Λάθος

8. Το $a+b$ δεν είναι δεκτό ως όνομα μιας μεταβλητής.

Σωστό Λάθος

9. Η ακολουθιακή δομή χρησιμοποιείται για την αντιμετώπιση απλών προβλημάτων που είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών.

Σωστό Λάθος

10. Η εντολή `print('x')` εμφανίζει το περιεχόμενο της μεταβλητής x .

Σωστό Λάθος

3.4 Διδακτικό σενάριο 2: Δομή επιλογής

3.4.1 Τίτλος διδακτικού σεναρίου

Διδακτικό σενάριο στη διδασκαλία των δομών επιλογής στη γλώσσα προγραμματισμού Python.

3.4.2 Εκτιμώμενη διάρκεια διδακτικού σεναρίου

Το διδακτικό σενάριο προβλέπεται να διαρκέσει συνολικά 3 διδακτικές ώρες.

3.4.3 Προαπαιτούμενες γνώσεις διδακτικού σεναρίου

Το διδακτικό σενάριο προορίζεται για εκπαιδευόμενους με κάποιες στοιχειώδεις γνώσεις προγραμματισμού. Συγκεκριμένα, θα πρέπει ήδη να γνωρίζουν το προγραμματιστικό περιβάλλον της συγκεκριμένης γλώσσας έχοντας ήδη πραγματοποιήσει σχετικές δραστηριότητες χρησιμοποιώντας τις ακολουθιακές δομές ώστε να επιλύουν προβλήματα που αντιμετωπίζουν καθημερινά. Πιο συγκεκριμένα οι έννοιες που πρέπει να έχουν ήδη διδαχθεί είναι: η εισαγωγή δεδομένων από το πληκτρολόγιο (`input()`), η εκτύπωση στην οθόνη (`print()`), η εντολή εκχώρησης (`=`), οι βασικοί τύποι δεδομένων και η εισαγωγή βιβλιοθήκης με την κλήση ενσωματωμένων συναρτήσεων (`import`).

3.4.4 Σκοποί και στόχοι του διδακτικού σεναρίου

Ο βασικός σκοπός του διδακτικού σεναρίου είναι η εισαγωγή των εκπαιδευόμενων στη δομή επιλογής και συγκεκριμένα στις τέσσερις μορφές της δομής επιλογής οι οποίες περιγράφονται σε τέσσερις ενότητες ως εξής:

1η Ενότητα : Απλή επιλογή

2η Ενότητα : Σύνθετη επιλογή

3η Ενότητα : Πολλαπλή επιλογή

4η Ενότητα : Εμφωλευμένη επιλογή

Το διδακτικό σενάριο έχει ως στόχο μέσω πειραματισμού και προσομοιώσεων την σύνδεση και κατανόηση βασικών προγραμματιστικών εννοιών και την διατύπωση υποθέσεων και πραγματοποίηση ελέγχων. Πιο συγκεκριμένα, στόχος είναι οι εκπαιδευόμενοι να:

- κατανοήσουν ότι ο κόσμος γύρω μας είναι γεμάτος επιλογές οι οποίες εξαρτώνται από διάφορες συνθήκες. Για παράδειγμα, «Αν είναι σαββατοκύριακο τότε ξεκουραζόμαστε αλλιώς πάμε σχολείο»

- κατανοήσουν ότι οι δομές επιλογής χρησιμοποιούνται όταν θέλουμε να εκτελέσουμε κάτι υπό συνθήκη
- διακρίνουν την αναγκαιότητα χρήσης της κάθε δομής επιλογής
- είναι ικανοί να προσδιορίζουν τη συνθήκη η οποία πρέπει να ισχύει για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν υπό συνθήκη
- περιγράφουν την λειτουργία της κάθε δομής επιλογής
- συντάσσουν σωστά την κάθε δομή επιλογής
- μπορούν να καταλάβουν πότε μία συνθήκη είναι αληθής και πότε είναι ψευδής
- μπορούν να αντιληφθούν συντακτικά και λογικά λάθη που οδηγούν στην αδυναμία εκτέλεσης των προγραμμάτων
- μπορούν να αναπαριστούν ένα πρόβλημα που εμπεριέχει δομές επιλογής κωδικοποιημένο σε Python βλέποντας τα αποτελέσματα άμεσα στον υπολογιστή
- εξοικειωθούν με τη χρήση ΤΠΕ
- μπορούν να δουλεύουν ομαδοσυνεργατικά.

3.4.5 Περιγραφή του διδακτικού σεναρίου

1η Ενότητα - Απλή δομή επιλογής

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή απλής επιλογής με ένα παράδειγμα της καθημερινότητας (Πίνακας 8) το οποίο είναι:

Πίνακας 8: Παράδειγμα καθημερινότητας για την απλή επιλογή

Αν βρέχει τότε παίρνουμε ομπρέλα

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή απλής επιλογής με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 9) το οποίο είναι ³⁴:

Πίνακας 9: Παράδειγμα με τη χρήση της πλακέτας για την απλή επιλογή

Ερμηνεία των εντολών	MicroPython
Εισάγεται η βιβλιοθήκη pyb	import pyb
Αν πατηθεί το κουμπί USR τότε	sw = pyb.Switch()
Θα παραχθεί μια τυχαία τιμή 0 ή 1	while True: if sw.value():

³⁴ Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνα 25).

	X=(pyb.rng()%2)
	break
Αν η τυχαία τιμή είναι ίση με 0 τότε	if X==0:
Θα ανάψει το κόκκινο led	pyb.LED(1).on()
Ανάβει το πράσινο led	pyb.LED(2).on()
Ανάβει το κίτρινο led	pyb.LED(3).on()
Ανάβει το μπλε led	pyb.LED(4).on()
Περιμένει 1 sec	pyb.delay(1000)
Σβήνει το κόκκινο led	pyb.LED(1).off()
Σβήνει το πράσινο led	pyb.LED(2).off()
Σβήνει το κίτρινο led	pyb.LED(3).off()
Σβήνει το μπλε led	pyb.LED(4).off()

Βήμα 2:

Επίσης, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι η δομή επιλογής *if* χρησιμοποιείται, όταν θέλουμε να εκτελεστεί μια ακολουθία εντολών, μόνον, εφόσον πληρείται μία συγκεκριμένη συνθήκη. Για να επιλύσουμε πιο σύνθετα προβλήματα, πρέπει να είμαστε σε θέση να δημιουργούμε στον αλγόριθμο λογικά μονοπάτια, εξετάζοντας απλά λογικά ερωτήματα για την εκτέλεση ή όχι μιας ομάδας εντολών (Αράπογλου Α. , et al., 2016).

Ο κόσμος γύρω μας είναι γεμάτος επιλογές οι οποίες εξαρτώνται από διάφορες συνθήκες. Η συνθήκη μπορεί να αποτιμηθεί είτε σε *True* (δηλαδή να ισχύει) είτε σε *False* (δηλαδή να μην ισχύει).

Η εντολή ελέγχου **if συντάσσεται** ως εξής (Αράπογλου Α. , et al., 2016):

if συνθήκη ελέγχου:

εντολές που θα εκτελεσθούν αν ισχύει (αληθής-True) η συνθήκη ελέγχου

Σημείωση: Οι ομάδες εντολών που θα εκτελεστούν, αν ισχύει μια συνθήκη, ορίζονται ως ένα μπλοκ με τη χρήση εσοχής (κενά). Οι διαδοχικές εντολές που έχουν την ίδια εσοχή ανήκουν στο ίδιο μπλοκ.

Η εντολή ελέγχου **if λειτουργεί** ως εξής (Αράπογλου Α. , et al., 2016):

Αν η συνθήκη είναι αληθής, τότε το σύνολο των εντολών που περιέχονται στην δομή *if*, θα εκτελεστούν. Αλλιώς, η ροή του προγράμματος θα προσπεράσει τη δομή *if* και θα συνεχίσει από την εντολή που βρίσκεται αμέσως μετά το τέλος της *if*.

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 10) το οποίο είναι:

Πίνακας 10: Παράδειγμα σε Python για την απλή επιλογή

Ερμηνεία των εντολών	Python
Εμφανίζεται το μήνυμα «Give a number» στην οθόνη και εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή <i>a</i>	<code>a = int(input ('Give a number'))</code>
Αν η τιμή της μεταβλητής <i>a</i> είναι αρνητική εμφανίζεται στην οθόνη το μήνυμα ('Negative number')	<code>if a < 0: print ('Negative number')</code>

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 5.

2η Ενότητα - Σύνθετη δομή επιλογής

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή σύνθετης επιλογής με ένα παράδειγμα της καθημερινότητας (Πίνακας 11) το οποίο είναι:

Πίνακας 11: Παράδειγμα καθημερινότητας για τη σύνθετη επιλογή

Αν είναι σαββατοκύριακο τότε ξεκουραζόμαστε αλλιώς πάμε σχολείο

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή σύνθετης επιλογής με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 12) το οποίο είναι ³⁵:

Πίνακας 12: Παράδειγμα με τη χρήση της πλακέτας για τη σύνθετη επιλογή

Ερμηνεία των εντολών	MicroPython
Εισάγεται η βιβλιοθήκη <i>pyb</i>	<code>import pyb</code>
Αν πατήσω το κουμπί <i>USR</i> τότε	<code>sw = pyb.Switch()</code>
Θα παραχθεί μια τυχαία τιμή 0 ή 1	<code>while True:</code>

³⁵ Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνα 26).

	if sw.value(): X=(pyb.rng()%2) break
Αν η τυχαία τιμή είναι ίση με 0 τότε	if X==0:
Θα ανάψει το κόκκινο led	pyb.LED(1).on()
αλλιώς	else:
Θα ανάψει το πράσινο led	pyb.LED(2).on()
Ανάβει το κίτρινο led	pyb.LED(3).on()
Ανάβει το μπλε led	pyb.LED(4).on()
Περιμένει 1 sec	pyb.delay(1000)
Σβήνει το κόκκινο led	pyb.LED(1).off()
Σβήνει το πράσινο led	pyb.LED(2).off()
Σβήνει το κίτρινο led	pyb.LED(3).off()
Σβήνει το μπλε led	pyb.LED(4).off()

Βήμα 2:

Επίσης, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι αν, ανάλογα με την αποτίμηση μιας συνθήκης, θέλουμε να εκτελεστούν διαφορετικές ακολουθίες εντολών, τότε μπορούμε να χρησιμοποιήσουμε τη δομή επιλογής `if...else`.

Η εντολή ελέγχου **`if...else`** συντάσσεται ως εξής (Αράπογλου Α. , et al., 2016):

if συνθήκη ελέγχου:

εντολές που θα εκτελεσθούν, αν η συνθήκη ελέγχου είναι αληθής

else:

εντολές που θα εκτελεσθούν αν η συνθήκη ελέγχου είναι ψευδής

Η εντολή ελέγχου **`if...else`** λειτουργεί ως εξής (Αράπογλου Α. , et al., 2016):

Αν ισχύει η συνθήκη (έχει τιμή TRUE), θα εκτελεστεί η A ομάδα εντολών της if, αλλιώς (αν δεν ισχύει-έχει τιμή FALSE), θα εκτελεστεί η B ομάδα εντολών της else.

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python το οποίο είναι:

Πίνακας 13: Παράδειγμα σε Python για τη σύνθετη επιλογή

Ερμηνεία των εντολών	Python
Εμφανίζεται το μήνυμα «Give your age» στην οθόνη και εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή a	<pre>a = int(input('Give your age:'))</pre>
Εμφανίζεται το μήνυμα «Give your age» στην οθόνη και εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής του στη μεταβλητή b	<pre>b = int(input('Give your age:'))</pre>
Αν η τιμή της μεταβλητής a είναι μεγαλύτερη από την τιμή της μεταβλητής b, εμφανίζεται στην οθόνη το μήνυμα «Age ... is bigger than ...» με τις αντίστοιχες τιμές των μεταβλητών a και b. Διαφορετικά, εμφανίζεται στην οθόνη το μήνυμα «Age ... is bigger or equal to ...» με τις αντίστοιχες τιμές των μεταβλητών b και a.	<pre>if a > b: print('Age', a, 'is bigger than', b) else: print('Age', b, 'is bigger or equal to', a)</pre>

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 6.

3η Ενότητα - Πολλαπλή Επιλογή

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή πολλαπλής επιλογής με ένα παράδειγμα της καθημερινότητας (Πίνακας 14) το οποίο είναι:

Πίνακας 14: Παράδειγμα της καθημερινότητας για την πολλαπλή επιλογή

Αν είναι καλοκαίρι **τότε** πάμε διακοπές, **αλλιώς αν** είναι φθινόπωρο **τότε** ξεκινούν τα σχολεία, **αλλιώς αν** είναι χειμώνας **τότε** κάνει κρύο **αλλιώς** ανθίζουν τα λουλούδια

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή πολλαπλής επιλογής με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 15) το οποίο είναι ³⁶:

Πίνακας 15: Παράδειγμα με τη χρήση της πλακέτας για την πολλαπλή επιλογή

Ερμηνεία των εντολών	MicroPython
Εισάγεται η βιβλιοθήκη pyb	<code>import pyb</code>
Αν πατήσω το κουμπί USR τότε	<code>sw = pyb.Switch()</code>
Θα παραχθεί μια τυχαία τιμή 0, 1, 2 ή 3	<code>while True:</code>
	<code>if sw.value():</code>
	<code>X=(pyb.rng()%4)</code>
	<code>break</code>
Αν η τυχαία τιμή είναι ίση με 0 τότε	<code>if X==0:</code>
Θα ανάψει το κόκκινο led	<code>pyb.LED(1).on()</code>
αλλιώς αν η τυχαία τιμή είναι ίση με 1 τότε	<code>elif X==1:</code>
Θα ανάψει το πράσινο led	<code>pyb.LED(2).on()</code>
αλλιώς αν η τυχαία τιμή είναι ίση με 2 τότε	<code>elif X==2:</code>
Θα ανάψει το κίτρινο led	<code>pyb.LED(3).on()</code>
αλλιώς	<code>else:</code>
Θα ανάψει το μπλε led	<code>pyb.LED(4).on()</code>
Περιμένει 1 sec	<code>pyb.delay(1000)</code>
Σβήνει το κόκκινο led	<code>pyb.LED(1).off()</code>
Σβήνει το πράσινο led	<code>pyb.LED(2).off()</code>
Σβήνει το κίτρινο led	<code>pyb.LED(3).off()</code>
Σβήνει το μπλε led	<code>pyb.LED(4).off()</code>

Βήμα 2:

Επίσης, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι αν, ανάλογα με την αποτίμηση μιας συνθήκης, θέλουμε να εκτελεστούν διαφορετικές ακολουθίες εντολών, τότε μπορούμε να χρησιμοποιήσουμε τη δομή επιλογής `if...elif...else`.

³⁶ Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνα 27).

Η εντολή ελέγχου **if...elif...else** συντάσσεται ως εξής (Αράπογλου Α. , et al., 2016):

if συνθήκη:

εντολές_1

elif συνθήκη2:

εντολές_2

else:

εντολές_3

Η εντολή ελέγχου **if...elif...else** λειτουργεί ως εξής (Αράπογλου Α. , et al., 2016):

Εκτελείται η ομάδα εντολών στο αντίστοιχο τμήμα όπου ισχύει η συνθήκη (έχει τιμή TRUE).

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 16) το οποίο είναι:

Πίνακας 16:Πρόγραμμα σε Python για την πολλαπλή επιλογή

Ερμηνεία των εντολών	Python
Εμφανίζεται το μήνυμα «Give a number» στην οθόνη και εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή a	a = input ('Give a number')
Εμφανίζεται το μήνυμα «Give a number» στην οθόνη και εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής του στη μεταβλητή b	b = input ('Give a number')
Αν η τιμή της μεταβλητής a είναι μεγαλύτερη από την τιμή της μεταβλητής b, εμφανίζεται στην οθόνη το μήνυμα «The number ... is bigger than ...» με τις αντίστοιχες τιμές των μεταβλητών a και b, αλλιώς αν η τιμή της μεταβλητής a είναι μικρότερη από την τιμή της μεταβλητής b, εμφανίζεται στην οθόνη το μήνυμα «The	if a > b: print ('The number', a, ' is bigger than ',b) elif a < b: print ('The number', b, ' is bigger than ',a) else: print ('The numbers are equal')

number is bigger than» με τις αντίστοιχες τιμές των μεταβλητών b και a, διαφορετικά εμφανίζεται στην οθόνη το μήνυμα ('The numbers are equal')

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 7.

4η Ενότητα - Εμφωλευμένες δομές επιλογής

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή εμφωλευμένης επιλογής με ένα παράδειγμα της καθημερινότητας (Πίνακας 17) το οποίο είναι:

Πίνακας 17: Παράδειγμα καθημερινότητας για την εμφωλευμένη επιλογή

Αν είναι πρωί **τότε**

θα πάω σχολείο

Αν έχω γυμναστική **τότε**

πρέπει να φορέσω αθλητικά ρούχα και παπούτσια

αλλιώς

μπορώ να φορέσω ότι θέλω

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή εμφωλευμένης επιλογής με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 18) το οποίο είναι ³⁷:

Πίνακας 18: Παράδειγμα με τη χρήση της πλακέτας για την εμφωλευμένη επιλογή

Ερμηνεία των εντολών	MicroPython
Εισάγεται η βιβλιοθήκη pyb	import pyb
Αν πατήσω το κουμπί USR τότε	sw = pyb.Switch()
Θα παραχθεί μια τυχαία τιμή 0, 1, 2 ή 3	while True: if sw.value(): X=(pyb.rng()%4) break
Αν η τυχαία τιμή είναι ίση με 0 τότε	if X==0:

³⁷ Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνα 28).

Θα ανάψει το κόκκινο led	<code>pyb.LED(1).on()</code>
αλλιώς	<code>else:</code>
αν η τυχαία τιμή είναι ίση με 1 τότε	<code>if X==1:</code>
Θα ανάψει το πράσινο led	<code> pyb.LED(2).on()</code>
αλλιώς	<code>else:</code>
αν η τυχαία τιμή είναι ίση με 2 τότε	<code>if X==2:</code>
Θα ανάψει το κίτρινο led	<code> pyb.LED(3).on()</code>
αλλιώς	<code>else:</code>
Θα ανάψει το μπλε led	<code> pyb.LED(4).on()</code>
Περιμένει 1 sec	<code>pyb.delay(1000)</code>
Σβήνει το κόκκινο led	<code>pyb.LED(1).off()</code>
Σβήνει το πράσινο led	<code>pyb.LED(2).off()</code>
Σβήνει το κίτρινο led	<code>pyb.LED(3).off()</code>
Σβήνει το μπλε led	<code>pyb.LED(4).off()</code>

Βήμα 2:

Έπειτα, ο εκπαιδευτικός προτείνεται να αναφέρει ότι:

Εμφωλευμένα AN ονομάζονται δύο ή περισσότερες εντολές της μορφής «AN...ΤΟΤΕ...ΑΛΛΙΩΣ» που περιέχονται η μία μέσα στην άλλη (Βακάλη, et al., 2017).

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 19) το οποίο είναι:

Πίνακας 19: Πρόγραμμα σε Python για την εμφωλευμένη επιλογή

Ερμηνεία των εντολών	Python
Εμφανίζεται το μήνυμα «Give a number» στην οθόνη και εισάγεται ένας ακέραιος αριθμός από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή a	<code>a = input('Give a number:')</code>
Αν η τιμή της μεταβλητής a είναι θετική εμφανίζεται στην οθόνη το μήνυμα «The	<code>if a > 0:</code> <code> print('The number is positive')</code>

<p>number is positive» και αν είναι άρτιος εμφανίζεται στην οθόνη το μήνυμα «The number is even», διαφορετικά εμφανίζεται στην οθόνη το μήνυμα «The number is uneven». Σε περίπτωση που δεν είναι θετικός εμφανίζεται στην οθόνη το μήνυμα «The number is negative'»</p>	<pre> if a%2==0: print ('The number is even') else: print (' The number is uneven') else: print ('The number is negative or zero')</pre>
--	--

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 8.

3.4.6 Αναπαραστάσεις εκπαιδευομένων / Πρόβλεψη δυσκολιών στο διδακτικό σενάριο

Μερικές δυσκολίες που έχουν καταγραφεί κατά τη χρήση των δομών επιλογής σε προγράμματα είναι τα παρακάτω (Ξυνόγαλος, 2005):

- Τα περισσότερα προβλήματα κατανόησης από τους εκπαιδευόμενους προκύπτουν κατά την ύπαρξη σύνθετων εκφράσεων, λογικών συναρτήσεων και συνδυασμών προτάσεων με τους λογικούς τελεστές and, not, or σε προγράμματα (Δαγδιλέλης, 1996).
- Το βάθος εμφώλευσης των εντολών αυξάνει τον βαθμό δυσκολίας των προγραμμάτων. Στο (Δαγδιλέλης, 1996) επισημαίνεται ότι πολλοί ερευνητές, όπως οι Barron D. W. και Green T.R.G., συνιστούν την κατάργηση των εμφωλευμένων εντολών επιλογής, γιατί προκαλούν σύγχυση στους εκπαιδευόμενους. Από την άλλη, υπάρχουν ερευνητές που διαφωνούν με την άποψη αυτή, γιατί θεωρούν ότι σε πολύ μικρό βαθμό και σε ορισμένες μόνο περιπτώσεις η παράθεση των δομών επιλογής είναι πιο κατανοητή από την εμφώλευσή τους (Δαγδιλέλης, 1996).
- Η εκμάθηση και χρήση των δομών επιλογής διευκολύνεται από τη σωστή οργάνωση στο χώρο του προγράμματος, δηλαδή τη στοίχιση και τη χρήση εσοχών.
- Οι εκπαιδευόμενοι με καλύτερο υπόβαθρο στα μαθηματικά μαθαίνουν ευκολότερα νέες δομές.
- Στις μελέτες των (Putman, Sleeman, Baxter, & Kuspa, 1989), (Sleeman, Putman, Baxter, & Kuspa, 1988) καταγράφηκαν οι παρακάτω παρανοήσεις σχετικά με τη συμπεριφορά ενός προγράμματος ή τη ροή ελέγχου μετά από την εκτέλεση μιας εντολής επιλογής:
 - Όταν η συνθήκη είναι False, ο έλεγχος πηγαίνει στην αρχή του προγράμματος.
 - Όταν μια εντολή If έχει ως αποτέλεσμα τη διακλάδωση σε μια εντολή Print, τότε η τιμή της μεταβλητής της εντολής Print και της έκφρασης της συνθήκης εμφανίζονται στην οθόνη.

- Εκτελείται το τμήμα των εντολών από το if μέχρι το else και το τμήμα των εντολών μετά το else.
- Το τμήμα των εντολών από το If μέχρι το else της εντολής εκτελείται ανεξάρτητα από το αν η συνθήκη είναι True ή False.
- Η εντολή: If <συνθήκη> : <εντολή1> <εντολή2> ερμηνεύεται ως If <συνθήκη> : <εντολή1> else: <εντολή2>.

Τέλος, διαπιστώθηκε ότι οι εκπαιδευόμενοι πιστεύουν ότι όταν η συνθήκη μιας εντολής If είναι False η εκτέλεση του προγράμματος τερματίζεται (Putman, Sleeman, Baxter, & Kuspa, 1989), αν δεν υπάρχει τμήμα else (Sleeman, Putman, Baxter, & Kuspa, 1988).

3.4.7 Επεκτάσεις / Διασυνδέσεις των εννοιών ή των δραστηριοτήτων

Μετά την ολοκλήρωση του παρόντος σεναρίου οι εκπαιδευόμενοι θα είναι προετοιμασμένοι για να εργαστούν με επόμενα σενάρια στα οποία είτε οι εντολές επιλογής είναι βασικές είτε περιλαμβάνουν τη χρήση πιο σύνθετων αλγοριθμικών δομών όπως επανάληψη.

3.4.8 Φύλλα εργασίας 5, 6 ,7,8 και 9.

Φύλλο εργασίας 5

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
x = int(input('Give a number'))
y = int(input('Give a number'))
x = abs(x - y)
y = abs(x - y)
if y % x <= 3:
    x = int(input('Give a number'))
    y = y + x // 2
if y < x:
    x = x - y // 5
    y = int(input('Give a number'))
print(x,y)
```

1. Να προσδιορίσεις ποιές είναι οι συνθήκες οι οποίες πρέπει να ισχύουν για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παρακάτω πρόγραμμα αν δοθούν ως είσοδος οι τιμές 5, 2, 9 και 1 και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής και λογικών εκφράσεων). Εκτέλεσε το πρόγραμμα στο αρχείο f5d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά στο έτος 2018 και στην είσοδο ενός ατόμου σε μια ελεγχόμενη περιοχή. Επομένως, το πρόγραμμα να εισάγει το έτος γέννησής του, να υπολογίζει και να εμφανίζει την ηλικία του και να εμφανίζει το μήνυμα «Μπορείτε να εισέλθετε», εάν η ηλικία του είναι πάνω από 25 ετών.

2. Αποθήκευσε το πρόγραμμα με όνομα `myf5d2.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f5d2.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε έναν τελικό διαγωνισμό επίδειξης με ποδήλατο όπου έλαβαν μέρος τρεις νεαροί αθλητές, οι οποίοι βαθμολογήθηκαν για τις εντυπωσιακές τους φιγούρες που πραγματοποίησαν. Επομένως, το πρόγραμμα εισάγει το επίθετο, το όνομα και το βαθμό που έλαβε ο καθένας από αυτούς και θα εμφανίζει το επίθετο και το όνομα του αθλητή με τον καλύτερο βαθμό. Παρατήρηση: Θεωρείστε ότι όλοι οι αθλητές έλαβαν διαφορετικό βαθμό στην κλίμακα 0-100.
2. Αποθήκευσε το πρόγραμμα με όνομα `myf5d3.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f5d3.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

Να τροποποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 3 χρησιμοποιώντας δομή απλής επιλογής όπου :

1. Αν η τιμή είναι το 0 να εμφανίζει το όνομά του και μήνυμα ότι ο πρώτος παίκτης έχει κόκκινη κάρτα διαφορετικά ότι ο πρώτος παίκτης έχει μπλε κάρτα. Αντίστοιχα και για τον δεύτερο και τον βοηθητικό παίκτη.
2. Συγκρίνει το χρώμα της κάρτας του πρώτου παίκτη με το χρώμα της τυχαίας κάρτας και σε περίπτωση που είναι το ίδιο παίρνει δύο πόντους διαφορετικά δεν παίρνει κανέναν πόντο. Στο τέλος, εμφανίζει το όνομά του και τον αριθμό των πόντων του. Αντίστοιχα και για τον δεύτερο παίκτη.
3. Να τροποποιήσεις το παραπάνω πρόγραμμα ώστε να εμφανίζει το όνομα του νικητή.
4. Αποθήκευσε το πρόγραμμα με όνομα `myf5d4.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f5d4.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 6

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
a= int(input('Give a number'))
```

```
b = 2 * a + 1
```

```
c = a + b
```

```
if c > b:
```

```
    b=c
```

```
else:
```

```
    c=b
```

```
print(a,b,c)
```

1. Να προσδιορίσεις ποια είναι η συνθήκη η οποία πρέπει να ισχύει για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παρακάτω πρόγραμμα αν δοθεί ως είσοδος α) η τιμή 10 και β) η τιμή -10, προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής και λογικών εκφράσεων). Εκτέλεσε το πρόγραμμα στο αρχείο f6d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε ένα εργοστάσιο όπου η πληρωμή των εργαζομένων πραγματοποιείται ανά εβδομάδα ανάλογα με τις ώρες που έχουν εργαστεί σε διάστημα μιας εβδομάδας ως εξής: αν ένας εργαζόμενος έχει εργαστεί μέχρι και 40 ώρες, τότε πληρώνεται με 7,5€ ανά ώρα εργασίας, ενώ για κάθε ώρα πέραν των 40 θεωρείται ως υπερωρία και αμείβεται με 10€ την ώρα. Επομένως, το πρόγραμμα να εισάγει το όνομα ενός εργαζομένου και το σύνολο των ωρών εργασίας του σε μια εβδομάδα και αν ο υπάλληλος έχει εργαστεί υπερωριακά, να εμφανίζει το μήνυμα «Ο ...όνομα... δούλεψε υπερωρίες» και να υπολογίζει και να εμφανίζει το ποσό που

αντιστοιχεί στις υπερωρίες του και το ποσό πληρωμής του υπαλλήλου. Αν ο υπάλληλος δεν έχει εργαστεί υπερωριακά να εμφανίζει το μήνυμα «Ο ...όνομα... δεν δούλεψε υπερωρίες» και να υπολογίζει και να εμφανίζει το ποσό πληρωμής του υπαλλήλου.

2. Αποθήκευσε το πρόγραμμα με όνομα myf6d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f6d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε ένα κατάστημα που πούλησε τηλεοράσεις καθέναν από τους μήνες Σεπτέμβριο, Οκτώβριο και Νοέμβριο. Το κατάστημα πούλησε διαφορετικό αριθμό τηλεοράσεων κάθε μήνα. Τον Σεπτέμβριο οι τηλεοράσεις κόστιζαν 400 ευρώ η μία και το κέρδος του καταστήματος ανά τηλεόραση ήταν 5% της τιμής πώλησης. Τον Οκτώβριο το κέρδος του καταστήματος ανά υπολογιστή ήταν 40 ευρώ. Τον Νοέμβριο το κατάστημα κέρδισε 7% από κάθε υπολογιστή που κόστιζε 600 ευρώ ο καθένας. Επομένως, το πρόγραμμα να εισάγει τον αριθμό των υπολογιστών που πούλησε ένα κατάστημα καθέναν από τους παραπάνω μήνες, να υπολογίζει το κέρδος του κάθε μήνα και να εμφανίζει το όνομα του μήνα που το κατάστημα κέρδισε τα περισσότερα χρήματα. Αν η διαφορά του κέρδους μεταξύ του πιο κερδοφόρου μήνα και του λιγότερου κερδοφόρου ξεπερνά τα 5000 ευρώ να εμφανίζεται το μήνυμα «ΥΠΑΡΧΕΙ ΜΕΓΑΛΗ ΟΙΚΟΝΟΜΙΚΗ ΔΙΑΦΟΡΑ», διαφορετικά να εμφανίζεται το μήνυμα «ΔΕΝ ΥΠΑΡΧΕΙ ΜΕΓΑΛΗ ΟΙΚΟΝΟΜΙΚΗ ΔΙΑΦΟΡΑ».

Παρατήρηση: Θεωρήστε ότι δεν υπάρχει περίπτωση σε δύο μήνες να έχουν κερδίσει τα ίδια χρήματα.

2. Αποθήκευσε το πρόγραμμα με όνομα myf6d3.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f6d3.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Να τροποποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 5 χρησιμοποιώντας δομή σύνθετης επιλογής. Εκτέλεσε το παραπάνω πρόγραμμα,

σύγκρινε τα αποτελέσματα με την 4η δραστηριότητα του φύλλου εργασίας 5, εντόπισε σε ποιο πρόγραμμα γίνονται οι λιγότερες συγκρίσεις και αιτιολόγησε την απάντησή σου.

2. Να τροποποιήσεις το πρόγραμμα έτσι ώστε να προσθέτει 4 επιπλέον πόντους σε περίπτωση που έχει κόκκινο χρώμα και 6 επιπλέον πόντους στον παίκτη που έχει μπλε χρώμα.
3. Αποθήκευσε το πρόγραμμα με όνομα `myf6d4.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f6d4.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 7

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
x = int(input('Give a number'))
```

```
y=x
```

```
if x % 2==0:
```

```
    x = x + 1
```

```
if x < 11:
```

```
    y = y // 2 - x // 3 + 1
```

```
    print(x,y)
```

```
elif x < 20:
```

```
    y = y / 2 - x // 3 % 2
```

```
    print(x,y)
```

```
elif x==11:
```

```
    y = (y + x) // 3
```

```
    print(x,y)
```

```
x = x - 1
```

```
y = pow(y,3) // x
```

```
print(x,y)
```

1. Να προσδιορίσεις ποια είναι η συνθήκη η οποία πρέπει να ισχύει για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παρακάτω πρόγραμμα αν δοθεί ως είσοδος α) η τιμή 10 και β) η τιμή 9 και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής και λογικών εκφράσεων). Εκτέλεσε το πρόγραμμα στο αρχείο f7d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε ένα κατάστημα αθλητικών ειδών που παρέχει μια ειδική προσφορά σε όποια ποδοσφαιρική ομάδα αγοράσει αθλητικά παπούτσια ως εξής: αν αγοραστούν πάνω από 18 ζευγάρια αθλητικών παπουτσιών, τότε δεν χρεώνονται τα 3. Αν αγοραστούν από 10 έως 18 ζευγάρια παπουτσιών, τότε δεν χρεώνονται τα 2. Ενώ για αγορά λιγότερων από 10 ζευγάρια αθλητικών παπουτσιών, τότε δεν χρεώνεται το ένα. Αν το ένα ζευγάρι αθλητικών παπουτσιών κοστίζει 120€, το πρόγραμμα να εισάγει τον αριθμό των ζευγαριών των αθλητικών παπουτσιών που αγόρασε μια ποδοσφαιρική ομάδα και να υπολογίζει και να εμφανίζει το ποσό που θα πληρώσει η ομάδα στο κατάστημα
2. Αποθήκευσε το πρόγραμμα με όνομα myf7d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f7d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε μια διαφημιστική εταιρεία για την παραγωγή μιας τηλεοπτικής διαφήμισης που χρεώνει τον κάθε υποψήφιο πελάτη ανά δευτερόλεπτο διάρκειας της διαφήμισης ως εξής:

Διάρκειας της διαφήμισης σε δευτερόλεπτα	Χρέωση σε ευρώ ανά δευτερόλεπτο
Μέχρι 10	10
11-15	20
16-25	40
26 και πάνω	55

Στη χρέωση που προκύπτει προστίθεται ΦΠΑ 23% και αν η διαφήμιση παιχτεί σε ζώνη υψηλής ακροαματικότητας υπάρχει και επιπλέον κόστος ύψους 150€. Επομένως, το πρόγραμμα για έναν πελάτη που επιθυμεί να παράγει μια τηλεοπτική διαφήμιση να εισάγει τη χρονική διάρκεια της διαφήμισης σε δευτερόλεπτα, να ρωτάει με κατάλληλο μήνυμα αν η διαφήμιση θα προβληθεί σε ζώνη υψηλής ακροαματικότητας ή όχι, να εισάγει την απάντηση που είναι «ΝΑΙ» ή «ΟΧΙ» και να υπολογίζει και να εμφανίζει το κόστος της διαφήμισης.

2. Να τροποποιήσεις το πρόγραμμά σου έτσι ώστε να υπολογίζει και να εμφανίζει τη διαφορά που θα υπήρχε στο κόστος αν η χρέωση ανά δευτερόλεπτο διάρκειας της διαφήμισης γινόταν κλιμακωτά. Για παράδειγμα, αν μια διαφήμιση διαρκούσε 18 λεπτά το κόστος μετά από κλιμακωτό υπολογισμό θα ήταν $10*10+5*20+3*40=320\text{€}$.
3. Αποθήκευσε το πρόγραμμα με όνομα `myf7d3.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f7d3.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Να τροποποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 6 έτσι ώστε το παιχνίδι να παίζεται με κάρτες τριών χρωμάτων (κόκκινο, μπλε και πράσινο) με την πράσινη κάρτα να μειώνει τους πόντους κατά 5 χρησιμοποιώντας δομή πολλαπλής επιλογής. Σε περίπτωση που προκύψουν σε κάποιον αρνητικοί πόντοι τότε απλά μηδενίζονται.
2. Αποθήκευσε το πρόγραμμα με όνομα `myf7d4.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.
3. Θα μπορούσε να λυθεί το παραπάνω πρόγραμμα με απλή δομή επιλογής; Αν συμφωνείς εξήγησε σε ποιο πρόγραμμα γίνονται οι λιγότερες συγκρίσεις και αιτιολόγησε την απάντησή σου.

Το πρόγραμμα στο αρχείο `f7d4.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 8

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
a= 15
```

```
b = a**2
```

```
c = (a + b)//2
```

```
if c % 2==0 or c>=25:
```

```
    c = c - 20
```

```
    if not(c>a):
```

```
        a=b % a
```

```
    else:
```

```
        c=c // 2
```

```
else:
```

```
    a = a + c
```

```
    c = c + b
```

```
print(a,b,c)
```

```
a = a // b + c
```

```
b = b % c + a
```

```
c = c % a + b
```

```
print(a,b,c)
```

1. Να προσδιορίσεις ποιες είναι οι συνθήκες που πρέπει να ισχύουν για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παρακάτω πρόγραμμα και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής και λογικών εκφράσεων). Εκτέλεσε το πρόγραμμα στο αρχείο f8d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε έναν κινηματογράφο όπου τα εισιτήρια εισόδου χωρίζονται σε «Παιδικό» και σε «Κανονικό». Για τα παιδιά υπάρχει μια συγκεκριμένη αίθουσα που η τιμή του εισιτηρίου εξαρτάται από την ηλικία του παιδιού αν είναι από 5 μέχρι και 9 χρονών, το εισιτήριο είναι 5€ και αν είναι από 10 έως 14 χρονών, το εισιτήριο είναι 6€. Για παιδιά μεγαλύτερα των 14 χρόνων, το παιδί πρέπει να πάει στις αίθουσες για ενήλικες και αν το παιδί είναι κάτω από 5 χρονών απαγορεύεται η είσοδος. Για το κανονικό εισιτήριο υπάρχουν τρεις αίθουσες με συγκεκριμένη τιμή εισιτηρίου . Η ΑΙΘΟΥΣΑ_1 με τιμή εισιτηρίου τα 8€, η ΑΙΘΟΥΣΑ_2 με τιμή εισιτηρίου τα 9€ και η ΑΙΘΟΥΣΑ_3 με τιμή εισιτηρίου τα 10€. Επομένως, το πρόγραμμα να εισάγει τον τύπο του εισιτηρίου που θα είναι «ΠΑΙΔΙΚΟ» ή «ΚΑΝΟΝΙΚΟ». Αν το εισιτήριο είναι «ΠΑΙΔΙΚΟ» τότε να εισάγει την ηλικία του παιδιού και να εμφανίζει το ποσό που πρέπει να καταβάλλει ή το αντίστοιχο μήνυμα. Αν το εισιτήριο είναι «ΚΑΝΟΝΙΚΟ», να εισάγει την αίθουσα που επιθυμεί ο πελάτης και να εμφανίζει το αντίστοιχο ποσό που πρέπει να καταβάλλει.
2. Αποθήκευσε το πρόγραμμα με όνομα myf8d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f8d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε έναν δήμο που έχει τοποθετήσει σε ορισμένα σημεία αυτόματα μηχανήματα για την εξυπηρέτηση πολιτών του, όσον αφορά στην εξόφληση διαφόρων λογαριασμών. Μέσω του μηχανήματος ένας πολίτης μπορεί να εξοφλήσει λογαριασμούς νερού, ΔΕΗ, ΟΤΕ. Για κάθε λογαριασμό που εξοφλείται μέσω του μηχανήματος ο πολίτης θα πρέπει να καταβάλλει μια μικρή εισφορά στον Δήμο η οποία προκύπτει από το είδος λογαριασμού και το ποσό πληρωμής του από τον πίνακα:

Λογαριασμός	Λογαριασμός ΔΕΗ	Λογαριασμός ΟΤΕ
ΝΕΡΟΥ		
Η εισφορά είναι	Ποσό πληρωμής	Εισφορά
		Ποσό πληρωμής
		Εισφορά

2% επί του ποσού πληρωμής	Μέχρι 100	2€	< 100	1% επί του ποσού
	Μέχρι 250	4€	>=100	3% επί του ποσού
	Πάνω από 250	7€		

Επομένως, το πρόγραμμα να εμφανίζει το παρακάτω μενού:

1. Πληρωμή λογαριασμού ΝΕΡΟΥ
2. Πληρωμή λογαριασμού ΔΕΗ
3. Πληρωμή λογαριασμού ΟΤΕ
4. Έξοδος

και να εισάγει την επιλογή του πολίτη , που θα είναι ένας ακέραιος αριθμός από το 1 έως το 4. Αν η επιλογή είναι η έξοδος, τότε να εμφανίζει το μήνυμα «ΤΕΛΟΣ», διαφορετικά να εισάγει το ποσό πληρωμής του λογαριασμού και να εμφανίζει το ποσό εισφοράς που θα πρέπει να καταβάλλει ο πολίτης.

2. Αποθήκευσε το πρόγραμμα με όνομα myf8d3.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f8d3.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Να τροποποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 7 έτσι ώστε ο παίκτης να παίρνει ή να χάνει τους πόντους μόνο σε περίπτωση που έχει το ίδιο χρώμα με τον βοηθητικό παίκτη χρησιμοποιώντας δομή εμφωλευμένης επιλογής.
2. Αποθήκευσε το πρόγραμμα με όνομα myf8d4.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f8d4.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 9 - Τεστ αυτοαξιολόγησης

Για κάθε μία από τις παρακάτω 10 προτάσεις να επιλέξεις λέξη Σωστό, αν είναι Σωστή, ή τη λέξη Λάθος, αν είναι λανθασμένη.

1. Η δομή σύνθετης επιλογής ελέγχει πάντα μια σύνθετη λογική έκφραση.

Σωστό **Λάθος**

2. Δύο αλφαριθμητικές μεταβλητές μπορούν να συγκριθούν μεταξύ τους.

Σωστό **Λάθος**

3. Η δομή επιλογής εκτελείται όταν θέλουμε μια ομάδα εντολών να εκτελεστεί πολλές φορές.

Σωστό **Λάθος**

4. Στη δομή επιλογής μπορεί μία ή περισσότερες εντολές να μην εκτελεστούν.

Σωστό **Λάθος**

5. Στη δομή επιλογής μετά την εντολή If τοποθετείται μια αριθμητική έκφραση.

Σωστό **Λάθος**

6. Μία εμφωλευμένη επιλογή μπορεί να υλοποιηθεί με διαδοχικές δομές απλής επιλογής.

Σωστό **Λάθος**

7. Δεν μπορούν να τοποθετηθούν περισσότερες από τρεις εμφωλευμένες δομές επιλογής.

Σωστό **Λάθος**

8. Σε μια πολλαπλή επιλογή ελέγχονται με τη σειρά όλες οι συνθήκες.

Σωστό **Λάθος**

9. Πολύ συχνά εντολές που έχουν γραφεί με εμφωλευμένα αν μπορούν να γραφούν πιο απλά χρησιμοποιώντας σύνθετες εκφράσεις.

Σωστό **Λάθος**

10. Πολλαπλές επιλογές μπορούν να γίνουν και με μία εμφωλευμένη δομή.

Σωστό **Λάθος**

3.5 Διδακτικό σενάριο 3: Δομή επανάληψης

3.5.1 Τίτλος διδακτικού σεναρίου

Διδακτικό σενάριο στη διδασκαλία των δομών επανάληψης στη γλώσσα προγραμματισμού Python.

3.5.2 Εκτιμώμενη διάρκεια διδακτικού σεναρίου

Το διδακτικό σενάριο προβλέπεται να διαρκέσει συνολικά 3 διδακτικές ώρες.

3.5.3 Προαπαιτούμενες γνώσεις διδακτικού σεναρίου

Το διδακτικό σενάριο προορίζεται για εκπαιδευόμενους με κάποιες στοιχειώδεις γνώσεις προγραμματισμού. Συγκεκριμένα, θα πρέπει ήδη να γνωρίζουν το προγραμματιστικό περιβάλλον της συγκεκριμένης γλώσσας έχοντας ήδη πραγματοποιήσει σχετικές δραστηριότητες χρησιμοποιώντας τις ακολουθιακές δομές και τις δομές επιλογής ώστε να επιλύουν προβλήματα που αντιμετωπίζουν καθημερινά.

3.5.4 Σκοποί και στόχοι του διδακτικού σεναρίου

Ο βασικός σκοπός του διδακτικού σεναρίου είναι η εισαγωγή των εκπαιδευομένων στη δομή επανάληψης και συγκεκριμένα στις δύο μορφές της δομής επανάληψης οι οποίες περιγράφονται σε τρεις ενότητες ως εξής:

1η Ενότητα : Δομή επανάληψης while

2η Ενότητα : Δομή επανάληψης for

3η Ενότητα : Εμφωλευμένοι βρόχοι

Το διδακτικό σενάριο έχει ως στόχο μέσω πειραματισμού και προσομοιώσεων την σύνδεση των ελέγχων και της δομής επιλογής με τις επαναληπτικές διαδικασίες. Πιο συγκεκριμένα, στόχος είναι οι εκπαιδευόμενοι να:

- κατανοήσουν ότι πολλές φορές υπάρχει ανάγκη χρήσης των επαναληπτικών διαδικασιών. Για παράδειγμα, «Ο καθηγητής πληροφορικής του σχολείου καταχωρεί τον αριθμό μητρώου και τους βαθμούς στα δύο τετράμηνα πολλών μαθητών από το μάθημα «Προγραμματισμός Υπολογιστών» με σκοπό οι βαθμοί να υποστούν περεταίρω επεξεργασία
- κατανοήσουν ότι οι δομές επανάληψης χρησιμοποιούνται όταν θέλουμε να εκτελέσουμε κάτι πολλές φορές
- διακρίνουν την αναγκαιότητα χρήσης της κάθε δομής επανάληψης

- είναι ικανοί να προσδιορίζουν τη συνθήκη η οποία πρέπει να ισχύει για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν επαναληπτικά υπό τη συγκεκριμένη συνθήκη
- περιγράφουν την λειτουργία της κάθε δομής επανάληψης
- συντάσσουν σωστά την κάθε δομή επανάληψης
- μπορούν να αντιληφθούν συντακτικά και λογικά λάθη που οδηγούν στην αδυναμία εκτέλεσης των προγραμμάτων
- διακρίνουν τον ατέρμονα βρόχο
- μπορούν να αντιληφθούν την αναγκαιότητα της αρχικοποίησης των μεταβλητών και μετρητών
- μπορούν να αναπαριστούν ένα πρόβλημα που εμπεριέχει δομές επανάληψης κωδικοποιημένο σε Python βλέποντας τα αποτελέσματα άμεσα στον υπολογιστή
- εξοικειωθούν με τη χρήση ΤΠΕ
- μπορούν να δουλεύουν ομαδοσυνεργατικά.

3.5.5 Περιγραφή του διδακτικού σεναρίου

1η Ενότητα - Δομή επανάληψης while

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή επανάληψης while με ένα παράδειγμα της καθημερινότητας (Πίνακας 20) το οποίο είναι:

Πίνακας 20: Παράδειγμα της καθημερινότητας για την while

Όσο βρέχει κρατάω την ομπρέλα μου

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή επανάληψης while με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 21) το οποίο είναι ³⁸:

Πίνακας 21: Παράδειγμα με την χρήση της πλακέτας για την while

Ερμηνεία των εντολών	MicroPython
Εισάγει τη βιβλιοθήκη pyb	import pyb
Ενεργοποιεί την LCD οθόνη και ανάβει το φως της	lcd = pyb.LCD('Y') lcd.light(True)
Εμφανίζει στην οθόνη τους αριθμούς 1,3,5,7,9	i=1

³⁸ Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνες 29, 30 και 31).

	<pre> while i<=10: lcd.write(str(i)) lcd.write(' ') i=i+2 </pre>
Εισάγει τη βιβλιοθήκη pyb	<code>import pyb</code>
Ενεργοποιεί την LCD οθόνη και ανάβει το φως της	<code>lcd = pyb.LCD('Y')</code> <code>lcd.light(True)</code>
Εμφανίζει στην LCD οθόνη το μήνυμα «Random leds»	<code>lcd.write('Random leds')</code>
Παράγει έναν τυχαίο ακέραιο αριθμό από 0 μέχρι 4	<code>i=pyb.rng()%5</code>
Όσο ο τυχαίος ακέραιος αριθμός δεν είναι μηδέν επαναλαμβάνει τα παρακάτω:	<code>while i!=0:</code>
Ανάβει το αντίστοιχο led που υποδεικνύει ο αριθμός	<code>pyb.LED(i).on()</code>
Περιμένει 500ms	<code>pyb.delay(500)</code>
Σβήνει το αντίστοιχο led	<code>pyb.LED(i).off()</code>
Περιμένει 500ms	<code>pyb.delay(500)</code>
Παράγει έναν τυχαίο ακέραιο αριθμό από 0 μέχρι 4	<code>i=pyb.rng()%5</code>

Βήμα 2:

Έπειτα, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι η δομή επανάληψης εφαρμόζεται στις περιπτώσεις όπου μια ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων που έχουν κάτι κοινό και επιτρέπει την εκτέλεση εντολών περισσότερες από μία φορές (Βακάλη, et al., 2017).

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 (Διόφαντος, 2013). Η δομή `while` χρησιμοποιείται για μη προκαθορισμένο αριθμό

επαναλήψεων και σε κάθε επανάληψη (και στην αρχική) πραγματοποιείται ο έλεγχος της συνθήκης, πριν από την εκτέλεση των εντολών του βρόχου, πράγμα που σημαίνει ότι υπάρχει περίπτωση να μην εκτελεστούν οι εντολές του βρόχου (Αράπογλου Α. , et al., 2016).

Η εντολή While **συντάσσεται** ως εξής (Αράπογλου Α. , et al., 2016):

***while* ονομα_μεταβλητής <συνθήκη>:**

Εντολή_1

Εντολή_2

....

Εντολή_κ

Η δομή While **λειτουργεί** ως εξής (Αράπογλου Α. , et al., 2016):

Μέσα στο μπλοκ εντολών πρέπει να υπάρχει κατάλληλη εντολή, ώστε να εξασφαλίζεται ότι κάποια στιγμή η συνθήκη θα γίνει ψευδής και θα διακοπεί ο βρόχος, διαφορετικά δε θα τερματίζει.

Πριν το βρόχο θα πρέπει αρχικά να έχει δοθεί μία τιμή στη μεταβλητή που ελέγχει τη συνθήκη του βρόχου, ώστε ανάλογα να εκτελεστεί ή όχι ο βρόχος.

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 22) το οποίο είναι:

Πίνακας 22:Πρόγραμμα γραμμένο σε Python για την while

Ερμηνεία των εντολών	Python
Εμφανίζεται το μήνυμα «Give a number» στην οθόνη, εισάγεται μια ακέραια τιμή από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή x	x=int(input('Give a number'))
Γίνεται ανάθεση της τιμής της στη μεταβλητή maxi	maxi=x
Όσο η τιμή της μεταβλητής x δεν είναι μηδέν επαναλαμβάνει τα παρακάτω:	while x!=0:
Αν η τιμή της μεταβλητής x μεγαλύτερη από την τιμή της μεταβλητής maxi γίνεται ανάθεση της	if x>maxi:
	maxi=x
	x=int(input('Give a number'))

τιμής της στη μεταβλητή `maxi`.

Εμφανίζεται το μήνυμα «Give a number» στην οθόνη, εισάγεται μια ακέραια τιμή από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή `x`

Μετά το τέλος της επανάληψης εμφανίζεται το `print ('The maximum number is', maxi)` μήνυμα «The maximum number is» στην οθόνη και η τιμή της μεταβλητή `maxi`

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 10.

2η Ενότητα - Δομή επανάληψης `for`

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή επανάληψης `for` με ένα παράδειγμα της καθημερινότητας (Πίνακας 23) το οποίο είναι:

Πίνακας 23: Παράδειγμα της καθημερινότητας για την `for`

Για 10 μαθητές μιας τάξης ρωτάω τα ονόματά τους

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τη δομή επανάληψης `for` με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 24) το οποίο είναι ³⁹:

Πίνακας 24: Παράδειγμα με την χρήση της πλακέτας για την `for`

Ερμηνεία των εντολών	MicroPython
Εισάγει τη βιβλιοθήκη <code>pyb</code>	<code>import pyb</code>
Ενεργοποιεί την LCD οθόνη και ανάβει το φως της	<code>lcd = pyb.LCD('Y')</code> <code>lcd.light(True)</code>
Εμφανίζει στην LCD οθόνη τις τιμές 2,4,6,8,10	<code>for i in range(2,11,2):</code> <code> lcd.write(str(i))</code> <code> lcd.write(' ')</code>
<hr/>	
Εισάγει τη βιβλιοθήκη <code>pyb</code>	<code>import pyb</code>
Ενεργοποιεί την LCD οθόνη και ανάβει το φως	<code>lcd = pyb.LCD('Y')</code>

³⁹ Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνες 32, 33, 34 και 35).

της	lcd.light(True)
Εμφανίζει στην LCD οθόνη το μήνυμα «3 Random leds»	lcd.write('3 random leds')
Για 3 φορές (Τιμές 0, 1, 2) επαναλαμβάνει τα παρακάτω:	for i in range(3):
Παράγει έναν τυχαίο ακέραιο αριθμό από 0 μέχρι 3	x=pyb.rng()%4
Ανάβει το αντίστοιχο led που υποδεικνύει ο αριθμός αυξημένο κατά ένα (Πιθανές τιμές 1,2,3,4)	pyb.LED(x+1).on() pyb.delay(500)
Περιμένει 500ms	pyb.LED(x+1).off() pyb.delay(500)
Σβήνει το αντίστοιχο led	
Περιμένει 500ms	

Βήμα 2:

Επίσης, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι η δομή For χρησιμοποιείται για προκαθορισμένο αριθμό επαναλήψεων, δηλαδή όπου το πλήθος των επαναλήψεων είναι δεδομένο, πριν αρχίσουν οι επαναλήψεις (Αράπογλου Α. , et al., 2016).

Η εντολή ελέγχου **for** **συντάσσεται** ως εξής (Αράπογλου Α. , et al., 2016):

for onoma_metavlitis in range (αρχή, μέχρι, βήμα):

Εντολή_1

Εντολή_2

.....

Εντολή_v

Η εντολή ελέγχου **for** **λειτουργεί** ως εξής (Αράπογλου Α. , et al., 2016):

Η range() είναι μια ενσωματωμένη συνάρτηση της γλώσσας Python, η οποία, ανάμεσα σε άλλα, χρησιμοποιείται για την υπόδειξη του αριθμού των επαναλήψεων που θα εκτελεστούν σε ένα βρόχο. Η δομή της είναι της μορφής range (αρχή, μέχρι, βήμα), όπου αρχή, μέχρι, βήμα ακέραιοι αριθμοί. Οι ενδείξεις της αρχής και του βήματος δεν είναι υποχρεωτικές και,

αν δεν αναφέρονται, θα αρχίσει από 0 και θα συνεχίσει με βήμα 1. Αντίθετα η ένδειξη μέχρι πρέπει πάντα να αναφέρεται. Για παράδειγμα η εντολή:

`range(10)`, παράγει τη λίστα: `[0,1,2,3,4,5,6,7,8,9]`.

`range(1, 8)`, παράγει τη λίστα: `[1,2,3,4,5,6,7]`

`range(8, -1, -1)`, παράγει τη λίστα `[8, 7, 6, 5, 4, 3, 2, 1, 0]`

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 25) το οποίο είναι:

Πίνακας 25: Πρόγραμμα σε Python για την for

Ερμηνεία των εντολών	MicroPython
Γίνεται ανάθεση της τιμής 0 στη μεταβλητή S	<code>S=0</code>
Για 10 φορές (Τιμές 0, 1, 2, ..., 9) επαναλαμβάνει τα παρακάτω:	<code>for i in range(10):</code> <code> name=input('Give your name')</code>
Εμφανίζεται το μήνυμα «Give your name» στην οθόνη, εισάγεται μια αλφαριθμητική τιμή από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή name	
Εμφανίζεται το μήνυμα «Give your grade» στην οθόνη, εισάγεται μια πραγματική τιμή από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή grade	<code> grade=float(input('Give your grade'))</code>
Αυξάνεται η μεταβλητή S κατά τη μεταβλητή grade και γίνεται ανάθεση της τιμής στη μεταβλητή S	<code> S = S + grade</code>
Υπολογίζεται ο μέσος όρος των βαθμολογιών και γίνεται ανάθεση του αποτελέσματος στην μεταβλητή average	<code> average=S/10</code>
Εμφανίζεται το μήνυμα «The average grade is» στην οθόνη και η τιμή της μεταβλητής average	<code> print('The average grade is', average)</code>

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 11.

3η Ενότητα – Εμφωλευμένοι βρόχοι

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τους εμφωλευμένους βρόχους με ένα παράδειγμα της καθημερινότητας (Πίνακας 26) το οποίο είναι:

Πίνακας 26: Παράδειγμα της καθημερινότητας για τους εμφωλευμένους βρόχους

Για 15 μαθητές μια τάξης

Καταχωρούνται τα ονόματά τους στη βάση δεδομένων του σχολείου

Για 10 μαθήματα του προγράμματος σπουδών

Καταχωρούνται οι βαθμολογίες τους στη βάση δεδομένων του σχολείου

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τους εμφωλευμένους βρόχους με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 27) το οποίο είναι ⁴⁰:

Πίνακας 27: Παράδειγμα με τη χρήση της πλακέτας για τους εμφωλευμένους βρόχους

Ερμηνεία των εντολών	MicroPython
Εισάγεται η βιβλιοθήκη pyb	<code>import pyb</code>
Ενεργοποιεί την LCD οθόνη και ανάβει το φως της	<code>lcd = pyb.LCD('Y')</code> <code>lcd.light(True)</code>
Γίνεται ανάθεση των τιμών της λίστας [5,6] στη μεταβλητή i	<code>for i in range(5,7):</code>
Για κάθε μια τιμή της μεταβλητής i, γίνεται ανάθεση των τιμών της λίστας [1,2,3,4,5,6,7,8,9,10] στη μεταβλητή j	<code>for j in range(11):</code>
Υπολογίζεται το γινόμενο της τιμής της μεταβλητής i με την τιμή της μεταβλητής j και γίνεται ανάθεση του αποτελέσματος στη μεταβλητή x	<code>x=i*j</code>
Εμφανίζει στην LCD οθόνη την τιμή της	<code>lcd.write(str(x))</code>

⁴⁰ Το αντίστοιχο φωτογραφικό υλικό βρίσκεται στο παράρτημα Α (Εικόνα 36).

Βήμα 2:

Έπειτα, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι εμφωλευμένος βρόχος δημιουργείται όταν ο ένας βρόχος εμπεριέχεται μέσα σε έναν άλλο.

Οι κανόνες που ισχύουν στους εμφωλευμένους βρόχους είναι (Βακάλη, et al., 2017):

- Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό. Ο βρόχος που ξεκινάει τελευταίος πρέπει να ολοκληρώνεται πρώτος.
- Η είσοδος σε κάθε βρόχο υποχρεωτικά γίνεται από την αρχή του.
- Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής 2 ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 28) το οποίο είναι:

Πίνακας 28: Πρόγραμμα σε Python για τους εμφωλευμένους βρόχους

Ερμηνεία των εντολών	Python
Για 20 φορές (Τιμές 0, 1, 2,, 19) επαναλαμβάνει τα παρακάτω:	<code>for i in range(20):</code>
Εμφανίζεται το μήνυμα «Give the name of the city» στην οθόνη, εισάγεται μια συμβολοσειρά από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή name	<code>name=input('Give the name of the city')</code>
Γίνεται ανάθεση της τιμής 0 στη μεταβλητή p	<code>p=0</code>
Για 31 φορές (Τιμές 0, 1, 2,, 30) επαναλαμβάνει τα παρακάτω για κάθε μία από τις προηγούμενες επαναλήψεις:	<code>for j in range(31):</code>
Εμφανίζεται το μήνυμα «Give the temperature» στην οθόνη, εισάγεται μια ακέραια τιμή από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή t	<code>t=int('Give the temperature)</code>
Αν η τιμή της μεταβλητής t είναι μικρότερη του 10	<code>if t<10:</code>

Αυξάνεται η τιμή του μετρητή p κατά 1 και γίνεται ανάθεση του αποτελέσματος στη μεταβλητή p .

$p=p+1$

Εμφανίζεται το μήνυμα «The days that the temperature was under 10°C in ... are...» στην οθόνη και η τιμή της μεταβλητής $name$ και p αντίστοιχα

print('The days that the temperature was under 10°C in', name, ' are', p)

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 12.

3.5.6 Αναπαραστάσεις εκπαιδευομένων / Πρόβλεψη δυσκολιών στο διδακτικό σενάριο

Τα σημαντικότερα και πιο συνηθισμένα προβλήματα που αντιμετωπίζουν οι εκπαιδευόμενοι, όπως προκύπτει από τη σχετική βιβλιογραφία, είναι (Ξυνογάλος, 2005):

- **Αδυναμία γενίκευσης:** Οι εκπαιδευόμενοι συνήθως επαναλαμβάνουν εντολές στα προγράμματά τους αντί να χρησιμοποιούν ένα βρόχο (Hoc, 1989).
- **Ανεπαρκή νοητά μοντέλα:** Όπως προκύπτει από την έρευνα των (Kessler & Anderson, 1989) οι εκπαιδευόμενοι διαθέτουν ανεπαρκή μοντέλα για τις επαναληπτικές δομές.
- **Οι αναλλοίωτες σχέσεις που διέπουν μια επαναληπτική δομή:** Οι εκπαιδευόμενοι βασίζονται στα μοντέλα των βρόχων στην αναπαράσταση μιας αλληλουχίας ενεργειών (δυναμικό μοντέλο) και όχι στην αναπαράσταση των αναλλοίωτων σχέσεων μεταξύ των μεταβλητών (στατικό μοντέλο) χρησιμοποιώντας διαφορετικά ονόματα σε κάθε βήμα της επανάληψης για τη σηματοδότηση μιας μεταβλητής με συγκεκριμένη λειτουργία (Rogalski & Samurcay, 1990).
- **Επιλογή κατάλληλης επαναληπτικής δομής:** Οι εκπαιδευόμενοι δυσκολεύονται να επιλέξουν την καταλληλότερη επαναληπτική δομή για ένα πρόβλημα (Δαγδιλέλης, 1996).
- **Έξοδος από ένα βρόχο.** Η χρήση των βρόχων στους οποίους η συνθήκη ελέγχεται πριν να εκτελεστεί το σώμα του βρόχου, όπως για παράδειγμα στη δομή While δυσκολεύει τους εκπαιδευόμενους και αυτό ίσως να οφείλεται στη δυσκολία αναπαράστασης και εκφοράς μιας συνθήκης για ένα αντικείμενο το οποίο ακόμα δεν έχουν επεξεργαστεί (Rogalski & Samurcay, 1990).

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

- Οι εκπαιδευόμενοι συνηθίζουν να συμπεριλαμβάνουν στον βρόχο τις εντολές Print που βρίσκονται αμέσως μετά από αυτόν (Sleeman, Putman, Baxter, & Kuspa, 1988), (Putman, Sleeman, Baxter, & Kuspa, 1989).
- Ελεγχόμενη από τα δεδομένα - εισόδου επανάληψη (Data-driven looping): Οι εκπαιδευόμενοι δυσκολεύονται όταν το πλήθος των δεδομένων είναι αυτό που καθορίζει πόσες επαναλήψεις θα γίνουν (Sleeman, Putman, Baxter, & Kuspa, 1988), (Putman, Sleeman, Baxter, & Kuspa, 1989).
- Προβλήματα εμβέλειας (scope problems): Οι εκπαιδευόμενοι συνηθίζουν να κάνουν λάθη που οφείλονται σε παρανοήσεις σχετικά με την αρχή και το τέλος των βρόχων, καθώς και τις εντολές που επαναλαμβάνονται σε αυτούς (Sleeman, Putman, Baxter, & Kuspa, 1988): α) μόνο η τελευταία εντολή ενός βρόχου εκτελείται επαναληπτικά, ενώ οι υπόλοιπες εντολές μία μόνο φορά, β) η στοίχιση καθορίζει την εμβέλεια ενός βρόχου και γ) μετά την εκτέλεση ενός βρόχου, ο έλεγχος πηγαίνει στην αρχή του προγράμματος.

Ειδικά για τους βρόχους For, έχουν καταγραφεί οι εξής δυσκολίες / παρανοήσεις: οι εκπαιδευόμενοι θεωρούν ότι η μεταβλητή ελέγχου δεν έχει τιμή μέσα στο βρόχο (Sleeman, Putman, Baxter, & Kuspa, 1988) και επομένως έχουν τη δυνατότητα να αλλάζουν την τιμή της μέσα στο σώμα του βρόχου (Putman, Sleeman, Baxter, & Kuspa, 1989) μη κατανοώντας ότι η μεταβλητή ελέγχου είναι μια μεταβλητή που αυξάνεται κατά ένα βήμα σε κάθε επανάληψη (Du Boulay, 1989), (Sleeman, Putman, Baxter, & Kuspa, 1988) και ότι το πεδίο τιμών της μεταβλητής ελέγχου περιορίζει τις τιμές που μπορούν να διαβαστούν στις μεταβλητές με την εντολή Input εντός του σώματος του βρόχου (Sleeman, Putman, Baxter, & Kuspa, 1988), (Putman, Sleeman, Baxter, & Kuspa, 1989).

Ειδικά για τους βρόχους While, πολλοί εκπαιδευόμενοι πιστεύουν ότι ένας βρόχος τερματίζεται τη στιγμή που η συνθήκη ελέγχου παύει να αληθεύει (daemon-exit) και όχι αφού η συνθήκη ελεγχθεί κατά την αμέσως επόμενη επανάληψη (Du Boulay, 1989), (Sleeman, Putman, Baxter, & Kuspa, 1988). Τέλος, δυσκολεύονται και σε προγράμματα στα οποία εισάγεται μια αλληλουχία τιμών που τερματίζεται με μια τιμή-σημαία (π.χ. 99999) και γίνεται κάποια επεξεργασία των τιμών αυτών, π.χ. υπολογίζεται η μέση τιμή των τιμών εκτός της τιμής-σημαίας (99999).

Επίσης, έρευνες που έχουν γίνει πάνω στα νοητικά μοντέλα των εκπαιδευομένων δείχνουν ότι οι αυθόρμητοι συλλογισμοί τους και οι αναπαραστάσεις που διαθέτουν σχετικά με τη δομή επανάληψης έχουν κάποια συγκεκριμένη δομή και σταθερή σειρά έκφρασης η οποία

είναι: α) περιγραφή της δράσης, β) μετρητής επαναλήψεων, γ) προσδιορισμός της επανάληψης και δ) συνθήκη ελέγχου. Η δομή While ακολουθεί αντίθετη πορεία συλλογισμού και για το λόγο αυτό οι εκπαιδευόμενοι δυσκολεύονται περισσότερο να την κατανοήσουν (Κόμης Β. , 2001).

Τέλος, η χρήση επαναληπτικών δομών στηρίζεται στη μεταφορά της στρατηγικής επίλυσης με χαρτί και μολύβι σε προγραμματιστική λύση οι οποίες δεν είναι πάντα συμβατές και αυτός αποτελεί έναν πολύ σημαντικό λόγο για τον οποίο η επίλυση τέτοιου είδους προβλημάτων είναι ίσως εξαιρετικά δυσχερής για τους εκπαιδευόμενους. Για παράδειγμα, στην περίπτωση του υπολογισμού αθροιστή εντός μιας δομής επανάληψης η λύση με χαρτί και μολύβι δεν έχει ανάγκη αποθήκευσης ενδιάμεσων αποτελεσμάτων (Διόφαντος, 2013).

3.5.7 Επεκτάσεις / Διασυνδέσεις των εννοιών ή των δραστηριοτήτων

Μετά την ολοκλήρωση του παρόντος σεναρίου οι εκπαιδευόμενοι θα είναι προετοιμασμένοι για να εργαστούν με επόμενα σενάρια στα οποία είτε οι εντολές επιλογής και επανάληψης είναι βασικές είτε περιλαμβάνουν τη χρήση εμφωλευμένων βρόχων είτε περιλαμβάνουν τη χρήση λιστών.

3.5.8 Φύλλα εργασίας 10, 11, 12 και 13.

Φύλλο εργασίας 10

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
z = 1
k = 2
a = 0
while k % 2 == 0:
    x = int(input('Give a number'))
    if k ** 2 == x // 3:
        a = a + x
        k = x // 2
    else:
        z = z * x
        k = k // 2 + 1
print(x,z,k)
print(k)
```

1. Να προσδιορίσεις ποια είναι η συνθήκη η οποία πρέπει να ισχύει για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παρακάτω πρόγραμμα αν δοθούν ως είσοδος οι τιμές 12, 8 και 9 και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής και λογικών εκφράσεων). Εκτέλεσε το πρόγραμμα στο αρχείο f10d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

3. Προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος αν από το πρόβλημα παραλειφθούν οι εντολές $k = k // 2 + 1$ και $k = k // 2$.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε ένα ιδιωτικό πάρκινγκ αυτοκινήτων όπου η χρέωση είναι 0.03€ ανά λεπτό παραμονής σε αυτό. Επομένως, το πρόγραμμα για κάθε αυτοκίνητο που εξέρχεται από το πάρκινγκ να εισάγει τον αριθμό του αυτοκινήτου και τα λεπτά παραμονής σε αυτό, να υπολογίζει και να εμφανίζει το ποσό που πρέπει να πληρώσει ο κάτοχος του αυτοκινήτου και να τερματίζει όταν δοθεί ως αριθμός αυτοκινήτου το κενό.
2. Να τροποποιήσεις το πρόγραμμα έτσι ώστε να υπολογίζει και να εμφανίζει το πλήθος των αυτοκινήτων που εξυπηρέτησε το πάρκινγκ και τις συνολικές εισπράξεις του πάρκινγκ.
3. Τι πιστεύεις ότι θα εμφανίζει το πρόγραμμα αν εισαχθεί κατευθείαν ως αριθμός αυτοκινήτου το κενό;
4. Αποθήκευσε το πρόγραμμα με όνομα myf10d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f10d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε μια δημοπρασία ενός οίκου που διεξήχθη τον Δεκέμβριο του 2018 και παρουσιάστηκαν έργα – πίνακες τριών γνωστών ελλήνων ζωγράφων με ονόματα Α, Β και Γ με σκοπό τη συγκέντρωση χρημάτων για ανέγερση ενός νοσοκομείου. Επομένως, το πρόγραμμα να εισάγει το όνομα του καλλιτέχνη και την τιμή πώλησης για κάθε πίνακα ζωγραφικής που δημοπρατήθηκε, να τερματίζει όταν δοθεί η τελεία ως όνομα καλλιτέχνη, να υπολογίζει και να εμφανίζει την υψηλότερη τιμή στην οποία πουλήθηκε ένας πίνακας ζωγραφικής και το όνομα του καλλιτέχνη που τον δημιούργησε, το πλήθος των έργων που δημοπρατήθηκαν από κάθε καλλιτέχνη και το συνολικό ποσό των χρημάτων που συγκεντρώθηκε από την δημοπράτηση των έργων.
2. Τι πιστεύεις ότι θα εμφανίζει το πρόγραμμα αν εισαχθεί κατευθείαν η τελεία ως όνομα καλλιτέχνη;
3. Αποθήκευσε το πρόγραμμα με όνομα myf10d3.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f10d3.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Να τροποποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 8 έτσι ώστε το παιχνίδι να επαναλαμβάνεται για πολλούς γύρους μέχρι ο ένας από τους δύο παίκτες να κερδίσει πρώτος 10 πόντους.
2. Να τροποποιήσεις το πρόγραμμα έτσι ώστε να υπολογίζει και να εμφανίζει για πόσους γύρους παίχτηκε το παιχνίδι.
3. Να τροποποιήσεις το πρόγραμμα ώστε να εμφανίζει τους πόντους που συγκέντρωσε ο κάθε παίκτης σε κάθε γύρο και συνολικά σε όλους τους γύρους.
4. Αποθήκευσε το πρόγραμμα με όνομα myf10d4.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f10d4.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 11

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
a = 0
```

```
b = 1
```

```
c = 2
```

```
for k in range(3,11,4):
```

```
    a = a + k
```

```
    if a <= 8:
```

```
        b = b + 1
```

```
    else:
```

```
        c = c * k
```

```
print(a,b,c)
```

1. Να προσδιορίσεις ποια είναι η συνθήκη η οποία πρέπει να ισχύει για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παρακάτω πρόγραμμα και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής και λογικών εκφράσεων). Εκτέλεσε το πρόγραμμα στο αρχείο f11d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

3. Προσπάθησε να δημιουργήσεις ένα ισοδύναμο τμήμα προγράμματος χρησιμοποιώντας τη δομή επανάληψης while.

Απάντηση:.....

.....

.....

.....

.....
.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε μία εταιρεία που απασχολεί 30 υπαλλήλους όπου οι μηνιαίες αποδοχές κάθε υπαλλήλου κυμαίνονται από 0€ έως και 3000€. Επομένως, το πρόγραμμα για κάθε υπάλληλο να εισάγει το ονοματεπώνυμο και τις μηνιαίες αποδοχές του και να υπολογίζει το ποσό του φόρου κλιμακωτά, σύμφωνα με τον παρακάτω πίνακα:

Μηνιαίες αποδοχές	Ποσοστό κράτησης φόρου
Έως και 700€	10%
Άνω των 700€ έως και 1000€	15%
Άνω των 1000€ έως και 1700€	30%
Άνω των 1700€	40%

να εμφανίζει το ονοματεπώνυμο, τις μηνιαίες αποδοχές, τον φόρο και τις καθαρές μηνιαίες αποδοχές που προκύπτουν μετά την αφαίρεση του φόρου, να υπολογίζει και θα εμφανίζει το συνολικό ποσό που αντιστοιχεί στον φόρο όλων των υπαλλήλων και το συνολικό ποσό που αντιστοιχεί στις καθαρές μηνιαίες αποδοχές όλων των υπαλλήλων.

2. Να τροποποιήσεις το πρόγραμμα έτσι ώστε να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του υπαλλήλου με τις περισσότερες καθαρές αποδοχές και το ποσοστό των υπαλλήλων με καθαρές αποδοχές από 1000€ έως και 2000€.
3. Αποθήκευσε το πρόγραμμα με όνομα myf11d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f11d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε έναν ιδιωτικό τηλεοπτικό σταθμό που διοργανώνει ακροάσεις για ερασιτέχνες τραγουδιστές, με σκοπό να επιλέξει ορισμένους από αυτούς για να συμμετάσχουν σε ένα τηλεπαιχνίδι. Η επιλογή των ατόμων γίνεται από μια πενταμελή κριτική επιτροπή με τον παρακάτω τρόπο: η επιτροπή θέτει ένα όριο βαθμών που πρέπει να λάβει ο κάθε διαγωνιζόμενος για να

επιλεγεί, ο κάθε υποψήφιος τραγουδάει ένα τραγούδι της αρεσκείας του και κάθε μέλος της επιτροπής δίνει και έναν μοναδικό βαθμό από 1 μέχρι 100, από τους πέντε βαθμούς αφαιρείται ο μεγαλύτερος και ο μικρότερος, αν ο μέσος όρος των βαθμών που απομένουν είναι τουλάχιστον ίσος με το όριο της επιτροπής μειωμένο κατά 5%, τότε ο διαγωνιζόμενος επιλέγεται. Επομένως, το πρόγραμμα να εισάγει το όριο που έχει θεσπίσει η επιτροπή, για κάθε ένα από τα 500 άτομα που προσήλθαν στα δοκιμαστικά να εισάγει το όνομά του και τον βαθμό που έλαβε από κάθε μέλος της επιτροπής, να ελέγχει αν επιλέγεται ή όχι εμφανίζοντας κατάλληλο μήνυμα δίπλα από το όνομά του και να υπολογίζει και να εμφανίζει το ποσοστό αυτών που δεν επιλέχτηκαν.

2. Να τροποποιήσεις το πρόγραμμά σου έτσι ώστε ο αριθμός των ατόμων να μην είναι σταθερός 500, αλλά να μπορεί ο χρήστης να τον προσδιορίζει κάθε φορά που ξεκινάει να εκτελεί το πρόγραμμα.
3. Να τροποποιήσεις το πρόγραμμά σου έτσι ώστε να εξασφαλίζει ότι ο αριθμός των ατόμων που εισάγει ο χρήστης είναι μια τιμή από 1 μέχρι 500.
4. Αποθήκευσε το πρόγραμμα με όνομα `myf11d3.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f11d3.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Να τροποποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 10 έτσι ώστε το παιχνίδι να επαναλαμβάνεται για 5 γύρους.
2. Να τροποποιήσεις το πρόγραμμα ώστε να εμφανίζει τον αριθμό του γύρου όπου κάθε παίκτης συγκέντρωσε του περισσότερους θετικούς πόντους και πόσοι ήταν αυτοί. Σε περίπτωση ισοβαθμίας να εμφανίζει τον πρώτο γύρο κατά σειρά. Διαφορετικά, να εμφανίζει μήνυμα ότι σε κανέναν γύρο δεν είχε θετικούς πόντους.
3. Αποθήκευσε το πρόγραμμα με όνομα `myf11d4.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f11d4.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 12

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
x = int(input('Give a number'))
```

```
a = 0
```

```
while x**2 > 5:
```

```
    if x % 2 == 0:
```

```
        k = 1
```

```
        while k <= 3 or 20 > a:
```

```
            a = a + x + k
```

```
            print(x,k,a)
```

```
            k = k + 2
```

```
        else:
```

```
            for l in range(-8,1,3):
```

```
                print(l)
```

```
    x = x - 1
```

1. Να προσδιορίσεις ποια είναι η συνθήκη η οποία πρέπει να ισχύει για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παρακάτω πρόγραμμα και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής και λογικών εκφράσεων). Εκτέλεσε το πρόγραμμα στο αρχείο f12d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά σε ένα περιοδικό αυτοκινήτων που έκανε μια μελέτη για τα αυτοκίνητα της μεσαίας κατηγορίας. Στα πλαίσια της έρευνας μελέτησε 25 αυτοκίνητα καταγράφοντας 10 ποσοτικούς δείκτες για

το καθένα (αξιολόγηση παθητικής και ενεργητικής ασφάλειας, πολυτέλειας, οικονομίας καυσίμου κλπ). Επομένως, το πρόγραμμα να εισάγει το όνομα κάθε μοντέλου και τις τιμές των 10 δεικτών, να εμφανίζει το όνομα του μοντέλου που προτείνεται από το περιοδικό ως η καλύτερη αγορά (αυτό με τη μεγαλύτερη μέση τιμή των δεικτών που είναι μόνο ένα) και να υπολογίζει και να εμφανίζει το πλήθος των μοντέλων που είχαν μέσο δείκτη μεγαλύτερο από 7.

Παρατήρηση: Θεωρείστε ότι το όνομα του μοντέλου που προτείνεται από το περιοδικό ως η καλύτερη αγορά είναι μοναδικό.

2. Αποθήκευσε το πρόγραμμα με όνομα myf12d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f12d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που αφορά στις ανάγκες του φετινού διαγωνισμού ΑΣΕΠ όπου έχουν δεσμευτεί 350 αίθουσες διαφορετικής χωρητικότητας σε εξεταστικά κέντρα σε ολόκληρη τη χώρα. Ο αριθμός των επιτηρητών που απαιτούνται ανά αίθουσα καθορίζεται από το πλήθος των εξεταζομένων που βρίσκονται σε αυτή, σύμφωνα με τον παρακάτω πίνακα:

Άτομα που διαγωνίζονται	Αριθμός επιτηρητών
μέχρι και 12	1
από 13 μέχρι και 22	2
περισσότερα από 22	3

Επομένως το πρόγραμμα για κάθε αίθουσα να εισάγει επαναληπτικά τα ονόματα των εξεταζομένων που θα βρίσκονται σε αυτήν, να υπολογίζει και να εμφανίζει το πλήθος των ατόμων που διαγωνίζονται στη συγκεκριμένη αίθουσα, να τερματίζει όταν δοθεί ως όνομα εξεταζομένου η λέξη "τέλος", να υπολογίζει και να εμφανίζει το πλήθος των επιτηρητών που απαιτούνται για αυτήν, να εμφανίζει το πλήθος των επιτηρητών που απαιτούνται συνολικά για τη διενέργεια των εξετάσεων και να εμφανίζει το μεγαλύτερο πλήθος εξεταζομένων που θα εξεταστούν σε κάποια αίθουσα.

Παρατήρηση: Θεωρείστε ότι η καταχώρηση των ατόμων για κάθε αίθουσα είναι σωστή και δεν υπάρχει περίπτωση να παραβιάζεται η χωρητικότητα κάποιας αίθουσας.

2. Αποθήκευσε το πρόγραμμα με όνομα `myf12d3.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f12d3.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Να τροποποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 11 έτσι ώστε σε κάθε γύρο του παιχνιδιού οι παίκτες να παίρνουν εναλλάξ από 3 κάρτες και οι επιπλέον πόντοι του να υπολογίζονται από το άθροισμα των πόντων ανάλογα με το χρώμα που του έτυχε.
2. Να τροποποιήσεις το πρόγραμμα ώστε να εμφανίζει τους πόντους που συγκέντρωσε ο κάθε παίκτης σε κάθε γύρο.
3. Να τροποποιήσεις το πρόγραμμα ώστε να εμφανίζει τον αριθμό του γύρου όπου κάθε παίκτης συγκέντρωσε του περισσότερους πόντους. Σε περίπτωση ισοβαθμίας να εμφανίζει τον πρώτο γύρο κατά σειρά.
4. Αποθήκευσε το πρόγραμμα με όνομα `myf12d4.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f12d4.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 13 - Τεστ αυτοαξιολόγησης

Για κάθε μία από τις παρακάτω 10 προτάσεις να επιλέξεις λέξη Σωστό, αν είναι Σωστή, ή τη λέξη Λάθος, αν είναι λανθασμένη.

1. Η μεταβλητή που συμμετέχει στη συνθήκη ελέγχου μιας δομής «while» δεν μπορεί να μεταβάλλεται εντός του βρόχου, γιατί τότε πραγματοποιούνται άπειρες επαναλήψεις.

Σωστό Λάθος

2. Οι εντολές που βρίσκονται σε μία επανάληψη while εκτελούνται τουλάχιστον μία φορά.

Σωστό Λάθος

3. Η δομή επανάληψης while εκτελείται όσο η συνθήκη είναι True.

Σωστό Λάθος

4. Τη δομή επανάληψης while μπορούμε να τη χρησιμοποιούμε όταν δε γνωρίζουμε τον αριθμό των επαναλήψεων.

Σωστό Λάθος

5. Οι εντολές που βρίσκονται σε μία επανάληψη for εκτελούνται τουλάχιστον μία φορά.

Σωστό Λάθος

6. Στη δομή επανάληψης for δεν είναι δυνατόν η αρχική τιμή να είναι μεγαλύτερη από την τελική.

Σωστό Λάθος

7. Στη δομή επανάληψης for αν το βήμα είναι αρνητικός αριθμός, τότε δεν εκτελείται καμία επανάληψη.

Σωστό Λάθος

8. Κάθε βρόχος while μπορεί να μετατραπεί σε for και κάθε βρόχος for μπορεί να μετατραπεί σε while.

Σωστό Λάθος

9. Με την εντολή for i in range(1,21,2) γίνονται 11 επαναλήψεις και η συνάρτηση range(1,21,2) επιστρέφει τις τιμές [1,3,5,7,9,11,13,15,17,19,21]

Σωστό Λάθος

10. Με την εντολή «x=0 while x!=0:» δεν γίνεται καμία επανάληψη.

Σωστό Λάθος

3.6 Διδακτικό σενάριο 4: Λίστες

3.6.1 Τίτλος διδακτικού σεναρίου

Διδακτικό σενάριο στη διδασκαλία των λιστών στη γλώσσα προγραμματισμού Python.

3.6.2 Εκτιμώμενη διάρκεια διδακτικού σεναρίου

Το διδακτικό σενάριο προβλέπεται να διαρκέσει συνολικά 3 διδακτικές ώρες.

3.6.3 Προαπαιτούμενες γνώσεις διδακτικού σεναρίου

Το διδακτικό σενάριο προορίζεται για εκπαιδευόμενους με κάποιες στοιχειώδεις γνώσεις προγραμματισμού. Συγκεκριμένα, θα πρέπει ήδη να γνωρίζουν το προγραμματιστικό περιβάλλον της συγκεκριμένης γλώσσας έχοντας ήδη πραγματοποιήσει σχετικές δραστηριότητες χρησιμοποιώντας τις ακολουθιακές δομές, τις δομές επιλογής και επανάληψης ώστε να επιλύουν προβλήματα που αντιμετωπίζουν καθημερινά.

3.6.4 Σκοποί και στόχοι του διδακτικού σεναρίου

Ο βασικός σκοπός του διδακτικού σεναρίου είναι η εισαγωγή των εκπαιδευόμενων στις λίστες οι οποίες περιγράφονται σε δύο ενότητες ως εξής:

1η Ενότητα: Εισαγωγή στις λίστες

2η Ενότητα: Βασικές επεξεργασίες λιστών

Το διδακτικό σενάριο έχει ως στόχο μέσω πειραματισμού και προσομοιώσεων την σύνδεση των δομών επανάληψης με τις λίστες. Πιο συγκεκριμένα, στόχος είναι οι εκπαιδευόμενοι να:

- περιγράφουν και να ορίζουν την δομή «λίστα»
- διακρίνουν την αναγκαιότητα χρήσης της λίστας
- δημιουργούν μια λίστα
- διασχίζουν, προσθέτουν, αφαιρούν ή τροποποιούν στοιχεία σε μια λίστα
- μπορούν να αντιληφθούν συντακτικά και λογικά λάθη που οδηγούν στην αδυναμία εκτέλεσης των προγραμμάτων
- μπορούν να αναπαριστούν ένα πρόβλημα που εμπεριέχει λίστες κωδικοποιημένο σε Python βλέποντας τα αποτελέσματα άμεσα στον υπολογιστή
- εξοικειωθούν με τη χρήση ΤΠΕ
- μπορούν να δουλεύουν ομαδοσυνεργατικά.

3.6.5 Περιγραφή του διδακτικού σεναρίου

1η Ενότητα – Εισαγωγή στις λίστες

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι μια δομή δεδομένων μπορεί να οριστεί ως ένα σχήμα οργάνωσης σχετικών μεταξύ τους στοιχείων δεδομένων και ένα παράδειγμα είναι η λίστα. Η λίστα είναι μια διατεταγμένη ακολουθία αντικειμένων, όχι απαραίτητα του ίδιου τύπου και αποτελεί τη βασική δομή δεδομένων της Python. Είναι μια δυναμική δομή στην οποία μπορούμε να προσθέτουμε ή να αφαιρούμε στοιχεία (mutable) (Αράπογλου Α. , et al., 2016).

Βήμα 2:

Έπειτα, ο εκπαιδευτικός προτείνεται να εξηγήσει ότι:

- Κάθε αντικείμενο (Αράπογλου Α. , et al., 2016) της λίστας χαρακτηρίζεται από ένα μοναδικό αύξοντα αριθμό, ο οποίος ορίζει τη θέση του στη λίστα, ενώ η προσπέλαση στα στοιχεία της λίστας γίνεται με το όνομα της λίστας και τον αύξοντα αριθμό του αντικείμενου μέσα σε αγκύλες. Η αρίθμηση των στοιχείων ξεκινάει από το 0. Για παράδειγμα η εντολή `L = [3, 5, 8, 13, 21, 34]` δημιουργεί τη μεταβλητή `L` που αναφέρεται στη λίστα `[3, 5, 8, 13, 21, 34]`. Άρα το 1ο στοιχείο της λίστας είναι το `L[0]`, το οποίο είναι ίσο με το 3, το 2ο το `L[1]` και το τελευταίο το `L[5]`.
- Στις λίστες χρησιμοποιείται ο τελεστής `in`:

item in List: επιστρέφει `True`, αν το στοιχείο `item` υπάρχει μέσα στη λίστα `List`, αλλιώς επιστρέφει `False`.

item not in List: επιστρέφει `True`, αν το στοιχείο `item` δεν υπάρχει μέσα στη λίστα `List`, αλλιώς, αν υπάρχει επιστρέφει `False`.

- Συναρτήσεις που παίρνουν ως όρισμα μια λίστα είναι:

len (List): Επιστρέφει το πλήθος των στοιχείων (ή μέγεθος) της λίστας.

list (String): Επιστρέφει μια λίστα με στοιχεία τους χαρακτήρες της συμβολοσειράς `String`.

- Οι λίστες διαθέτουν μεγάλη ποικιλία μεθόδων:

L.append(object): προσθήκη του στοιχείου `object` στο τέλος της λίστας `L`.

L.insert(index, object): προσθήκη του στοιχείου `object`, στη θέση `index` της λίστας `L`, μετακινώντας όλα τα στοιχεία από τη θέση `index` και μετά, κατά μια θέση.

L.pop(index): Αφαίρεση από τη λίστα του στοιχείου που βρίσκεται στη θέση index. Αν δεν δοθεί θέση, τότε θα αφαιρεθεί το τελευταίο στοιχείο της λίστας.

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 14.

2η Ενότητα – Βασικές Επεξεργασίες λιστών

Βήμα 1:

Αρχικά, ο εκπαιδευτικός προτείνεται να συσχετίσει τη λίστα με ένα παράδειγμα της καθημερινότητας (Πίνακας 29) το οποίο είναι:

Πίνακας 29: Παράδειγμα της καθημερινότητας για την λίστα

Λίστα σούπερ μάρκετ: καφές, ζάχαρη, μακαρόνια, σάλτσα

Έπειτα, ο εκπαιδευτικός προτείνεται να συσχετίσει τη λίστα με ένα παράδειγμα με τη χρήση της πλακέτας (Πίνακας 30) το οποίο είναι:

Πίνακας 30: Παράδειγμα με τη χρήση της πλακέτας για την λίστα

Ερμηνεία των εντολών	MicroPython (Damien & Sokolovsky, 2018)
Εισάγεται η βιβλιοθήκη pyb	<code>import pyb</code>
Δημιουργείται μια κενή λίστα με όνομα leds	<code>leds=[]</code>
Για 4 φορές (Τιμές 0,1,2,3) επαναλαμβάνει τα παρακάτω:	<code>for i in range(4):</code>
Προστίθεται στη λίστα το LED(1), LED(2), LED(3), LED(4)	<code>leds.append(pyb.LED(i+1))</code>
Για 100 φορές (Τιμές 0, 1, 2, ..., 99) επαναλαμβάνει τα παρακάτω:	<code>for i in range(100):</code>
Παράγει μια τυχαία ακέραια τιμή από το 0 μέχρι το 3	<code>X=(pyb.rng())% (len(leds))</code>
Αναβοσβήνει το led που υποδεικνύει ο τυχαίος αριθμός	<code>leds[X].toggle()</code>
Καθυστερεί 50ms	<code>pyb.delay(50)</code>
<hr/>	
Εισάγεται η βιβλιοθήκη pyb	<code>import pyb</code>
Δημιουργείται μια κενή λίστα με όνομα leds	<code>leds = []</code>

Για 4 φορές (Τιμές 1,2,3,4) επαναλαμβάνει τα `for i in range(1,5):`
παρακάτω:

Προστίθεται στη λίστα το LED(1), LED(2), LED(3), LED(4) `leds.append(pyb.LED(i))`

Γίνεται ανάθεση της τιμής 0 στη μεταβλητή n `n = 0`

Επαναλαμβάνει συνεχώς τα παρακάτω: `while True:`

Γίνεται ανάθεση των τιμών 1, 2, 3 και 0 επαναληπτικά στη μεταβλητή n `n = (n + 1) % 4`

Αναβοσβήνει το led που υποδεικνύει ο αριθμός n `leds[n].toggle()`

Καθυστερεί 500ms `pyb.delay(500)`

Βήμα 2:

Επίσης, ο εκπαιδευτικός προτείνεται να εξηγήσει τον τρόπο με τον οποίο γίνεται η διάσχιση μιας λίστας (Πίνακας 31).

Πίνακας 31: Διάσχιση μιας λίστας

1ος τρόπος (Αράπογλου Α. , et al., 2016)

2ος τρόπος (Αγγελιδάκης, 2015)

for item **in** List :

`i=0`

<Εντολές Επεξεργασίας του αντικειμένου `while i<len(List):`
item>

<Εντολές Επεξεργασίας του αντικειμένου
List[i]>

`i=i+1`

Τέλος, ο εκπαιδευτικός προτείνεται να δείξει ένα πρόγραμμα γραμμένο σε Python (Πίνακας 32) το οποίο είναι:

Πίνακας 32: Πρόγραμμα σε Python για την λίστα

Ερμηνεία των εντολών

Python

Δημιουργείται μια κενή λίστα

`L=[]`

Για 10 φορές (Τιμές 0, 1, 2, 3, 4, 5, 6, 7, 8, 9) επαναλαμβάνει τα παρακάτω: `For i in range (10):`

Εμφανίζεται το μήνυμα «Give a number» στην `x=int(input('Give a number'))`

οθόνη, εισάγεται μια ακέραια τιμή από το πληκτρολόγιο και γίνεται ανάθεση της τιμής της στη μεταβλητή x

Προστίθεται η τιμή της μεταβλητής x στη λίστα L `L.append(x)`

Γίνεται ανάθεση της τιμής 0 στη μεταβλητή s `s = 0`

Για κάθε στοιχείο που βρίσκεται στη λίστα L `for item in L :`

Προστίθεται η τιμή της μεταβλητής item στην τιμή της μεταβλητής s και γίνεται ανάθεση του αποτελέσματος στη μεταβλητή s. `s = s + item`

Διαιρείται η τιμή της μεταβλητής s με το μέγεθος της λίστας L και γίνεται ανάθεση του αποτελέσματος στη μεταβλητή average `average = s / len(L)`

Εμφανίζεται στην οθόνη το μήνυμα «The average is» και η τιμή της μεταβλητής average `print ('The average is', average)`

Βήμα 3: Ακολουθούνται οι οδηγίες του φύλλου εργασιών 15.

3.6.6 Αναπαραστάσεις εκπαιδευομένων / Πρόβλεψη δυσκολιών στο διδακτικό σενάριο

Οι λίστες αποτελούν ένα σημαντικό αντικείμενο του προγραμματισμού, που εισάγει τους εκπαιδευόμενους στις δομές δεδομένων και στην ανάπτυξη της αφαιρετικής σκέψης. Οι δυσκολίες που παρατηρούνται κατά τη χρήση της, αφορούν στον έλεγχο της και στον προγραμματισμό βασικών λειτουργιών της. Οι εκπαιδευόμενοι αντιμετωπίζουν δυσκολίες στο χειρισμό των θέσεων και των στοιχείων της λίστας. Γενικά, παρατηρείται μια σύγχυση ανάμεσα στο περιεχόμενο του στοιχείου και στη θέση που αποθηκεύεται (Αμανατίδης, 2014).

3.6.7 Επεκτάσεις / Διασυνδέσεις των εννοιών ή των δραστηριοτήτων

Μετά την ολοκλήρωση του παρόντος σεναρίου οι εκπαιδευόμενοι θα είναι προετοιμασμένοι για να εργαστούν με επόμενα σενάρια στα οποία περιλαμβάνονται αλγόριθμοι με τις βασικές επεξεργασίες επί των λιστών όπως ταξινόμηση, αναζήτηση κτλ.

3.6.8 Φύλλα εργασίας 14,15 και 16.

Φύλλο εργασίας 14

Να προβλέψεις τις απαντήσεις των παρακάτω ερωτημάτων, να τα εκτελέσεις στον υπολογιστή και να συγκρίνεις τα αποτελέσματα με τις προβλέψεις σου.

Δραστηριότητα 1

1. Δημιούργησε τη λίστα $L=[15,6,10,23]$ στο Python Idle και πληκτρολόγησε το L για να το επαληθεύσεις.
2. Ποιο είναι το περιεχόμενο του πρώτου και το τελευταίου στοιχείου της λίστας.

Απάντηση:.....

3. Εκτέλεσε στο Python Shell τα παρακάτω:

```
>>> id(L[0])
```

Απάντηση:.....

```
>>> id(L[1])
```

Απάντηση:.....

Τί παρατηρείς; Το κάθε στοιχείο της λίστας δείχνει σε διαφορετική θέση μνήμης ή όχι;

Απάντηση:.....

4. Να προσθέσεις τα ονόματα 3 φίλων σου σε μία λίστα $L1$. Τι θα συμβεί αν εκτελέσεις στον υπολογιστή την εμφάνιση του στοιχείου με δείκτη 3;

Απάντηση:.....

5. Να προσθέσεις τα ονόματα 2 φίλων σου σε μία λίστα $L2$, να προσθέσεις τις δύο λίστες $L1$ και $L2$ δημιουργώντας μία λίστα L και να την εμφανίσεις .

Απάντηση:.....

Δραστηριότητα 2

1. Να υπολογίσεις το μέγεθος της λίστας L με τη χρήση της συνάρτησης $len(list)$.

Απάντηση:.....

2. Να ελέγξεις αν η 'Μαρία' είναι αποθηκευμένη στη λίστα με τη χρήση του τελεστή in .

Απάντηση:.....

3. Να μετατρέψεις σε λίστα το 1ο στοιχείο της λίστας L με τη χρήση της συνάρτησης $list()$.

Απάντηση:.....

Δραστηριότητα 3

1. Να προσθέσεις έναν ακόμη φίλο σου στο τέλος της λίστας με τη χρήση της μεθόδου $append()$ και να εμφανίσεις τη λίστα.

Απάντηση:.....

2. Να προσθέσεις τον «Γιάννη» στην 4η θέση της λίστας με τη χρήση της μεθόδου $insert()$ και να εμφανίσεις τη λίστα.

Απάντηση:.....

3. Να αφαιρέσεις το 1ο στοιχείο της λίστας με τη χρήση της μεθόδου pop() και να εμφανίσεις τη λίστα.

Απάντηση:.....

Δραστηριότητα 4

Να προσθέσεις τον αριθμό 100 στο τέλος της λίστας με τη χρήση της μεθόδου append() και να εμφανίσεις τη λίστα. Υπάρχει δυνατότητα μία λίστα να περιέχει στοιχεία διαφορετικού τύπου;

Απάντηση:.....

Φύλλο εργασίας 15

Δραστηριότητα 1

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
b=[]  
  
for i in range(8):  
    x=int(input('Give a number'))  
    b.append(x)  
  
print(b[0])  
  
a=b[0]  
  
for i in range(1,8):  
    if i%2==0:  
        a=a+b[i]  
    else:  
        print(b[i]+i+1)  
  
print(a)
```

1. Να προσδιορίσεις ποια είναι η συνθήκη η οποία πρέπει να ισχύει για να εκτελεστούν οι εντολές και ποιες είναι αυτές οι εντολές που θα εκτελεστούν.

Απάντηση:.....

2. Να εκτελέσεις με το μυαλό σου το παρακάτω πρόγραμμα αν ως είσοδος δοθούν οι τιμές -4, 6, -1, 2, 8, 3, 1, 5 και προσπάθησε να προβλέψεις την εξέλιξη της εκτέλεσης του προγράμματος (τιμές κάθε μεταβλητής και λογικών εκφράσεων). Εκτέλεσε το πρόγραμμα στο αρχείο f15d1.py στον υπολογιστή και σύγκρινε τα αποτελέσματα με τις προβλέψεις σου.

Απάντηση:.....

Δραστηριότητα 2

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που να αφορά σε ένα εργοστάσιο που παράγει αναψυκτικό τύπου «κόλα» και το εμφιαλώνει σε γυάλινα μπουκάλια των 750ml. Επομένως, το πρόγραμμα για κάθε μία ημέρα του μήνα Ιουνίου να εισάγει τον αριθμό των μπουκαλιών που εμφιαλώθηκαν και τα αποθηκεύει σε μία λίστα.

2. Να τροποποιήσεις το πρόγραμμα έτσι ώστε να υπολογίζει και να εμφανίζει τη συνολική ποσότητα μπουκαλιών που εμφιαλώθηκαν και τη μέση ημερήσια ποσότητα εμφιάλωσης των μπουκαλιών.
3. Να τροποποιήσεις το πρόγραμμα έτσι ώστε να υπολογίζει και να εμφανίζει την ημέρα (1 έως 30) του μήνα που εμφιαλώθηκαν τα λιγότερα μπουκάλια.
4. Αποθήκευσε το πρόγραμμα με όνομα myf15d2.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Παρατήρηση: Θεωρείστε ότι δεν υπάρχουν δύο μέρες που να εμφιαλώθηκε ο ίδιος αριθμός μπουκαλιών.

Το πρόγραμμα στο αρχείο f15d2.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 3

1. Προσπάθησε να δημιουργήσεις ένα πρόγραμμα που να αφορά σε ένα ερασιτεχνικό πρωτάθλημα Α κατηγορίας που συμμετέχουν 24 ομάδες. Επομένως, να γίνει πρόγραμμα που για κάθε μία ομάδα του πρωταθλήματος να εισάγει το όνομα, τον αριθμό των νικών, τον αριθμό των ισοπαλιών και τον αριθμό των ηττών που πραγματοποίησε και τα αποθηκεύει σε λίστες.
2. Να τροποποιήσεις το πρόγραμμα έτσι ώστε ο αριθμός των ομάδων να μην είναι σταθερός 24, αλλά να μπορεί ο χρήστης να τον προσδιορίζει τερματίζοντας την επανάληψη όταν δώσει ως όνομα ομάδας τη λέξη «Τέλος».
3. Να τροποποιήσεις το πρόγραμμα έτσι ώστε με δεδομένο ότι η ισοπαλία δίνει έναν βαθμό, η νίκη τρεις βαθμούς και η ήττα κανέναν, να υπολογίζει τους βαθμούς που συγκέντρωσε η κάθε ομάδα και να τους αποθηκεύει σε λίστα. Να εμφανίζει το όνομα της κάθε ομάδας και τους βαθμούς που συγκέντρωσε.
4. Να τροποποιήσεις το πρόγραμμα έτσι ώστε να εμφανίζει το όνομα της πρωταθλήτριας ομάδας.

Παρατήρηση: Θεωρείστε ότι δεν υπάρχουν ισοβαθμίες.

5. Αποθήκευσε το πρόγραμμα με όνομα myf15d3.py, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο f15d3.py είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Δραστηριότητα 4

1. Να υλοποιήσεις το πρόγραμμα της 4ης δραστηριότητας του φύλλου εργασίας 12 με τη χρήση λιστών έτσι ώστε το παιχνίδι να μπορεί παίζεται και με περισσότερους παίκτες που ο αριθμός τους θα δίνεται από τον χρήστη.
2. Αποθήκευσε το πρόγραμμα με όνομα `myf14d4.py`, εκτέλεσέ το και παρατήρησε τα αποτελέσματα.

Το πρόγραμμα στο αρχείο `f15d4.py` είναι το πρόγραμμα του παραπάνω ερωτήματος για να το συγκρίνεις με το δικό σου.

Φύλλο εργασίας 16 - Τεστ αυτοαξιολόγησης

Για κάθε μία από τις παρακάτω 10 προτάσεις να επιλέξεις λέξη Σωστό, αν είναι Σωστή, ή τη λέξη Λάθος, αν είναι λανθασμένη.

1. Η λίστα είναι μια δομή που μπορεί να περιέχει στοιχεία διαφορετικού τύπου δεδομένων.
Σωστό Λάθος
2. Το μέγεθος της λίστας καθορίζεται κατά την εκτέλεση του προγράμματος.
Σωστό Λάθος
3. Η προσπέλαση στα στοιχεία της λίστας γίνεται με το όνομα της λίστας και τον αύξοντα αριθμό του αντικείμενου μέσα σε αγκύλες.
Σωστό Λάθος
4. Η αρίθμηση των στοιχείων της λίστας ξεκινάει από το 1.
Σωστό Λάθος
5. Σε μία λίστα $L=[4,9,8,5,7]$, η εντολή **8 in L** επιστρέφει την τιμή False.
Σωστό Λάθος
6. Σε μία λίστα $L=[4,9,8,5,7]$, η εντολή **len (L)** επιστρέφει την τιμή 6.
Σωστό Λάθος
7. Με μία συμβολοσειρά `String = 'computer'` η εντολή **list (String)** επιστρέφει τη λίστα `['c', 'o', 'm', 'p', 'u', 't', 'e', 'r']`.
Σωστό Λάθος
8. Σε μία λίστα $L=[4,9,8,5,7]$ η εντολή **L.append(18)** επιστρέφει τη λίστα $L=[18,4,9,8,5,7]$.
Σωστό Λάθος
9. Σε μία λίστα $L=[4,9,8,5,7]$ η εντολή **L.insert(0,10)** επιστρέφει τη λίστα $L=[4,9,8,5,7,10]$.
Σωστό Λάθος
10. Σε μία λίστα $L=[4,9,8,5,7]$ η εντολή **L.pop(4)** επιστρέφει τη λίστα $L=[9,8,5,7]$.
Σωστό Λάθος

Κεφάλαιο 4: Αξιολόγηση διδακτικών σεναρίων

4.1 Στόχος της έρευνας

Στόχος της έρευνας αποτελεί η αξιολόγηση του σχεδιασμού των διδακτικών σεναρίων τα οποία βασίζονται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας ως προς την ενίσχυση των κινήτρων των εκπαιδευομένων με σκοπό την επίτευξη των μαθησιακών στόχων και συγκεκριμένα την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

4.2 Ερευνητικές μεταβλητές

Οι μεταβλητές που θα ερευνηθούν περιγράφονται παρακάτω (Prayaga, 2007):

Κίνητρα

Ο (Keller J. , 1987a) ανέπτυξε το μοντέλο ARCS για να διερευνήσει και να δομήσει τα κίνητρα των εκπαιδευομένων σε μια προσπάθεια να βελτιώσει τη μάθηση και την απόδοση. Προσδιόρισε τα τέσσερα στοιχεία του μοντέλου ARCS (προσοχή, συσχέτιση, εμπιστοσύνη και ικανοποίηση) ως προϋποθέσεις για την παραγωγή μιας ενδιαφέρουσας και αποτελεσματικής διδασκαλίας που ενισχύει τα κίνητρα των εκπαιδευομένων.

Το κίνητρο διαδραματίζει σημαντικό ρόλο στη μάθηση, προωθώντας την επιθυμία και την περιέργεια στον εκπαιδευόμενο να συμμετάσχει και να διερευνήσει το εκπαιδευτικό περιεχόμενο. Τα κλασικά παραδείγματα κινήτρων είναι η προσέγγιση του stick-and-carrot, η θεωρία του Maslow για την κλινική συμπεριφορά, η καθοδηγούμενη από την τεχνολογία διδασκαλία σε σχέση με την απλή διάλεξη και τα παιχνίδια για την παροχή διδασκαλίας. Σε μια ατμόσφαιρα παιχνιδιού, το κίνητρο για τον εκπαιδευόμενο προκύπτει επειδή το εκπαιδευτικό περιεχόμενο είναι διαμορφωμένο μέσα στο περιβάλλον του παιχνιδιού.

Προσοχή

Η προσοχή είναι η εφαρμογή του μυαλού σε ένα αντικείμενο νοήματος ή σκέψης. Ο (Keller J. , 1987a) πρότεινε ότι το πρώτο βήμα της διδασκαλίας είναι να προσελκύσει την προσοχή των εκπαιδευομένων. Για να κερδίσει την προσοχή του εκπαιδευόμενου, ο Keller πρότεινε ότι οι εκπαιδευτές πρέπει να δημιουργούν ένα περιβάλλον τόνωσης, χρησιμοποιώντας μια ποικιλία από μεθόδους για την παροχή διδασκαλίας, συμπεριλαμβανομένης της διάλεξης, hands-on έργων, ήχου, και εικόνες. Άλλοι σημαντικοί παράγοντες για την ανάκτηση της προσοχής είναι οι μέθοδοι για την αύξηση της αντίληψης, έρευνα και μεταβλητότητα (Keller). Η αντιληπτική διέγερση περιλαμβάνει στοιχεία έκπληξης, περιέργεια, αβεβαιότητα ή

πρόκληση. Η έρευνα περιλαμβάνει προκλήσεις και προβλήματα που διεγείρουν την περιέργεια. Η μεταβλητότητα σχετίζεται με τη χρήση διαφόρων μεθόδων για την παρουσίαση διδακτικού περιεχομένου και τη διατήρηση του ενδιαφέροντος μεταβάλλοντας τα στοιχεία της διδασκαλίας.

Συσχέτιση

Το δεύτερο στοιχείο στο μοντέλο ARCS του (Keller J. , 1987a) είναι η συσχέτιση. *Η διαδικασία της συσχέτισης περιλαμβάνει τη σύνδεση νέου υλικού με τις ανάγκες του χρήστη και την κατάσταση.* Ο Keller πρότεινε έξι στρατηγικές για τη δημιουργία ενός μαθησιακού περιβάλλοντος σχετικού με τον εκπαιδευόμενο που σχετίζεται με τον τρόπο με τον οποίο απεικονίζεται ο εκπαιδευτικός στόχος στον εκπαιδευόμενο, αντί να επηρεάζει άμεσα το περιεχόμενο. Ο Keller περιέγραψε τις στρατηγικές συσχέτισης με τις ακόλουθες κατηγορίες: (α) εμπειρία, (β) παρούσα αξία, (γ) μελλοντική χρησιμότητα, (δ) ανάγκη αντιστοίχισης, (ε) μοντελοποίηση και (στ) επιλογή. Η συσχέτιση είναι ένα ιδιαίτερα πολύτιμο εργαλείο, διότι αποδεικνύει την ανάγκη για νέες πληροφορίες, τη σημασία αυτών των νέων πληροφοριών και τη σχέση με παλιές ή υπάρχουσες πληροφορίες. Τα παιχνίδια παρουσιάζουν τα στοιχεία της εμπειρίας, της μοντελοποίησης, της επιλογής και της λήψης αποφάσεων και παρέχουν την απαραίτητη συσχέτιση, την οποία διατήρησε ο (Keller J. , 1987b).

Εμπιστοσύνη

Το τρίτο στοιχείο του μοντέλου ARCS είναι η εμπιστοσύνη. *Ο (Keller J. , 1987a) όρισε την εμπιστοσύνη ως την πεποίθηση στην επιτυχία.* Ο (Davies, 2000) πρότεινε ότι η εμπιστοσύνη είναι απαραίτητη στη μάθηση. Ένα μεγάλο εμπόδιο που αντιμετωπίζουν οι εκπαιδευόμενοι είναι η έλλειψη εμπιστοσύνης στο ταξίδι τους στη μάθηση.

Μια κατάσταση παιχνιδιού είναι ιδανική για να δημιουργηθεί μια πλατφόρμα για την ενίσχυση της εμπιστοσύνης. Παίζοντας με τον δικό τους ρυθμό, οι παίκτες εξοικειώνονται με τους κανόνες του παιχνιδιού, γεγονός που αυξάνει την αυτοεκτίμησή τους (Dempsey, Lucassen, Gilley, & Rasmussen, 1994). Η εκτέλεση του παιχνιδιού είναι ασφαλής επειδή δεν υπάρχει εξωτερική πίεση και ο παίκτης δημιουργεί και αναπτύσσει μια ζώνη άνεσης από την οποία μπορεί να πλοηγηθεί. Οι (Mitchell & Savill-Smith, 2004) παρατήρησαν ότι το σενάριο τυχερών παιχνιδιών προσφέρει επίσης άμεση ανατροφοδότηση, ένα βασικό στοιχείο μιας καλής εκπαιδευτικής στρατηγικής (Chickering & Ehrmann, 1996), μέσω στοιχείων παιχνιδιών όπως μια κάρτα σκορ και άλλα συναφή μηνύματα. Αυτό το χαρακτηριστικό μπορεί να χρησιμοποιηθεί με πλεονεκτήματα για τη δημιουργία εκπαιδευτικών υλικών μέσα

σε περιβάλλον παιχνιδιών για την παρακίνηση του μαθητή και την παροχή συμβουλών, προτάσεων και παραδειγμάτων για να κατευθύνει τον εκπαιδευόμενο προς τη σωστή κατεύθυνση προς τη μάθηση.

Ικανοποίηση

Το τελευταίο στοιχείο του μοντέλου ARCS είναι η ικανοποίηση. Ο (Keller J. , 1987a) υποστήριξε ότι η ικανοποίηση των εκπαιδευομένων αποτελεί το κλειδί της μάθησης. *Η ικανοποίηση των εκπαιδευομένων μπορεί να επιτευχθεί με την αίσθηση εκπλήρωσης των μαθησιακών αποτελεσμάτων, με καλή απόδοση σε ένα δεδομένο μάθημα ή με τη λήψη θετικών ανατροφοδοτήσεων ενώ βρίσκονται στη διαδικασία εκμάθησης* (Gunter, Kenny, & Vick, 2006). Ο Keller (όπως παραθέτει ο (Ragan & Smith, 1999)) πρότεινε τους ακόλουθους τρεις τρόπους για να ικανοποιήσει τον εκπαιδευόμενο:

- Μέσω φυσικών συνεπειών: Παρέχετε στους εκπαιδευόμενους ρεαλιστικές καταστάσεις να εφαρμόζουν τη νέα γνώση που έχουν μάθει.
- Μέσα από θετικές συνέπειες: Παρέχετε στους εκπαιδευόμενους ανατροφοδότηση όταν είναι απαραίτητο.
- Μέσω της δικαιοσύνης: Να κάνετε τις προσδοκίες σαφείς και να είστε δίκαιοι με τους εκπαιδευόμενους.

Το περιβάλλον παιχνιδιού φυσικά προσφέρεται για την παροχή ικανοποίησης στον εκπαιδευόμενο και με τους τρεις τρόπους. Τα παιχνίδια μπορούν να χρησιμοποιηθούν για να δημιουργήσουν ένα πλαίσιο που να (α) να εισαγάγει ρεαλιστικές καταστάσεις, (β) να παράσχει άμεση ανατροφοδότηση για να καθοδηγήσει τους εκπαιδευόμενους μέσω θετικής ενίσχυσης και συμβουλών, και (γ) να παράσχει ισότητα.

4.3 Δείγμα της έρευνας

Το δείγμα της έρευνας αποτέλεσαν 15 προπτυχιακοί φοιτητές του τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας, 6 φοιτητές του 1ου εξαμήνου, 7 φοιτητές του 3ου εξαμήνου και 2 φοιτητές του 7ου εξαμήνου. Συγκεκριμένα στο 1ο διδακτικό σενάριο συμμετείχαν 9 φοιτητές, στο 2ο διδακτικό σενάριο 6 φοιτητές, στο 3ο διδακτικό σενάριο 10 φοιτητές και στο 4ο διδακτικό σενάριο 5 φοιτητές.

4.4 Μέσα συλλογής δεδομένων

Η διεξαγωγή της έρευνας έγινε με την χρήση ενός αυτοσχέδιου ερωτηματολογίου αξιολόγησης του μαθήματος στο εργαστήριο το οποίο χρησιμοποιήθηκε για τη μέτρηση των

κινήτρων των εκπαιδευομένων αναφορικά με το μάθημα στο εργαστήριο και συμπληρώθηκε από τους εκπαιδευόμενους μετά την ολοκλήρωση του κάθε διδακτικού σεναρίου. Βασίστηκε στο εργαλείο CIS (Course Interest Survey) το οποίο σχεδιάστηκε από τον Keller το 1987 για τη μέτρηση υποκινητικών αντιδράσεων των εκπαιδευομένων σε ένα εκπαιδευτικό περιβάλλον καθοδηγούμενο από τον εκπαιδευτή (Πίνακας 74). Αποτελείται από 34 προτάσεις από τις οποίες οι 8 αναφέρονται στη μεταβλητή «προσοχή», οι 9 στη μεταβλητή «συσχέτιση», οι 8 στη μεταβλητή «εμπιστοσύνη» και οι 9 στη μεταβλητή «ικανοποίηση» του μοντέλου ARCS. Οι απαντήσεις βασίζονται στην ψυχομετρική 5βάθμια κλίμακα Likert που είναι η εξής: 1=Διαφωνώ Απόλυτα, 2=Διαφωνώ, 3=Αδιαφορώ, 4=Συμφωνώ, 5=Συμφωνώ Απόλυτα. Το ερωτηματολόγιο βρίσκεται στο παράρτημα Β.

4.5 Μέθοδος ανάλυσης δεδομένων

Η έρευνα βαθμολογήθηκε για κάθε μία από τις τέσσερις επιμέρους κλίμακες ανά δύο και για τη βαθμολογία της συνολικής κλίμακας (Πίνακας 33). Η κλίμακα απόκρισης κυμαίνεται από 1 έως 5. Μια μέθοδος βαθμολόγησης είναι ο υπολογισμός του μέσου σκορ για κάθε υποκλίμακα και της συνολικής κλίμακας. Για κάθε ερωτώμενο, διαιρείται η συνολική βαθμολογία σε μια δεδομένη κλίμακα από τον αριθμό των στοιχείων της κλίμακας αυτής. Οι αποδόσεις καθορίζονται με άθροιση των απαντήσεων για κάθε υποκλίμακα και τη συνολική κλίμακα. Τα αντικείμενα που σημειώνονται αντιστρόφως (Πίνακας 33) αναφέρονται αρνητικά και οι απαντήσεις πρέπει να αντιστραφούν πριν μπορέσουν να προστεθούν στο σύνολο των απαντήσεων. Δηλαδή, για αυτά τα στοιχεία, 5 = 1, 4 = 2, 3 = 3, 2 = 4, και 1 = 5.

Πίνακας 33: CIS οδηγός σκορ

Attention	Relevance	Confidence	Satisfaction
1	2	3	7 (reverse)
4 (reverse)	5	6 (reverse)	12
10	8 (reverse)	9	14
15	13	11 (reverse)	16
21	20	17 (reverse)	18
24	22	27	19
26 (reverse)	23	30	31 (reverse)
29	25 (reverse)	34	32
	28		33

4.6 Ερευνητικά ερωτήματα

Τα ερευνητικά ερωτήματα είναι τα παρακάτω:

1ο ερευνητικό ερώτημα

Ο σχεδιασμός των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας μπορεί να ενισχύσει τα **κίνητρα** των εκπαιδευομένων με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python;

2ο ερευνητικό ερώτημα

Υπάρχει στατιστικά σημαντική επίδραση στην **προσοχή** και την **συσχέτιση** των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python;

3ο ερευνητικό ερώτημα

Υπάρχει στατιστικά σημαντική επίδραση στην **εμπιστοσύνη** και στην **ικανοποίηση** των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python;

4.7 Ερευνητικές υποθέσεις

Οι ερευνητικές υποθέσεις είναι:

Υποθέσεις 1ου ερευνητικού ερωτήματος

H₀: Ο σχεδιασμός των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας **δεν μπορεί** να ενισχύσει τα **κίνητρα** των εκπαιδευομένων με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

H₁: Ο σχεδιασμός των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας **μπορεί** να ενισχύσει τα **κίνητρα** των εκπαιδευομένων με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

Υποθέσεις 2ου ερευνητικού ερωτήματος

H₀: Δεν υπάρχει στατιστικά σημαντική επίδραση στην **προσοχή** και στην **συσχέτιση** των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

H₁: Υπάρχει στατιστικά σημαντική επίδραση στην **προσοχή** και στην **συσχέτιση** των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

Υποθέσεις 3ου ερευνητικού ερωτήματος

H₀: Δεν υπάρχει στατιστικά σημαντική επίδραση στην **εμπιστοσύνη** και στην **ικανοποίηση** των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

H₁: Υπάρχει στατιστικά σημαντική επίδραση στην **εμπιστοσύνη** και στην **ικανοποίηση** των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

4.8 Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης

Προκειμένου να αναλυθούν τα δεδομένα της έρευνας χρησιμοποιήθηκε το στατιστικό πακέτο ανάλυσης δεδομένων SPSS 16.0 . Πιο συγκεκριμένα χρησιμοποιήθηκε (Τριανταφύλλου, 2017):

- ο συντελεστής εσωτερικής συνέπειας Cronbach's alpha για να μετρήσει την αξιοπιστία του ερωτηματολογίου και να δείξει κατά πόσο διαφορετικές προτάσεις (items) μετρούν την ίδια έννοια (μεταβλητή) δηλαδή την ομοιογένεια μιας κλίμακας. Για να θεωρείται αποδεκτή η τιμή του Cronbach's alpha θα πρέπει να είναι >0.7 . Όσο μεγαλύτερη είναι η

τιμή του συντελεστή Cronbach's alpha τόσο μεγαλύτερη είναι η αξιοπιστία εσωτερικής συνοχής.

- ο έλεγχος κανονικότητας των δεδομένων του δείγματος Shapiro-Wilk για να εξετάσει αν οι διαθέσιμες μετρήσεις προσαρμόζονται ικανοποιητικά στην κανονική κατανομή. Οι δύο υποθέσεις που έρχονται σε αντιπαράθεση είναι:

H₀: Τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

H₁: Τα δεδομένα δεν προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

- ο έλεγχος One-Sample T-Test για τη μέση τιμή του δείγματος ως προς μια προκαθορισμένη τιμή μ₀ που ορίζεται από τον ερευνητή και στη συγκεκριμένη έρευνα ορίζεται η τιμή 3.

4.8.1 Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης 1ου Διδακτικού Σεναρίου 1ο ερευνητικό ερώτημα

Πίνακας 34: Υπολογισμός αξιοπιστίας της μεταβλητής Motivation με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,895	34

Όπως φαίνεται στον πίνακα 34 ο συντελεστής Cronbach's alpha είναι $0.895 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 35: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Motivation

	Tests of Normality					
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Motivation	,166	9	,200*	,947	9	,660

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Όπως φαίνεται στον πίνακα 35 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο με $0,660 > 0,05$ και αποδεχόμαστε την H₀. Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 36: One-Sample T-Test για τη μεταβλητή Motivation

One-Sample Test						
Test Value = 3						
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean Difference	Lower	Upper
Motivation	6,930	8	,000	1,06209	,7087	1,4155

Όπως φαίνεται στον πίνακα 36 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,000 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι ο σχεδιασμός των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας μπορεί να ενισχύσει τα κίνητρα των εκπαιδευομένων με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

2ο ερευνητικό ερώτημα

Πίνακας 37: Υπολογισμός αξιοπιστίας της μεταβλητής Attention-Relevance με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,884	17

Όπως φαίνεται στον πίνακα 37 ο συντελεστής Cronbach's alpha είναι $0.884 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 38: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Attention-Relevance

Tests of Normality						
Kolmogorov-Smirnov ^a			Shapiro-Wilk			
	Statistic	df	Sig.	Statistic	df	Sig.
Attention_Relevance	,282	9	,038	,907	9	,297

a. Lilliefors Significance Correction

Όπως φαίνεται στον πίνακα 38 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,297 > 0,05$ και αποδεχόμαστε την H_0 . Με

πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 39: One-Sample T-Test για τη μεταβλητή Attention-Relevance

One-Sample Test						
Test Value = 3						
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean Difference	Lower	Upper
Attention_Relevance	7,382	8	,000	1,20915	,8314	1,5869

Όπως φαίνεται στον πίνακα 39 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,000 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι υπάρχει στατιστικά σημαντική επίδραση στην προσοχή και στην συσχέτιση των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

3ο ερευνητικό ερώτημα

Πίνακας 40: Υπολογισμός αξιοπιστίας της μεταβλητής Confidence-Satisfaction με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,752	17

Όπως φαίνεται στον πίνακα 40 ο συντελεστής Cronbach's alpha είναι $0.752 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 41: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Confidence-Satisfaction

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Confidence_Satisfaction	,241	9	,141	,910	9	,318

a. Lilliefors Significance Correction

Όπως φαίνεται στον πίνακα 41 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,318 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 42 : One-Sample T-Test για τη μεταβλητή Confidence-Satisfaction

One-Sample Test						
Test Value = 3						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Confidence_Satisfaction	5,607	8	,001	,91503	,5387	1,2914

Όπως φαίνεται στον πίνακα 42 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,001 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι υπάρχει στατιστικά σημαντική επίδραση στην εμπιστοσύνη και στην ικανοποίηση των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

Πίνακας 43:Υπολογισμός των μέσων όρων των μεταβλητών

One-Sample Statistics				
	N	Mean	Std. Deviation	Std. Error Mean
Motivation	9	4,0621	,45977	,15326
Attention_Relevance	9	4,2092	,49137	,16379
Confidence_Satisfaction	9	3,9150	,48961	,16320

Όπως φαίνεται στον πίνακα 43 στο 1ο διδακτικό σενάριο ο μέσος όρος της μεταβλητής Motivation ισούται με 4,0621, ο μέσος όρος της μεταβλητής Attention_Relevance ισούται με 4,2092 και ο μέσος όρος της μεταβλητής Confidence_Satisfaction ισούται με 3,9150.

4.8.2 Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης 2ου Διδακτικού Σεναρίου 1ο ερευνητικό ερώτημα

Πίνακας 44: Υπολογισμός αξιοπιστίας της μεταβλητής Motivation με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,888	34

Όπως φαίνεται στον πίνακα 44 ο συντελεστής Cronbach's alpha είναι $0.888 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 45: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Motivation

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Motivation	,179	6	,200*	,960	6	,819

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Όπως φαίνεται στον πίνακα 45 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,819 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 46: One-Sample T-Test για τη μεταβλητή Motivation

One-Sample Test						
Test Value = 3						
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean Difference	Lower	Upper
Motivation	10,107	5	,000	1,25980	,9394	1,5802

Όπως φαίνεται στον πίνακα 46 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,000 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι ο σχεδιασμός των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας μπορεί να ενισχύσει τα κίνητρα των εκπαιδευομένων με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

2ο ερευνητικό ερώτημα

Πίνακας 47: Υπολογισμός αξιοπιστίας της μεταβλητής Attention - Relevance με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,846	17

Όπως φαίνεται στον πίνακα 47 ο συντελεστής Cronbach's alpha είναι $0.846 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 48: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Attention-Relevance

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Attention_Relevance	,276	6	,170	,880	6	,268

a. Lilliefors Significance Correction

Όπως φαίνεται στον πίνακα 48 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,268 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 49: One-Sample T-Test για τη μεταβλητή Attention_Relevance

One-Sample Test						
Test Value = 3						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Attention_Relevance	8,889	5	,000	1,26471	,8990	1,6304

Όπως φαίνεται στον πίνακα 49 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,000 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι υπάρχει στατιστικά σημαντική επίδραση στην προσοχή και στην συσχέτιση

των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

3ο ερευνητικό ερώτημα

Πίνακας 50: Υπολογισμός αξιοπιστίας της μεταβλητής Confidence-Satisfaction με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,762	17

Όπως φαίνεται στον πίνακα 50 ο συντελεστής Cronbach's alpha είναι $0.762 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 51: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Confidence-Satisfaction

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Confidence_Satisfaction	,232	6	,200*	,930	6	,579

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Όπως φαίνεται στον πίνακα 51 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,579 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 52: One-Sample T-Test για τη μεταβλητή Confidence_Satisfaction

One-Sample Test						
Test Value = 3						
					95% Confidence Interval of the Difference	
t	df	Sig. (2-tailed)	Mean Difference	Lower	Upper	

One-Sample Test

Test Value = 3						
					95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean Difference	Lower	Upper
Confidence_Satisfaction	10,044	5	,000	1,25490	,9337	1,5761

Όπως φαίνεται στον πίνακα 52 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,000 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι υπάρχει στατιστικά σημαντική επίδραση στην εμπιστοσύνη και στην ικανοποίηση των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

Πίνακας 53: Υπολογισμός των μέσων όρων των μεταβλητών

One-Sample Statistics				
	N	Mean	Std. Deviation	Std. Error Mean
Motivation	6	4,2598	,30533	,12465
Attention_Relevance	6	4,2647	,34850	,14228
Confidence_Satisfaction	6	4,2549	,30603	,12494

Όπως φαίνεται στον πίνακα 53 στο 2ο διδακτικό σενάριο ο μέσος όρος της μεταβλητής Motivation ισούται με 4,2598, ο μέσος όρος της μεταβλητής Attention_Relevance ισούται με 4,2647 και ο μέσος όρος της μεταβλητής Confidence_Satisfaction ισούται με 4,2549.

4.8.3 Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης 3ου Διδακτικού Σεναρίου 1ο ερευνητικό ερώτημα

Πίνακας 54: Υπολογισμός αξιοπιστίας της μεταβλητής Motivation με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,922	34

Όπως φαίνεται στον πίνακα 54 ο συντελεστής Cronbach's alpha είναι $0.922 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 55: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Motivation

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Motivation	,139	10	,200*	,968	10	,867

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Όπως φαίνεται στον πίνακα 55 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,867 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 56: One-Sample T-Test για τη μεταβλητή Motivation

One-Sample Test						
Test Value = 3						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Motivation	7,821	9	,000	1,00294	,7128	1,2930

Όπως φαίνεται στον πίνακα 56 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,000 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι ο σχεδιασμός των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας μπορεί να ενισχύσει τα κίνητρα των εκπαιδευομένων με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

2ο ερευνητικό ερώτημα

Πίνακας 57: Υπολογισμός αξιοπιστίας της μεταβλητής Attention-Relevance με τον δείκτη Cronbach's alpha

Reliability Statistics

Cronbach's Alpha	N of Items
,834	17

Όπως φαίνεται στον πίνακα 57 ο συντελεστής Cronbach's alpha είναι $0.834 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 58: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Attention_Relevance

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Attention_Relevance	,108	10	,200*	,983	10	,979

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Όπως φαίνεται στον πίνακα 58 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,979 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 59: One-Sample T-Test για τη μεταβλητή Attention_Relevance

One-Sample Test						
Test Value = 3						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Attention_Relevance	8,118	9	,000	1,08235	,7807	1,3840

Όπως φαίνεται στον πίνακα 59 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,000 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι υπάρχει στατιστικά σημαντική επίδραση στην προσοχή και στην συσχέτιση των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

3ο ερευνητικό ερώτημα

Πίνακας 60: Υπολογισμός αξιοπιστίας της μεταβλητής Confidence-Satisfaction με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,886	17

Όπως φαίνεται στον πίνακα 60 ο συντελεστής Cronbach's alpha είναι $0.886 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 61: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Confidence_Satisfaction

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Confidence_Satisfaction	,208	10	,200*	,908	10	,270

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Όπως φαίνεται στον πίνακα 61 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,270 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 62: One-Sample T-Test για τη μεταβλητή Confidence_Satisfaction

One-Sample Test						
Test Value = 3						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Confidence_Satisfaction	6,783	9	,000	,92353	,6156	1,2315

Όπως φαίνεται στον πίνακα 62 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,000 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι υπάρχει στατιστικά σημαντική επίδραση στην εμπιστοσύνη και στην ικανοποίηση των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

Πίνακας 63: Υπολογισμός των μέσων όρων των μεταβλητών

One-Sample Statistics				
	N	Mean	Std. Deviation	Std. Error Mean
Motivation	10	4,0029	,40552	,12824
Attention_Relevance	10	4,0824	,42164	,13333
Confidence_Satisfaction	10	3,9235	,43053	,13614

Όπως φαίνεται στον πίνακα 63 στο 3ο διδακτικό σενάριο ο μέσος όρος της μεταβλητής Motivation ισούται με 4,0029, ο μέσος όρος της μεταβλητής Attention ισούται με 4,0824 και ο μέσος όρος της μεταβλητής Confidence_Satisfaction ισούται με 3,9235.

4.8.4 Στατιστική ανάλυση αποτελεσμάτων της αξιολόγησης 4ου Διδακτικού Σεναρίου 1ο ερευνητικό ερώτημα

Πίνακας 64: Υπολογισμός αξιοπιστίας της μεταβλητής Motivation με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,908	34

Όπως φαίνεται στον πίνακα 64 ο συντελεστής Cronbach's alpha είναι $0.908 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 65: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Motivation

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Motivation	,248	5	,200*	,893	5	,375

Tests of Normality

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Motivation	,248	5	,200*	,893	5	,375

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Όπως φαίνεται στον πίνακα 65 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,375 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 66: One-Sample T-Test για τη μεταβλητή Motivation

One-Sample Test						
Test Value = 3						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Motivation	7,151	4	,002	1,23529	,7556	1,7149

Όπως φαίνεται στον πίνακα 66 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,002 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι ο σχεδιασμός των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδοποίηση και τη χρήση αναδυόμενης τεχνολογίας μπορεί να ενισχύσει τα κίνητρα των εκπαιδευομένων με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

2ο ερευνητικό ερώτημα

Πίνακας 67: Υπολογισμός αξιοπιστίας της μεταβλητής Attention-Relevance με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,805	17

Όπως φαίνεται στον πίνακα 67 ο συντελεστής Cronbach's alpha είναι $0.805 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 68: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Intention_Relevance

	Tests of Normality					
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Attention_Relevance	,180	5	,200*	,956	5	,779

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Όπως φαίνεται στον πίνακα 68 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,779 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 69: One-Sample T-Test για τη μεταβλητή Attention_Relevance

	One-Sample Test					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Attention_Relevance	7,889	4	,001	1,35294	,8768	1,8291

Όπως φαίνεται στον πίνακα 69 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,001 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι υπάρχει στατιστικά σημαντική επίδραση στην προσοχή και στην συσχέτιση των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

3ο ερευνητικό ερώτημα

Πίνακας 70: Υπολογισμός αξιοπιστίας της μεταβλητής Confidence-Satisfaction με τον δείκτη Cronbach's alpha

Reliability Statistics	
Cronbach's Alpha	N of Items
,824	17

Όπως φαίνεται στον πίνακα 70 ο συντελεστής Cronbach's alpha είναι $0.824 > 0.7$ που δείχνει την αξιοπιστία της εσωτερικής συνοχής της κλίμακας.

Πίνακας 71: Υπολογισμός κανονικότητας δείγματος της μεταβλητής Motivation

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Confidence_Satisfaction	,323	5	,095	,802	5	,084

a. Lilliefors Significance Correction

Όπως φαίνεται στον πίνακα 71 το p-value για τον έλεγχο της κανονικότητας των δεδομένων με χρήση του ελέγχου Shapiro-Wilk είναι ίσο $0,084 > 0,05$ και αποδεχόμαστε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% τα δεδομένα προσαρμόζονται ικανοποιητικά στην κανονική κατανομή.

Πίνακας 72: One-Sample T-Test για τη μεταβλητή Confidence_Satisfaction

One-Sample Test						
Test Value = 3						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Confidence_Satisfaction	6,298	4	,003	1,11765	,6250	1,6103

Όπως φαίνεται στον πίνακα 72 το p-value του στατιστικού ελέγχου One-Sample T-test είναι $0,003 < 0,05$. Συνεπώς, απορρίπτουμε την H_0 . Με πιθανότητα λανθασμένης απόρριψης 5% εκτιμάται ότι υπάρχει στατιστικά σημαντική επίδραση στην εμπιστοσύνη και στην ικανοποίηση των εκπαιδευομένων μέσα από τον σχεδιασμό των διδακτικών σεναρίων που βασίζεται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης

τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python.

Πίνακας 73:Υπολογισμός των μέσων όρων των μεταβλητών

One-Sample Statistics				
	N	Mean	Std. Deviation	Std. Error Mean
Motivation	5	4,2353	,38629	,17276
Attention_Relevance	5	4,3529	,38348	,17150
Confidence_Satisfaction	5	4,1176	,39679	,17745

Όπως φαίνεται στον πίνακα 73 στο 4ο διδακτικό σενάριο ο μέσος όρος της μεταβλητής Motivation ισούται με 4,2353, ο μέσος όρος της μεταβλητής Attention_Relevance ισούται με 43529 και ο μέσος όρος της μεταβλητής Confidence_Satisfaction ισούται με 4,1176.

Κεφάλαιο 5: Συμπεράσματα

Στην παρούσα έρευνα έγινε μια προσπάθεια να μελετηθούν όλα μαζί και ανά δύο τα τέσσερα στοιχεία του μοντέλου ARCS μέσα από τον σχεδιασμό των τεσσάρων διδακτικών σεναρίων που βασίζονται στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας έτσι ώστε οι εκπαιδευόμενοι να αναπτύξουν κίνητρα για την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python. Μετά την συλλογή, επεξεργασία και ανάλυση των δεδομένων της έρευνας προκύπτουν τα εξής συμπεράσματα:

- ✓ Ο σχεδιασμός των διδακτικών σεναρίων που βασίστηκε στη βιωματική μάθηση, την παιχνιδιοποίηση και τη χρήση αναδυόμενης τεχνολογίας με σκοπό την οικοδόμηση των βασικών γνώσεων του προγραμματισμού στη γλώσσα προγραμματισμού Python είχε σημαντική επίδραση στα κίνητρα, στην προσοχή και συσχέτιση, στην εμπιστοσύνη και ικανοποίηση των εκπαιδευομένων.
- ✓ Η νέα αυτή προσέγγιση προσέλκυσε την προσοχή των εκπαιδευομένων μέσα από τη χρησιμοποίηση διαφόρων μεθόδων κατά τη διάρκεια της διδασκαλίας, όπως βιντεοπροβολέας, φωτογραφίες, παραδείγματα της καθημερινής ζωής κτλ. Το παιχνίδι των χρωματιστών καρτών που παίχτηκε μέσα στην τάξη και κυρίως η σταδιακή εξέλιξή του διέγειρε την περιέργειά τους και αποτέλεσε πρόκληση για τους εκπαιδευόμενους να επιτύχουν τον προγραμματισμό του.
- ✓ Η συσχέτιση του κλασσικού παιχνιδιού των χρωματιστών καρτών με την αλληλεπίδραση των εκπαιδευομένων με την πλακέτα Pyboard μέσω των πιεζόμενων κουμπιών σε συνδυασμό με τη σύνδεση του χρωματισμού του RGB Led με τον χρωματισμό της κάρτας κέντริσε το ενδιαφέρον τους και απέδειξε την ανάγκη εκμάθησης του προγραμματισμού.
- ✓ Οι εκπαιδευόμενοι έπαιξαν πολλές φορές το παιχνίδι ανά δύο άτομα μέχρι να το κατανοήσουν πλήρως και να εξοικειωθούν με αυτό αυξάνοντας την αυτοεκτίμησή τους. Το συγκεκριμένο παιχνίδι ήταν ένα παιχνίδι βασισμένο στον παράγοντα της τύχης το οποίο προσέφερε άμεση ανατροφοδότηση και παρακίνηση του εκπαιδευόμενου για τον επιτυχή προγραμματισμό του.
- ✓ Η ικανοποίηση των εκπαιδευομένων ήταν εμφανής για τους εκπαιδευόμενους που μπόρεσαν να ανταποκριθούν στις δραστηριότητες των φύλλων εργασιών και στο τέλος κατάφεραν να υλοποιήσουν τον κώδικα και να επαληθεύσουν άμεσα τα αποτελέσματά του.

Βιβλιογραφία

Ξενόγλωσση

Association, E. (2014). *Essential Facts about the Computer and Video Game*. Ανάκτηση 10 14, 2018, από theesa.com

Beeland, W. (2002). *Student Engagement, Visual Learning and Technology: Can Interactive Whiteboards Help?*

Boekaerts, M. (2002). *Motivation to learn*. Ανάκτηση 10 14, 2018, από http://www.ibe.unesco.org/fileadmin/user_upload/archive/Publications/educationalpracticesseriespdf/prac10e.pdf

Brown, S., & Vaughan, C. (2009). *Play: How it Shapes the Brain, Opens the Imagination, and Invigorates the Soul*. Penguin.

Camarero, C., Rodríguez, J., & San José, R. (2010). A comparison of the learning effectiveness of live cases and classroom projects. *International Journal of Management Education*(8), 83-94.

Carroll, J., & Thomas, J. (1988). *J.M. FUN*. ACM SIGCHI.

Cass, S. (2018, July 31). *The 2018 Top Programming Languages*. Ανάκτηση 10 14, 2018, από <https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>

Chickering, A., & Ehrmann, S. (1996). *Implementing the seven principles: Technology as lever*. *AAHE Bulletin*.

Cisco. (2008). *Multimodal Learning Through Media: What the Research Says*. Ανάκτηση 10 14, 2018, από https://www.cisco.com/c/dam/en_us/solutions/industries/docs/education/Multimodal-Learning-Through-Media.pdf

Damien, G., & Sokolovsky, P. (2018, 6 10). *MicroPython Documentation*. Ανάκτηση 10 14, 2018, από <https://media.readthedocs.org/pdf/micropython-pfalcon/stable/micropython-pfalcon.pdf>

Davies, T. (2000). *Confidence! Its role in the creative teaching and learning of design and technology [Electronic version]*. *Journal of Technology Education*.

Dempsey, J., Lucassen, B., Gilley, W., & Rasmussen, K. (1994). *Since Malone's theory of intrinsically motivating instruction: What's the score in the gaming literature?* Journal of Educational Technology Systems.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, (σσ. 9–15.).

Du Boulay. (1989). *Some Difficulties Of Learning To Program*, In E. Soloway, J. Sprohrer (Eds.) *Studying The Novice Programmer*. Lawrence Erlbaum Associates.

Gee, J. (2008). Learning and games. *Cambridge, MA: MIT Press*.

Gentry, J. (1990). *What is experiential learning". Guide to business gaming and experiential learning*.

Gioia, D., & Brass, D. (1985). Teaching the TV generation: The case for observational learning. (10), 11-18.

Gunter, G., Kenny, R., & Vick, E. (2006, 4). *A case for a formal design paradigm for serious games. Paper presented at IDMC+IMS Conference. Oxford, OH. 73.* Ανάκτηση 4 15, 2006, από www.units.muohio.edu/codeconference/

Guo, P. (2008, July 2). Ανάκτηση 10 14, 2018, από Why Python is a great language for teaching beginners in introductory programming classes: <http://pgbovine.net/python-teaching.htm>

Guo, P. (2013). *Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE)*, (σσ. 579-584). Colorado, USA.

Hawtrej, K. (2007). Using experiential learning techniques. *The Journal of Economic Education*(38), 143-152.

Hoc. (1989). *Do We Really Have Conditionals In Our Brains?*, In E. Soloway, J. Sprohrer (Eds.) *Studying The Novice Programmer*. Lawrence Erlbaum Associates.

Iosup, A., & Epema, D. (2014). *An experience report on using gamification in technical higher education*.

Kapp, K. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons.

- Kark, R. (2011).** Games managers play: Play as a form of leadership development. (10), 507-527.
- Keller, J. (1987a).** *Development and use of the ARCS model of motivational design.* Journal of Instructional Development.
- Keller, J. (1987b).** *Strategies for stimulating the motivation to learn.* Performance and Instruction.
- Kessler, & Anderson. (1989).** *Learning Flow of Control: Recursive and Iterative Procedures, In E. Soloway, J. Sprohrer (Eds.) Studying The Novice Programmer.* Lawrence Erlbaum Associates.
- Kolb, D. (2014).** *Experiential learning: Experience as the source of learning and development.* FT Press.
- Kumar, B., & Khurana, P. (2012).** Gamification in education – Learn computer programming with fun. *International Journal of Computers and Distributed Systems.*
- Lee, J., & Hammer, J. (2011).** *Gamification in Education: What, How, Why Bother? .* Academic Exchange Quarterly.
- Lee, K. (2017).** *Foundations of Programming Languages.* Springer International Publishing.
- Lutz, M. (2013).** *Learning Python.* O'Reilly Media.
- Maia Filev Rodrigo , & Graeml Reis Felipe . (2015).** Playing and learning with gamification: an in-class concurrent and distributed programming activity. *IEEE.*
- McDonald, M., Musson, R., & Smith, R. (2008).** Using Productivity Games to Prevent Defects. In *The Practical Guide to Defect Prevention. Microsoft Press, 79-95.*
- Mészárosová, E. (2015).** Is python an appropriate programming language for teaching programming in secondary schools? *ICTE Journal(4), 5-14.*
- Mitchell, A., & Savill-Smith, C. (2004).** *Why use computer games for learning? Ανάκτηση* 5 15, 2007, από <http://www.masternewmedia.org/news/>
- Morrison, B., & DiSalvo, B. (2014).** *Khan academy gamifies computer science.*
- Pane, & Myers. (1996).** *Usability Issues in the Design of Novice Programming Systems.* Technical Report CMU-CS-96-132, CarnegieMellon University.

Prayaga, L. (2007). *Game Development environment to teach*. Ανάκτηση 12 3, 2018, από http://etd.fcla.edu/WF/WFE0000089/Prayaga_Lakshmi_200708_PhD.pdf

Proserpio, L., & Gioia, D. (2007). Teaching the Virtual Generation. (6), 69-80.

Putano, B. (2017, December 18). *Most Popular and Influential Programming Languages of 2018*. Ανάκτηση 10 14, 2018, από <https://stackify.com/popular-programming-languages-2018/>

Putman, R., Sleeman, D., Baxter, J., & Kuspa, L. (1989). *A Summary Of Misconceptions Of High-School BASIC Programmers*. In E. Soloway, J. Sprohrer (Eds.) *Studying The Novice Programmer*. Lawrence Erlbaum Associates.

Ragan, T., & Smith, P. (1999). *Instructional design (2nd ed.)*. Upper Saddle River, NJ: Prentice Hall.

Rogalski, & Samurcay. (1990). Acquisition of Programming Knowledge and Skills. In J. Hoc, T. Green, R. Samurcay & D. Gilmore (Eds.), *Psychology of Programming*. Academic Press, 157-174.

Samurcay. (1989). *The Concept of Variable in Programming: its Meaning and Use in Problem-Solving by Novice Programmers*, In E. Soloway, J. Sprohrer (Eds.) *Studying The Novice Programmer*. Lawrence Erlbaum Associates.

Sleeman, D., Putman, R., Baxter, J., & Kuspa, L. (1988). *An Introductory Pascal Class: A Case Study Of Students' Errors*, In R. Mayer (Ed.), *Teaching and Learning Computer Programming*. Lawrence Erlbaum Associates.

Sprint, G., & Cook, D. (2015). Enhancing the CS1 Student Experience with Gamification. *5th IEEE Integrated STEM Conference*.

Swaroop, C. (2008). *A Byte Of Python (el)*. Ανάκτηση 10 14, 2018, από http://dide.flo.sch.gr/Plinet/Meetings/Meeting23/A_Byte_of_Python-el.pdf

Tollervey, N. (2015). *Python in Education Teach, Learn, Program*. O'Reilly Media.

Zhang, Y. (2015). *An Introduction to Python and Computer Programming*. Springer Science+Business Media Singapore.

Ελληνόγλωσση

Αγγελιδάκης, Ν. (2015, Αύγουστος). *Εισαγωγή στον προγραμματισμό με την Python*.

Ανάκτηση 10 14, 2018, από <http://aggelid.mysch.gr/pythonbook/>

Αμανατίδης, Μ. (2014). Διδακτικό σενάριο με θέμα: Δημιουργία λίστας σε προγραμματιστικό περιβάλλον scratch. *7ο Πανελλήνιο Συνεδρίο Διδακτική της Πληροφορικής*. Πανεπιστήμιο Κρήτης, Ρέθυμνο.

Αράπογλου, Α., Βραχνός, Ε., Κανίδης, Ε., Λέκκα, Δ., Μακρυγιάννης, Π., Μπελεσιώτης, Β., και συν. (2016). *Προγραμματισμός Υπολογιστών, Γ' ΕΠΑ.Α., Τομέας Πληροφορικής*. Αθήνα: ΙΤΥΕ - Εκδόσεις «Διόφαντος».

Αράπογλου, Α. (2016). Η έννοια της μεταβλητής στο προγραμματισμό. Ιδιαίτερα χαρακτηριστικά της έννοιας στη γλώσσα προγραμματισμού Python και διαφοροποιήσεις ως προς τη διδακτική προσέγγιση της. *8ο Πανελλήνιο Συνεδρίου «Διδακτική της Πληροφορικής»*. Ιωάννινα.

Αράπογλου, Α., Βραχνός, Ε., Κανίδης, Ε., Μακρυγιάννης, Π., Μπελεσιώτης, Β., & Τζήμας, Δ. (2015). *Οδηγίες για τον εκπαιδευτικό για τα μαθήματα «Αρχές Προγραμματισμού Υπολογιστών» & «Προγραμματισμός Υπολογιστών» των ΕΠΑ.Α.* Ανάκτηση 10 14, 2018, από <http://www.iep.edu.gr/el/component/k2/19-tomeas-pliroforikis>

Βακάλη, Α., Γιαννόπουλος, Η., Ιωαννίδης, Ν., Κοίλιας, Χ., Μάλαμας, Κ., Μανωλόπουλος, Ι., και συν. (2017). *Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον, Γ' τάξη Γενικού Λυκείου, Προσανατολισμός Οικονομίας και Πληροφορικής*. Αθήνα: ΙΤΥΕ - Εκδόσεις «Διόφαντος».

Δαγδιλέλης, Β. (1996). *Διδακτική της πληροφορικής. Η διδασκαλία του προγραμματισμού: αντιλήψεις των σπουδαστών για την κατασκευή και επικύρωση προγραμμάτων και διδακτικές καταστάσεις για τη διαμόρφωσή τους*. Διδακτορική διατριβή, Τμήμα Εφ. Πληροφορικής Πανεπιστήμιο Μακεδονίας.

Δημητριάδης, Σ. (2016). Η Python και η «τέχνη του υπολογίζεин»: πρόταση για ένα λεξικό μοντέλων διδασκαλίας της γλώσσας. *8ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής» - ΕΤΠΕ*. Ιωάννινα.

Διόφαντος, Ι. (2013, Απρίλιος). *Επιμόρφωση εκπαιδευτικών για την Αξιοποίηση και Εφαρμογή των ΤΠΕ στη διδακτική πράξη*. Ανάκτηση 10 14, 2018, από https://blogs.sch.gr/kse799/files/2013/09/ylikoKSE_PE1920_6B_201304.pdf

ΙΤΥΕ, Δ. (2013, Μάρτιος). *Επιμόρφωση εκπαιδευτικών για την Αξιοποίηση και Εφαρμογή των ΤΠΕ στη διδακτική πράξη.* Ανάκτηση 10 14, 2018, από

https://blogs.sch.gr/kse799/files/2013/05/ylikoKSE_Geniko_Meros_C_201304.pdf

Καρκαμάνης, Γ. (2015). *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον.*

Θεσσαλονίκη: Μαθηματική Βιβλιοθήκη.

Κόμης, Β. (2001). *Διδακτική της Πληροφορικής* (Τόμ. Α'). Πάτρα: Ομάδα εκτέλεσης έργου ΕΑΠ.

Κόμης, Β. (2015). *Διδακτική της Πληροφορικής: ερευνητικές προσεγγίσεις στη μάθηση και στη διδασκαλία.* Ανάκτηση 12 2018, από ΤΕΕΑΠΗ Πανεπιστήμιο Πατρών:

<https://eclass.upatras.gr/modules/document/file.php/PN1425/%CE%95%CF%81%CE%B3%CE%B1%CF%83%CE%AF%CE%B5%CF%82/ProjectInstructionsOpenCourses2.pdf>

Κοτίνη, Ι., & Τζαλέπη, Σ. (2013, 5). *Εκπαιδευτικό σενάριο «Εισαγωγή στη δομή επιλογής στο προγραμματιστικό περιβάλλον SCRATCH».* Ανάκτηση 10 14, 2018, από

<https://blogs.sch.gr/kse799/>

Λαζαρίνης, Φ. (2015). *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον.* Αθήνα: Ελληνοεκδοτική.

Ντζιός, Κ., & Κοψίνης, Ι. (2008). *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον.* Αθήνα: Σαββάλας.

Ξυνόγαλος, Σ. (2005). *Η διδασκαλία των αλγοριθμικών δομών στα πλαίσια του μαθήματος Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον. 3ο Συνέδριο ΤΠΕ στην εκπαίδευση.* Σύρος.

Τριανταφύλλου, Ι. Σ. (2017). *Στατιστική Ανάλυση Δεδομένων με χρήση του πακέτου IBM SPSS Statistics.* . Ανάκτηση από <http://eclass.uth.gr/>

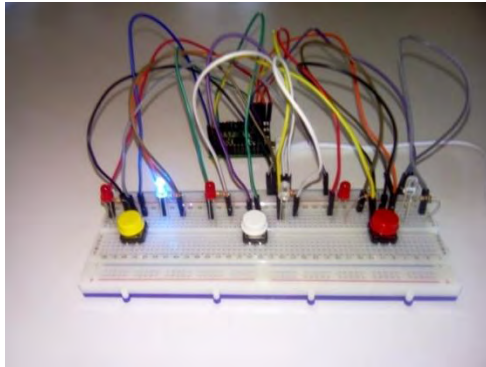
Τσιωτάκης, Π. (2008). *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον.* Αθήνα: Σαββάλας.

Παράρτημα Α

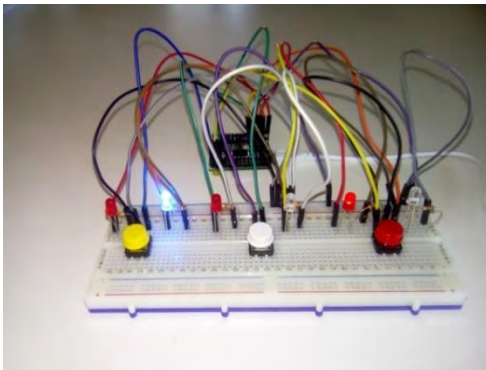
Οδηγίες παιχνιδιού και φωτογραφικό υλικό

Φύλλο εργασίας 5

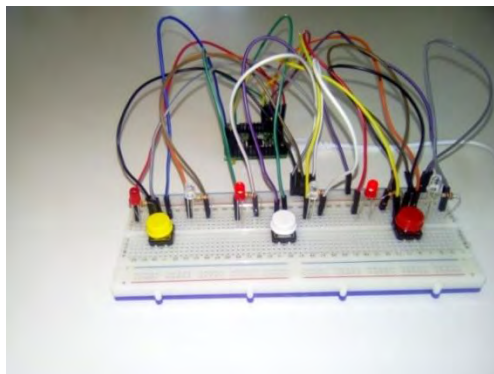
Οι εκπαιδευόμενοι παίζουν το παιχνίδι με τις χρωματιστές κάρτες. Οι εκπαιδευόμενοι χρησιμοποιούν την πλακέτα για να παίξουν το παιχνίδι όπως απεικονίζεται και στις εικόνες 14, 15 και 16. Ακολουθεί το πρόγραμμα του παιχνιδιού σε MicroPython.



Εικόνα 14: Μπλε led αναμμένο (1ος παίκτης)



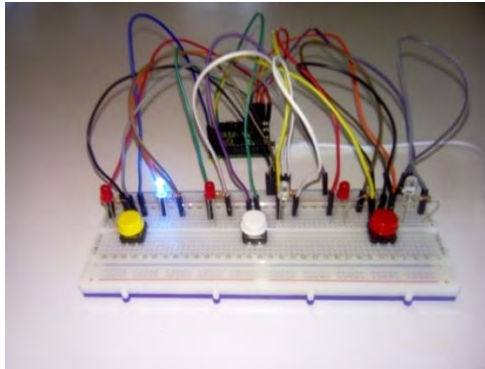
Εικόνα 15: Κόκκινο led αναμμένο (2ος παίκτης)



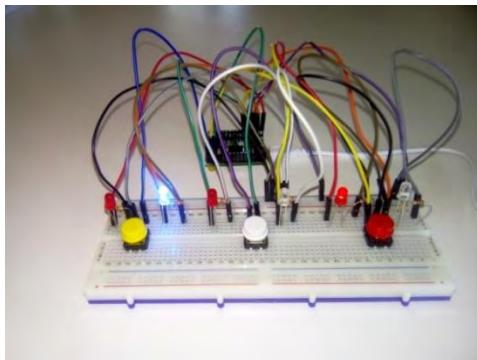
Εικόνα 16: Κόκκινο led αναμμένο (βοηθητικός παίκτης). Αποτέλεσμα: Νικητής 2ος παίκτης

Φύλλο εργασίας 6

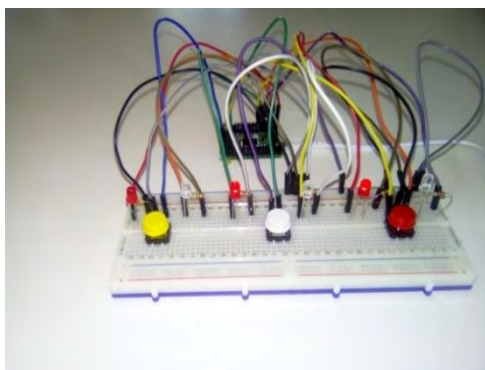
Οι εκπαιδευόμενοι παίζουν το παιχνίδι με τις χρωματιστές κάρτες όπως και στο φύλλο εργασίας 5 με τη διαφορά ότι η κάθε κάρτα έχει έναν αριθμό και προσθέτει 4 επιπλέον πόντους στον παίκτη που έχει κόκκινο χρώμα και 6 επιπλέον πόντους στον παίκτη που έχει μπλε χρώμα. Οι εκπαιδευόμενοι χρησιμοποιούν την πλακέτα για να παίξουν το παιχνίδι όπως απεικονίζεται και στις εικόνες 17, 18 και 19. Ακολουθεί το πρόγραμμα του παιχνιδιού σε MicroPython.



Εικόνα 17: Μπλε led αναμμένο – 6 πόντοι (1ος παίκτης)



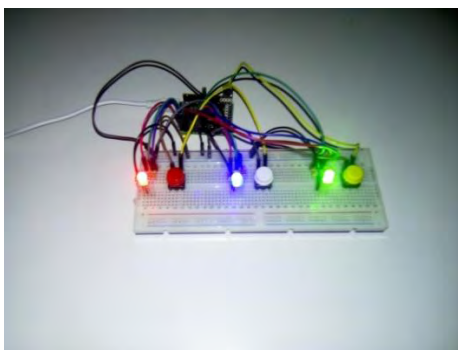
Εικόνα 18: Κόκκινο led αναμμένο – 4 πόντοι (2ος παίκτης)



Εικόνα 19: Κόκκινο led αναμμένο (βοηθητικός παίκτης). Αποτέλεσμα: Ισοπαλία 6 - 6

Φύλλο εργασίας 7

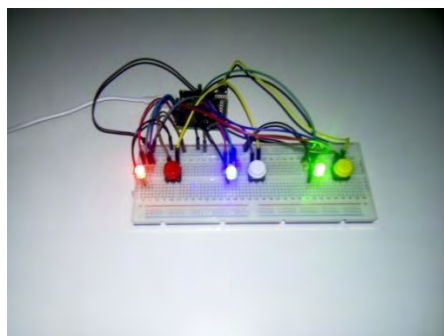
Οι εκπαιδευόμενοι παίζουν το παιχνίδι με τρεις χρωματιστές κάρτες σε σχέση με το φύλλο εργασίας 6 με τη διαφορά ότι η πράσινη κάρτα που προστίθεται στο παιχνίδι αφαιρεί 5 πόντους από τον παίκτη. Σε περίπτωση που προκύψουν σε κάποιον αρνητικοί πόντοι τότε απλά μηδενίζονται. Οι εκπαιδευόμενοι χρησιμοποιούν την πλακέτα για να παίξουν το παιχνίδι όπως απεικονίζεται στην εικόνα 20, στην οποία έχουν αντικατασταθεί τα 6 led λαμπάκια με 3 RGB led λαμπάκια κοινής ανόδου. Το λαμπάκι του νικητή γίνεται άσπρο. Ακολουθεί το πρόγραμμα του παιχνιδιού σε MicroPython.



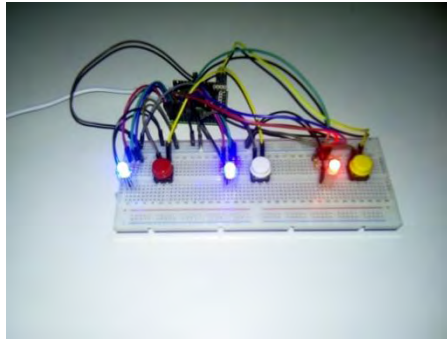
Εικόνα 20: Κόκκινο RGB led αναμμένο (1ος παίκτης), πράσινο RGB led αναμμένο (2ος παίκτης) και μπλε RGB led αναμμένο (βοηθητικός παίκτης). Αποτέλεσμα: Νικητής ο 1ος παίκτης 4 – 0.

Φύλλο εργασίας 8

Οι εκπαιδευόμενοι παίζουν το παιχνίδι με τρεις χρωματιστές κάρτες όπως και στο φύλλο εργασίας 7 με τη διαφορά ότι ο παίκτης παίρνει ή χάνει τους πόντους μόνο σε περίπτωση που έχει το ίδιο χρώμα με τον βοηθητικό παίκτη. Οι εκπαιδευόμενοι χρησιμοποιούν την πλακέτα για να παίξουν το παιχνίδι όπως απεικονίζεται στις εικόνες 21 και 22. Ακολουθεί το πρόγραμμα του παιχνιδιού σε MicroPython.



Εικόνα 21: Κόκκινο RGB led αναμμένο (1ος παίκτης), πράσινο RGB led αναμμένο (2ος παίκτης) και μπλε RGB led αναμμένο (βοηθητικός παίκτης). Αποτέλεσμα: Ισοπαλία 0 – 0



Εικόνα 22: Μπλε RGB led αναμμένο (1ος παίκτης), κόκκινο RGB led αναμμένο (2ος παίκτης) και μπλε RGB led αναμμένο (βοηθητικός παίκτης). Αποτέλεσμα: Νικητής 1ος παίκτης 8 - 0

Φύλλο εργασίας 10

Οι εκπαιδευόμενοι παίζουν το παιχνίδι με τρεις χρωματιστές κάρτες όπως και στο φύλλο εργασίας 8 με τη διαφορά ότι το παιχνίδι επαναλαμβάνεται για πολλούς γύρους μέχρι ο ένας από τους δύο παίκτες να κερδίσει πρώτος 10 πόντους. Επομένως, θα χρειαστεί οι εκπαιδευόμενοι να σημειώνουν κάθε φορά τη βαθμολογία τους σε ένα σημειωματάριο. Οι εκπαιδευόμενοι χρησιμοποιούν την πλακέτα για να παίξουν το παιχνίδι. Ακολουθεί το πρόγραμμα του παιχνιδιού σε MicroPython.

Φύλλο εργασίας 11

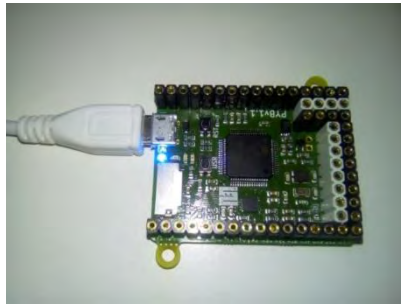
Οι εκπαιδευόμενοι παίζουν το παιχνίδι με τρεις χρωματιστές κάρτες όπως και στο φύλλο εργασίας 10 με τη διαφορά ότι το παιχνίδι επαναλαμβάνεται για 5 γύρους. Οι εκπαιδευόμενοι χρησιμοποιούν την πλακέτα για να παίξουν το παιχνίδι. Ακολουθεί το πρόγραμμα του παιχνιδιού σε MicroPython.

Φύλλο εργασίας 12

Οι εκπαιδευόμενοι παίζουν το παιχνίδι με τρεις χρωματιστές κάρτες όπως και στο φύλλο εργασίας 11 με τη διαφορά ότι σε κάθε γύρο του παιχνιδιού οι παίκτες παίρνουν εναλλάξ από 3 κάρτες. Οι εκπαιδευόμενοι χρησιμοποιούν την πλακέτα για να παίξουν το παιχνίδι. Ακολουθεί το πρόγραμμα του παιχνιδιού σε MicroPython.

Φωτογραφικό υλικό διδακτικών σεναρίων

Διδακτικό σενάριο 1



Εικόνα 23: Μπλε led αναμμένο



Εικόνα 24: Πράσινο led αναμμένο (Τυχαίος αριθμός 2)

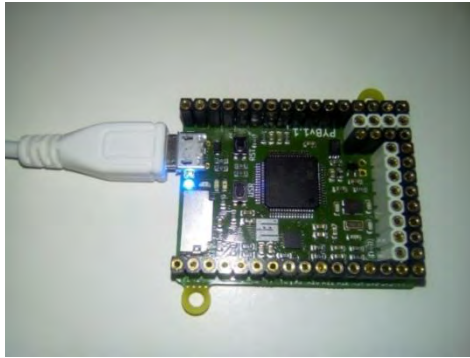
Διδακτικό σενάριο 2



Εικόνα 25: Τέσσερα αναμμένα led που σημαίνει ότι η συνθήκη είναι True.



Εικόνα 26: Τρία αναμμένα led (μαζί με το πράσινο) που σημαίνει ότι η συνθήκη είναι False.



Εικόνα 27: Ένα αναμμένο led (μπλε) που σημαίνει ότι όλες οι συνθήκες είναι False.



Εικόνα 28: Ένα αναμμένο led (πράσινο) που σημαίνει ότι η δεύτερη συνθήκη είναι True.

Διδακτικό σενάριο 3

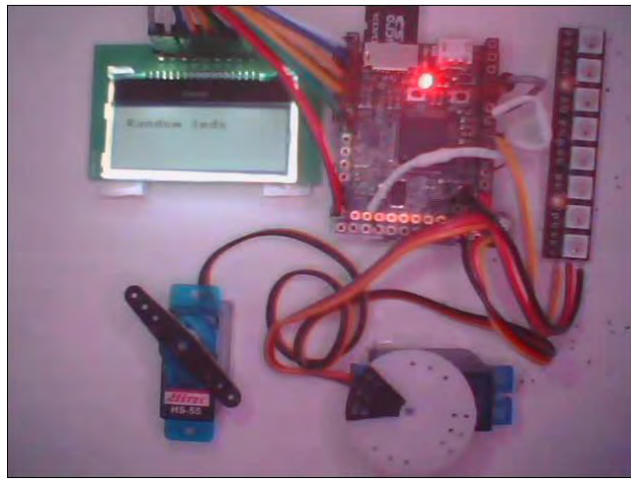


Εικόνα 29: Εμφάνιση των θετικών μονοψήφιων περιττών αριθμών στην LCD οθόνη⁴¹

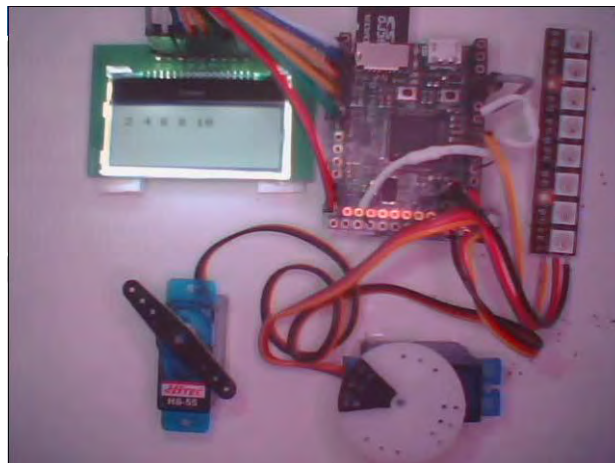
⁴¹ <http://micropython.org/live/>



Εικόνα 30: Εμφάνιση του μηνύματος «Random leds» στην LCD οθόνη και άναμμα του μπλε led (τυχαίο)⁴²



Εικόνα 31: Εμφάνιση του μηνύματος «Random leds» στην LCD οθόνη, άναμμα του κόκκινου led (τυχαίο) και τερματισμός της επανάληψης⁴³



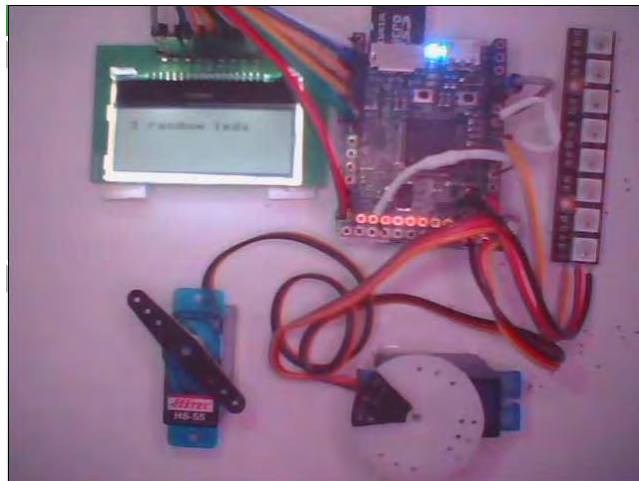
Εικόνα 32: Εμφάνιση των θετικών μονοψήφιων άρτιων αριθμών στην LCD οθόνη⁴⁴

⁴² <http://micropython.org/live/>

⁴³ <http://micropython.org/live/>



Εικόνα 33: Εμφάνιση του μηνύματος «3 Random leds» στην LCD οθόνη και άναμμα του μπλε led (τυχαίο)⁴⁵

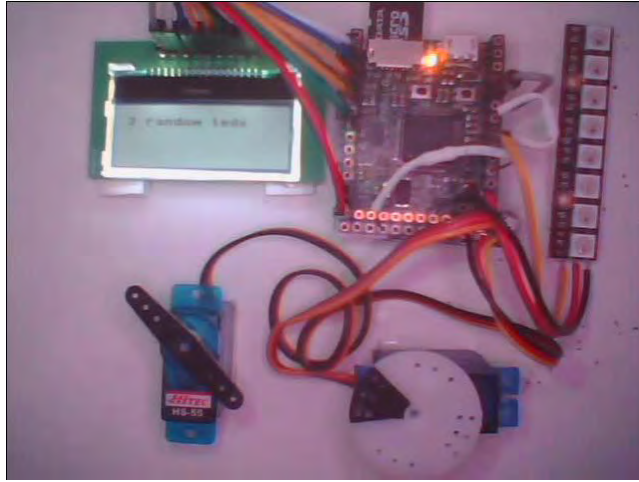


Εικόνα 34: Εμφάνιση του μηνύματος «Random leds» στην LCD οθόνη και άναμμα του μπλε led (τυχαίο)⁴⁶

⁴⁴ <http://micropython.org/live/>

⁴⁵ <http://micropython.org/live/>

⁴⁶ <http://micropython.org/live/>



Εικόνα 35: Εμφάνιση του μηνύματος «Random leds» στην LCD οθόνη και άναμμα του κίτρινου led (τυχαίο)⁴⁷



Εικόνα 36: Εμφάνιση της προπαίδειας του 5 και του 6 στην LCD οθόνη⁴⁸

⁴⁷ <http://micropython.org/live/>

⁴⁸ <http://micropython.org/live/>

Παράρτημα Β

Πίνακας 74: Αυτοσχέδιο ερωτηματολόγιο CIS (Keller)

Ερωτηματολόγιο αξιολόγησης του μαθήματος των διδακτικών σεναρίων στο εργαστήριο (Modified CIS).

Όνοματεπώνυμο:

Φύλο:

Ηλικία:

Ημερομηνία:

1=Διαφωνώ Απόλυτα 2=Διαφωνώ 3=Αδιαφορώ 4=Συμφωνώ 5=Συμφωνώ Απόλυτα

1. Ο εκπαιδευτής γνώριζε πώς να μας κάνει να αισθανόμαστε ενθουσιασμένοι με το αντικείμενο αυτού του μαθήματος, δηλαδή τον προγραμματισμό σε Python.
 2. Όλα όσα έμαθα σε αυτό το μάθημα θα είναι χρήσιμα για μένα.
 3. Αισθάνομαι σιγουριά ότι πήγα καλά σε αυτό το μάθημα.
 4. Το μάθημα είχε πολύ λίγα πράγματα που τράβηξαν την προσοχή μου.
 5. Ο εκπαιδευτής έκανε το αντικείμενο αυτού του μαθήματος, δηλαδή τον προγραμματισμό σε Python, να φαίνεται σημαντικό για την εποχή μας.
 6. Έπρεπε να είσαι τυχερός για να αποδώσεις σε αυτό το μάθημα.
 7. Έπρεπε να εργαστώ πολύ σκληρά για να πετύχω σε αυτό το μάθημα.
 8. Δεν καταλαβαίνω πώς το περιεχόμενο αυτού του μαθήματος, δηλαδή ο προγραμματισμός, σχετίζεται με αυτά που ήδη γνωρίζω.
 9. Το αν πέτυχα ή όχι σε αυτό το μάθημα εξαρτήθηκε κυρίως από μένα.
 10. Ο εκπαιδευτής χρησιμοποίησε τρόπους που δημιουργούσαν αγωνία κάθε φορά που έφτανε σε ένα σημείο του μαθήματος.
 11. Το αντικείμενο αυτού του μαθήματος, δηλαδή ο προγραμματισμός σε Python, ήταν πολύ δύσκολο για μένα.
 12. Αισθανόμουν μεγάλη ικανοποίηση κάθε φορά που τελείωνε το μάθημα.
 13. Σε αυτό το μάθημα προσπάθησα να θέσω και να επιτύχω υψηλά πρότυπα αριστείας.
 14. Πιστεύω ότι η αναγνώριση που έλαβα ήταν δίκαιη σε σύγκριση με τους άλλους εκπαιδευόμενους.
 15. Οι εκπαιδευόμενοι του μαθήματος έδειχναν περιέργεια για να μάθουν προγραμματισμό με τη βοήθεια της πλακέτας pyboard.
-

-
16. Μου άρεσε να εργάζομαι για αυτό το μάθημα.
 17. Είναι δύσκολο να προβλέψει κανείς πώς ο εκπαιδευτής θα αξιολογήσει την προσπάθειά του εκπαιδευόμενου.
 18. Είμαι ευχαριστημένος με τις αξιολογήσεις του εκπαιδευτή για το έργο μου σε σύγκριση με το πόσο καλά πίστευα ότι απέδωσα.
 19. Αισθάνομαι ικανοποιημένη/ος με αυτά που έμαθα σε αυτό το μάθημα.
 20. Το περιεχόμενο αυτού του μαθήματος, δηλαδή ο προγραμματισμός, σχετίζεται με τις προσδοκίες και τους στόχους μου.
 21. Ο εκπαιδευτής χρησιμοποίησε παραδείγματα της καθημερινής ζωής που ήταν ενδιαφέροντα.
 22. Οι εκπαιδευόμενοι συμμετείχαν ενεργά στο μάθημα.
 23. Για να πετύχω τους στόχους μου, ήταν σημαντικό να πάω καλά σε αυτό το μάθημα.
 24. Ο εκπαιδευτής χρησιμοποίησε την τεχνική της παιχνιδοποίησης που ήταν ενδιαφέρουσα.
 25. Δεν νομίζω ότι ωφελήθηκα πολύ από αυτό το μάθημα.
 26. Συχνά κάτι μου αποσπούσε την προσοχή ενώ βρισκόμουν στο μάθημα.
 27. Καθώς παρακολουθούσα το μάθημα, πίστευα ότι μπορώ να πετύχω αν προσπαθήσω αρκετά σκληρά.
 28. Τα προσωπικά οφέλη αυτού του μαθήματος ήταν σαφή για μένα.
 29. Η περιέργειά μου συχνά ενθαρρυνόταν από τις ερωτήσεις που υποβάλλονταν ή τα προβλήματα που παρουσιάζονταν σχετικά με το αντικείμενο αυτού του μαθήματος, δηλαδή τον προγραμματισμό σε Python.
 30. Θεωρώ ότι το επίπεδο πρόκλησης σε αυτό το μάθημα ήταν σωστό: ούτε πολύ εύκολο ούτε πολύ δύσκολο.
 31. Αισθάνομαι μάλλον απογοητευμένη/ος από αυτό το μάθημα.
 32. Αισθάνομαι ότι έλαβα αρκετή αναγνώριση της δουλειάς μου σε αυτό το μάθημα με σχόλια.
 33. Ο όγκος των δραστηριοτήτων που έπρεπε να κάνω ήταν κατάλληλος για αυτόν τον τύπο μαθημάτων.
 34. Είχα αρκετή ανατροφοδότηση για να μάθω πόσο απέδιδα στις δραστηριότητες των φύλλων εργασιών μέσω των διαθέσιμων απαντήσεων.
-

Ευρετήριο

A		K	
Αλγόριθμος	47	Κλίμακα Likert	130
Αντικειμενοστραφής προγραμματισμός	20	Κοινωνικός εποικοδομισμός.....	39
B		Λ	
Βιβλιοθήκες	54	Λίστα.	116
Βιωματική μάθηση.....	34	Λογικός Προγραμματισμός	22
Γ		Λογισμικό	18
Γλώσσα MicroPython	28, 39	M	
Γλώσσα προγραμματισμού Python.....	23	Μοντέλο ARCS	127
Δ		Π	
Δηλωτικός προγραμματισμός	19	Παιχνιδοποίηση	34, 35, 37
Διδακτικό σενάριο	38	Πλακέτα MicroPython Pyboard v1.1.....	39
Δομή ακολουθίας	53	Προγραμματισμός.....	18
Δομή επανάληψης.....	95	Προσοχή	127
Δομή επιλογής.....	71	Προστακτικός προγραμματισμός	19
Δομημένος προγραμματισμός.....	19	Σ	
Ε		Συναρτησιακός προγραμματισμός.....	21
Εικονική μηχανή	23	Συντελεστής εσωτερικής συνέπειας	
Έλεγχος One-Sample T-Test	133	Cronbach's alpha	132
Έλεγχος κανονικότητας των δεδομένων		Συσχέτιση	128
του δείγματος Shapiro-Wilk	133	T	
Εμπιστοσύνη	128	Τρόποι αναπαράστασης αλγορίθμου	48
Εμφωλευμένος βρόχος.....	101	Υ	
Εργαλείο CIS	130	Υλικό	18
Ι <td>Υπολογιστές.....</td> <td>18</td>		Υπολογιστές.....	18
Ικανοποίηση.....	129		