

ΟΔΗΓΟΣ ΓΙΑ ΤΟΝ ΕΚΠΑΙΔΕΥΤΙΚΟ

ΠΛΗΡΟΦΟΡΙΚΗ

(Τάξη: Γ΄)

ΓΕΝΙΚΟ ΛΥΚΕΙΟ

2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

ΕΙΔΙΚΟΙ ΕΠΙΣΤΗΜΟΝΕΣ

ΑΞΙΟΛΟΓΗΣΗΣ:

Μαυρίδης Ιωάννης, Αν. Καθηγητής Πανεπιστημίου Μακεδονίας (Συντονιστής)

Βαρζάκας Παναγιώτης, Αν. Καθηγητής ΑΤΕΙ Στερεάς Ελλάδας

Παπαδάκης Σπυρίδων, Σχολικός Σύμβουλος ΠΕ19

Μαραγκός Κωνσταντίνος, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ19

ΕΠΙΣΤΗΜΟΝΙΚΗΣ
ΕΠΟΠΤΕΙΑΣ:

Μπελεσιώτης Βασίλειος, Σχολικός Σύμβουλος ΠΕ19

ΕΚΠΟΝΗΣΗΣ:

Αράπογλου Αριστείδης, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ19

Βραχνός Ευριπίδης, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ19

«ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21ου αιώνα) – Νέο Πρόγραμμα Σπουδών»
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ «ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ»

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Σωτήριος Γκλαβιάς

Πρόεδρος του Ινστιτούτου Εκπαιδευτικής Πολιτικής

Υπεύθυνη Πράξης

Γεωργία Φέρμελη

Σύμβουλος Α΄ Ινστιτούτου Εκπαιδευτικής Πολιτικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Το παρόν συγχρηματοδοτήθηκε από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και εθνικούς πόρους στο πλαίσιο της πράξης «ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21ου αιώνα) – Νέο Πρόγραμμα Σπουδών» του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Διά Βίου Μάθηση»

Μάθημα
«Πληροφορική»
Γ' τάξης Γενικού Λυκείου

**Οδηγός για τον
Εκπαιδευτικό
και
Πρόγραμμα Σπουδών**

Έκδοση 1^η, Ιανουάριος 2015

Περιεχόμενα

Πρόλογος	8
Σκοπός και Στόχοι του Οδηγού Εκπαιδευτικού.....	8
Παραδοχές – Συμβάσεις	8
Η ταυτότητα του μαθήματος και ΠΣ.....	2
Σκοπός του μαθήματος.....	2
Προτεινόμενη πορεία διδασκαλίας.....	4
Η Αρχιτεκτονική και η Ειδική φιλοσοφία του ΠΣ	4
Το σύστημα και τα κριτήρια αξιολόγησης	5
Διδακτική μεθοδολογία	5
Θεματικές ενότητες και Χρονοπρογραμματισμός διδασκαλίας.....	6
1. Εισαγωγή στην Υπολογιστική Σκέψη	14
1.1 Λέξεις Κλειδιά.....	14
1.2 Προτεινόμενος Χρόνος	14
1.3 Σκοπός-Στόχοι.....	14
1.4 Πλαίσιο	14
1.5 Διδακτικές Προτάσεις.....	15
1.6 Χώρος.....	15
1.7 Εποπτικά Μέσα-Διδακτικά Μέσα.....	15
1.8 Ενδεικτικές Δραστηριότητες	16
1.8.1 Δραστηριότητα 1: Διαχείριση προβλημάτων.....	16
1.8.2 Δραστηριότητα 2: Μελέτη προβλήματος «Ασφαλής πλοήγηση στο Διαδίκτυο, σύνταξη ενός δεκάλογου συμβουλευτικών οδηγιών».....	16
1.8.3 Δραστηριότητα 3: Ανάλυση σύνθετου υπολογιστικού προβλήματος	17
1.8.4 Δραστηριότητα 4: Επιλογή συντομότερης διαδρομής	17
1.8.5 Δραστηριότητα 5: Διαδικασία λήψης απόφασης για το σχεδιασμό ενός εκπαιδευτικού παιχνιδιού	18
1.9 Βιβλιογραφία	18
1.10 Πηγές-Πρόσθετο Υλικό - Χρήσιμοι Εκπαιδευτικοί Πόροι	18
1.11 Σημειώσεις εκπαιδευτικού	19
2. Υπολογιστής Γενικού Σκοπού: Υλικό και Λογισμικό	20
2.1 Λέξεις Κλειδιά:.....	20
2.2 Προτεινόμενος Χρόνος	20
2.3 Σκοπός-Στόχοι.....	20

2.4	Πλαίσιο	20
2.5	Διδακτικές Προτάσεις.....	21
2.6	Χώρος.....	22
2.7	Εποπτικά Μέσα-Διδακτικά Μέσα.....	22
2.8	Ενδεικτικές Δραστηριότητες	22
2.8.1	Δραστηριότητα 1: Αρχιτεκτονική von Neumann	22
2.8.2	Δραστηριότητα 2: Επίπεδα ιεραρχίας ενός υπολογιστικού συστήματος.....	23
2.8.3	Δραστηριότητα 3: Αναπαράσταση δεδομένων στον υπολογιστή.....	23
2.8.4	Δραστηριότητα 4: Μετατροπή δεκαεξαδικών αριθμών σε δυαδικούς.....	24
2.9	Βιβλιογραφία	24
2.10	Πηγές - Πρόσθετο Υλικό - Χρήσιμοι Εκπαιδευτικοί Πόροι	25
2.11	Σημειώσεις ενότητας εκπαιδευτικού	25
3.	Βασικά στοιχεία γλώσσας προγραμματισμού	26
3.1	Λέξεις Κλειδιά:.....	26
3.2	Προτεινόμενος Χρόνος	26
3.3	Σκοπός-Στόχοι:.....	26
3.4	Πλαίσιο	26
3.5	Διδακτικές Προτάσεις.....	26
3.6	Χώρος.....	28
3.7	Εποπτικά Μέσα-Διδακτικά Μέσα:	28
3.8	Ενδεικτικές Δραστηριότητες	28
3.8.1	Δραστηριότητα 1: Ο διερμηνευτής της γλώσσας Python	28
3.8.2	Δραστηριότητα 2: Ο διερμηνευτής της γλώσσας Python ως αριθμομηχανή	28
3.8.3	Δραστηριότητα 4: Κανόνες σύνταξης στη γλώσσα Python	29
3.8.4	Δραστηριότητα 4: Ασκήσεις χειρισμού μιγαδικών αριθμών	31
3.8.5	Δραστηριότητα 5: Υλοποίηση απλού προγράμματος σε Python.....	31
3.9	Βιβλιογραφία	31
3.10	Πηγές-Πρόσθετο Υλικό- Χρήσιμοι Εκπαιδευτικοί Πόροι.....	32
3.11	Σημειώσεις ενότητας εκπαιδευτικού	33
4.	Λογική οργάνωση δεδομένων κατά την αποθήκευσή τους.....	34
4.1	Λέξεις Κλειδιά:.....	34
4.2	Προτεινόμενος Χρόνος	34
4.3	Σκοπός-Στόχοι.....	34
4.4	Πλαίσιο	34
4.5	Διδακτικές Προτάσεις.....	35

4.5.1	Διδακτικές Προτάσεις: Μεταβλητή, Πίνακες-Λίστες και Κύρια Μνήμη	35
4.5.2	Ενδεικτικές Δραστηριότητες: Κύρια Μνήμη-Μεταβλητές-Πίνακες-Λίστες	36
4.5.3	Διδακτικές Προτάσεις για την υποενότητα της Συμπίεσης Δεδομένων	38
4.5.4	Ενδεικτικές Δραστηριότητες: Υποενότητα Συμπίεση Δεδομένων	40
4.6	Βιβλιογραφία	41
4.7	Πηγές-Πρόσθετο Υλικό- Χρήσιμοι Εκπαιδευτικοί Πόροι	41
4.8	Σημειώσεις ενότητας εκπαιδευτικού	43
5.	Από το πρόβλημα στον αλγόριθμο	44
5.1	Λέξεις Κλειδιά:.....	44
5.2	Προτεινόμενος Χρόνος:	44
5.3	Σκοπός-Στόχοι.....	44
5.4	Πλαίσιο-Διδακτικές Προτάσεις	44
5.5	Χώρος.....	46
5.6	Εποπτικά Μέσα-Διδακτικά Μέσα.....	46
5.7	Ενδεικτικές Δραστηριότητες	46
5.7.1	Δραστηριότητα 1	46
5.7.2	Δραστηριότητα 2	46
5.8	Δραστηριότητες-Ασκήσεις με απλά προβλήματα προς επίλυση	47
5.8.1	Δραστηριότητα 1η	48
5.8.2	Δραστηριότητα 2 ^η	48
5.8.3	Δραστηριότητα 3η	48
5.8.4	Δραστηριότητα 4η	48
5.8.5	Δραστηριότητα 5η	48
5.8.6	Δραστηριότητα 6η	49
5.8.7	Δραστηριότητα 7η	49
5.8.8	Δραστηριότητα 8η	49
5.8.9	Δραστηριότητα 9η	49
5.8.10	Δραστηριότητα 10η	49
5.8.11	Δραστηριότητα 11 ^η	49
5.8.12	Δραστηριότητα 12η	49
5.8.13	Άσκηση Αυτοαξιολόγησης.....	49
5.9	Βιβλιογραφία	50
5.10	Πηγές - Πρόσθετο Υλικό - Χρήσιμοι Εκπαιδευτικοί Πόροι	51
5.11	Σημειώσεις ενότητας εκπαιδευτικού	51
6.	Από τον αλγόριθμο στο πρόγραμμα	52

6.1	Λέξεις Κλειδιά.....	52
6.2	Προτεινόμενος Χρόνος	52
6.3	Σκοπός-Στόχοι.....	52
6.4	Πλαίσιο	52
6.5	Διδακτικές Προτάσεις.....	53
6.6	Χώρος.....	53
6.7	Εποπτικά Μέσα-Διδακτικά Μέσα:	53
6.8	Ενδεικτικές Δραστηριότητες	54
6.8.1	Δραστηριότητα: Τα βασικά είδη προγραμματισμού, αναζήτηση και οργάνωση πληροφοριών.	54
6.8.2	Δραστηριότητα: Δομή ακολουθίας.....	54
6.8.3	Δραστηριότητα: Δομή ελέγχου - Ηλικία	55
6.8.4	Δραστηριότητα Δομή ελέγχου - Εμφωλευμένο Αν –Αλλιώς.....	55
6.8.5	Δραστηριότητα: Δομή ελέγχου - Έλεγχος θερμοστάτη για ψύξη θέρμανση	56
6.8.6	Δραστηριότητα: Δομή ελέγχου	56
6.8.7	Δραστηριότητα: Δομή Επανάληψης.....	57
6.8.8	Δραστηριότητα: Αλφαριθμητικά	60
6.8.9	Δραστηριότητα: Λίστες	61
6.9	Πηγές-Πρόσθετο Υλικό- Χρήσιμοι Εκπαιδευτικοί Πόροι.....	66
6.10	Σημειώσεις ενότητας εκπαιδευτικού	67
7.	Πληροφοριακά Συστήματα.....	69
7.1	Λέξεις Κλειδιά.....	69
7.2	Προτεινόμενος Χρόνος	69
7.3	Σκοπός-Στόχοι.....	69
7.4	Πλαίσιο	69
7.5	Διδακτικές Προτάσεις.....	70
7.6	Χώρος.....	70
7.7	Εποπτικά Μέσα-Διδακτικά Μέσα.....	71
7.8	Ενδεικτικές Δραστηριότητες	71
7.8.1	Δραστηριότητα 1 ^η	71
7.8.2	Δραστηριότητα 2 ^η	71
7.8.3	Δραστηριότητα 3 ^η	71
7.8.4	Δραστηριότητα 4 ^η	71
7.8.5	Δραστηριότητα 5 ^η	71
7.8.6	Δραστηριότητα 6 ^η	72
7.8.7	Δραστηριότητα 7 ^η	72

7.8.8	Δραστηριότητα 8 ^η	72
7.8.9	Δραστηριότητα 9 ^η	73
7.9	Βιβλιογραφία	73
7.10	Πηγές - Πρόσθετο Υλικό - Χρήσιμοι Εκπαιδευτικοί Πόροι	73
1.	Σχέδιο Δραστηριότητας: Η έννοια του Αλγορίθμου	76
1.1	Προσδοκώμενα μαθησιακά αποτελέσματα	77
1.2	Οδηγίες για τον εκπαιδευτικό.....	77
1.2.1	Φύλλο εργασίας: Εισαγωγή στην έννοια του αλγόριθμου	78
2.	Σχέδιο Δραστηριότητας: Αριθμομηχανή	80
2.1	Προσδοκώμενα μαθησιακά αποτελέσματα	80
2.2	Οδηγίες για τον εκπαιδευτικό.....	80
2.3	Φύλλο εργασίας: Αριθμομηχανή	81
3.	Σχέδιο Δραστηριότητας: Εισαγωγή στις συναρτήσεις της Python	84
3.1	Προσδοκώμενα μαθησιακά αποτελέσματα	84
3.2	Οδηγίες για τον εκπαιδευτικό.....	84
3.3	Φύλλο εργασίας: Εισαγωγή στις Συναρτήσεις της Python.....	85
4.	Σχέδιο Δραστηριότητας: Παράμετροι και εμβέλεια συναρτήσεων	87
4.1	Προσδοκώμενα μαθησιακά αποτελέσματα	87
4.2	Οδηγίες για τον/την εκπαιδευτικό	87
4.3	Φύλλο εργασίας: Παράμετροι και Εμβέλεια Συναρτήσεων	88
4.3.1	Δραστηριότητα 1	88
4.3.2	Δραστηριότητα 2	88
4.3.3	Δραστηριότητα 3	89
4.3.4	Δραστηριότητα 4	89
5.	Σχέδιο Δραστηριότητας: Δομή Επιλογής	90
5.1	Προσδοκώμενα μαθησιακά αποτελέσματα	90
5.2	Οδηγίες για τον εκπαιδευτικό.....	90
5.3	Φύλλο εργασίας: Δομή Επιλογής.....	91
6.	Σχέδιο Δραστηριότητας: Πίνακες και Λίστες	94
6.1	Προσδοκώμενα μαθησιακά αποτελέσματα	94
6.2	Οδηγίες για τον εκπαιδευτικό.....	94
6.3	Φύλλο Εργασίας: Πίνακες και Λίστες	96
6.3.1	Δραστηριότητα 1	96
6.3.2	Δραστηριότητα 2	96
6.3.3	Δραστηριότητα 3	97

6.3.4	Δραστηριότητα 4	97
6.3.5	Δραστηριότητα 5	97
7.	Σχέδιο Δραστηριότητας: Αλγόριθμος συμπίεσης δεδομένων RLE	98
7.1	Προσδοκώμενα μαθησιακά αποτελέσματα	98
7.2	Οδηγίες για τον εκπαιδευτικό.....	98
7.2.1	Δραστηριότητα 1	98
7.2.2	Δραστηριότητα 2	99
7.2.3	Δραστηριότητα 3	99
7.3	Φύλλο Εργασίας: Αλγόριθμος Συμπίεσης Δεδομένων RLE	100
8.	Αναφορές Μέρους II	102

Πρόλογος

Ο οδηγός για τον εκπαιδευτικό αποτελεί «εργαλείο» για τον διδάσκοντα και περιέχει διδακτικές-παιδαγωγικές προσεγγίσεις και πρακτικές υλοποίησης του νέου Προγράμματος Σπουδών (ΠΣ) στη σχολική πραγματικότητα.

Σκοπός και Στόχοι του Οδηγού Εκπαιδευτικού

Ο οδηγός του εκπαιδευτικού λειτουργεί συμπληρωματικά με το ΠΣ και έχει στόχο να βοηθήσει τον εκπαιδευτικό στη διδακτική διαχείριση των ενοτήτων και στην υλοποίηση της φιλοσοφίας του. Χρησιμεύει στον εκπαιδευτικό τόσο στο σχεδιασμό της διδασκαλίας όσο και κατά τη διεξαγωγή της, προς την επίτευξη υψηλών μαθησιακών αποτελεσμάτων, παρέχοντας πρόσθετες γνώσεις και πηγές σχετικά με τις θεματικές ενότητες του ΠΣ ή επικαιροποιώντας υπάρχουσες.

Αποτελείται, με βάση τις οδηγίες του ΙΕΠ, από δύο μέρη.

Στο πρώτο μέρος περιλαμβάνονται:

- Συνοπτική αναφορά στο φιλοσοφικό πλαίσιο του μαθήματος.
- Βασικές μεθοδολογικές προσεγγίσεις, κατευθύνσεις και παραδείγματα για την αξιολόγηση.
- Συνοπτικές οδηγίες για τη διδασκαλία κάθε ενότητας του Προγράμματος Σπουδών.
- Σχετική βιβλιογραφία.

Στο δεύτερο μέρος περιλαμβάνονται, χαρακτηριστικά παραδείγματα σχεδίων μαθημάτων (ή και σχεδίων εργασίας) σύμφωνα με τα προτεινόμενα στο ΠΣ.

Παραδοχές – Συμβάσεις

Στο παρόν τεύχος, το γένος χρησιμοποιείται γενικά, εκτός χαρακτηριστικών περιπτώσεων, υπονοώντας ισότιμα και τα δύο γένη.

Οι όροι «Πληροφορική» και «Επιστήμη Υπολογιστών» θα χρησιμοποιούνται εναλλακτικά και ισοδύναμα. Παρόμοια χρησιμοποιούνται οι όροι ψευδοκώδικας/ψευδογλώσσα και αλφαριθμητικά/συμβολοσειρές.

Ο εκπαιδευτικός πρέπει να ελέγχει τους προτεινόμενους υπερσυνδέσμους και το περιεχόμενο των δικτυακών πόρων πριν την αξιοποίησή τους στην τάξη, για την όποια πιθανότητα αλλαγής τους στο μέλλον. Οι αναφερόμενοι σύνδεσμοι προσπελάστηκαν στις 5/1/2015 από τους συγγραφείς.

Συνίσταται η όσο το δυνατόν χρήση ανοικτού λογισμικού και λογισμικού καθώς και υπηρεσιών του ΠΣΔ.

Όλα τα παραδείγματα σε Python που παρουσιάζονται στον οδηγό έχουν αναπτυχθεί με την έκδοση 2.7.9 της γλώσσας και το περιβάλλον IDLE, χωρίς αυτό να περιορίζει τον εκπαιδευτικό στην προσαρμογή τους σε επόμενη έκδοση.

Η προτεινόμενη διάρκεια κάθε ενότητας στο ΠΣ αποτελεί το μέσο όρο των διδακτικών ωρών. Ο εκπαιδευτικός μπορεί να αναπροσαρμόζει τη διάρκεια ανάλογα με το γνωστικό υπόβαθρο των μαθητών, στο προτεινόμενο εύρος των διδακτικών ωρών στο ΠΣ (από-έως).

Η όλη ομάδα ευελπιστεί στη Διαδικτυακή Ανάρτηση του Οδηγού, σε ηλεκτρονική μορφή για να είναι δυνατή η βελτίωση και η επέκτασή του, από την κοινότητα των εκπαιδευτικών Πληροφορικής.

Μέρος



Το ΠΣ και η φιλοσοφία του μαθήματος

Ο μαθητής - πολίτης τον οποίο θέλουμε να εκπαιδεύσουμε πρέπει να αποκτήσει από την εκπαίδευσή του το απαραίτητο σύνολο γνώσεων, δεξιοτήτων/ικανοτήτων και στάσεων, ώστε να αντιμετωπίζει τις αυξανόμενες ανάγκες τόσο κατά τη συνέχιση των σπουδών του όσο και σε αυτές της καθημερινής ζωής. Να μπορεί να μαθαίνει αυτόνομα, αλλά και να συνεργάζεται, να επικοινωνεί και να αντιμετωπίζει προβληματικές καταστάσεις σε μεταβαλλόμενες συνθήκες.

Σύμφωνα δε με την πρόταση του Υπουργείου Παιδείας και Θρησκευμάτων (2012) για το «Νέο Σχολείο (Σχολείο του 21^{ου} αιώνα)» θα πρέπει να δοθεί ειδική μέριμνα στην εμπάθουση των υπό διδασκαλία γνωστικών αντικειμένων που θα δίνει τη δυνατότητα στο μαθητή να εμπλουτίσει κατάλληλα τις γνώσεις, ειδικά σε γνωστικά αντικείμενα που σχετίζονται με τις προτιμήσεις του σε επιστημονικές κατευθύνσεις και πεδία εξειδίκευσης.

Σε έναν κόσμο που περιβάλλεται από την τεχνολογία και τη συνεχή αλληλεπίδραση με υπολογιστές στις καθημερινές δραστηριότητες, η εκμάθηση των βασικών εννοιών και δομών προγραμματισμού ως συνάρτηση του τρόπου λειτουργίας του υπολογιστή, αποτελεί ουσιαστική προϋπόθεση για την αποτελεσματική αξιοποίησή τους. Επιπλέον, η ανάπτυξη δεξιοτήτων συγγραφής «κώδικα» βοηθάει στην καλλιέργεια του κριτικού τρόπου σκέψης, στην εξερεύνηση νέων ιδεών και στη βελτίωση της δημιουργικότητας, τόσο για την εργασία όσο και για την επικοινωνία.

Στη σημερινή διαδικτυωμένη κοινωνία που εξελίσσεται ραγδαία, είναι απαραίτητη η εμπλοκή των μαθητών της Γ' ΓΕΛ, ιδιαίτερα αυτών που θα ακολουθήσουν στην Τριτοβάθμια Εκπαίδευση συγγενή πεδία με την Επιστήμη των Υπολογιστών (ΕΥ), να έρθουν σε επαφή με τη θεματολογία και τη διεπιστημονική φύση της ΕΥ. Στο σημείο αυτό διευκρινίζουμε ότι οι όροι «Πληροφορική» και «Επιστήμη Υπολογιστών» θα χρησιμοποιούνται εναλλακτικά και ισοδύναμα. Ο συνδυασμός θεωρίας αλλά και πρακτικής στο εργαστήριο της κάθε σχολικής μονάδας, έχει ως στόχο να προσφέρει στους μαθητές την ευκαιρία τόσο να αφομοιώσουν τις έννοιες της ΕΥ όσο και να αναπτύξουν δεξιότητες και στάσεις μέσα σε ένα συνεργατικό και αλληλεπιδραστικό περιβάλλον μάθησης και δημιουργίας.

Η ταυτότητα του μαθήματος και ΠΣ

Το μάθημα «Πληροφορική» σύμφωνα με το άρθρο 57 Νόμου 4310 (ΦΕΚ 258/8-12-2014) έχει τα ακόλουθα χαρακτηριστικά:

- Είναι μάθημα της Γ' τάξης του Γενικού Λυκείου, μάθημα προσανατολισμού προς τις θετικές και τις τεχνολογικές επιστήμες. Απευθύνεται στους μαθητές οι οποίοι επιλέγουν το Επιστημονικό Πεδίο Εξειδίκευσης (ΕΠΕ) το οποίο έχει το μάθημα αυτό ως προϋπόθεση για την εισαγωγή τους σε Πανεπιστήμια και ΤΕΙ. Διδάσκεται έξι (6) ώρες ανά εβδομάδα.
- Είναι πανελλαδικά γραπτά εξεταζόμενο, όπως αναφέρεται στο Φ.Ε.Κ 193/Α/17-9-2013 «Αναδιάρθρωση της Δευτεροβάθμιας Εκπαίδευσης και λοιπές διατάξεις», άρθρο 4 «Εισαγωγή στην Τριτοβάθμια Εκπαίδευση αποφοίτων Γενικού Λυκείου» και στο ΦΕΚ 258/Α/8-12-2014, άρθρο 57 «Ρύθμιση θεμάτων διάρθρωσης εκπαιδευτικών προγραμμάτων Γενικού Λυκείου Παράγραφος 2».

Σκοπός του μαθήματος

Σκοπός του μαθήματος είναι να αναπτύξουν οι μαθητές αναλυτική και συνθετική σκέψη, να αποκτήσουν ικανότητες μεθοδολογικού χαρακτήρα και να μπορούν να επιλύουν προβλήματα και να δημιουργούν τα αντίστοιχα προγράμματα σε προγραμματιστικό περιβάλλον. Να προσφέρει ένα επιστημονικό υπόβαθρο για την Επιστήμη Υπολογιστών και την αξιοποίησή της σε άλλες επιστήμες, παράλληλα με μια εφαρμοσμένη προσέγγιση όπου χρησιμοποιείται μια πραγματική γλώσσα προγραμματισμού. Αποτελεί δε τη σύνδεση και συνέχεια μεταξύ των μαθημάτων Πληροφορικής που προηγήθηκαν, όπως:

- Το μάθημα Γενικής Παιδείας «Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ» της Β' Λυκείου, με μία (1) ώρα ανά εβδομάδα (Πρόγραμμα Σπουδών του μαθήματος «Εισαγωγή στις Αρχές

της Επιστήμης των Η/Υ» Β΄ Τάξης Γενικού Λυκείου και Επαγγελματικού Λυκείου, ΦΕΚ 934/Β/14-4-2014). Μάθημα που εισάγει τους μαθητές σε βασικές έννοιες της Επιστήμης Υπολογιστών, έτσι ώστε να γνωρίσουν τομείς και θεμελιώδεις έννοιες και να αναπτύξουν την αναλυτική και συνθετική τους σκέψη.

- Το μάθημα επιλογής «Εφαρμογές Πληροφορικής» της Α΄ Λυκείου (2 ώρες ανά εβδομάδα), το οποίο ολοκληρώνει τις βασικές γνώσεις που έχουν αποκτήσει οι μαθητές από το Γυμνάσιο (Πρόγραμμα Σπουδών του μαθήματος «Εφαρμογές Πληροφορικής» Α΄ τάξης Γενικού Λυκείου, ΦΕΚ 932/Β/14-4-2014).
- Τα μαθήματα Πληροφορικής στο Γυμνάσιο και ειδικά το μάθημα «Πληροφορική» της Γ΄ Γυμνασίου στο οποίο οι μαθητές εμπλέκονται με θεμελιώδεις έννοιες προγραμματισμού υπολογιστών και αποκτούν βασικές δεξιότητες.
- Το μάθημα «Τεχνολογίες της Πληροφορίας και των Επικοινωνιών (Τ.Π.Ε.) στην Πρωτοβάθμια Εκπαίδευση και στα ολοήμερα δημοτικά σχολεία που λειτουργούν με Ενιαίο Αναμορφωμένο Εκπαιδευτικό Πρόγραμμα (ΕΑΕΠ) (ΦΕΚ 1139/Β/28-7-2010) με στοιχεία προγραμματιστικής λογικής.

Προκειμένου το μάθημα να παρέχει ένα γόνιμο περιβάλλον μάθησης και πρακτικής προτείνεται και θεωρείται αναγκαίο, όπως αναφέρθηκε, να διδάσκεται στο Εργαστήριο Πληροφορικής, με τις έξι (6) ώρες διδασκαλίας να διαρθρώνονται σε τέσσερις (4) ώρες για την κάλυψη της ύλης με όλες τις κατάλληλες δραστηριότητες και σε δύο (2) για επιπλέον δραστηριότητες εμπέδωσης και εμβάθυνσης.

Η εξοικείωση με μια σύγχρονη γλώσσα προγραμματισμού και αντίστοιχα προγραμματιστικά περιβάλλοντα ανάπτυξης εφαρμογών, αποτελεί ουσιώδες συστατικό της εκπαίδευσης στο πλαίσιο του μαθήματος «Πληροφορική». Οι μαθητές θα πρέπει να αξιοποιήσουν προγραμματιστικά περιβάλλοντα για να αναπτύξουν περαιτέρω ικανότητες για τη δημιουργία προγραμμάτων, καθώς και την επιλογή κατάλληλων μορφών αρχείων ανάλογα με το κάθε υπολογιστικό πρόβλημα. Ακόμη, θα πρέπει να αναπτύξουν δεξιότητες στον προγραμματισμό και τη χρήση κατάλληλων διεπαφών προγραμματισμού εφαρμογών - γνωστών ως API (Application-Programming Interface), καθώς και εργαλείων λογισμικού και βιβλιοθηκών για την υποβοήθηση της επίλυσης προβλημάτων με αξιοποίηση υπολογιστών. Πρέπει να εξοικειωθούν με την αξιοποίηση διαδικαστικών και αντικειμενοστρεφών γλωσσών προγραμματισμού σε ανάλογα προγραμματιστικά περιβάλλοντα ανάπτυξης εφαρμογών, καθώς και με τα βασικά τους χαρακτηριστικά. Ακόμη, θα πρέπει να τους δοθούν οι αφορμές για να προβληματιστούν και να αναλογιστούν τις κοινωνικές και ηθικές επιπτώσεις της εργασίας τους, υιοθετώντας κατάλληλες συμπεριφορές επικοινωνίας, κυρίως όταν εργάζονται ως μέλη μιας ομάδας.

Το μάθημα βασίζεται τόσο στην παράσταση αλγορίθμων σε ψευδοκώδικα, εξελληνισμένο περιβάλλον, όσο και σε πραγματική γλώσσα προγραμματισμού, όπου χρησιμοποιείται η γλώσσα Python, για την ανάπτυξη των προγραμμάτων, προκειμένου να δώσει στο μαθητή μια καθαρή και σε βάθος ανάλυση θεμάτων κύρια αλγοριθμικής και προγραμματισμού. Αυτό διότι η Python υποστηρίζει διαφορετικά είδη προγραμματισμού όπως διαδικασιακός, δηλωτικός, αντικειμενοστρεφής, αλλά και διότι αξιοποιείται ευρέως, λόγω της ευκολίας χρήσης της, από τη διεθνή εκπαιδευτική κοινότητα. Επίσης παρουσιάζει μεγάλες ομοιότητες με τη χρήση ψευδοκώδικα, ενώ διαθέτει πληθώρα βιβλιοθηκών (modules) για περαιτέρω προχωρημένη αξιοποίησή της, στοιχεία που την κάνουν κατάλληλη για ένα εισαγωγικό μάθημα στην ΕΥ.

Οι επιμέρους σκοποί του ΠΣ του μαθήματος είναι η ανάπτυξη:

- του αναγκαίου γνωστικού υποβάθρου των μαθητών
 - της κριτικής τους σκέψης
 - της δημιουργικότητας
 - της ανθρώπινης επικοινωνίας και συνεργασίας
 - θετικών στάσεων στη μάθηση
- καθώς και η
- δημιουργία ενεργών, δημοκρατικών πολιτών.

Ως προς τα προσδοκώμενα αποτελέσματα το ΠΣ εκτείνεται σε τέσσερις διαστάσεις:

- στα μαθησιακά αποτελέσματα, δηλαδή στις γνώσεις, δεξιότητες και στάσεις που θα αποκτήσουν οι μαθητές
- στα βασικά περιεχόμενα και τα αντικείμενα με τα οποία αναμένεται να ασχοληθούν οι μαθητές στο μάθημα, καθώς και την οργάνωσή τους
- στις μαθησιακές δραστηριότητες που προτείνονται, καθώς και την παιδαγωγική φιλοσοφία που τις διέπει
- στο εκπαιδευτικό υλικό, όπου προτείνεται η αξιοποίηση πολλαπλού εκπαιδευτικού υλικού.

Ειδικότερα το ΠΣ του μαθήματος «Πληροφορική» καθώς και ο Οδηγός για τον Εκπαιδευτικό που το συνοδεύει, στοχεύουν στο να:

- δίνουν το γενικό πλαίσιο με βάση το οποίο καλείται να προγραμματίσει, σχεδιάσει και οργανώσει τη διδασκαλία του ο εκπαιδευτικός Πληροφορικής
- δίνουν τη δυνατότητα ευελιξίας στον εκπαιδευτικό να παρεμβαίνει και να αυτενεργεί στο περιεχόμενο και στη μέθοδο διδασκαλίας.
- περιγράφουν με σαφήνεια τα προσδοκώμενα αποτελέσματα για τις ικανότητες που θα πρέπει να αποκτήσουν οι μαθητές, ικανότητες από τις οποίες απορρέει η επιλογή περιεχομένων και διάρθρωσης της ύλης, καθώς και οι μέθοδοι διδασκαλίας και αξιολόγησης.
- εξασφαλίζουν τη συνέχεια και τη σύνδεση της γνώσης από προηγούμενες τάξεις και βαθμίδες εκπαίδευσης.
- διακρίνονται από συνοπτικότητα, ώστε να αποτελούν εργαλείο επικοινωνίας και καθοδήγησης της εκπαιδευτικής πράξης, προσιτό για τον εκπαιδευτικό, διατηρώντας την ισορροπία ανάμεσα στα διαφορετικά είδη μάθησης.
- υποστηρίζουν την εμβάθυνση ώστε να καλλιεργούνται περαιτέρω οι γνώσεις, δεξιότητες-ικανότητες, αλλά και στάσεις των μαθητών στα υπό μελέτη θέματα.
- λαμβάνουν υπόψη διαφορετικές παιδαγωγικές μεθόδους και τους διαφορετικούς ρυθμούς μάθησης των μαθητών, τις ιδιαιτερότητες στην τάξη, τις διαφορετικές κοινωνικό-πολιτισμικές αναπαραστάσεις και όλα τα άλλα στοιχεία που καταστούν τη διδασκαλία μια μοναδική, μη-τυποποιημένη διαδικασία
- προωθούν μεθοδολογίες, που καλλιεργούν το βιωματικό και συνεργατικό τρόπο μάθησης, ενσωματώνοντας στοιχεία της σύγχρονης ζωής
- αξιοποιούν τις νέες τεχνολογίες, που αποτελούν βασικό συντελεστή της νέας πραγματικότητας του ψηφιακού σχολείου, στη διδασκαλία.
- προτείνουν σχέδια εργασίας, που να υποστηρίζουν την έρευνα και όχι την αποστήθιση.
- χρησιμοποιούν πλούσιο εκπαιδευτικό υλικό (πολλαπλές αναπαραστάσεις) από διάφορες πηγές
- προτείνουν διάφορες εναλλακτικές μεθόδους και μορφές (διαγνωστικής, διαμορφωτικής και τελικής) αξιολόγησης
- έχουν δομημένη διάρθρωση και να κάνουν ορθή χρήση της γλώσσας και της ορολογίας του γνωστικού αντικειμένου
- λαμβάνουν υπόψη τη χρονική διάρκεια της διδασκαλίας και το αντίστοιχο μέγεθος του εκπαιδευτικού υλικού.

Προτεινόμενη πορεία διδασκαλίας

Η διδασκαλία εξελίσσεται παράλληλα σε δυο άξονες. Ο πρώτος άξονας αφορά στην κάλυψη της ύλης και περιλαμβάνει τις δυνατές διδακτικές δραστηριότητες, ενώ ο δεύτερος εστιάζει στην εμπέδωση της διδαχθείσας ύλης. Και οι δύο άξονες είναι δυνατόν να περιέχουν κάθε διδακτική δραστηριότητα, όπως και εργασία σε υπολογιστές.

Η Αρχιτεκτονική και η Ειδική φιλοσοφία του ΠΣ

Το ΠΣ αποτελείται από:

- α) Θεματικές Ενότητες με το χρονικό εύρος που προτείνεται να αφιερωθεί σε καθεμιά.
 - β) Τα αναμενόμενα Μαθησιακά Αποτελέσματα ανά ενότητα.
 - γ) Ενδεικτικές Εκπαιδευτικές Δραστηριότητες ανά ενότητα.
 - δ) Πηγές αναζήτησης διδακτικού υλικού.
- Τέλος, αναφέρεται στο τρόπο και τα κριτήρια αξιολόγησης.

Το ΠΣ συνοδεύεται από τον Οδηγό Εκπαιδευτικού που αποτελεί αναπόσπαστο τμήμα του και «εργαλείο» στον διδάσκοντα. Στοχεύει στην ανάδειξη της γενικής φιλοσοφίας και του εκπαιδευτικού προσανατολισμού του ΠΣ, καθώς και των διδακτικών προσεγγίσεων που προτείνονται. Επίσης, στην αποτελεσματική υποστήριξη των εκπαιδευτικών κατά το σχεδιασμό και την οργάνωση, τόσο της διδασκαλίας τους όσο και της μαθησιακής πορείας και καθοδήγησης των μαθητών, σύμφωνα με τους στόχους και τις παιδαγωγικές στρατηγικές του νέου ΠΣ. Περιέχει διδακτικές-παιδαγωγικές προσεγγίσεις και πρακτικές υλοποίησης του ΠΣ στη σχολική πραγματικότητα.

Η διαμόρφωση της παρούσας πρότασης για το ΠΣ:

- Βασίζεται κυρίως στο διεθνές πρότυπο CSTA / ACM K-12 Computer Science Standard [Computer Science Teachers Association (CSTA) 2011. K-12 Computer Science Standards. Revised 2011. Retrieved August 1, 2014. http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf], της Ομάδας Προτύπων (The CSTA Standards Task Force) της Εταιρείας Εκπαιδευτικών Επιστήμης Υπολογιστών (Computer Science Teachers Association (CSTA)) των ΗΠΑ σε συνεργασία με την ACM (Association for Computing Machinery).
- Προσαρμόζει τα παραπάνω στην προηγούμενη εμπειρία των μαθητών και στην ελληνική πραγματικότητα, όπως αναφέρθηκε σχετικά.
- Πληροί τους σκοπούς που αναφέρονται στο άρθρο 1 του Νόμου 4186 «Αναδιάρθρωση της Δευτεροβάθμιας Εκπαίδευσης και λοιπές διατάξεις» (Φ.Ε.Κ 193/Α/17-9-2013).
- Πληροί τα κριτήρια που αναφέρθηκαν σε προηγούμενη παράγραφο.

Το σύστημα και τα κριτήρια αξιολόγησης

Οι γραπτές προαγωγικές εξετάσεις του μαθήματος γίνονται με τέσσερα (4) θέματα από την εξεταστέα ύλη, με τα οποία ελέγχεται η γνώση και η κατανόηση εννοιών και ορολογίας, η κριτική ικανότητα, η δυνατότητα αναπαραγωγής γνωστικών στοιχείων, η ικανότητα του μαθητή να αναλύει, να συνθέτει και να δημιουργεί, συνδυάζοντας γνώσεις και δεξιότητες που απέκτησε για την επίλυση προβλημάτων. Σημαντική θεωρείται και πρέπει να ακολουθείται η διαμορφωτική αξιολόγηση κατά τη διάρκεια της διδασκαλίας. Σε κάθε περίπτωση διαμορφώνεται με τις τρέχουσες οδηγίες του Υ.ΠΑΙ.Θ.

Διδακτική μεθοδολογία

Η διδακτική του μαθήματος γίνεται με βάση τον κοινωνικό εποικοδομισμό και τις σύγχρονες θεωρήσεις για την «επεξεργασία των πληροφοριών». Το μάθημα προτείνεται και πρέπει να γίνεται στο εργαστήριο πληροφορικής ακολουθώντας μεθόδους αναζήτησης και ανακάλυψης για την οικοδόμηση της γνώσης. Με κατάλληλα απλά παραδείγματα επιδιώκεται οι μαθητές να αντιληφθούν τις διάφορες έννοιες και να είναι ικανοί να αξιολογούν και οργανώνουν τις έννοιες αυτές σε νοητικές δομές, συστηματοποιώντας και ολοκληρώνοντας τις πρότερες εμπειρίες τους. Ως κοινωνική οργάνωση της τάξης ακολουθείται και η ομαδοσυνεργατική προσέγγιση ιδιαίτερα σε θέματα αναζήτησης, διερεύνησης και ανάπτυξης εφαρμογών. Όσον αφορά τις εκπαιδευτικές τεχνικές προτείνεται να ακολουθούνται οι πλέον ενεργητικές εξ αυτών. Χρησιμοποιούνται συχνά αυθεντικά παραδείγματα από τον πραγματικό κόσμο. Ο καθηγητής λειτουργεί ως διευκολυντής για την οικοδόμηση της γνώσης από τον μαθητή κινούμενος με τεχνικές του «πλαϊσίου στήριξης» (scaffolding).

Θεματικές ενότητες και Χρονοπρογραμματισμός διδασκαλίας

Οι ενότητες του ΠΣ και ο προτεινόμενος χρονοπρογραμματισμός είναι ο ακόλουθος:

	Θεματικές Ενότητες	Ώρες Προόδου ύλης Προτεινόμενες (Από –Εως)	Ώρες εμπέδωσης Προτεινόμενες (Από –Εως)
1	Εισαγωγή στην Υπολογιστική Σκέψη	4 (3-5)	-
2	Υπολογιστής Γενικού Σκοπού: Υλικό και Λογισμικό	6 (5-7)	4 (3-5)
3	Βασικά στοιχεία γλώσσας προγραμματισμού	2 (1-3)	4 (3-5)
4	Λογική οργάνωση δεδομένων κατά την αποθήκευσή τους	4 (3-5)	4 (3-5)
5	Από το πρόβλημα στον αλγόριθμο	8 (7-9)	4 (3-5)
6	Από τον αλγόριθμο στο πρόγραμμα	6 (5-7)	8 (7-9)
7	Προηγμένα στοιχεία γλώσσας προγραμματισμού	6 (5-7)	8 (7-9)
8	Δομές Δεδομένων	10 (9-11)	4 (3-5)
9	Κλασικοί αλγόριθμοι	12 (10-14)	6 (5-7)
10	Επίδοση και πολυπλοκότητα αλγορίθμων	12 (10-14)	8 (7-9)
11	Αντικειμενοστρεφής Προγραμματισμός	10 (9-11)	6 (5-7)
12	Περιβάλλοντα ανάπτυξης λογισμικού	4 (3-5)	2 (1-3)
13	Εφαρμογές σε γλώσσα προγραμματισμού με χρήση API	2 (1-3)	6 (5-7)
14	Πληροφοριακά Συστήματα	8 (7-9)	-

Οι στόχοι, θεματικές ενότητες και ενδεικτικές δραστηριότητες του ΠΣ φαίνονται στον ακόλουθο πίνακα.

<p style="text-align: center;">ΣΤΟΧΟΙ</p> <p style="text-align: center;">Οι μαθητές πρέπει να είναι ικανοί:</p>	<p style="text-align: center;">ΘΕΜΑΤΙΚΕΣ ΕΝΟΤΗΤΕΣ</p> <p style="text-align: center;">[Ενδεικτικές Ώρες (εύρος) προόδου ύλης + Ενδεικτικές Ώρες (εύρος) εμπέδωσης ύλης]</p>	<p style="text-align: center;">ΠΑΡΑΤΗΡΗΣΕΙΣ/ΕΝΔΕΙΚΤΙΚΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ</p>
<ul style="list-style-type: none"> • Να περιγράφουν τι περιλαμβάνει η έννοια της υπολογιστικής σκέψης (ΥΣ). • Να περιγράφουν τη σημασία της ΥΣ στην επίλυση προβλημάτων της καθημερινής ζωής. • Να περιγράφουν τα χαρακτηριστικά της υπολογιστικής σκέψης. • Να αναλύουν ένα υπολογιστικό πρόβλημα περιγράφοντας τις βασικές διαδικασίες με τις οποίες αντιμετωπίζεται (διάσπαση, αναγνώριση προτύπων, γενίκευση και σχεδίαση αλγορίθμου). 	<p>1. Εισαγωγή στην Υπολογιστική Σκέψη</p> <p style="text-align: center;">[4 (από 3 έως 5) + 0 Ω]</p> <p>1.1 Η έννοια της υπολογιστικής σκέψης</p> <p>1.2 Χαρακτηριστικά (π.χ. αφαίρεση - abstraction, μοντελοποίηση, γενίκευση, μετασχηματισμός, πρότυπα - patterns).</p> <p>1.3 Υπολογιστική σκέψη και επίλυση προβλημάτων</p>	<ul style="list-style-type: none"> • Επεξήγηση σκοπού και στόχων μαθήματος και τρόπου εργασίας. • Σύνδεση υπολογιστικής σκέψης με την επίλυση προβλημάτων καθημερινότητας και από τη σχολική εμπειρία των μαθητών μέσω συζήτησης και καταιγισμού ιδεών. • Μέσα από μια μελέτη περίπτωσης ενός σύνθετου προβλήματος από τις εμπειρίες και τις γνώσεις των μαθητών, οι μαθητές, εργαζόμενοι σε ομάδες, αναλύουν το πρόβλημα σε απλούστερα, εντοπίζουν τα δεδομένα και σχεδιάζουν την επεξεργασία που απαιτείται για την επίλυση του προβλήματος.
<ul style="list-style-type: none"> • Να περιγράφουν με τη βοήθεια ενός διαστρωματικού αφαιρετικού μοντέλου μελέτης τα βασικά στοιχεία και τις λειτουργίες που απαιτούνται για την επίλυση ενός προβλήματος μέσω υπολογιστή. • Να εξηγούν την οργάνωση της κλασικής von Neumann μηχανής και των κύριων λειτουργικών μονάδων της. • Να εξηγούν το ρόλο και τους περιορισμούς του υλικού και του λογισμικού στην επίλυση προβλημάτων σε ένα υπολογιστικό σύστημα. • Να περιγράφουν τον τρόπο που τα δεδομένα (ακέραιοι, πραγματικοί, χαρακτήρες, κλπ) αναπαριστώνται σε έναν υπολογιστή. • Να αντιστοιχίζουν τους τύπους δεδομένων με ανάλογες εντολές ISA. 	<p>2. Υπολογιστής Γενικού Σκοπού: Υλικό και Λογισμικό</p> <p style="text-align: center;">[6 (5-7) + 4 (3- 5) Ω]</p> <p>2.1 Από την επιθυμητή συμπεριφορά (λύση προβλήματος) στα ηλεκτρονικά στοιχεία</p> <p>2.2 Επίπεδα αφαίρεσης:</p> <p style="margin-left: 20px;">2.2.1 Πρόβλημα</p> <p style="margin-left: 20px;">2.2.2 Αλγόριθμος</p> <p style="margin-left: 20px;">2.2.3 Πρόγραμμα</p> <p style="margin-left: 20px;">2.2.4 Αρχιτεκτονική Μηχανής (ISA)</p> <p style="margin-left: 20px;">2.2.5 Μικρο-αρχιτεκτονική</p> <p style="margin-left: 20px;">2.2.6 Λογικά Κυκλώματα</p> <p style="margin-left: 20px;">2.2.7 Ηλεκτρονικά Στοιχεία</p> <p>2.3 Μοντέλο von Neumann</p> <p>2.4 Αντιστοιχία τύπων δεδομένων και εντολών ISA (αναφορά)</p> <p>2.5 Αναπαράσταση δεδομένων</p> <p style="margin-left: 20px;">2.5.1 Δυαδικό, δεκαδικό</p>	<ul style="list-style-type: none"> • Οι μαθητές εργαζόμενοι σε ομάδες να αναζητήσουν πηγές και πόρους για α) την αρχιτεκτονική von Neumann και β) τα επίπεδα αφαιρετικής μελέτης ενός υπολογιστικού συστήματος. • Επίδειξη του κύκλου εκτέλεσης εντολής (π.χ $c = a + b$). • Με ενδεικτικά παραδείγματα να εξηγηθούν οι τρόποι αναπαράστασης των δεδομένων. • Με χρήση Hexdump να διερευνήσουν την αναπαράσταση του περιεχομένου διαφόρων τύπων αρχείων.

	<p>2.5.2 Ακέραιοι αριθμοί</p> <ul style="list-style-type: none"> • μη προσημασμένοι • συμπλήρωμα ως προς 1 και 2 • αριθμητικές και λογικές πράξεις • υπερχείλιση (overflow) <p>2.5.3 Πραγματικοί αριθμοί (κινητής υποδιαστολής με μονή καθώς και με διπλή ακρίβεια)</p> <p>2.5.4 Οκταδικό, δεκαεξαδικό</p> <p>2.5.5 Unicode, ASCII</p>	
<ul style="list-style-type: none"> • Να περιγράφουν τους βασικούς τύπους δεδομένων και τις βασικές εντολές της γλώσσας προγραμματισμού. • Να αναγνωρίζουν και να χρησιμοποιούν τις βασικές μαθηματικές συναρτήσεις γλώσσας προγραμματισμού. • Να χρησιμοποιούν το διερμηνευτή γλώσσας προγραμματισμού για να κάνουν απλές πράξεις. • Να χρησιμοποιούν το προγραμματιστικό περιβάλλον για σύνταξη κώδικα προγράμματος και μετέπειτα εκτέλεσής του. 	<p>3. Βασικά στοιχεία γλώσσας προγραμματισμού [2 (1-3) + 4 (3-5) Ω]</p> <p>3.1 Τύποι δεδομένων</p> <p>3.2 Βασικές εντολές</p> <p>3.3 Αριθμητικές και λογικές πράξεις</p> <p>3.4 Βασικές συναρτήσεις (έτοιμες μαθηματικές)</p> <p>3.5 Περιβάλλοντα προγραμματισμού</p> <p>3.6 Απλά προγράμματα</p>	<ul style="list-style-type: none"> • Οι μαθητές εργαζόμενοι ατομικά ή ανά δύο (pair programming) να πειραματιστούν με τον διερμηνευτή της γλώσσας εκτελώντας πράξεις με χρήση των δομικών στοιχείων της Python (η Python ως αριθμομηχανή). • Οι μαθητές, με χρήση του περιβάλλοντος προγραμματισμού της Python IDLE, αναπτύσσουν απλά προγράμματα (αποθηκευμένα σενάρια εντολών), ως μια πρώτη πρακτική άσκηση για την επίλυση προβλημάτων.
<ul style="list-style-type: none"> • Να περιγράφουν τον τρόπο αναπαράστασης της μεταβλητής στην κύρια μνήμη. • Να περιγράφουν τον τρόπο αποθήκευσης τιμών στην κύρια μνήμη ώστε να υλοποιούν τη δομή του Πίνακα και της Λίστας. • Να αναγνωρίζουν τους διαφορετικούς τύπους αρχείων και να τους αντιστοιχούν με τις εφαρμογές που τα δημιούργησαν. • Να αναφέρουν και επεξηγούν τρόπους και μεθοδολογίες κωδικοποίησης και συμπίεσης δεδομένων σε αρχεία. 	<p>4. Λογική οργάνωση δεδομένων και αποθήκευσή τους [4 (3-5) + 4 (3-5) Ω]</p> <p>4.1 Κύρια Μνήμη</p> <p>4.1.1 Μεταβλητές</p> <p>4.1.2 Δομές Δεδομένων:</p> <ul style="list-style-type: none"> • Στατικές: πίνακας • Δυναμικές: λίστα <p>4.2 Αρχεία</p> <p>4.2.1 Τύποι αρχείων και αντιστοιχία με εφαρμογές.</p> <p>4.2.2 Συμπίεση</p> <ul style="list-style-type: none"> • μη-απωλεστική • απωλεστική 	<ul style="list-style-type: none"> • Να επιδειχθούν (π.χ. με προσομοίωση) α) η δομή της κύριας μνήμης και διευθύνσεις και β) οι διαφορές αποθήκευσης μεταξύ πίνακα και λίστας. • Να γίνει επίδειξη προσομοίωσης της λειτουργίας ενός αλγόριθμου συμπίεσης. • Να αναζητήσουν-πειραματιστούν οι μαθητές, εργαζόμενοι σε ομάδες, αλγόριθμοι απωλεστικής (π.χ. αρχεία MP3) και μη απωλεστικής (LZW, RLE) συμπίεσης και να τις συνδέσουν με μορφές αρχείων που χρησιμοποιούν συχνά (εικόνας, βίντεο).

<ul style="list-style-type: none"> • Να αναγνωρίζουν ένα σύνθετο πρόβλημα και τα πλεονεκτήματα που προκύπτουν από την ανάλυσή του σε απλούστερα. • Να επιλύουν προβλήματα χρησιμοποιώντας τα βήματα ανάλυσης σε επιμέρους προβλήματα, με αξιοποίηση υπαρχουσών λύσεων. • Να επιλέγουν κατά τον σχεδιασμό αλγορίθμου τις κατάλληλες αλγοριθμικές δομές, όπως η ακολουθία, η επιλογή, η επανάληψη και η αναδρομή. • Να περιγράφουν έναν αλγόριθμο μέσω ψευδοκώδικα ή διαγράμματος ροής. 	<p>5. Από το πρόβλημα στον αλγόριθμο [8 (7-9)+4 (3-5) Ω]</p> <p>5.1 Εισαγωγή στη διαχείριση της πολυπλοκότητας ενός προβλήματος</p> <p>5.1.1 Η αξία της αφαίρεσης</p> <p>5.1.2 Ανάλυση ενός προβλήματος σε απλούστερα υποπροβλήματα</p> <p>5.2 Περιγραφή με ψευδοκώδικα / διάγραμμα ροής</p> <p>5.3 Αλγοριθμικές δομές: ακολουθία, επιλογή, επανάληψη, αναδρομή</p>	<ul style="list-style-type: none"> • Να συζητηθούν παραδείγματα απλών και πολύπλοκων - σύνθετων προβλημάτων. • Μέσω παραδείγματος να αναδειχθεί η αξία της «αφαίρεσης» για τη διαχείριση της πολυπλοκότητας ενός προβλήματος. • Επίδειξη έτοιμων αλγορίθμων σε ψευδοκώδικα και διάγραμμα ροής. • Εξάσκηση των μαθητών στην αξιοποίηση βασικών αλγοριθμικών δομών, με εναλλακτικούς τρόπους παρουσίασης.
<ul style="list-style-type: none"> • Να επεξηγούν τη λογική συγγραφής προγραμμάτων ανά είδος προγραμματισμού (Διαδικασιακός, Αντικειμενοστρεφής και Συναρτησιακός προγραμματισμός). • Να υλοποιούν έναν πλήρη κύκλο ανάπτυξης προγράμματος σε περιβάλλον προγραμματισμού (συγγραφή κώδικα, αποσφαλμάτωση, μετάφραση/διερμηνεία, εκτέλεση). • Να επιλέγουν και να χρησιμοποιούν βασικές αλγοριθμικές δομές, όπως ακολουθία, επιλογή, επανάληψη και δομές δεδομένων, όπως οι Συμβολοσειρές και οι Λίστες στη γλώσσα προγραμματισμού. 	<p>6. Από τον αλγόριθμο στο πρόγραμμα [6 (5-7) + 8 (7-9) Ω]</p> <p>6.1 Η λογική συγγραφής προγράμματος ανάλογα με το είδος προγραμματισμού (η χρήση της γλώσσας προγραμματισμού ως εργαλείο έκφρασης της σχεδίασης)</p> <ul style="list-style-type: none"> • Διαδικασιακός προγραμματισμός • Αντικειμενοστρεφής προγραμματισμός • Συναρτησιακός προγραμματισμός <p>6.2 Διαδικασία συγγραφής, μετάφρασης και εκτέλεσης προγράμματος</p> <p>6.3 Σύνταξη εντολών γλώσσας προγραμματισμού</p> <p>6.3.1 Αλγοριθμικές δομές (ακολουθία, επιλογή, επανάληψη)</p> <p>6.3.2 Δομές δεδομένων</p> <ul style="list-style-type: none"> • Συμβολοσειρές (ή Αλφαριθμητικά) (Strings) • Λίστες (Επεξεργασία λιστών, Τελεστές, Μέθοδοι) 	<ul style="list-style-type: none"> • Αναζήτηση πληροφοριών (ιστοεξερεύνηση) για τα είδη προγραμματισμού, όπως ο Διαδικασιακός, Αντικειμενοστρεφής, Συναρτησιακός προγραμματισμός και του τρόπου λειτουργίας του καθενός. • Επίδειξη σε Python και εφαρμογή της διαδικασίας συγγραφής, μετάφρασης και εκτέλεσης προγράμματος. • Παρουσίαση και πειραματισμός των αλγοριθμικών δομών της Python με παράλληλη αναφορά σε αντίστοιχες δομές της ψευδογλώσσας. • Ανάλυση των βασικών κανόνων συγγραφής ενός προγράμματος σε Python (χρήση εσοχών, σχόλια, εισαγωγικά κλπ). • Επεξεργασία συμβολοσειρών (π.χ συνένωση). • Βασικές λειτουργίες σε λίστες (εμφάνιση, εισαγωγή, διαγραφή στοιχείου κ.λπ).

<ul style="list-style-type: none"> • Να αναγνωρίζουν αντικείμενα και να περιγράφουν τα χαρακτηριστικά τους. • Να αναγνωρίζουν τις συναρτήσεις και τη χρήση των παραμέτρων. • Να συντάσσουν συναρτήσεις που πραγματοποιούν μια συγκεκριμένη εργασία και να αξιοποιούν τα πλεονεκτήματα που προκύπτουν. • Να εντοπίζουν-τεκμηριώνουν την εμβέλεια των μεταβλητών σε ένα πρόγραμμα ή σε μια συνάρτηση. • Να επεξηγούν απλούς αναδρομικούς αλγόριθμους. • Να εφαρμόζουν λειτουργίες στις δομές δεδομένων: Λεξικό, Πλειάδα. • Να εφαρμόζουν βασικές λειτουργίες σε αρχεία που χρησιμοποιεί η γλώσσα προγραμματισμού. 	<p>7. Προηγμένα στοιχεία γλώσσας προγραμματισμού [6 (5-7) + 8 (7-9) Ω]</p> <p>7.1 Αντικείμενα</p> <p>7.2 Ορισμός συναρτήσεων</p> <ul style="list-style-type: none"> • τυπικές-πραγματικές παράμετροι • εμβέλεια μεταβλητών <p>7.3 Αναδρομή</p> <p>7.4 Δομημένη συγγραφή-αρθρώματα (modules)</p> <p>7.5 Ενσωματωμένες (Built-in) δομές δεδομένων</p> <ul style="list-style-type: none"> • Λεξικό • Πλειάδες (Tuples) <p>7.6 Αρχεία δεδομένων</p> <ul style="list-style-type: none"> • Τύποι αρχείων στη γλώσσα προγραμματισμού • Χειρισμός αρχείων (άνοιγμα, κλείσιμο) • Λειτουργίες σε αρχεία (ανάγνωση - εγγραφή κλπ.) 	<ul style="list-style-type: none"> • Αναφορά στις έννοιες Αντικείμενο και Μέθοδοι του αντικειμένου, με παραδείγματα. • Να επεξηγηθεί ότι όλα στην Python είναι αντικείμενα με μεθόδους επί αυτών. • Παρουσίαση παραδειγμάτων με αντικείμενα (συμβολοσειρές, λίστες κ.λπ). • Εξάσκηση στο χειρισμό μεθόδων επί αντικειμένου (π.χ μέθοδοι χειρισμού συμβολοσειράς). • Να εξηγηθεί ο ορισμός συναρτήσεων, η χρήση των παραμέτρων, η κλήση τους και η επιστροφή τιμών καθώς και η εμβέλεια των μεταβλητών. • Επίδειξη αναδρομικών αλγορίθμων και υλοποίηση με παράδειγμα (π.χ. αριθμοί Fibonacci). • Επίδειξη χρήσης βιβλιοθηκών με παραδείγματα. • Εξάσκηση σε χειρισμό βασικών λειτουργιών σε δομές δεδομένων όπως το Λεξικό και η Πλειάδα. • Αναφορά σε τύπους αρχείων στην Python, δυνατούς τρόπους χειρισμού τους (άνοιγμα, κλείσιμο) και λειτουργίες (ανάγνωση, εγγραφή). • Εξάσκηση στο χειρισμό αρχείων.
<ul style="list-style-type: none"> • Να περιγράφουν τους τύπους δομών δεδομένων: Πίνακας, Σύνολο, Στοιβά - Σωρός - Ουρά, Δένδρο, Γράφος. • Να συγκρίνουν και να επιλέγουν την καταλληλότερη δομή ανάλογα με τον τύπο του προβλήματος. • Να εφαρμόζουν βασικές λειτουργίες σε δομές δεδομένων. 	<p>8. Δομές δεδομένων [10 (9-11) + 4 (3-5) Ω]</p> <p>8.1 Πίνακες (μονοδιάστατοι - πολυδιάστατοι)</p> <p>8.2 Σύνολα</p> <p>8.3 Στοιβά - Σωρός - Ουρά</p> <p>8.4 Δένδρα</p> <p>8.5 Γράφοι</p> <ul style="list-style-type: none"> • Πίνακας Γειτνίασης • Λίστα Γειτνίασης • Σύγκριση αναπαραστάσεων (μνήμη και πολυπλοκότητα βασικών λειτουργιών) 	<ul style="list-style-type: none"> • Μέσω παραδειγμάτων σχεδιασμού σε ψευδογλώσσα και υλοποίησης σε Python να αναπαρασταθούν δομές δεδομένων όπως: πίνακας, σύνολο, στοιβά, ουρά, δένδρο, γράφος (π.χ. δίκτυο δεδομένων ή οδικό δίκτυο με γράφους ή ιεραρχική δομή ενός συστήματος αρχείων). • Μέσω παραδειγμάτων να συγκρίνουν και να επιλέγουν (ανάπτυξη προβληματισμού) την καταλληλότερη δομή ανάλογα με τον τύπο του προβλήματος. • Να εφαρμόσουν

		ομαδοσυνεργατικά βασικές λειτουργίες στις δομές δεδομένων (είσοδος, εύρεση κλπ).
<ul style="list-style-type: none"> • Να υλοποιούν σε γλώσσα προγραμματισμού τους κλασικούς αλγόριθμους: Αναζήτησης, Ταξινόμησης και Διάσχισης Γράφων. • Να συγκρίνουν και να επιλέγουν τον κατάλληλο αλγόριθμο ανάλογα με το είδος του προβλήματος. 	<p>9. Κλασικοί Αλγόριθμοι [12 (10-14)+6 (5-7) Ω]</p> <p>9.1 Αναζήτηση</p> <ul style="list-style-type: none"> • σειριακή • δυαδική <p>9.2 Ταξινόμηση</p> <ul style="list-style-type: none"> • γρήγορη (quick) • με συγχώνευση (merge) <p>9.3 Διάσχιση γράφων</p> <ul style="list-style-type: none"> • κατά πλάτος • κατά βάθος 	<ul style="list-style-type: none"> • Δραστηριότητα (εννοιολογική χαρτογράφηση) αναζήτησης και κατηγοριοποίησης κλασικών αλγορίθμων, με τους μαθητές να εργάζονται σε ομάδες. • Δραστηριότητα σχεδιασμού κλασικών αλγορίθμων σε Ψευδογλώσσα και υλοποίηση σε Python με ή χωρίς αναδρομικότητα.
<ul style="list-style-type: none"> • Να συγκρίνουν την επίδοση δύο αλγορίθμων για την επίλυση συγκεκριμένου προβλήματος (benchmarking). • Να αναφέρουν περιορισμούς ως προς το χώρο και το χρόνο ανά κατηγορία αλγορίθμων. • Να υπολογίζουν την επίδοση αλγορίθμων αναζήτησης και ταξινόμησης. • Να εξηγούν τι σημαίνει ανάλυση χειρότερης περίπτωσης ενός αλγόριθμου. 	<p>10. Επίδοση και πολυπλοκότητα Αλγορίθμων [12 (10-14)+8 (7-9) Ω]</p> <p>10.1 Περιορισμοί ως προς χρόνο και χώρο</p> <p>10.2 Ανάλυση πολυπλοκότητας ως προς χρόνο/χώρο</p> <p>10.2.1 Υπολογισμός βημάτων:</p> <ul style="list-style-type: none"> • αναζήτηση <ul style="list-style-type: none"> ο σειριακή (sequential search) ο δυαδική (binary search) • ταξινόμηση <ul style="list-style-type: none"> ο γρήγορη ταξινόμηση (quick sort) ο ταξινόμηση με συγχώνευση (merge sort) <p>10.2.2 Ανάλυση χειρότερης περίπτωσης</p> <p>10.2.3 Ασυμπτωτικός συμβολισμός</p> <p>10.3.4 Βασικές ασυμπτωτικές κλάσεις αποδοτικότητας</p>	<ul style="list-style-type: none"> • Να δοθούν- συζητηθούν παραδείγματα που αναδεικνύουν την σχέση χώρου-χρόνου στην εκτέλεση αλγορίθμων. • Πρακτικό πρόβλημα, π.χ. password cracking με αργή κρυπτογράφηση (25 φορές DES) σε Python (βήματα: δημιουργία υποψήφιου συνθηματικού, κρυπτογράφηση, σύγκριση με αποθηκευμένο) <ul style="list-style-type: none"> ο εξαντλητική αναζήτηση (exhaustive search) - online (χρόνος) ο αναζήτηση με προϋπολογισμένους πίνακες (dictionary) - offline (χώρος) ο Αξιολόγηση/Σύγκριση • Μελέτη περίπτωσης- Υπολογισμός βημάτων για την ανάλυση αλγορίθμων αναζήτησης και ταξινόμησης ως προς το χρόνο εκτέλεσης. • Μελέτη επαναληπτικού και αναδρομικού αλγορίθμου της δυαδικής αναζήτησης • Παρουσίαση και συζήτηση για κλασικά προβλήματα και υπολογισμός πολυπλοκότητας αντίστοιχων αλγορίθμων (π.χ. γρήγορη ταξινόμηση, αλγόριθμος του Ευκλείδη κλπ).

<ul style="list-style-type: none"> • Να περιγράψουν τις διαφορές ανάμεσα στον Διαδικαστικό και τον Αντικειμενοστρεφή προγραμματισμό. • Να αξιοποιούν τεχνικές του αντικειμενοστρεφούς προγραμματισμού. • Να ορίζουν και να χρησιμοποιούν κλάσεις καθώς και ιδιότητες και μεθόδους. • Να ξεχωρίζουν την έννοια της κλάσης από εκείνη του αντικειμένου. • Να αντιλαμβάνονται το ρόλο της κληρονομικότητας και του πολυμορφισμού στη δημιουργία επαναχρησιμοποιήσιμου κώδικα. 	<p>11. Αντικειμενοστρεφής Προγραμματισμός [10 (9-11) + 6 (5-7) Ω]</p> <p>11.1 Αντικείμενα και κλάσεις</p> <p>11.2 Στιγμιότυπα (αυτόματη αρχικοποίηση αντικειμένων)</p> <p>11.3 Ιδιότητες και μέθοδοι</p> <p>11.4 Κληρονομικότητα και πολυμορφισμός</p> <p>11.5 Ενθυλάκωση και απόκρυψη πληροφορίας</p>	<ul style="list-style-type: none"> • Μέσω ενδεικτικού παραδείγματος, να αναδειχθούν οι έννοιες του αντικειμενοστρεφούς προγραμματισμού π.χ η κλάση Όχημα (Αυτοκίνητα, αεροπλάνα, πλοία) ή η κλάση Ζώο (Θηλαστικά, πουλιά κ.λπ). • Να εξηγηθεί, με τη χρήση παραδειγμάτων σε Python, ο τρόπος ορισμού κλάσεων, ιδιοτήτων και μεθόδων και ως “μαύρα κουτιά” να εισαχθεί η έννοια των υποκλάσεων και της κληρονομικότητας. • Οι μαθητές να εξασκηθούν στη δημιουργία κλάσεων και στη χρήση αντικειμένων μέσα από παραδείγματα από την καθημερινή ζωή που είναι κοντά στα βιώματά τους.
<ul style="list-style-type: none"> • Να διακρίνουν τα πλεονεκτήματα χρήσης ολοκληρωμένου περιβάλλοντος ανάπτυξης. • Να περιγράψουν τα εργαλεία που περιλαμβάνει ένα ολοκληρωμένο περιβάλλον ανάπτυξης. • Να συγγράφουν ποιοτικό κώδικα ακολουθώντας κανόνες ορθής συγγραφής (ονόματα μεταβλητών, εσοχές, τεκμηρίωση). • Να εφαρμόζουν τεχνικές αποσφαλμάτωσης προγραμμάτων με χρήση εργαλείων ενός περιβάλλοντος ανάπτυξης. 	<p>12. Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης λογισμικού – IDE [4 (3-5) + 2 (1-3) Ω]</p> <p>12.1 Περιβάλλοντα - Εργαλεία</p> <p>12.2 Ορθή συγγραφή κώδικα</p> <p>12.2 Τεχνικές Αποσφαλμάτωσης</p>	<ul style="list-style-type: none"> • Αναζήτηση και οργάνωση πληροφοριών για την έννοια του Ολοκληρωμένου Περιβάλλοντος Ανάπτυξης (Integrated Development Environment- IDE). Συμπλήρωση πίνακα σχετικά με το ρόλο, τη χρησιμότητά τους και τα μέρη που αποτελούνται. • Σύναψη συμβολαίου κανόνων συγγραφής ποιοτικού κώδικα λαμβάνοντας υπόψη διεθνή πρότυπα. • Σύγκριση, ως προς τις δυνατότητες, διαφορετικών προγραμματιστικών περιβαλλόντων για τη γλώσσα Python. • Να δοθούν παραδείγματα προγραμμάτων στα οποία να εντοπιστούν και να συζητηθούν περιπτώσεις συντακτικών και λογικών λαθών.
<ul style="list-style-type: none"> • Να χρησιμοποιούν περιβάλλοντα, Application Program Interfaces (APIs) και βιβλιοθήκες για την ανάπτυξη και τροποποίηση εφαρμογών λογισμικού. • Να ενισχύσουν την δημιουργική τους σκέψη, να 	<p>13. Εφαρμογές σε Γλώσσα Προγραμματισμού με χρήση API [2 (1-3) + 6 (5-7) Ω]</p>	<ul style="list-style-type: none"> • Παρουσίαση - επεξήγηση της Διεπαφής Προγραμματισμού Εφαρμογών (Application Programming Interface - API). • Αναζήτηση έτοιμων APIs υπηρεσιών κοινωνικής δικτύωσης τα οποία

<p>ελέγξουν μεγάλο όγκο πληροφοριών και τις διασυνδέσεις τους μέσω της οπτικοποίησης δεδομένων</p> <ul style="list-style-type: none"> • Να αναγνωρίσουν την συμβολή της πληροφορικής στην επίλυση μεγάλων επιστημονικών διαθεματικών προβλημάτων μέσω της οπτικοποίησης δεδομένων. • Να αποκτήσουν εμπειρία στην συγγραφή και διαχείριση ενός σύνθετου, διαθεματικού έργου. 		<p>επιτρέπουν την ανάπτυξη εφαρμογών ανάσχυσης από αυτές στοιχείων.</p> <ul style="list-style-type: none"> • Ανάπτυξη εφαρμογής όπως: <ul style="list-style-type: none"> • Εφαρμογή οπτικοποίησης δεδομένων. • Γραφική Διεπαφή Χρήστη (GUI). • Διαδικτυακή εφαρμογή.
<ul style="list-style-type: none"> • Να περιγράψουν τι είναι ένα Πληροφοριακό Σύστημα (ΠΣ). • Να απαριθμούν τα συστατικά στοιχεία ενός ΠΣ. • Να γνωρίσουν τις κατηγορίες εφαρμογής ενός ΠΣ και τις περιοχές που αξιοποιείται. • Να κατονομάζουν και να αναλύουν τις φάσεις και τα στάδια ανάπτυξης ενός ΠΣ. • Να υιοθετήσουν μια ευρύτερη οπτική για τα προβλήματα και τις λύσεις-προγράμματα χωρίς να εστιάζουν μόνο στο επίπεδο των τεχνικών χαρακτηριστικών, αλλά λαμβάνοντας υπόψη τον πραγματικό κόσμο (επιχειρήσεις, διοίκηση κλπ). 	<p>14. Πληροφοριακά Συστήματα [8 (7-9) + 0 Ω]</p> <p>14.1 Τύποι Πληροφοριακών Συστημάτων</p> <p>14.2 Μέρη – συνιστώσες ΠΣ (Υλικό - συστήματα, Λογισμικό, Δεδομένα, Διαδικασίες, Ανθρώπινο δυναμικό)</p> <p>14.3 Ο κύκλος ζωής ανάπτυξης ΠΣ</p> <p>14.4 Θέματα διαχείρισης ΠΣ</p> <ul style="list-style-type: none"> • Γενικές αρχές στην ανάπτυξη ΠΣ • Θέματα ασφάλειας πληροφοριών • Ο ρόλος των βάσεων δεδομένων στα Πληροφοριακά Συστήματα Διοίκησης • Ο ανθρώπινος παράγοντας στην αποδοχή ενός ΠΣ 	<ul style="list-style-type: none"> • Να αναζητήσουν εργαζόμενοι ομαδοσυνεργατικά Πληροφοριακά Συστήματα σε διάφορες περιοχές, όπως: <ul style="list-style-type: none"> ο συστήματα υποστήριξης αποφάσεων ο συστήματα διαχείρισης γνώσης • Να συζητηθούν τα πλεονεκτήματα χρήσης ΠΣ. • Να διακρίνουν τις βασικές οντότητες και τις σχέσεις μεταξύ τους σε ένα ΠΣ για μια θεματική περιοχή, όπου να αναδεικνύονται οι συνιστώσες του, όπως το Λογισμικό, το Υλικό, το ανθρώπινο δυναμικό και οι ρόλοι του. • Με παιχνίδι ρόλων να συνεργαστούν σε μια απλή μελέτη περίπτωσης για τη δημιουργία μίας μικρής μελέτης σκοπιμότητας για την ανάπτυξη ενός οικείου ΠΣ στους μαθητές.

Ενότητες ΠΣ

1. Εισαγωγή στην Υπολογιστική Σκέψη

1.1 Λέξεις Κλειδιά

Υπολογιστική Σκέψη, Επίλυση Προβλήματος, Ανάλυση προβλήματος, Υποπροβλήματα, Αφαίρεση (abstraction), Αναγνώριση Προτύπων (patterns), Μετασχηματισμός, Μοντελοποίηση, Γενίκευση.

1.2 Προτεινόμενος Χρόνος

Τέσσερις (4) διδακτικές ώρες προόδου ύλης.

1.3 Σκοπός-Στόχοι

Μετά το τέλος της ενότητας οι μαθητές πρέπει να είναι σε θέση:

- Να περιγράφουν τι περιλαμβάνει η έννοια της υπολογιστικής σκέψης (ΥΣ).
- Να περιγράφουν τη σημασία της ΥΣ στην επίλυση προβλημάτων της καθημερινής ζωής.
- Να περιγράφουν τα χαρακτηριστικά της υπολογιστικής σκέψης.
- Να αναλύουν ένα υπολογιστικό πρόβλημα περιγράφοντας τις βασικές διαδικασίες με τις οποίες αντιμετωπίζεται (διάσπαση, αναγνώριση προτύπων, γενίκευση και σχεδίαση αλγόριθμου).

1.4 Πλαίσιο

Σύμφωνα με τους Cuny, Snyder, Wing (2010) η υπολογιστική σκέψη αφορά «στις διεργασίες της σκέψης που εμπλέκονται στη μορφοποίηση προβλημάτων και των λύσεών τους έτσι ώστε οι λύσεις να μπορούν να αναπαρασταθούν σε μορφή που μπορεί αποτελεσματικά να υλοποιηθούν από έναν επεξεργαστή πληροφορίας». Η Wing (2010), που έκανε δημοφιλή την ιδέα της υπολογιστικής σκέψης στην ευρύτερη εκπαιδευτική κοινότητα (Wing 2006), επεξηγεί ότι «η υπολογιστική σκέψη περιγράφει τη νοητική δραστηριότητα μορφοποίησης ενός προβλήματος ώστε να επιδέχεται υπολογιστικές λύσεις. Οι λύσεις αυτές μπορεί να υλοποιηθούν είτε από τον άνθρωπο είτε από τη μηχανή ή το συνδυασμό ανθρώπων και μηχανών μαζί». Η υπολογιστική σκέψη είναι μια προσέγγιση επίλυσης προβλήματος που ενισχύει το συνδυασμό των ψηφιακών τεχνολογιών με τις ανθρώπινες ιδέες. Δεν αντικαθιστά την έμφαση στη δημιουργία, στη λογική και στη κριτική σκέψη, αλλά αντίθετα τονίζει αυτές τις δεξιότητες μέσα από την ανάδειξη τρόπων για την οργάνωση ενός προβλήματος, ώστε να μπορεί να υποστηριχθεί η επίλυσή τους από υπολογιστές.

Η υπολογιστική σκέψη περιλαμβάνει, αλλά δεν περιορίζεται, τα παρακάτω χαρακτηριστικά (Selby and Woollard, 2014):

- Μορφοποίηση προβλημάτων με τρόπο ώστε να επιτρέπει τη χρήση υπολογιστή και άλλων εργαλείων για την επίλυσή τους.
- Εννοιολογική αφαίρεση και επιλογή της απαραίτητης πληροφορίας για την αντιμετώπιση της πολυπλοκότητας των σύνθετων προβλημάτων.
- Διάσπαση ενός προβλήματος σε απλούστερα υποπροβλήματα.
- Λογική οργάνωση και ανάλυση δεδομένων.
- Αναπαράσταση δεδομένων μέσα από μοντέλα και προσομοιώσεις.
- Αυτοματοποίηση των λύσεων μέσα από την αλγοριθμική σκέψη.

- Αναγνώριση, ανάλυση και υλοποίηση πιθανών λύσεων με σκοπό την επίτευξη των αποτελεσματικότερων και αποδοτικότερων συνδυασμών βημάτων και πόρων.
- Γενίκευση και μεταφορά της διαδικασίας επίλυσης του συγκεκριμένου προβλήματος σε μια ευρύτερη ποικιλία από προβλήματα.

1.5 Διδακτικές Προτάσεις

Ο εκπαιδευτικός αναπτύσσει στους μαθητές τη διερευνητική και συμμετοχική διάθεση, δίνοντάς τους προβλήματα προς επίλυση που είναι κοντά στα βιώματά τους από την καθημερινή ζωή και σχετίζονται με διάφορα γνωστικά αντικείμενα. Μέσα από μια δημιουργική διαδικασία, η διδασκαλία πρέπει να προάγει την ανάπτυξη της κριτικής σκέψης των μαθητών, προτρέποντάς τους να αναπτύσσουν πρωτοβουλίες, να πειραματίζονται και να διερευνούν. Μέσα από κατάλληλα επιλεγμένες μελέτες περίπτωσης, οι μαθητές ενθαρρύνονται να κατανοήσουν το προς επίλυση πρόβλημα, να το αναλύσουν σε απλούστερα, να εντοπίσουν τα απαραίτητα δεδομένα, να το μορφοποιήσουν κατάλληλα, να συνθέσουν τη λύση του και στη συνέχεια να τη γενικεύσουν ώστε να επιλύουν γρήγορα νέα προβλήματα. Παράλληλα ο εκπαιδευτικός καλλιεργεί το πνεύμα συνεργασίας καθώς και την αυτοπεποίθηση των μαθητών για την επίλυση σύνθετων προβλημάτων μαθαίνοντάς τους να εμμένουν στη λύση τους.

Οι διδακτικές τεχνικές που μπορούν να χρησιμοποιηθούν για την προσέγγιση των εννοιών της ενότητας είναι: η μελέτη περίπτωσης για την επίλυση προβλημάτων, οι ερωτήσεις, ο καταγιγισμός ιδεών, το παιχνίδι ρόλων, η συζήτηση, η δημιουργία μοντέλων, η οπτικοποίηση της ανάλυσης προβλήματος και της συλλογιστικής με διαγράμματα (πίνακες, εννοιολογικούς χάρτες, διαγράμματα οργάνωσης δομής). Η διδακτική τεχνική της εισήγησης προτείνεται να περιοριστεί στον ελάχιστο χρόνο για την παρουσίαση ή ολοκλήρωση της ενότητας και των νέων εννοιών.

Είναι σημαντικό να εμπλουτιστεί το μάθημα με ποικίλες δραστηριότητες διαφορετικού τύπου. Οι προτεινόμενες δραστηριότητες πρέπει να έχουν νόημα για τους μαθητές, να έχουν σχέση με την κουλτούρα τους, τις εμπειρίες και τις ευρύτερες γνώσεις τους. Σε κάθε διαδικασία μάθησης τα άτομα είναι επιλεκτικά ως προς το τι και το πώς θα μάθουν. Οι μαθητές φιλτράρουν, επεξεργάζονται και οργανώνουν τα ερεθίσματα που δέχονται, προσπαθώντας να βρουν νόημα σε αυτό που αντιλαμβάνονται. Τα κίνητρα, οι αντιλήψεις και οι γνώσεις που ήδη έχουν είναι καθοριστικοί παράγοντες για τη μάθηση.

Στο συνολικό διαθέσιμο χρόνο πρέπει να ληφθεί υπόψη ο απαραίτητος χρόνος για τη γνωριμία του εκπαιδευτικού με την τάξη στα πρώτα μαθήματα. Με την έναρξη των μαθημάτων, ο εκπαιδευτικός πρέπει να ανιχνεύσει τις ατομικές διαφορές στη μάθηση για κάθε μαθητή, τις τυχόν μαθησιακές δυσκολίες ή προβλήματα που μπορεί να αντιμετωπίζουν ορισμένοι μαθητές, καθώς και τις πρότερες γνώσεις τους, ώστε να τις λάβει υπόψη του στο σχεδιασμό των μαθημάτων. Η εικόνα της τάξης μπορεί να συμπληρωθεί αντλώντας πληροφορίες από το σύλλογο των διδασκόντων για το γνωστικό επίπεδο των μαθητών, καθώς και για το ευρύτερο κοινωνικό πλαίσιο που είναι ενταγμένο το σχολείο. Με βάση τα παραπάνω στοιχεία ο εκπαιδευτικός θα μπορέσει να κτίσει τη διδακτική προσέγγιση για την παρουσίαση των νέων εννοιών, επεκτείνοντας τις πρότερες γνώσεις και αναπαραστάσεις των μαθητών ή τροποποιώντας κατάλληλα τυχόν παρανοήσεις τους.

1.6 Χώρος

Εργαστήριο Πληροφορικής – Αίθουσα.

1.7 Εποπτικά Μέσα-Διδακτικά Μέσα

Υπολογιστές, Πρόσβαση στο Διαδίκτυο, Πίνακας Μαρκαδόρου ή Διαδραστικός πίνακας, Βιντεοπροβολέας ή Επιδιασκόπιο (Overhead Projector), Λογισμικό Υπολογιστικών Φύλλων,

Διαδικτυακά εργαλεία συνεργατικής γραφής wikis της ηλεκτρονικής σχολικής τάξης, κατάλληλα λογισμικό εννοιολογικής χαρτογράφησης και υπολογιστικών φύλλων.

1.8 Ενδεικτικές Δραστηριότητες

1.8.1 Δραστηριότητα 1: Διαχείριση προβλημάτων

Για τα προβλήματα που δίνονται στη συνέχεια να συμπληρωθεί η ακόλουθη πληροφορία:

Κατανόηση του προβλήματος:

- Ποια δεδομένα είναι γνωστά;
- Τι δεν είναι γνωστό;
- Ποιο είναι το ζητούμενο;
- Ποιες είναι οι συνθήκες;
- Να σχεδιαστεί η λύση
- Ποιο το πλάνο εργασίας για την επίλυση του προβλήματος.

Υλοποίηση της λύσης

- Χρησιμοποιώντας το πλάνο εργασίας, να παρουσιαστεί η όλη εργασία και η λύση.

Να γίνει ανακεφαλαίωση και συζήτηση της λύσης.

Να γίνει καταγραφή των συλλογισμών που έγιναν πάνω στη λύση αυτή.

Πρόβλημα 1. Εύρεση των κάλπικων νομισμάτων με μια ζύγιση

Υπάρχουν δέκα σακιά που περιέχουν 100 νομίσματα το καθένα. Το κάθε νόμισμα ζυγίζει 10 γραμμάρια. Το ένα από τα δέκα σακιά έχει μέσα μόνο κάλπικα νομίσματα τα οποία ζυγίζουν 9 γραμμάρια το καθένα. Πώς μπορούμε με μία μόνο ζύγιση σε μία ηλεκτρονική ζυγαριά ακριβείας, να βρούμε ποιο σακί περιέχει τα κάλπικα νομίσματα;

Πρόβλημα 2. Το πρόβλημα της Χειραψίας

Ας υποθέσουμε ότι είκοσι άνθρωποι βρίσκονται μαζί με έναν μαθητή σε ένα δωμάτιο και πρέπει ο μαθητής να ανταλλάξει χειραψία με κάθε έναν από αυτούς. Με πόσους ανθρώπους τελικά θα ανταλλάξει χειραψία; Εάν υπάρχουν N ($N > 0$) άνθρωποι μαζί με το μαθητή στο δωμάτιο, με πόσους τελικά θα έρθει αυτός σε επαφή;

1.8.2 Δραστηριότητα 2: Μελέτη προβλήματος «Ασφαλής πλοήγηση στο Διαδίκτυο, σύνταξη ενός δεκάλογου συμβουλευτικών οδηγιών».

Οι μαθητές χωρισμένοι σε ομάδες των πέντε ατόμων αναλαμβάνουν να μελετήσουν ένα πρόβλημα ως μελέτη περίπτωσης. Για παράδειγμα να μελετήσουν το πρόβλημα: «Ασφαλής πλοήγηση στο Διαδίκτυο – δημιουργία ενός δεκάλογου συμβουλευτικών οδηγιών». Για το σκοπό αυτό καλούνται να συλλέξουν πληροφορίες, να επιλέξουν την ουσιώδη πληροφορία, να αξιοποιήσουν στατιστικά δεδομένα, να οργανώσουν τα δεδομένα που εντόπισαν. Για τη συλλογή και την οργάνωση των δεδομένων αξιοποιούν τους υπολογιστές του εργαστηρίου Πληροφορικής και το Διαδίκτυο. Στη συνέχεια αναλύουν το πρόβλημα σε απλούστερα προβλήματα, συντάσσουν τον συμβουλευτικό οδηγό και παρουσιάζουν την εργασία τους στη τάξη.

Για την υποστήριξη της συνεργασίας των ομάδων μπορεί να αξιοποιηθεί ένα Διαδικτυακό εργαλείο συνεργατικής γραφής. Για παράδειγμα μπορεί να αξιοποιηθεί το εργαλείο wiki της ηλεκτρονικής σχολικής τάξης (<http://eclass.sch.gr/>). Οι μαθητές οργανώνουν το υλικό τους και δημιουργούν ένα κοινό έγγραφο που χωρίζεται σε δύο μέρη: Το πρώτο μέρος περιέχει την ανάλυση του προβλήματος και το δεύτερο το δεκάλογο των συμβουλευτικών οδηγιών.

Προτεινόμενες Πηγές:

Δικτυακός κόμβος του συνηγόρου του παιδιού (<http://www.0-18.gr>). Δείτε τη «συνταγή» για ασφαλή πλοήγηση στο Διαδίκτυο, όπως προέκυψε από το ηλεκτρονικό φόρουμ της Κοινότητας Εφήβων Συμβούλων του Συνηγόρου του Παιδιού <http://www.0-18.gr/gia-paidia/nea/syntagi-gia-asfali-ploigisi-sto-internet> (τελευταία πρόσβαση 12/2014).

Ελληνικό Κέντρο Ασφαλούς Διαδικτύου με δραστηριότητες, άρθρα, εκπαιδευτικούς πόρους, προτάσεις (<http://www.saferinternet.gr>).

Ασφάλεια στο Διαδίκτυο: Ενημερωτικός κόμβος Πανελληνίου Σχολικού Δικτύου (<http://internet-safety.sch.gr/>).

1.8.3 Δραστηριότητα 3: Ανάλυση σύνθετου υπολογιστικού προβλήματος

Οι μαθητές, χωρισμένοι σε μικρές ομάδες των τριών ατόμων, καλούνται να αναλύσουν ένα σύνθετο υπολογιστικό πρόβλημα που συνδέεται με τις εμπειρίες τους και επηρεάζει την καθημερινή τους ζωή. Για παράδειγμα το πρόβλημα της πρόγνωσης του καιρού. Αρχικά προσπαθούν να χωρίσουν το πρόβλημα σε απλούστερα. Με την καθοδήγηση του εκπαιδευτικού μορφοποιούν κατάλληλα ένα υπολογιστικό φύλλο, ώστε να αναλύσουν τις βασικές κατηγορίες και τα δεδομένα που πιστεύουν ότι είναι απαραίτητα για τη επίλυση του προβλήματος.

Εκτός από το υπολογιστικό φύλλο για την καταγραφή της συλλογιστικής των μαθητών και την αναπαράσταση των κατηγοριών του προβλήματος μπορεί να χρησιμοποιηθούν εργαλεία εννοιολογικής χαρτογράφησης (όπως το Cmap-tool <http://cmaptools.en.softonic.com/>)

1.8.4 Δραστηριότητα 4: Επιλογή συντομότερης διαδρομής

Οι μαθητές χωρισμένοι σε μικρές ομάδες των δύο ατόμων αναλύουν ένα καθημερινό πρόβλημα που συνδέεται με τις εμπειρίες τους. Για παράδειγμα, η επιλογή της συντομότερης διαδρομής που πρέπει να επιλέξουν για να μεταβούν από το σπίτι τους στη πλατεία της περιοχής τους ή στο σχολείο τους. Οι μαθητές αναλύουν τα δεδομένα, που πρέπει να λάβουν υπόψη τους για την επίλυση του προβλήματος.

Οδηγίες

Ο εκπαιδευτικός οπτικοποιεί το πρόβλημα στο πίνακα με κατάλληλη γραφική αναπαράσταση (επίδειξη ενός απλού γράφου με τους βασικούς κόμβους και τις αντίστοιχες ακμές με βάρη) χρησιμοποιώντας ένα παράδειγμα. Για οικονομία του διδακτικού χρόνου προτείνεται το παράδειγμα και το αντίστοιχο διάγραμμα να έχει ετοιμαστεί ήδη και να παρουσιαστεί με τη βοήθεια ενός εποπτικού μέσου (βιντεοπροβολέα).

Για την υλοποίηση της δραστηριότητας μπορεί να αξιοποιηθεί ένας ψηφιακός ή έντυπος χάρτης της περιοχής. Οι μαθητές αναγνωρίζουν τους βασικούς κόμβους που μεσολαβούν από την αφετηρία μέχρι να φτάσουν στον τελικό προορισμό τους, τις ακμές μεταξύ δύο κόμβων και τα βάρη τους. Σημειώνονται πάνω στο χάρτη οι βασικοί κόμβοι με μια πινέζα (ψηφιακή στη περίπτωση ενός ψηφιακού χάρτη). Με ένα εργαλείο υπογράμμισης σημειώνονται οι ακμές που συνδέουν τους κόμβους και τα μονοπάτια που δημιουργούνται. Με ένα εργαλείο μέτρησης της απόστασης οι μαθητές μετρούν τη κάθε ακμή και σημειώνουν τις αντίστοιχες αποστάσεις (βάρη). Στη συνέχεια προσθέτουν τα βάρη για κάθε μονοπάτι που σχηματίζεται από την αφετηρία μέχρι το τελικό προορισμό και συγκρίνουν τα τελικά αθροίσματα, ώστε να βρουν τη συντομότερη διαδρομή. Με φυσική γλώσσα με βήματα περιγράφουν τη διαδικασία επίλυσης του προβλήματος και γενικεύουν τη λύση.

1.8.5 Δραστηριότητα 5: Διαδικασία λήψης απόφασης για το σχεδιασμό ενός εκπαιδευτικού παιχνιδιού

Ανάπτυξη συζήτησης στην τάξη με θέμα τη διαδικασία λήψης αποφάσεων για τον σχεδιασμό ενός εκπαιδευτικού παιχνιδιού. Οι μαθητές αρχικά διατυπώνουν με σαφήνεια το πρόβλημα και αναλύουν τις βασικές συνιστώσες του. Στη συνέχεια περιγράφουν με απλά βήματα τη διαδικασία λήψης των αποφάσεων τους, λαμβάνοντας υπόψη διάφορες μεταβλητές (γνωστές και άγνωστες) όπως: πολυπλοκότητα του παιχνιδιού, διαθέσιμοι πόροι, αναμενόμενος χρόνος για την ολοκλήρωσή του, αριθμός διαθέσιμων ανθρώπων που θα εμπλακούν στον σχεδιασμό, είδος του παιχνιδιού, γνωστικά αντικείμενα που θα προσεγγίζει το παιχνίδι κ.ά.

1.9 Βιβλιογραφία

Cuny, J., Snyder, L., & Wing, J.M. (2010). *Demystifying computational thinking for noncomputer scientists*. Δημοσίευτη μελέτη, αναφέρεται στο <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf> (τελευταία προσπέλαση 23/12/2014).

Report of a Workshop on the Scope and Nature of Computational Thinking, National Research Council, (2010) <http://www8.nationalacademies.org/cp/projectview.aspx?key=48969> (τελευταία προσπέλαση 22/12/2014).

Selby, Cynthia, Dorling, Mark and Woollard, John (2014) Evidence of assessing computational thinking. *Brookes eJournal of Learning and Teaching*, 1-12. <http://eprints.soton.ac.uk/366152/> (τελευταία προσπέλαση 28/12/2014).

Wing, J. (2010). Research Notebook: Computational Thinking - What and Why? *The Link*. Pittsburgh, PA: Carneige Mellon. Διαθέσιμο: <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf> (τελευταία προσπέλαση 23/12/2014).

Wing, J. (March 2006) "Computational Thinking," *Communications of the ACM, CACM* vol. 49, no. 3, pp. 33, <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf> (τελευταία προσπέλαση 23/12/2014).

1.10 Πηγές-Πρόσθετο Υλικό - Χρήσιμοι Εκπαιδευτικοί Πόροι

Computer Science Teacher Association- Εκπαιδευτικοί Πόροι για την Υπολογιστική Σκέψη: <http://csta.acm.org/Curriculum/sub/CompThinking.html> (τελευταία προσπέλαση 2/01/2015).

Δικτυακός τόπος της Google για την εξερεύνηση της υπολογιστικής σκέψης με εκπαιδευτικούς πόρους και συνδέσμους για πρόσθετο υλικό για τους εκπαιδευτικούς (Exploring Computational Thinking website (www.google.com/edu/computational-thinking)) (τελευταία προσπέλαση 2/01/2015).

Δικτυακός τόπος για τη διδασκαλία της επιστήμης της Πληροφορικής χωρίς υπολογιστές Computer Science Unplugged: <http://www.csunplugged.org> (τελευταία προσπέλαση 2/01/2015).

Κέντρο για την Υπολογιστική σκέψη του Carnegie Mellon. Διάφοροι προτεινόμενοι σύνδεσμοι με πρόσθετο εκπαιδευτικό υλικό για την υπολογιστική σκέψη: <http://www.cs.cmu.edu/~CompThink/resources/education.html> (τελευταία προσπέλαση 2/01/2015).

Εκπαιδευτικά παιχνίδια με θέμα την υπολογιστική σκέψη και την επίλυση προβλήματος: <http://games.thinkingmyself.com/> (τελευταία προσπέλαση 2/01/2015).

2. Υπολογιστής Γενικού Σκοπού: Υλικό και Λογισμικό

2.1 Λέξεις Κλειδιά:

Αρχιτεκτονική Μηχανής (ISA), Μικρο-αρχιτεκτονική, Λογικά Κυκλώματα, Ηλεκτρονικά Στοιχεία, Μοντέλο von Neumann, Τρόποι αναπαράστασης των δεδομένων.

2.2 Προτεινόμενος Χρόνος

Δέκα (10) διδακτικές ώρες: Έξι (6) ώρες κάλυψη της ύλης και τέσσερις (4) ώρες εμβάθυνση.

2.3 Σκοπός-Στόχοι

Μετά το τέλος της ενότητας οι μαθητές πρέπει να είναι σε θέση:

- Να περιγράψουν με τη βοήθεια ενός διαστρωματικού αφαιρετικού μοντέλου μελέτης τα βασικά στοιχεία και τις λειτουργίες που απαιτούνται για την επίλυση ενός προβλήματος μέσω υπολογιστή.
- Να εξηγούν την οργάνωση της κλασικής von Neumann μηχανής και των κύριων λειτουργικών μονάδων της.
- Να επεξηγούν το ρόλο και τους περιορισμούς του υλικού και του λογισμικού στην επίλυση προβλημάτων σε ένα υπολογιστικό σύστημα.
- Να περιγράψουν τον τρόπο που τα δεδομένα (ακέραιοι, πραγματικοί, χαρακτήρες, κλπ) αναπαριστώνται σε έναν υπολογιστή.
- Να αντιστοιχίζουν τους τύπους δεδομένων με ανάλογες εντολές ISA.

2.4 Πλαίσιο

Η ενότητα αυτή έχει ως στόχο να λειτουργήσει ως βασικό υπόβαθρο για τις ενότητες της αλγοριθμικής σκέψης και του προγραμματισμού που ακολουθούν. Για την κατανόηση των προγραμματιστικών δομών είναι ιδιαίτερα σημαντικό οι μαθητές να έχουν οικοδομήσει ένα ολοκληρωμένο εννοιολογικό μοντέλο των βασικών στοιχείων και των διαδικασιών επεξεργασίας ενός υπολογιστικού συστήματος. Η ανάπτυξη προγραμμάτων που επιλύουν υπολογιστικά προβλήματα απαιτεί ένα συνδυασμό γνώσεων που δεν περιορίζονται μόνο στα πλαίσια των συντακτικών κανόνων χρήσης μιας γλώσσας προγραμματισμού.

Οι μαθητές έχουν δημιουργήσει πολλές φορές λανθασμένες αντιλήψεις για τη λειτουργία του υπολογιστή ως υπολογιστικό σύστημα, δίνοντάς του αρκετές φορές ανθρωπόμορφες ιδιότητες, ως προς το τρόπο επεξεργασίας (ο «εγκέφαλος» του υπολογιστή) ή την αποθήκευση των δεδομένων στη μνήμη. Αυτές οι αντιλήψεις εμφανίζονται αρκετές φορές ως εμπόδια στην κατανόηση νέων εννοιών στον προγραμματισμό. Για παράδειγμα, οι μαθητές δεν αντιλαμβάνονται την ανάγκη της αναλυτικής περιγραφής μιας εντολής, σύμφωνα με τους κανόνες της γλώσσας, πιστεύοντας ότι ο υπολογιστής μπορεί να πραγματοποιεί νοητικά «άλλατα» και να την καταλάβει, όπως ο ανθρώπινος συμπληρώνοντας τα όποια λογικά κενά. Άλλη συχνή παρανόηση, είναι η αντίληψη των μαθητών για τη μεταβλητή στον προγραμματισμό, καθώς την ταυτίζουν συχνά με αυτήν από το χώρο των Μαθηματικών, χωρίς σαφή προσδιορισμό των χαρακτηριστικών που αυτή έχει.

Με την ενότητα αυτή προτείνεται η οικοδόμηση (συμπλήρωση ή τροποποίηση) ενός απλοποιημένου μοντέλου της λειτουργίας του υπολογιστή ως υπολογιστικό σύστημα, που θα βοηθήσει τους μαθητές στην ανάπτυξη αναλυτικής σκέψης για την επίλυση προβλημάτων προγραμματίζοντας τη μηχανή. Για την οικοδόμηση αυτού του μοντέλου, θα πρέπει οι μαθητές να

κατανοήσουν τις λειτουργίες που εκτελούνται στο εσωτερικό του υπολογιστή κατά την εκτέλεση ενός προγράμματος, έχοντας ως βάση ότι ο υπολογιστής είναι μια ψηφιακή μηχανή που λειτουργεί με ηλεκτρικό ρεύμα που τα πάντα μετατρέπονται σε μια σειρά από 0 και 1.

Αξίζει να τονιστεί ότι δεν αποτελεί σκοπό της ενότητας η λεπτομερής ανάπτυξη της αρχιτεκτονικής Μηχανής ISA, η εμβάθυνση στα ψηφιακά κυκλώματα ή στο μοντέλο von Neumann, αλλά η αναφορά στα βασικά δομικά στοιχεία και στις λειτουργίες, σε ένα πρώτο επίπεδο αφαίρεσης.

2.5 Διδακτικές Προτάσεις

Για την προσέγγιση και την κατανόηση των εννοιών της ενότητας ο εκπαιδευτικός αρχικά χρησιμοποιεί δραστηριότητες αφόρμησης για την κινητοποίηση της ενεργούς συμμετοχής των μαθητών, καθώς και για την ανίχνευση και στη συνέχεια σύνδεση των πρότερων γνώσεων τους με τις νέες έννοιες. Με σύντομη εισήγηση παρουσιάζει τις βασικές έννοιες της ενότητας. Στη συνέχεια προτείνεται να χρησιμοποιήσει ποικίλες διδακτικές τεχνικές ανάλογα με το γνωστικό υπόβαθρο των μαθητών και την έννοια που πρέπει να προσεγγιστεί. Ανάμεσα στις διδακτικές τεχνικές που προτείνονται είναι:

- Επίδειξη, με τη χρήση εποπτικών μέσων. Για παράδειγμα η επίδειξη των τυπικών επιπέδων αφαίρεσης ενός υπολογιστικού συστήματος ή η επίδειξη του κύκλου εκτέλεσης μιας εντολής (π.χ. $c=a+b$).
- Παρουσίαση χαρακτηριστικών παραδειγμάτων. Μέσα από παραδείγματα αναδεικνύονται και εξηγούνται οι τρόποι αναπαράστασης και αποθήκευσης των δεδομένων, η αναπαράσταση των ακεραίων στο δυαδικό, δεκαδικό, οκταδικό και δεκαεξαδικό σύστημα, οι πράξεις δυαδικών ακεραίων με πρόσημο, οι μετατροπές βάσης. Με χρήση Hexdump παρουσιάζεται η αναπαράσταση του περιεχομένου διαφόρων τύπων αρχείων.
- Διερεύνηση μέσω προσομοίωσης του τρόπου λειτουργίας των βασικών υποσυστημάτων της αρχιτεκτονικής Von Neumann και της αναπαράστασης των δεδομένων. Για το σκοπό αυτό μπορεί να αξιοποιηθούν ψηφιακές διαδραστικές εφαρμογές από το Διαδίκτυο, όπως μαθησιακά αντικείμενα από το «Ψηφιακό Σχολείο», ή εκπαιδευτικό λογισμικό που είναι διαθέσιμο για την εκπαιδευτική κοινότητα και προσεγγίζει κατάλληλα τις έννοιες της ενότητας (για παράδειγμα το εκπαιδευτικό λογισμικό «Δ.Ε.Λ.Υ.Σ» ή το εκπαιδευτικό λογισμικό «ΤΡΙΤΩΝ» από την εκπαιδευτική πύλη του Υ.ΠΑΙ.Θ –«e-yliko», για περισσότερα δείτε τις προτεινόμενες πηγές-εκπαιδευτικούς πόρους στο τέλος της ενότητας).
- Δημιουργία ή συμπλήρωση διαγραμμάτων ή/και εννοιολογικών χαρτών για την αναπαράσταση αφαιρετικών μοντέλων του υπολογιστικού συστήματος και την οργάνωση της κλασικής von Neumann μηχανής.
- Ερωταποκρίσεις για ανατροφοδότηση και κινητοποίηση των μαθητών.
- Σύντομες δραστηριότητες αναζήτησης, συλλογής και οργάνωσης ψηφιακών εκπαιδευτικών πολυμεσικών πόρων και πληροφοριών σχετικές με τις έννοιες της ενότητας. Για την οργάνωση της διαδικασίας και τη συντόμευση του απαραίτητου διδακτικού χρόνου για την υλοποίηση της δραστηριότητας, το υλικό μπορεί να έχει δομηθεί σε μια ιστοεξερεύνηση ή σε ένα Wiki (η ηλεκτρονική σχολική τάξη του Πανελληνίου Σχολικού Δικτύου <http://eclass.sch.gr/> προσφέρει μεταξύ άλλων εργαλείο Wiki, συζήτηση ομάδων, διαχείριση εκπαιδευτικού υλικού και υπερσυνδέσμων). Οι μαθητές συμμετέχουν στις δραστηριότητες σε μικρές ομάδες, επιλέγουν το υλικό, το οργανώνουν και το παρουσιάζουν σύντομα στην τάξη. Ακολουθεί συζήτηση και ανακεφαλαίωση των κυριότερων σημείων.
- Μικρές ασκήσεις με κατάλληλα διαμορφωμένα φύλλα εργασίας, καθώς και απαντητικά φύλλα ανατροφοδότησης.

Σημαντική είναι η αξιοποίηση ποικίλου οπτικοποιημένου υλικού (βίντεο, σχεδιαγράμματα, κινούμενες ψηφιακές εικόνες), ώστε οι μαθητές να συμπληρώσουν ή να τροποποιήσουν μέσα από την οπτική αναπαράσταση τα νοητικά τους μοντέλα για τις λειτουργίες που εκτελούνται στο εσωτερικό του υπολογιστή και την αναπαράσταση των δεδομένων.

2.6 Χώρος

Εργαστήριο Πληροφορικής, Αίθουσα.

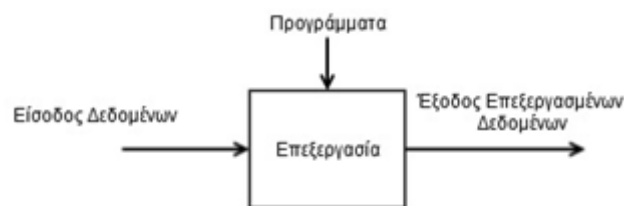
2.7 Εποπτικά Μέσα-Διδακτικά Μέσα

Υπολογιστές με πρόσβαση στο Διαδίκτυο, Διαδραστικός πίνακας ή Πίνακας Μαρκαδόρου, Βιντεοπροβολέας ή Επιδιασκόπιο (Overhead Projector), Εκπαιδευτικό λογισμικό (π.χ. ΔΕΛΥΣ), κατάλληλα διαμορφωμένα φύλλα εργασίας.

2.8 Ενδεικτικές Δραστηριότητες

2.8.1 Δραστηριότητα 1: Αρχιτεκτονική von Neumann

Δίνεται ένα διάγραμμα επεξεργασίας δεδομένων για προγραμματιζόμενο επεξεργαστή δεδομένων, με την επεξεργασία να προσεγγίζεται ως «μαύρο» αδιάφανο κουτί.



Οι μαθητές, σε μικρές ομάδες των τριών ατόμων, καλούνται να διερευνήσουν το υπολογιστικό σύστημα με βάση την αρχιτεκτονική von Neumann, ώστε να είναι σε θέση να περιγράψουν τα βασικά υποσυστήματα και τη διαδικασία επεξεργασίας. Κάθε ομάδα αναλαμβάνει να βρει ψηφιακούς πόρους για την αρχιτεκτονική von Neumann. Οι μαθητές αναζητούν μαθησιακά αντικείμενα στο «Ψηφιακό Σχολείο», την εκπαιδευτική πύλη ή άλλα αποθετήρια με ψηφιακούς εκπαιδευτικούς πόρους. Ανάμεσα στα μαθησιακά αντικείμενα μπορούν να είναι σύντομα βίντεο, αλληλεπιδραστικές εφαρμογές προσομοίωσης, ψηφιακά μοντέλα αναπαράστασης - σχεδιαγράμματα των βασικών υποσυστημάτων της αρχιτεκτονικής von Neumann. Κάθε ομάδα οργανώνει το υλικό που συνέλεξε και στη συνέχεια το παρουσιάζει στην τάξη. Αναπτύσσεται συζήτηση για τα βασικά στοιχεία της αρχιτεκτονικής και τη βασική λειτουργία της. Συμπληρώνεται κατάλληλα το αρχικό σχήμα, ώστε να είναι εμφανή τα βασικά στοιχεία της διαδικασίας επεξεργασίας. Η δραστηριότητα ολοκληρώνεται με ανακεφαλαίωση από τον εκπαιδευτικό των βασικών σημείων.

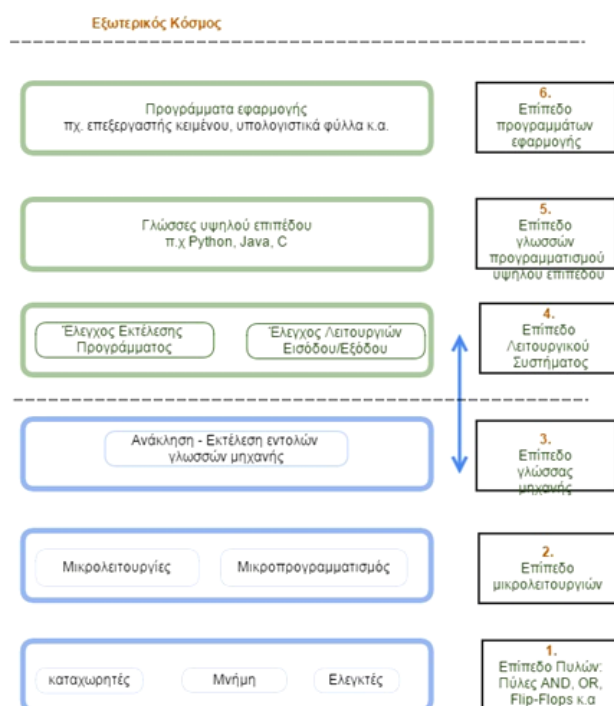
Ενδεικτική ερώτηση εμπέδωσης: Οι μαθητές καλούνται να απαντήσουν και να τεκμηριώσουν την απάντησή τους στην ερώτηση, αν ένας υπολογιστής μπορεί να λειτουργήσει χωρίς κύρια μνήμη.

Σημείωση: Δεν αποτελεί στόχο η λεπτομερής αναφορά στη λειτουργία του μοντέλου της αρχιτεκτονικής von Neumann, αλλά σε ένα επίπεδο αφαίρεσης, η περιγραφή των βασικών υποσυστημάτων.

2.8.2 Δραστηριότητα 2: Επίπεδα ιεραρχίας ενός υπολογιστικού συστήματος

Παρουσίαση, μέσω σχηματικής παράστασης, των επιπέδων ιεραρχίας ενός υπολογιστικού συστήματος. Οι μαθητές χωρίζονται σε έξι ομάδες. Κάθε ομάδα αναλαμβάνει από ένα επίπεδο και στη συνέχεια συλλέγει και οργανώνει σχετικό ψηφιακό υλικό και πληροφορίες από το Διαδίκτυο, ενώ ένας εκπρόσωπος της ομάδας παρουσιάζει το υλικό στη τάξη. Οι ομάδες ενώνουν το υλικό σε ένα κοινό αρχείο παρουσίασης, ώστε να έχουν ολοκληρωμένη εικόνα για το υπολογιστικό σύστημα. Στη συνέχεια, ο εκπαιδευτικός επιλύει τυχόν απορίες και με ερωταποκρίσεις ελέγχει το βαθμό κατανόησης από τους μαθητές. Στο τέλος της δραστηριότητας ανακεφαλαιώνει τα κυριότερα σημεία.

Για την υποστήριξη της συνεργασίας των ομάδων μπορεί να αξιοποιηθεί ένα Διαδικτυακό εργαλείο συνεργατικής γραφής. Οι μαθητές οργανώνουν το υλικό τους και δημιουργούν ένα κοινό έγγραφο με διαγράμματα, πληροφορίες και υπερσυνδέσεις σε ένα Wiki. Για παράδειγμα μπορεί να αξιοποιηθεί το εργαλείο wiki της ηλεκτρονικής σχολικής τάξης (<http://eclass.sch.gr/>).



Εναλλακτική διδακτική πρόταση προσέγγισης: Ο εκπαιδευτικός παρουσιάζει σύντομα τα επίπεδα του μοντέλου του υπολογιστικού συστήματος στους μαθητές χωρίς να αναφερθεί στην ιεραρχία τους. Στη συνέχεια με την βοήθεια βιντεοπροβολέα και ενός εργαλείου εννοιολογικής χαρτογράφησης (ή οποιουδήποτε άλλου προγράμματος που δημιουργεί εύκολα ιεραρχικά διαγράμματα) καλεί τους μαθητές, με καταιγισμό ιδεών, να τοποθετήσουν πάνω στο γράφημα τα επίπεδα με τη σειρά τεκμηριώνοντας την άποψη τους για τις επιλογές τους. Παρουσιάζεται οπτικοποιημένη η τελική μορφή του μοντέλου και αναπτύσσεται συζήτηση για τη δομή του.

2.8.3 Δραστηριότητα 3: Αναπαράσταση δεδομένων στον υπολογιστή

Οι μαθητές, με κατάλληλα διαμορφωμένα φύλλα εργασίας, αξιοποιούν προσομοιώσεις, διαδραστικές εφαρμογές και εκπαιδευτικό λογισμικό από το Διαδίκτυο, όπως το εκπαιδευτικό λογισμικό που είναι διαθέσιμο στην εκπαιδευτική πύλη εκπαιδευτική πύλη του Υ.ΠΑΙ.Θ (e-γίλιο) και στο αποθετήριο μαθησιακών αντικειμένων του Ψηφιακού Σχολείου (Φωτόδεντρο).

Ενδεικτικά παραδείγματα

Παράδειγμα 1: Αξιοποίηση του εκπαιδευτικού λογισμικού “ΔΕΛΥΣ” που είναι διαθέσιμο στην εκπαιδευτική πύλη e-γλικο (<http://tinyurl.com/o4hktsc>). Για την οπτικοποίηση της διαδικασίας επεξεργασίας των δεδομένων και της αναπαράστασή τους στον υπολογιστή, οι μαθητές πειραματίζονται με μικρές ασκήσεις-δραστηριότητες με την ενότητα «Επεξεργασία δεδομένων» του λογισμικού. Συγκεκριμένα με κατάλληλα διαμορφωμένο φύλλο εργασίας προτείνεται στους μαθητές να πειραματιστούν με τις εφαρμογές που παρέχει το λογισμικό στην ενότητα για την αποθήκευση/ανάκτηση χαρακτήρα, τη δυαδική παράσταση αριθμών, την πρόσθεση αριθμών και τις λογικές πύλες. Για την καλύτερη λειτουργία του λογισμικού προτείνεται να εκτελείται απευθείας από CD χωρίς να γίνει εγκατάσταση του στους υπολογιστές του εργαστηρίου.

Παράδειγμα 2: Αξιοποίηση του εκπαιδευτικού λογισμικού “ΤΡΙΤΩΝ” από την εκπαιδευτική πύλη e-γλικο (<http://tinyurl.com/lwxsucb>). Για την οπτικοποίηση της αρχιτεκτονικής μηχανής ISA οι μαθητές πειραματίζονται με έτοιμα παραδείγματα με την εφαρμογή «ΑΒΑΚΑΣ» (προσομοίωση της λειτουργίας υπολογιστή) του εκπαιδευτικού λογισμικού “ΤΡΙΤΩΝ”. Για παράδειγμα χρησιμοποιώντας τον συμβολομεταφραστή της εφαρμογής φορτώνουν ένα έτοιμο πρόγραμμα (απόλυτη τιμή) και το μετατρέπουν σε γλώσσα μηχανής σε δυαδικό κώδικα. Στη συνέχεια μεταφέρουν τον κώδικα στη μνήμη του «ΑΒΑΚΑ» και παρακολουθούν βήμα προς βήμα την εκτέλεση των εντολών. Ο εκπαιδευτικός καθοδηγεί τους μαθητές όπου χρειαστεί και επεξηγεί τα βασικά σημεία της διαδικασίας εκτέλεσης.

Παράδειγμα 3: Αξιοποίηση της προσομοίωσης «Μυθική Μηχανή» και του μεταγλωττιστή που τη συνοδεύει για μετατροπή απλών εντολών Python σε κώδικα assembly. Η προσομοίωση είναι γραμμένη σε python και αποτελεί μέρος του έργου Openbook Python for Fun <http://www.openbookproject.net/courses/python4fun/simulator.html>

2.8.4 Δραστηριότητα 4: Μετατροπή δεκαεξαδικών αριθμών σε δυαδικούς

Δίνονται ασκήσεις μετατροπής δεκαεξαδικών αριθμών στο δυαδικό σύστημα και αντίστροφα. Για την οπτικοποίηση της διαδικασίας και την κατανόησή της, ενδείκνυται η χρήση κατάλληλα διαμορφωμένου φύλλου εργασίας ή διαμορφωμένου υπολογιστικού φύλλου σε έντυπη ή προτιμότερο σε ηλεκτρονική μορφή.

2.9 Βιβλιογραφία

Du Boulay, B. (1989), Some difficulties of learning to program, In E. Saloway & J. C.Spohrer (Eds), Studying the Novice Programmer, Hillsdale, NJ, Lawrence Erlbaum Associates, 283-299.

Hayes ,P., J.(1998) “Computer Architecture and Organization”, McGraw–Hill, Third edition

Patterson, A., Hennessy, J. (1994) «Computer Organization & Design, The Hardware/ Software Interface», Morgan Kaufmann

Samurcay, R. (1989) “The Concept of variable in programming: Its meaning and use in problem solving by novice programmers, In E. Saloway & J. C.Spohrer (Eds), Studying the Novice Programmer.

Tanenbaum, A. (2000) «Η Αρχιτεκτονική των Υπολογιστών - Μια δομημένη προσέγγιση», Αθήνα, Εκδ. Κλειδάριθμος.

Mano, M. (1992) «Ψηφιακή Σχεδίαση», Εκδόσεις Παπασωτηρίου, 1992.

Νικολός, Δ. (2001) «Αρχιτεκτονική Υπολογιστών Ι», Πάτρα, Ε.Α.Π

Παπακωνσταντίνου Γ., Τσανάκας, Π., Φραγκάκης, Γ. (1989) “Αρχιτεκτονική Υπολογιστών”, Αθήνα, εκδ. Συμμετρία.

Patt, Y., Sanjay, P. (2003) "Introduction to Computing Systems: From bits & gates to C & beyond", Mc Graw-Hill Science, Computer Engineering series.

2.10 Πηγές - Πρόσθετο Υλικό - Χρήσιμοι Εκπαιδευτικοί Πόροι

- Ανοικτά Μαθήματα από το MIT: Computation Structures MIT OPEN COURSE WARE <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-004-computation-structures-spring-2009/index.htm> (τελευταία προσπέλαση 02/01/2015).
- Δικτυακός τόπος με συλλογή από εργαστηριακές ασκήσεις και προσομοιώσεων με Java applets για τα υπολογιστικά συστήματα και την αναπαράσταση δεδομένων <http://math.hws.edu/TMCM/java/index.html>(τελευταία προσπέλαση 02/01/2015).
- Δικτυακός τόπος με ποικίλο εκπαιδευτικό υλικό και υπερσυνδέσεις για τη προσέγγιση των δυαδικών αριθμών <http://csunplugged.org/binary-numbers> (τελευταία προσπέλαση 02/01/2015).
- Εκπαιδευτικό Λογισμικό ΔΕΛΥΣ- Διαδραστικό εκπαιδευτικό λογισμικό για Υπολογιστικά Συστήματα. Περιλαμβάνει μεταξύ άλλων μικρόκοσμος διερεύνησης της κωδικοποίησης δεδομένων. Διαθέσιμο για την εκπαιδευτική κοινότητα από την εκπαιδευτική πύλη του Υ.ΠΑΙ.Θ e-γliko: <http://tinyurl.com/o4hktsc> (τελευταία προσπέλαση 02/01/2015).
- Εκπαιδευτικό Λογισμικό «ΤΡΙΤΩΝ» διαθέσιμο στην εκπαιδευτική Πύλη του Υ.ΠΑΙ.Θ, e-γliko (<http://www.e-γliko.gr> Το λογισμικό "ΤΡΙΤΩΝ" προσεγγίζει μεταξύ άλλων θέματα Αρχιτεκτονικής Ηλεκτρονικών Υπολογιστών. Έχει δημιουργηθεί από το εργαστήριο Υπολογιστικών Συστημάτων του Ε.Μ.Π. Μεταξύ άλλων περιέχει την εφαρμογή προσομοίωσης "Αβακας" για την εσωτερική οργάνωση και λειτουργία του υπολογιστή. <http://tinyurl.com/lwxsucb> (τελευταία προσπέλαση 02/01/2015).
- Προσομοίωση υπολογιστικής μηχανής «Μυθική Μηχανή» και μεταγλώττισης εντολών ρυθον σε κώδικα assembly για τη «Μυθική Μηχανή». Η προσομοίωση είναι γραμμένη σε ρυθον και αποτελεί μέρος του έργου Openbook Python for Fun <http://www.openbookproject.net/courses/python4fun/simulator.html> (τελευταία προσπέλαση 03/01/2015).
- Πλατφόρμα ηλεκτρονικής σχολικής τάξης. Μεταξύ άλλων περιέχει εργαλεία για την ενορχήστρωση συνεργατικών δραστηριοτήτων, εργαλεία συνεργατικής γραφής (wikis) και οργάνωσης διαδικτυακών εκπαιδευτικών πόρων (<http://eclass.sch.gr/>).

2.11 Σημειώσεις ενότητας εκπαιδευτικού

.....
.....
.....
.....
.....
.....

3. Βασικά στοιχεία γλώσσας προγραμματισμού

3.1 Λέξεις Κλειδιά:

Περιβάλλοντα προγραμματισμού, γνωριμία με τη γλώσσα Python, Τύποι δεδομένων, Βασικές εντολές, Αριθμητικές και λογικές πράξεις, Βασικές έτοιμες μαθηματικές συναρτήσεις.

3.2 Προτεινόμενος Χρόνος

Έξι (6) διδακτικές ώρες: Δύο (2) ώρες διδασκαλία ύλης και τέσσερις (4) ώρες εμπάθунση.

3.3 Σκοπός-Στόχοι:

Μετά το τέλος της ενότητας οι μαθητές πρέπει να είναι σε θέση:

- Να περιγράφουν τους βασικούς τύπους δεδομένων και τις βασικές εντολές της γλώσσας προγραμματισμού.
- Να αναγνωρίζουν και να χρησιμοποιούν τις βασικές μαθηματικές συναρτήσεις γλώσσας προγραμματισμού.
- Να χρησιμοποιούν το διερμηνευτή γλώσσας προγραμματισμού για να κάνουν απλές πράξεις.
- Να χρησιμοποιούν το προγραμματιστικό περιβάλλον για σύνταξη κώδικα προγράμματος και μετέπειτα εκτέλεσής του.

3.4 Πλαίσιο

Βασικός σκοπός της ενότητας είναι η εξοικείωση των μαθητών με ένα προγραμματιστικό περιβάλλον μιας γλώσσας προγραμματισμού που χρησιμοποιείται ευρέως. Να εκτελούν σε αυτό απλές εντολές και μαθηματικές συναρτήσεις. Για το σκοπό αυτό προτείνεται ως προγραμματιστικό περιβάλλον, το περιβάλλον της γλώσσας Python (εδώ η έκδοση 2.7.9, με την παρουσία δύο κυρίαρχων εκδόσεων της γλώσσας Python, τις 2.7.x και 3.4.x κατά το χρόνο γραφής του οδηγού). Η επιλογή της έκδοσης αυτού του προγραμματιστικού περιβάλλοντος δεν είναι περιοριστική, αρκεί να είναι αποδεκτοί οι συντακτικοί κανόνες των προηγούμενων εκδόσεων της γλώσσας. Εξάλλου, σκοπός δεν είναι η εξάντληση των νέων δυνατοτήτων και βιβλιοθηκών που συνοδεύουν κάθε έκδοση, αλλά η μεταφορά μιας αλγοριθμικής λύσης ενός προβλήματος σε ένα σύγχρονο και ευρέως γνωστό προγραμματιστικό περιβάλλον για την εκτέλεσή της από ένα υπολογιστικό σύστημα.

Ο εκπαιδευτικός μπορεί να μεταφορτώσει και να εγκαταστήσει το προγραμματιστικό περιβάλλον από το δικτυακό τόπο υποστήριξης της γλώσσας Python <https://www.python.org/>. Αρχικά επιλέγει την έκδοση της γλώσσας και το λειτουργικό σύστημα που έχει ο υπολογιστής που θα εγκαταστήσει το περιβάλλον προγραμματισμού, στη συνέχεια το μεταφορτώνει και το εγκαθιστά. Το ολοκληρωμένο περιβάλλον προγραμματισμού ονομάζεται IDLE από τα αρχικά των λέξεων Integrated Development Environment - Ολοκληρωμένο Περιβάλλον Ανάπτυξης.

3.5 Διδακτικές Προτάσεις

Ο εκπαιδευτικός με σύντομη επίδειξη εξηγεί στους μαθητές:

- τους βασικούς τύπους δεδομένων (int, float και bool)
- τις εντολές εξόδου και εκχώρησης τιμής
- τους τελεστές (αριθμητικοί, συγκριτικοί, λογικοί) και τις εκφράσεις (αριθμητικές, λογικές)
- την προτεραιότητα των πράξεων

- τις βασικές μαθηματικές συναρτήσεις της βιβλιοθήκης math της Python (fabs, exp, log, sqrt, cos, sin, tan) και της συνάρτησης type().

Παράλληλα, με κατάλληλα διαμορφωμένα φύλλα εργασίας, αναθέτει στους μαθητές απλές ασκήσεις για πειραματισμό με το προγραμματιστικό περιβάλλον. Οι μαθητές πειραματίζονται αρχικά με το Python Shell που είναι μέρος του ολοκληρωμένου περιβάλλοντος προγραμματισμού. Το Python Shell ουσιαστικά επιτρέπει τη συγγραφή κάθε εντολής ξεχωριστά (η εντολή ακολουθεί μετά τα τρία σύμβολα >>>) και την άμεση εκτέλεσή της με τη βοήθεια του διερμηνευτή της γλώσσας. Για παράδειγμα, οι μαθητές μπορούν να ξεκινήσουν από τη σύνταξη της εντολής (print) για την εμφάνιση του κλασικού μηνύματος χαιρετισμού «Καλημέρα σε όλους» και να συνεχίσουν με την εκτέλεση απλών και σταδιακά πιο σύνθετων αριθμητικών πράξεων.

```
>>> print "Καλημέρα σε όλους"
```

```
Καλημέρα σε όλους
```

Για την επεξεργασία και την αποθήκευση ενός προγράμματος θα πρέπει να χρησιμοποιηθεί το περιβάλλον IDLE της Python. Γενικά η διαδικασία έχει ως εξής: Άνοιγμα νέου αρχείου (New File), σύνταξη του κώδικα του προγράμματος, αποθήκευση του προγράμματος (Save, όνομα αρχείου με κατάληξη .py), εκτέλεση του προγράμματος (από την επιλογή Run-Run Module), εμφάνιση των αποτελεσμάτων στο Python Shell.

Κατά τη διάρκεια της υλοποίησης των ασκήσεων ο εκπαιδευτικός συντονίζει τη διαδικασία, λειτουργεί ως διευκολυντής αλλά δεν παρεμβαίνει, αφήνοντας τους μαθητές να πειραματιστούν. Παρακινεί τους μαθητές να ψάχνουν τρόπους να κάνουν τα απλά προγράμματα να δουλέψουν και να παρακολουθούν την ανατροφοδότηση που παίρνουν από το προγραμματιστικό περιβάλλον. Με αυτό το τρόπο οι μαθητές προσεγγίζουν τις επιτυχημένες ή αποτυχημένες προσπάθειές τους όχι ως αμοιβή ή ποινή αλλά ως χρήσιμη πληροφορία. Η προσέγγιση αυτή συντελεί στην ανάπτυξη εσωτερικών κινήτρων στο μαθητή προκειμένου να εμπλακεί επιτυχώς στη μαθησιακή διαδικασία και να εμπλουτίσει σταδιακά το «ρεπερτόριο» της επικοινωνίας του με τον υπολογιστή μέσα από τη χρήση της γλώσσας Python.

Μετά την υλοποίηση κάθε άσκησης – δραστηριότητας οι μαθητές συμπληρώνουν τις απαντήσεις τους σε φύλλο εργασίας. Ο εκπαιδευτικός χρησιμοποιώντας τη τεχνική της συζήτησης καλεί τους μαθητές να παρουσιάσουν τις απαντήσεις και να συμμετάσχουν στην εξαγωγή χρήσιμων συμπερασμάτων, όπως για παράδειγμα για την προτεραιότητα των αριθμητικών πράξεων. Στο πίνακα της τάξης ή με την αξιοποίηση ενός βιντεοπροβολέα ή ενός overhead Projector δίνονται οι ορθές απαντήσεις και ανακεφαλαιώνονται τα κυριότερα σημεία.

Οι μεταβλητές και η εντολή εκχώρησης τιμής.

Η γλώσσα Python παρέχει εντυπωσιακές εναλλακτικές δυνατότητες έκφρασης για τη διαχείριση μεταβλητών, που διευκολύνουν τον προγραμματιστή. Για την χρησιμοποίηση μιας μεταβλητής δεν απαιτείται η δήλωσή της, ενώ μπορεί να εκχωρήσουμε διαφορετικούς τύπους τιμών (ακέραιες, κινητής υποδιαστολής, συμβολοσειρές, λίστα, αντικείμενο κ.ά.). Στη πραγματικότητα η Python διαχειρίζεται τη μεταβλητή ως αντικείμενο.

Οι δυνατότητες αυτές σε συνδυασμό με την ευκολία σύνταξης μιας εντολής και τη δυνατότητα άμεσης διερμηνείας της δημιουργούν ένα γόνιμο περιβάλλον πειραματισμού και διερεύνησης για τους μαθητές μέσα από την υλοποίηση κατάλληλων δραστηριοτήτων. Παρόλα αυτά η ποικιλία διαφορετικών εκφράσεων σε συμβολικό επίπεδο μπορεί να αποτελέσουν εμπόδια για τον αρχάριο προγραμματιστή ως προς την κατανόηση της έννοιας της μεταβλητής αν αυτές δεν εισαχθούν από τον εκπαιδευτικό σταδιακά. Για παράδειγμα η γλώσσα Python παρέχει τις δυνατότητες:

Ανάθεσης τιμής σε περισσότερες από μία μεταβλητές με μια εντολή `metavliti1=metavliti2=metavliti3=15` ή ανάθεσης τιμών με τη σειρά διατυπώνοντας την έκφραση: `x, y=10, 18`. Στο παράδειγμα το `x` παίρνει τη τιμή 10 ενώ το `y` την τιμή 18. Η έκφραση `x, y, z = 10, 20, "Μυρτώ"` εκχωρεί στο `x` το 10, στο `y` το 20 και στο `z` τη συμβολοσειρά Μυρτώ. Άλλο χαρακτηριστικό παράδειγμα είναι η έκφραση **για την αντιμετάθεση τιμών μεταξύ δύο μεταβλητών: `x, y = y, x`**.

Για την καλύτερη κατανόηση της έννοιας της μεταβλητής και το τρόπο που η Python τη διαχειρίζεται προτείνεται ο εκπαιδευτικός να παρουσιάσει αρχικά πιο παραδοσιακούς τρόπους εκχώρησης (`x = 10`, `ονομα = "Μυρτώ"`) σε αντιστοιχία με τις εντολές του αντίστοιχου αλγορίθμου εκφρασμένου με ψευδοκώδικα (`x ← 10`) και στη συνέχεια να παρουσιάσει σταδιακά τις νέες δυνατότητες της γλώσσας.

3.6 Χώρος

Εργαστήριο Πληροφορικής.

3.7 Εποπτικά Μέσα-Διδακτικά Μέσα:

Υπολογιστές με εγκατεστημένο το ολοκληρωμένο περιβάλλον ανάπτυξης της γλώσσας Python (IDLE), Διαδραστικός πίνακας ή Πίνακας Μαρκαδόρου, Βιντεοπροβολέας ή Επιδιασκόπιο (Overhead Projector), κατάλληλα διαμορφωμένα φύλλα εργασίας.

3.8 Ενδεικτικές Δραστηριότητες

3.8.1 Δραστηριότητα 1: Ο διερμηνευτής της γλώσσας Python

Οι μαθητές πειραματίζονται στο διερμηνευτή της γλώσσας Python και εμφανίζουν βασικά μηνύματα χαιρετισμού με χρήση της εντολής `print`.

3.8.2 Δραστηριότητα 2: Ο διερμηνευτής της γλώσσας Python ως αριθμομηχανή

Οι μαθητές εργαζόμενοι ατομικά πειραματίζονται με τον διερμηνευτή της γλώσσας Python (Python Shell) χρησιμοποιώντας τον ως αριθμομηχανή. Μέσα από κατάλληλα διαμορφωμένο φύλλο εργασίας αρχικά εικάζουν τα αποτελέσματα της πράξης που τους ζητείται και επαληθεύουν το αποτέλεσμα χρησιμοποιώντας τη γλώσσα. Οι ερωτήσεις είναι διαβαθμισμένης δυσκολίας εισάγοντας τους μαθητές στις αριθμητικές πράξεις, στους αριθμητικούς τύπους δεδομένων, στην προτεραιότητα των πράξεων, σε απλές μαθηματικές συναρτήσεις.

Ενδεικτικό Φύλλο Εργασίας

Τάξη..... Τμήμα:.....

Ημερομηνία:.....

Όνομα:..... Επώνυμο:.....

Συμπλήρωσε κατάλληλα τον παρακάτω πίνακα. Στην αριστερή στήλη παρουσιάζεται μια πράξη που πρέπει να εκτελεστεί στον υπολογιστή χρησιμοποιώντας το προγραμματιστικό περιβάλλον της γλώσσας Python. Αρχικά να συμπληρωθεί η μεσαία στήλη με τα αποτελέσματα που πιστεύουμε ότι θα εμφανιστούν στην οθόνη του υπολογιστή μετά την εκτέλεση της πράξης. Να γίνει επαλήθευση των αποτελεσμάτων που συμπληρώθηκαν πληκτρολογώντας και εκτελώντας κάθε πράξη ξεχωριστά μέσα στο περιβάλλον της γλώσσας Python.

Πράξεις	Αποτέλεσμα που πιστεύουμε ότι θα εμφανιστεί	Τι εμφανίζεται τελικά στην οθόνη
2 + 3		
3560 – 130		
.....		
5 * 3 + 2		
5 * (3 + 2)		
5 * (3 + 2) / 10		
3 * 3.75 / 1.5		
.....		
5 ** 2		
3 ** 4		
.....		

Ερωτήσεις:

- Από τα παραπάνω μπορεί να διαμορφωθούν οι βασικοί κανόνες για την προτεραιότητα των πράξεων στη γλώσσα Python;
.....
- Ποιοι οι βασικοί αριθμητικοί τελεστές που χρησιμοποιούνται στην Python;
.....
- Χρησιμοποιώντας τη βοήθεια του περιβάλλοντος της γλώσσας Python βρείτε τις απλές μαθηματικές συναρτήσεις που παρέχει (όπως fabs, exp, log, sqrt, cos, sin, tan).
.....

3.8.3 Δραστηριότητα 4: Κανόνες σύνταξης στη γλώσσα Python

Εισαγωγή στους κανόνες σύνταξης της γλώσσας Python. Να προσεγγιστούν οι έννοιες:

- Μεταβλητές (variables), οι οποίες αποθηκεύουν τιμές για μελλοντική χρήση.
- Τύποι δεδομένων (data types), όπως αριθμοί (numbers) και λογικοί (booleans).
- Σχόλια που κάνουν τον κώδικα ευανάγνωστο.
- Αριθμητικοί τελεστές (Arithmetic operations), +, -, *, /, **, %
- Λογικοί τελεστές σύγκρισης ==, !=, <, >, <=, >=
- Τελεστές λογικών πράξεων not, or, and.

Μετά από πρακτική εξάσκηση:

- Να γράψετε μια απλή γραμμή σχολίων της αρεσκείας σας (προσοχή να ξεκινάει με #)

- Να οριστεί η μεταβλητή `arithmos` να περιέχει την τιμή 1.234 (αριθμός κινητής υποδιαστολής).
- Να οριστεί η μεταβλητή `arithmos_tetragwno` να περιέχει το τετράγωνο της `arithmos`.
- Να εμφανιστούν στην οθόνη οι τιμές των μεταβλητών `arithmos`, `arithmos_tetragwno`.
- Να οριστεί η μεταβλητή `logikh` να περιέχει τη τιμή `True`.
- Να οριστεί η μεταβλητή `paixnidia` να περιέχει την ακέραια τιμή 10.
- Να γίνουν λογικοί έλεγχοι για το περιεχόμενο της μεταβλητής `paixnidia`, με χρήση των λογικών τελεστών σύγκρισης σε συνδυασμό με τους τελεστές λογικών πράξεων (`paixnidia == 8`, `paixnidia > 8`,..., `paixnidia == 8 or paixnidia == 10`, `paixnidia == 9 and paixnidia == 10`).

```
>>> # ύψωσε σε δύναμη
>>> arithmos = 1.234
>>> arithmos_tetragwno = arithmos ** 2
>>> print arithmos, arithmos_tetragwno
```

Αποτέλεσμα στην οθόνη: 1.234 1.522756

```
>>> logikh = True
>>> print logikh
Αποτέλεσμα στην οθόνη: True
```

```
>>> paixnidia = 10
>>> paixnidia == 8
False
>>> paixnidia == 8 or paixnidia == 10
True
>>> not paixnidia == 8
True
>>> paixnidia >= 8
True
>>> paixnidia != 8
True
>>> paixnidia <= 8
False
>>>
```

3.8.4 Δραστηριότητα 4: Ασκήσεις χειρισμού μιγαδικών αριθμών

Οι μαθητές με χρήση του περιβάλλοντος προγραμματισμού IDLE μπορούν να πειραματιστούν και να υλοποιήσουν απλές ασκήσεις χειρισμού μιγαδικών αριθμών.

3.8.5 Δραστηριότητα 5: Υλοποίηση απλού προγράμματος σε Python

Οι μαθητές με χρήση του περιβάλλοντος προγραμματισμού IDLE πειραματίζονται και υλοποιούν απλά προγράμματα προγραμματισμού. Με κατάλληλα διαμορφωμένο φύλλο εργασίας δίνεται στους μαθητές ένα απλό πρόβλημα (για παράδειγμα η εύρεση του εμβαδού τριγώνου). Οι μαθητές σχεδιάζουν με απλά βήματα τη λύση του και στη συνέχεια τη μεταφέρουν με κατάλληλες εντολές στο προγραμματιστικό περιβάλλον. Ο εκπαιδευτικός αναπτύσσει θέματα σχετικά με τη μεταβλητή, τους βασικούς τύπους δεδομένων (αριθμοί, λογικοί), τους αριθμητικούς τελεστές, την εντολή εκχώρησης τιμής και εμφάνισης (print). Παράλληλα επεξηγεί τη σημασία της τήρησης των κανόνων σωστής συγγραφής και τεκμηρίωσης του κώδικα (προσθήκη σχολίων που τον κάνουν ευανάγνωστο, επιλογή κατάλληλων ονομάτων για τις μεταβλητές κ.α).

Να γραφεί αλγόριθμος που να υπολογίζει και να εκτυπώνει το εμβαδό τριγώνου βάσης 10 και ύψους 15. Το εμβαδό δίνεται από τον τύπο $E = (\beta * \upsilon) / 2$. Στη συνέχεια να γραφεί πρόγραμμα που να υλοποιεί τον αλγόριθμο εμβαδό τριγώνου σε γλώσσα Python.

Ψευδοκώδικας	Πρόγραμμα σε Python
Αλγόριθμος Εμβαδό_Τριγώνου ΒΑΣΗ ← 10 ΥΨΟΣ ← 15 ΕΜΒΑΔΟ ← (ΒΑΣΗ * ΥΨΟΣ) / 2 Εκτύπωσε «ΤΟ ΕΜΒΑΔΟ ΤΟΥ ΤΡΙΓΩΝΟΥ ΕΙΝΑΙ:», ΕΜΒΑΔΟ ΤΕΛΟΣ Εμβαδό_Τριγώνου	>>> # ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΕΜΒΑΔΟΥ ΤΡΙΓΩΝΟΥ >>> BASH = 10 >>> YPSOS = 15 >>> EMBADO = (BASH * YPSOS) / 2 >>> print 'ΤΟ ΕΜΒΑΔΟ ΤΟΥ ΤΡΙΓΩΝΟΥ ΕΙΝΑΙ: ', EMBADO Αποτέλεσμα στη οθόνη: ΤΟ ΕΜΒΑΔΟ ΤΟΥ ΤΡΙΓΩΝΟΥ ΕΙΝΑΙ: 75

Οδηγία σχετικά με την ονοματολογία μιας μεταβλητής: Η εισαγωγική δραστηριότητα μπορεί μεταξύ άλλων να αποτελέσει αφορμή για την ανάπτυξη συζήτησης σχετικά με την επιλογή χαρακτηριστικού ονόματος για μια μεταβλητή. Για παράδειγμα για τη μεταβλητή της βάσης του τριγώνου επιλέγουμε το όνομα ΒΑΣΗ. Συχνά οι μαθητές πιστεύουν ότι οι μεταβλητές που έχουν ένα χαρακτηριστικό όνομα δε μπορούν να αλλάξουν ρόλο και να χρησιμοποιηθούν για κάτι άλλο. Για το λόγο αυτό προτείνεται να διευκρινιστεί ότι η τακτική επιλογής χαρακτηριστικού ονόματος είναι ένας κανόνας καλής πρακτικής αλλά δεν απαιτείται από τη μηχανή. Οι μαθητές μπορούν να πειραματιστούν και στη μεταβλητή της βάσης να εκχωρήσουν την τιμή του ύψους και αντίστροφα και στη συνέχεια να ελέγξουν το αποτέλεσμα. Επιπρόσθετα ο εκπαιδευτικός μπορεί να παρουσιάσει τους κανόνες που ορίζει η γλώσσα Python για την επιλογή ονόματος μίας μεταβλητής.

3.9 Βιβλιογραφία

Αβούρης Ν. Κουκιάς Μ., Παλιουράς Β, Σγάρμπας Κ. (2013) «Εισαγωγή στους υπολογιστές με τη γλώσσα Python», Εκδόσεις Πανεπιστημίου Πατρών, Πάτρα.

Downey, A. (2012) «Think Python, How to think like a computer scientist», Ο' Reilly.

<http://www.greenteapress.com/thinkPython/>

Severance C., Python for Informatics: Exploring Information, 2013.

<http://www.Pythonlearn.com/book.php>

Swaroop, C., H. (2013) "A byte of Python", Διαδικτυακή έκδοση ηλεκτρονικού βιβλίου

<http://www.swaroopch.com/notes/Python/> με άδεια Creative Commons Attributions – ShareAlike 4.0 International License. Το βιβλίο έχει μεταφραστεί και στα Ελληνικά από την Ελληνική κοινότητα του Ubuntu. <http://ubuntu-gr.org/>

3.10 Πηγές-Πρόσθετο Υλικό- Χρήσιμοι Εκπαιδευτικοί Πόροι

Δικτυακός κόμβος υποστήριξης της γλώσσας Python, <https://www.Python.org/> (τελευταία προσπέλαση 3/01/2015)

Wiki από το δικτυακό κόμβο της γλώσσας Python με συνδέσμους σε εκπαιδευτικό υλικό, tutorials και εκπαιδευτικά βίντεο για την εκμάθηση της γλώσσας Python από αρχάριους προγραμματιστές χωρίς προηγούμενη εμπειρία στον προγραμματισμό.

<https://wiki.Python.org/moin/BeginnersGuide/NonProgrammers> (τελευταία προσπέλαση 3/01/2015)

Δικτυακός κόμβος ελληνικής κοινότητας προγραμματιστών Python. Περιέχει μεταξύ άλλων εκπαιδευτικό υλικό, συμβουλές, tutorials. <http://Python.org.gr/> (τελευταία προσπέλαση 3/01/2015).

Δικτυακός κόμβος υποστήριξης εκπαιδευτικών για τη διδασκαλία της γλώσσας Python, <http://www.Pythonschool.net/> (τελευταία προσπέλαση 02/01/2015)

Tutorial από το δικτυακό κόμβο της γλώσσας Python για την γνωριμία με τη γλώσσα Python – Χρησιμοποιώντας την Python ως αριθμομηχανή, <https://docs.Python.org/2/tutorial/introduction.html> (τελευταία προσπέλαση 3/01/2015)

Tutorial για την εκμάθηση της έκδοσης 2.7.x. της Python, <http://www.Python-course.eu/course.php> (τελευταία προσπέλαση 02/01/2015)

Tutorial για την εισαγωγή στη γλώσσα Python, <http://www.codecademy.com/tracks/Python> (τελευταία προσπέλαση 4/01/2015)

Σχετικό υλικό και χρήσιμοι σύνδεσμοι από την ελεύθερη εγκυκλοπαίδεια Βικιπαίδεια, <http://el.wikipedia.org/wiki/Python> (τελευταία προσπέλαση 4/01/2015)

4. Λογική οργάνωση δεδομένων κατά την αποθήκευσή τους

4.1 Λέξεις Κλειδιά:

Κύρια Μνήμη, Μεταβλητή, Στατικές και Δυναμικές Δομές Δεδομένων, Πίνακας, Λίστα, Τύποι Αρχείων, Κωδικοποίηση ανά τύπο αρχείου/εφαρμογής, Συμπύεση απωλεστική, συμπύεση μη-απωλεστική.

4.2 Προτεινόμενος Χρόνος

Οκτώ (8) διδακτικές ώρες. (4) ώρες προόδου της ύλης και (4) για εμπέδωση.

4.3 Σκοπός-Στόχοι

Μετά το τέλος της ενότητας οι μαθητές πρέπει να είναι σε θέση:

- Να περιγράψουν τον τρόπο αναπαράστασης της μεταβλητής στην κύρια μνήμη.
- Να περιγράψουν τον τρόπο αποθήκευσης τιμών στην κύρια μνήμη, ώστε να συνιστούν τη δομή του Πίνακα και της Λίστας.
- Να αναγνωρίζουν τους διαφορετικούς τύπους αρχείων και να τους αντιστοιχούν με τις αντίστοιχες εφαρμογές που τα δημιούργησαν.
- Να αναφέρουν και επεξηγούν τρόπους και μεθοδολογίες κωδικοποίησης και συμπύεσης δεδομένων σε αρχεία.

4.4 Πλαίσιο

Η ενότητα αυτή έχει ως σκοπό να λειτουργήσει ως βασικό υπόβαθρο για τις ενότητες της αλγοριθμικής σκέψης και του προγραμματισμού συμπληρώνοντας το μοντέλο λειτουργίας και αναπαράστασης των δεδομένων του υπολογιστή, που προσεγγίστηκε στη 2^η ενότητα «Υπολογιστής Γενικού Σκοπού: Υλικό και Λογισμικό». Οι μαθητές μέσα από την ενότητα αυτή έχουν την ευκαιρία να εμβαθύνουν στη λογική οργάνωση των δεδομένων που αποθηκεύονται στο υπολογιστικό σύστημα σε επίπεδο κύριας μνήμης, καθώς και στους διαφορετικούς τύπους αρχείων σε σχέση με τις εφαρμογές που τα δημιούργησαν και στις μεθοδολογίες κωδικοποίησης και συμπύεσης δεδομένων σε αρχεία.

Οι μαθητές εμφανίζεται να παρουσιάζουν δυσκολίες στην κατανόηση εννοιών της κύριας μνήμης και συγκεκριμένα:

- Στη λειτουργία της κύριας μνήμης μέσα σε έναν υπολογιστή.
- Στην οργάνωση της μνήμης και τη χρησιμότητα των διευθύνσεων μνήμης για την προσπέλαση στο περιεχόμενό της.
- Στην κατανόηση της προσωρινής αποθήκευσης (δεδομένων και προγραμμάτων) στην κύρια μνήμη.

Οι δυσκολίες αυτές δημιουργούν αρκετές παρανοήσεις στον προγραμματισμό για τη διαχείριση των μεταβλητών ή των δομών δεδομένων του πίνακα και της λίστας. Αναφορικά με τη μεταβλητή, παρότι οι μαθητές έχουν αντιμετωπίσει την έννοια και σε προηγούμενες τάξεις, παρατηρείται οι αντιλήψεις τους για τη μεταβλητή στο προγραμματισμό να μην είναι ξεκάθαρη και να ταυτίζουν τις ιδιότητες της με αυτές που έχει στα μαθηματικά. Πολλές φορές οι μαθητές έχουν οικοδομήσει ημιτελείς αναπαραστάσεις σχετικά με την μεταβλητή προσεγγίζοντας την για παράδειγμα ως ένα «κουτί που τοποθετούμε μέσα κάποιο περιεχόμενο» χωρίς να μπορούν να περιγράψουν με σαφήνεια τα χαρακτηριστικά της.

Επιπρόσθετα οι αναπαραστάσεις αυτές δημιουργούν προβλήματα κατανόησης για τον τύπο μιας μεταβλητής, τη διαδικασία εκχώρησης τιμής ή τη δυνατότητα επαναχρησιμοποίησής της μέσα σε ένα πρόγραμμα με τον ίδιο ή διαφορετικό ρόλο. Οι δυσκολίες αυτές εμφανίζονται συχνά σε πιο σύνθετες αλγοριθμικές δομές (επανάληψη, αναδρομή) ή όταν προσεγγίζονται θέματα εμβέλειας των μεταβλητών που χρησιμοποιούνται σε συναρτήσεις.

Με στόχο τη συμπλήρωση ή την τροποποίηση των νοητικών αναπαραστάσεων προτείνεται ως διδακτική στρατηγική αρχικά η οικοδόμηση ενός νοητικού μοντέλου για τη μνήμη του υπολογιστή και των διαδικασιών που πραγματοποιούνται κατά την εκτέλεση μιας απλής εντολής που σχετίζεται με τη μεταβλητή (εκχώρηση τιμής, εμφάνιση κ.ά). Η προσέγγιση πρέπει να διατηρείται σε ένα πρώτο επίπεδο αφαίρεσης παρουσιάζοντας με εμφανή τρόπο το όλον, χωρίς να δίνεται ιδιαίτερη έμφαση σε λεπτομερείς πληροφορίες από το χώρο της αρχιτεκτονικής υπολογιστών.

Στις επόμενες ενότητες, οι μαθητές θα έχουν την ευκαιρία να εμβαθύνουν με τις έννοιες της μεταβλητής, του πίνακα και της λίστας πειραματιζόμενοι με το προγραμματιστικό περιβάλλον. Εκμεταλλευόμενοι το γνωστικό τους υπόβαθρο που έχουν δομήσει από τις προηγούμενες ενότητες, μετασχηματίζουν και συμπληρώνουν τις ιδέες τους επιλύοντας προβλήματα και ασκήσεις με ανατροφοδότηση από τον υπολογιστή. Ο εκπαιδευτικός ανιχνεύει τυχόν γνωστικές παρανοήσεις, ενορχηστρώνει τις δραστηριότητες, καθοδηγεί και επεξηγεί όπου κρίνει ότι αυτό είναι σκόπιμο. Η αξιοποίηση τεχνικών και εργαλείων για την οπτικοποίηση των διαδικασιών (πίνακα μεταβλητών, οπτικοποιημένα βήματα προς βήματα εκτέλεση εντολών) μπορεί να συντελέσει σε αυτή την κατεύθυνση.

4.5 Διδακτικές Προτάσεις

Η ενότητα χωρίζεται σε δύο βασικές υποενότητες:

- Μεταβλητή, Πίνακες-Λίστες και Κύρια Μνήμη
- Τύποι αρχείων και συμπίεση δεδομένων.

4.5.1 Διδακτικές Προτάσεις: Μεταβλητή, Πίνακες-Λίστες και Κύρια Μνήμη

Εισαγωγή -Αφόρμηση

Ο εκπαιδευτικός αρχικά μέσα από ερωταποκρίσεις κινητοποιεί τους μαθητές και ανιχνεύει τις πρότερες γνώσεις τους. Με σύντομη εισήγηση παρουσιάζει τις βασικές έννοιες και κάνει σύνδεση με τις έννοιες που αναπτύχθηκαν στη δεύτερη ενότητα.

Μέρος μαθήματος κάλυψης ύλης

Κατά τη διάρκεια του μέρους του μαθήματος κάλυψης της ύλης θα πρέπει να προσεγγιστούν:

- Η μεταβλητή σε σχέση με τη δομή της κύριας μνήμης με αναφορά στις θέσεις μνήμης και στη διεύθυνσή τους.
- Η δομή του πίνακα ως στατική δομή και ο τρόπος αναπαράστασης και αποθήκευσης των περιεχομένων του (με σύντομη αναφορά στην τοπικότητα της αναφοράς για τα προγράμματα και τα δεδομένα).
- Η δομή της λίστας και ο τρόπος αναπαράστασης και αποθήκευσης των περιεχομένων της ως δυναμική δομή.

Για την ανάπτυξη του κυρίου μέρους του μαθήματος προτείνεται να χρησιμοποιηθούν ποικίλες διδακτικές τεχνικές ανάλογα με το γνωστικό υπόβαθρο των μαθητών και την έννοια που πρέπει να προσεγγιστεί. Ανάμεσα στις διδακτικές τεχνικές που προτείνονται είναι:

- Η επίδειξη με χρήση εποπτικών μέσων. Για παράδειγμα η επίδειξη με κατάλληλα απλοποιημένα σχεδιαγράμματα της κύριας μνήμης, των γραμμών διευθύνσεων, των γραμμών δεδομένων (προβάλλοντας απλοποιημένα οπτικά αναπαραστασιακά μοντέλα).
- Η παρουσίαση και η διερεύνηση απλών χαρακτηριστικών παραδειγμάτων για τη μεταβλητή, τους πίνακες, τις λίστες.
- Ερωταποκρίσεις για ανατροφοδότηση και κινητοποίηση των μαθητών.
- Σημαντική είναι η αξιοποίηση ποικίλου οπτικοποιημένου υλικού (βίντεο, σχεδιαγράμματα, κινούμενες ψηφιακές εικόνες, προσομοιώσεις) για την αναπαράσταση των εννοιών της ενότητας, ώστε οι μαθητές να συμπληρώσουν ή να τροποποιήσουν μέσα από την οπτικοποίηση τα νοητικά τους μοντέλα για τη κύρια μνήμη του υπολογιστή και τις βασικές λειτουργίες, που εκτελούνται στο εσωτερικό του υπολογιστή.

Εμβάθυνση-Εφαρμογή-Επεκτάσεις

Κατά τη διάρκεια της εφαρμογής στο σχολικό εργαστήριο Πληροφορικής προτείνεται η υλοποίηση ποικίλων δραστηριοτήτων για την ενεργή συμμετοχή των μαθητών, την ανάπτυξη πνεύματος συνεργασίας, τον πειραματισμό. Μέσα από δραστηριότητες οι μαθητές προτείνεται να εισάγονται στο αυθεντικό πρόβλημα της αναπαράστασης της πληροφορίας.

Προτεινόμενες Δραστηριότητες:

Σύντομες δραστηριότητες αναζήτησης, συλλογής και οργάνωσης ψηφιακών εκπαιδευτικών πολυμεσικών πόρων και πληροφοριών σχετικές με τις έννοιες της ενότητας. Οι μαθητές συμμετέχουν στις δραστηριότητες σε μικρές ομάδες, επιλέγουν το υλικό, το οργανώνουν και το παρουσιάζουν σύντομα στη τάξη. Ακολουθεί συζήτηση και ανακεφαλαίωση των κυριότερων σημείων.

Δραστηριότητες προσομοίωσης. Ο εκπαιδευτικός μπορεί να αξιοποιήσει το λογισμικό υπολογιστικών φύλλων, κατάλληλες ψηφιακές εφαρμογές προσομοίωσης από το Διαδίκτυο ή εκπαιδευτικό λογισμικό που είναι διαθέσιμο για την εκπαιδευτική κοινότητα (για παράδειγμα το «ΤΡΙΤΩΝ» εφαρμογή «ΑΒΑΚΑΣ» διαθέσιμο στην εκπαιδευτική πύλη «e-γliko» του Υ.ΠΑΙ.Θ). Στις προτεινόμενες Πηγές-Πρόσθετο Υλικό- Χρήσιμοι Εκπαιδευτικοί Πόροι στο τέλος της ενότητας δίνονται πληροφορίες και σχετικοί υπερσύνδεσμοι. Ο εκπαιδευτικός θα πρέπει να σχεδιάσει την αξιοποίησή τους και να ελέγξει τους προτεινόμενους υπερσυνδέσμους πριν τους χρησιμοποιήσει στη τάξη. Για την αποτελεσματικότερη αξιοποίησή τους προτείνεται να τις εντάξει στο διδακτικό σχεδιασμό του σύμφωνα με το γνωστικό υπόβαθρο των μαθητών, την υλικοτεχνική υποδομή του εργαστηρίου, το διαθέσιμο διδακτικό χρόνο.

Δραστηριότητες απλής εφαρμογής ανάθεσης τιμών (ακεραίων, κινητής υποδιαστολής, strings και λίστες) σε μεταβλητές και εμφάνισή τους στο προγραμματιστικό περιβάλλον της Python (εδώ, μια και ο διδακτικός χρόνος είναι περιορισμένος ο εκπαιδευτικός μπορεί να δώσει περισσότερη έμφαση στην 6^η ενότητα).

4.5.2 Ενδεικτικές Δραστηριότητες: Κύρια Μνήμη-Μεταβλητές-Πίνακες-Λίστες

4.5.2.1 Δραστηριότητα 1^η: Δραστηριότητα αντιμετάθεσης μεταβλητών με αξιοποίηση υπολογιστικών φύλλων.

Σκοπός: Η προσέγγιση της μεταβλητής μέσα από μια απλοποιημένη προσομοίωση σε ένα πρώτο αφαιρετικό επίπεδο με υπολογιστικό φύλλο για τη θέση και τη διεύθυνση της μεταβλητής στην κύρια μνήμη του υπολογιστή.

Ο εκπαιδευτικός με κατάλληλα διαμορφωμένο φύλλο εργασίας αναθέτει σε μικρές ομάδες των δύο ατόμων να πραγματοποιήσουν την αντιμετάθεση δύο ακέραιων τιμών που περιέχονται σε δύο μεταβλητές x και y με τη βοήθεια του υπολογιστικού φύλλου.

Έστω ότι έχουμε τη μεταβλητή x με περιεχόμενο τον ακέραιο αριθμό 12 και τη μεταβλητή y με περιεχόμενο τον ακέραιο αριθμό 20. Χρησιμοποιώντας ένα υπολογιστικό φύλλο αντιστοιχήστε τη μεταβλητή x στη θέση μνήμης (κελί) με διεύθυνση B(2) και εκχωρήστε (στο κελί) την τρέχουσα ακέραια τιμή της 12 (περιεχόμενο). Παρόμοια αντιστοιχήστε τη μεταβλητή y στη θέση μνήμης (κελί) με διεύθυνση B(3) και εκχωρήστε (στο κελί) την τρέχουσα ακέραια τιμή 20 (περιεχόμενο). Προσπαθήστε να αντιμεταθέσετε τις τιμές αυτές ώστε η μεταβλητή x (στη θέση B(2)) να πάρει τη τιμή της μεταβλητής y και η μεταβλητή y (στη θέση B(3)) την τιμή της μεταβλητής x . Στη συνέχεια περιγράψτε τη διαδικασία με φυσική γλώσσα με βήματα.

Οδηγία: Οι μαθητές αφού φθάσουν στη λύση, την περιγράφουν σε φυσική γλώσσα με βήματα, ενώ στη συνέχεια παρουσιάζουν τη λύση στη τάξη. Ο εκπαιδευτικός με χρήση εποπτικού μέσου παρουσιάζει τις αναλογίες και τις διαφορές που έχει η δραστηριότητα με τη διαχείριση μεταβλητών στην κύρια μνήμη, χρησιμοποιώντας ένα χαρακτηριστικό διάγραμμα απεικόνισης των διευθύνσεων και του περιεχομένου μιας περιοχής της κύριας μνήμης. Με ερωτήσεις κινητοποιεί τους μαθητές να εκφράσουν τις σκέψεις τους και διευκρινίζει τυχόν απορίες. Στο τέλος της δραστηριότητας γίνεται ανακεφαλαίωση των κυριότερων σημείων που αναπτύχθηκαν.

4.5.2.2 Δραστηριότητα 2^η: Αξιοποίηση εκπαιδευτικού λογισμικού και εφαρμογών προσομοιώσεων που είναι διαθέσιμα στο Διαδίκτυο

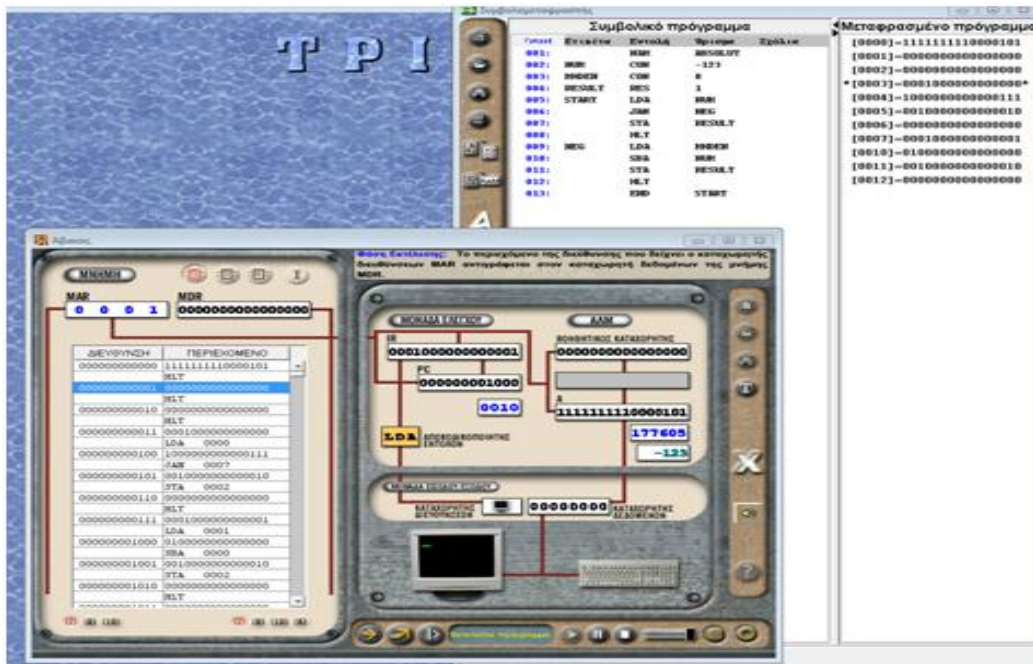
Οι μαθητές με κατάλληλα διαμορφωμένα φύλλα εργασίας αξιοποιούν προσομοιώσεις και διαδραστικές εφαρμογές και εκπαιδευτικό λογισμικό από το Διαδίκτυο, όπως το εκπαιδευτικό λογισμικό που είναι διαθέσιμο στην εκπαιδευτική εκπαιδευτική πύλη του Υ.ΠΑΙ.Θ και του αποθετηρίου μαθησιακών αντικειμένων (Φωτόδεντρο) του Ψηφιακού Σχολείου.

Παράδειγμα

Αξιοποίηση του εκπαιδευτικού λογισμικού “ΤΡΙΤΩΝ” από την εκπαιδευτική πύλη e-γίλιο (<http://tinyurl.com/lwxsucb>). Για την οπτικοποίηση των βασικών λειτουργιών της κύριας μνήμης, οι μαθητές χωρισμένοι σε μικρές ομάδες των δύο ατόμων πειραματίζονται με έτοιμα παραδείγματα με την εφαρμογή «ΑΒΑΚΑΣ» (προσομοίωση λειτουργίας υπολογιστή) του εκπαιδευτικού λογισμικού “ΤΡΙΤΩΝ”. Χρησιμοποιώντας το συμβολομεταφραστή της εφαρμογής φορτώνουν ένα έτοιμο πρόγραμμα (απόλυτη τιμή) και το μετατρέπουν σε γλώσσα μηχανής σε δυαδικό κώδικα. Στη συνέχεια πατώντας το κουμπί μεταφοράς από το πίνακα ελέγχου του συμβολομεταφραστή, μεταφέρουν τον κώδικα στη μνήμη του ΑΒΑΚΑ. Ο εκπαιδευτικός επεξηγεί τη διαδικασία και δίνει έμφαση στην περιγραφή και τις βασικές λειτουργίες της κύριας μνήμης όπως αυτή απεικονίζεται στο λογισμικό. Στο τέλος της δραστηριότητας ανακεφαλαιώνει τα κυριότερα σημεία.

Οδηγίες -Σημειώσεις

Αν δεν υπάρχει ο διαθέσιμος διδακτικός χρόνος, ο εκπαιδευτικός μπορεί να περιοριστεί στην επίδειξη της διαδικασίας με βιντεοπροβολέα, στην περιγραφή του τρόπου απεικόνισης στην κύρια μνήμη. Δεν αποτελεί στόχο της δραστηριότητας η αναλυτική περιγραφή και η διερεύνηση των λειτουργιών που εκτελούνται στην Κεντρική Μονάδα Επεξεργασίας. Ο εκπαιδευτικός αναπτύσσει γόνιμο διάλογο στην τάξη για τις αναλογίες και τις διαφορές που έχει η προσομοίωση με την πραγματική διαχείριση της κύριας μνήμης ως προς τις μεταβλητές, τις θέσεις μνήμης και τη διευθυνσιοδότηση. Για τον τρόπο χρήσης του λογισμικού του «ΑΒΑΚΑ» προτείνεται να μελετήσετε έγκαιρα το εγχειρίδιο χρήσης που συνοδεύει το λογισμικό «Τρίτων» σε ηλεκτρονική μορφή.



Σχήμα: Στιγμιότυπο από τον «ΑΒΑΚΑ» του λογισμικού «ΤΡΙΤΩΝ»

4.5.2.3 Δραστηριότητα 3^η: Άντληση και οργάνωση πληροφοριών και σχηματικών αναπαραστάσεων για την κύρια μνήμη του υπολογιστή και για τις δομές δεδομένων του πίνακα και της λίστας από το Διαδίκτυο.

Οι μαθητές χωρίζονται σε μικρές ομάδες των τριών ατόμων. Με κατάλληλα διαμορφωμένο φύλλο εργασίας καλούνται να αναζητήσουν από το Διαδίκτυο, πληροφορίες για την κύρια μνήμη του υπολογιστή, τη διευθυνσιοδότησή της, καθώς και για τους τρόπους αναπαράστασης και αποθήκευσης των περιεχομένων του πίνακα και τη λίστας. Οργανώνουν τις πληροφορίες και τις παρουσιάζουν σύντομα στη τάξη. Αναπτύσσεται συζήτηση σχετικά με τη διαφορά ανάμεσα στον πίνακα και τη λίστα και αναφέρονται χαρακτηριστικά παραδείγματα. Με την ολοκλήρωση της δραστηριότητας ο εκπαιδευτικός ανακεφαλαιώνει τα κυριότερα σημεία στην τάξη.

4.5.3 Διδακτικές Προτάσεις για την υποεπένδυση της Συμπύεσης Δεδομένων

Εισαγωγή -Αφόρμηση

Ο εκπαιδευτικός με σύντομη εισήγηση παρουσιάζει μερικά παραδείγματα εφαρμογής συμπύεσης δεδομένων. Ανάλογα με τις εμπειρίες, τις γνώσεις και τα ενδιαφέροντα των μαθητών, τους καλεί με ερωτοαποκρίσεις να απαντήσουν μέσα από παραδείγματα σε ερωτήματα σχετικά με την χρησιμότητα συμπύεσης των δεδομένων (ως προς την αποθήκευση, τη μετάδοση, το κόστος, την ποιότητα-ικανοποίηση χρήσης εφαρμογών πολυμέσων). Η υψηλή συχνότητα δειγματοληψίας και το μεγάλο μέγεθος δείγματος δημιουργεί ένα ψηφιακό αρχείο μεγάλου μεγέθους για το οποίο πρέπει:

- Να βρεθεί κατάλληλος αποθηκευτικός χώρος.
- Η πληροφορία του ψηφιακού αρχείου να μεταφέρεται αρκετά γρήγορα προς αναπαραγωγή στα αντίστοιχα υποσυστήματα (ήχου, εικόνας, βίντεο).
- Η πληροφορία να μεταφέρεται ικανοποιητικά γρήγορα στα δίκτυα υπολογιστών.

Για την ικανοποίηση των χρηστών θα μπορούσε να ισχυριστεί κανείς ότι μπορούν να χρησιμοποιηθούν μέσα αποθήκευσης με μεγαλύτερη χωρητικότητα και δίκτυα μετάδοσης πληροφοριών με μεγαλύτερο εύρος (bandwidth). Στην περίπτωση αυτή λαμβάνεται υπόψη ο

συντελεστής του κόστους. Τελικά συμπιέζουμε δεδομένα για να γλυτώσουμε χώρο και χρόνο και να ελαχιστοποιήσουμε το κόστος.

Με βάση τους παραπάνω άξονες ο εκπαιδευτικός μπορεί να θέσει μια σειρά από ερωτήματα αφόρμησης για τη χρησιμότητα συμπίεσης δεδομένων. Ενδεικτικά αναφέρονται:

- Πόση διάρκεια ασυμπίεστου βίντεο μπορεί να αποθηκευτεί σε ένα DVD;
- Τι χωρητικότητας αποθηκευτικό μέσο απαιτείται για να διατηρηθούν ψηφιακά αντίγραφα ασφαλείας για όλα τα ψηφιοποιημένα αρχιτεκτονικά σχέδια που έχουν δημιουργηθεί σε ένα αρχιτεκτονικό γραφείο ή για να αποθηκευτούν οι απεικονιστικές εξετάσεις ασθενών (μαγνητικές ακτινογραφίες, υπέρηχοι) σε ένα διαγνωστικό ιατρικό κέντρο.
- Πόσο πρέπει να περιμένει ο χρήστης για να δει ένα βίντεο με το κινητό του τηλέφωνο σε μια ιστοσελίδα που έχει επισκεφθεί;
- Τι προβλήματα μπορεί να παρουσιαστούν όταν σε μια τηλεδιάσκεψη ο χρήστης ακούει την απάντηση των συνομιλητών μετά από τη διέλευση αρκετού χρόνου;
- Πόσο επιβαρύνεται ο καταναλωτής αν έχει χρέωση με βάση τη διάρκεια σύνδεσης ή με βάση τον όγκο της διακινούμενης πληροφορίας;

Εφαρμόζοντας τη τεχνική του καταγισμού ιδεών καταγράφονται οι απαντήσεις των μαθητών στον πίνακα της τάξης. Στη συνέχεια οι μαθητές συνεργάζονται σε ομάδες των πέντε ατόμων για να επιλέξουν και να οργανώσουν τις απαντήσεις στις αντίστοιχες κατηγορίες. Για την επιλογή των κατάλληλων απαντήσεων μπορεί να ανατρέξουν στο Διαδίκτυο για περισσότερες πληροφορίες. Παρουσιάζεται στην τάξη ολοκληρωμένος ο πίνακας με τις απαντήσεις οργανωμένες σε κατηγορίες. Ακολουθεί συζήτηση στη τάξη. Ο εκπαιδευτικός ανακεφαλαιώνει τα κυριότερα σημεία.

Μέρος μαθήματος κάλυψης ύλης

Με σύντομη εισήγηση ο εκπαιδευτικός παρουσιάζει τη συμπίεση δεδομένων και τρόπους απωλεστικής και μη απωλεστικής συμπίεσης. Αξιοποιώντας τις προτεινόμενες πηγές του Διαδικτύου μέσα από χαρακτηριστικά οπτικοποιημένα παραδείγματα προσομοίωσης παρουσιάζει και επεξηγεί χαρακτηριστικούς τρόπους συμπίεσης δεδομένων (προτείνεται να επιλεγούν ένα με δύο χαρακτηριστικά παραδείγματα από αλγόριθμους μη απωλεστικής συμπίεσης όπως του αλγορίθμου LZW). Στη συνέχεια οργανώνει τους μαθητές σε ομάδες των τριών ατόμων για να αναζητήσουν σχετικές πληροφορίες για τους τρόπους συμπίεσης από το Διαδίκτυο (δείτε τη 1η δραστηριότητα της υποενότητας) και να δημιουργήσουν ένα λεξικό βασικών όρων.

Για την περίπτωση απωλεστικών τρόπων συμπίεσης που αφορούν κυρίως τα πολυμέσα (εικόνα, ήχος, βίντεο) προτείνεται να γίνει σύνδεση με πρότερες γνώσεις των μαθητών σε σχέση με τη ψηφιακή αναπαράσταση, τη δειγματοληψία, τον όγκο των δεδομένων. Στα δεδομένα αυτά, που αφορούν πολυμεσικές εφαρμογές η μικρή διαφοροποίηση από την αρχική μορφή των δεδομένων δεν επιφέρει σημαντικές αλλαγές. Καθώς το τελικό αποτέλεσμα αξιολογείται από τις ανθρώπινες αισθήσεις (ακουστό φάσμα, ψυχοακουστικά μοντέλα, όραση), μπορεί να εφαρμοστεί απωλεστική συμπίεση χωρίς πολλές φορές να γίνουν αντιληπτές οι αλλοιώσεις στην ποιότητα των δεδομένων. Αξίζει να σημειωθεί ότι δεν αποτελεί στόχος της ενότητας η αναλυτική περιγραφή των απωλεστικών τρόπων συμπίεσης.

Εμβάθυνση-Εφαρμογή-Επεκτάσεις

Κατά τη διάρκεια της εφαρμογής στο σχολικό εργαστήριο Πληροφορικής, οι μαθητές υλοποιούν τη 2^η δραστηριότητα και πειραματίζονται με τη μετατροπή πάνω σε δεδομένα αρχεία. Επίσης μπορούν να αξιοποιηθούν προσομοιωτές υπολογισμού του μεγέθους ενός αρχείου (ήχου ή βίντεο) τροποποιώντας διάφορες παραμέτρους. Αν υπάρχει διαθέσιμος χρόνος αξιοποιούνται οι αλληλοεπιδραστικές εφαρμογές επίδειξης (δείτε σχετικές πηγές-πόρους) για τη διερεύνηση και το πειραματισμό με τρόπους συμπίεσης δεδομένων.

Σημείωση-Οδηγία: Δεν αποτελεί σκοπό της ενότητας η σείρα απομνημόνευση τρόπων συμπίεσης δεδομένων. Επίσης δεν αποτελεί σκοπό της ενότητας η ανάπτυξη ασκησιολογίου πάνω στους αλγόριθμους απωλεστικής και μη απωλεστικής συμπίεσης δεδομένων.

Χώρος: Εργαστήριο Πληροφορικής με υπολογιστές κατάλληλα διαμορφωμένους για την ανάπτυξη συνεργατικών δραστηριοτήτων.

Εποπτικά Μέσα-Διδακτικά Μέσα: Υπολογιστές με πρόσβαση στο Διαδίκτυο, Διαδραστικός πίνακας ή πίνακας μαρκαδόρου, Βιντεοπροβολέας ή Επιδιασκόπιο (Overhead Projector), Λογισμικό Υπολογιστικών Φύλλων, κατάλληλα διαμορφωμένα φύλλα εργασίας, εκπαιδευτικό λογισμικό προσομοίωσης «Τρίτων».

4.5.4 Ενδεικτικές Δραστηριότητες: Υποενότητα Συμπίεση Δεδομένων

4.5.4.1 Δραστηριότητα 1η: Αναζήτηση, οργάνωση πληροφοριών και δημιουργία wiki λεξικού για απωλεστικούς και μη απωλεστικούς τρόπους συμπίεσης δεδομένων

Οι μαθητές χωρίζονται σε μικρές ομάδες των τριών ατόμων. Με κατάλληλα διαμορφωμένο φύλλο εργασίας καλούνται να αναζητήσουν από το Διαδίκτυο, πληροφορίες για εναλλακτικούς τρόπους απωλεστικής (MPEG-1 Audio Layer 3 για τη ψηφιακή κωδικοποίηση αρχείων ήχου MP3, πρότυπο συμπίεση εικόνας JPEG) και μη απωλεστικής (LZW, RLE) συμπίεσης και να τις συνδέσουν με μορφές αρχείων που χρησιμοποιούν συχνά (ήχου, εικόνας, βίντεο κ.α).

Στη συνέχεια οι μαθητές με την υποστήριξη του εκπαιδευτικού οργανώνουν το υλικό τους και δημιουργούν σε ένα Wiki ένα λεξικό όρων για τους βασικούς τρόπους συμπίεσης. Για την υλοποίηση του λεξικού μπορεί να αξιοποιηθεί το εργαλείο wiki της ηλεκτρονικής σχολικής τάξης (<http://eclass.sch.gr/>). Το λεξικό είναι εμπλουτισμένο με πολυτροπικά κείμενα, με διαγράμματα, υπερσυνδέσεις για πρόσθετο εκπαιδευτικό υλικό, παραδείγματα, σύντομα εκπαιδευτικά βίντεο. Αφού ολοκληρωθεί η οργάνωση των χρήσιμων πληροφοριών και η σύνταξη του λεξικού γίνεται σύντομη παρουσίαση στην τάξη. Αναπτύσσεται συζήτηση σχετικά με την αξιοποίηση των απωλεστικών και μη απωλεστικών τρόπων συμπίεσης ανά περίπτωση. Αναφέρονται χαρακτηριστικά παραδείγματα. Με την ολοκλήρωση της δραστηριότητας ο εκπαιδευτικός ανακεφαλαιώνει τα κυριότερα σημεία στην τάξη.

4.5.4.2 Δραστηριότητα 2η: Σύγκριση διάφορων τύπων αρχείων πριν και μετά τη συμπίεση τους

Οι μαθητές χωρισμένοι σε ομάδες των δύο ατόμων καλούνται να πειραματιστούν με προγράμματα συμπίεσης πάνω σε δεδομένα αρχεία. Για την υλοποίηση της δραστηριότητας στους υπολογιστές του εργαστηρίου έχουν εγκατασταθεί διάφορα προγράμματα συμπίεσης αρχείων (π.χ. .zip) καθώς και προγράμματα μετατροπής αρχείων πολυμέσων σε άλλη συμπιεσμένη μορφή (για παράδειγμα για μετατροπή ενός αρχείου ήχου .wav σε .mp3, ή τη μετατροπή μιας εικόνας σε .jpg μορφή). Στους μαθητές δίνονται έτοιμα αρχεία για τη μετατροπή ή προτρέπονται να μεταφορτώσουν συγκεκριμένα αρχεία από αξιόπιστους δικτυακούς τόπους. Στη συνέχεια οι μαθητές βλέπουν από τις ιδιότητες του αρχείου το μέγεθός του πριν και μετά τη συμπίεση και το συγκρίνουν. Ακολουθεί συζήτηση στη τάξη για τη συμπίεση αρχείων δεδομένων και τους τύπους αρχείων.

4.5.4.3 Δραστηριότητα 3η: Διερεύνηση τρόπων συμπίεσης δεδομένων μέσα από αλληλεπιδραστικές εφαρμογές προσομοίωσης κα σύντομα βίντεο.

Διερεύνηση παραδειγμάτων τρόπων συμπίεσης δεδομένων μέσα από οπτικοποιημένες αλληλεπιδραστικές εφαρμογές προσομοίωσης-επίδειξης.

Με κατάλληλο διαμορφωμένο φύλλο εργασίας οι μαθητές διερευνούν μικρές εφαρμογές οπτικοποιημένων επιδείξεων τρόπων συμπίεσης δεδομένων (προτείνεται να περιοριστεί η

δραστηριότητα σε μη απωλεστικούς τρόπους). Ακολουθεί σύντομη συζήτηση για την εξαγωγή συμπερασμάτων. Στην υποενότητα 4.7 «πηγές-πρόσθετο υλικό-εκπαιδευτικοί πόροι» μπορείτε να βρείτε διαδικτυακές πηγές με προσομοιώσεις, βίντεο και σχεδιαγράμματα χαρακτηριστικών αλγορίθμων μη απωλεστικών τρόπων συμπίεσης.

4.6 Βιβλιογραφία

Παπακωνσταντίνου Γ., Τσανάκας, Π., Φραγκάκης, Γ. (1989) "Αρχιτεκτονική Υπολογιστών", Αθήνα, εκδ. Συμμετρία.

Chapman, N., Jenny Chapman, J. (2009) "Digital Multimedia", Addison-Wesley, 3rd edition, <http://www.digitalmultimedia.org/downloads/DMM3-Sampler.pdf>.

Du Boulay, B. (1989), Some difficulties of learning to program, In E. Saloway & J. C.Spohrer (Eds), Studying the Novice Programmer, Hillsdale, NJ, Lawrence Erlbaum Associates, 283-299.

Patterson, A., Hennessy, J. (1994) «Computer Organization & Design, The Hardware/ Software Interface», Morgan Kaufmann.

Samurcay, R. (1989) "The Concept of variable in programming: Its meaning and use in problem solving by novice programmers, , In E. Saloway & J. C.Spohrer (Eds), Studying the Novice Programmer, Hillsdale, NJ, Lawrence Erlbaum Associates, 161-178.

Steinmetz, R., Nahrestedt, K. (2002) "Πολυμέσα Θεωρία και Πράξη», Αθήνα, Εκδόσεις Μ. Γκιούρδας.

Sedgewick, P., Wayne, K. (2011) "Algorithms"- Chapter 5.5 Data Compression" , Addison-Wesley, 4th edition

<http://www.cs.princeton.edu/courses/archive/fall14/cos226/lectures/55DataCompression.pdf>

4.7 Πηγές-Πρόσθετο Υλικό- Χρήσιμοι Εκπαιδευτικοί Πόροι

- Λογισμικό «ΤΡΙΤΩΝ» διαθέσιμο στην εκπαιδευτική Πύλη του Υ.ΠΑΙ.Θ, e-γliko (<http://www.e-γliko.gr>) Το λογισμικό "ΤΡΙΤΩΝ" προσεγγίζει μεταξύ άλλων θέματα Αρχιτεκτονικής Ηλεκτρονικών Υπολογιστών. Έχει δημιουργηθεί από το εργαστήριο Υπολογιστικών Συστημάτων του Ε.Μ.Π. Μεταξύ άλλων περιέχει την εφαρμογή προσομοίωσης "ΑΒΑΚΑΣ" για την εσωτερική οργάνωση και λειτουργία του υπολογιστή (τελευταία προσπέλαση 02/01/2015) <http://tinyurl.com/lwxsucb>
- Προσομοίωση κρυφής μνήμης από το Ε.Κ.Π.Α Τμήμα Πληροφορικής. <http://hermes.di.uoa.gr/simulation.html> (τελευταία προσπέλαση 02/01/2015)
- Διαδραστική επίδειξη αλγορίθμων συμπίεσης δεδομένων, δειγματοληψίας, υπολογισμού μεγέθους αρχείων ήχου με οπτικοποίηση και δυνατότητα ανατροφοδότησης. Ο Δικτυακός Τόπος αποτελεί τον κόμβο διανομής του Wolfram Demonstrations Project, έμπνευση του Stephen Wolfram δημιουργού του Mathematica. Το πρόγραμμα έχει ως στόχο τη παραγωγή και διανομή ΕΛ/ΛΑΚ εκπαιδευτικών διαδραστικών και οπτικοποιημένων επιδείξεων για την επιστήμη, την τεχνολογία, τα μαθηματικά, την τέχνη και για αρκετά άλλα επιστημονικά πεδία. Για την εκτέλεση των αρχείων επιδείξης στον υπολογιστή απαιτείται η εγκατάσταση του Wolfram CDF Player (παρέχεται δωρεάν). Στη συνέχεια μπορεί κανείς να κατεβάσει και να εκτελέσει το αρχείο επίδειξης ή να το εκτελέσει απ'ευθείας (on-line) από το φυλλομετρητή-browser). <http://demonstrations.wolfram.com/topic.html?topic=Data+Compression&limit=20> (τελευταία προσπέλαση 02/01/2015)
- Demos για την επίδειξη του μη απωλεστικού αλγορίθμου συμπίεσης δεδομένων LZW και του αλγορίθμου Huffman από το δικτυακό τόπο του Princeton University για την

υποστήριξη του μαθημάτων Αλγοριθμικής και δομών δεδομένων (τελευταία προσπέλαση 02/01/2015).

<http://www.cs.princeton.edu/courses/archive/fall14/cos226/demo/55DemoLZW.mov>

<http://www.cs.princeton.edu/courses/archive/fall14/cos226/demo/55DemoHuffman.mov>

- Διαδραστική εφαρμογή για την οπτικοποιημένη επίδειξη (με Applet) συμπίεσης δεδομένων. Από το δικτυακό κόμβο του Πανεπιστημίου Mannheim -Praktische Informatik IV (Prof. Dr. Wolfgang Effelsberg): <http://pi4.informatik.uni-mannheim.de/pi4.data/content/animations/losslesscompression/index.html>
- Δικτυακός τόπος Computer Science Unplugged με διδακτικές προτάσεις, υλικό, και πηγές για τη προσέγγιση της συμπίεσης δεδομένων: <http://csunplugged.org/text-compression> (τελευταία προσπέλαση 02/01/2015).
- On line Tutorials του Καθηγητή D.Marshall από το Πανεπιστήμιο του Cardiff σχετικά με τη συμπίεση δεδομένων (τελευταία προσπέλαση 02/01/2015).

http://www.cs.cf.ac.uk/Dave/Multimedia_CM0340/PDF/08_CM0340_Basic_Compression_Algorithms.pdf

http://www.cs.cf.ac.uk/Dave/Multimedia_CM0340/PDF/10_CM0340_JPEG.pdf

http://www.cs.cf.ac.uk/Dave/Multimedia_CM0340/PDF/11_CM0340_MPEG_Video.pdf

- Άρθρο χωρισμένο σε τρία κύρια μέρη για την συμπίεση δεδομένων που εστιάζεται στη συμπίεση εικόνας, βίντεο και ήχου. Περιέχει επίσης on-line εφαρμογή υπολογισμού μεγέθους αρχείων βίντεο (τελευταία προσπέλαση 02/01/2015).

http://dvd-hq.info/data_compression.php

http://dvd-hq.info/bitrate_calculator.php

5. Από το πρόβλημα στον αλγόριθμο

5.1 Λέξεις Κλειδιά:

Αλγοριθμική σκέψη, Ανάλυση προβλήματος, Αφαίρεση, Περιγραφή και Αναπαράσταση αλγορίθμου, Ψευδοκώδικας, Διάγραμμα ροής, δομή ακολουθίας, δομή επιλογής, δομή επανάληψης, αναδρομή.

5.2 Προτεινόμενος Χρόνος:

Δώδεκα (12) διδακτικές ώρες. Οκτώ (8) ώρες προόδου της ύλης και τέσσερις (4) για εμπέδωση.

5.3 Σκοπός-Στόχοι

Μετά το τέλος της ενότητας οι μαθητές πρέπει να είναι σε θέση:

- Να αναγνωρίζουν ένα σύνθετο πρόβλημα και τα πλεονεκτήματα που προκύπτουν από την ανάλυσή του σε απλούστερα.
- Να επιλύουν προβλήματα χρησιμοποιώντας τα βήματα ανάλυσης σε επιμέρους προβλήματα, με αξιοποίηση υπαρχουσών λύσεων.
- Να επιλέγουν κατά τον σχεδιασμό αλγορίθμου τις κατάλληλες αλγοριθμικές δομές, όπως η ακολουθία, η επιλογή, η επανάληψη και η αναδρομή.
- Να περιγράψουν έναν αλγόριθμο μέσω ψευδοκώδικα ή διαγράμματος ροής.

5.4 Πλαίσιο-Διδακτικές Προτάσεις

Στην ενότητα αυτή προσεγγίζονται θέματα για τη δομή ενός προβλήματος και της πολυπλοκότητάς του, για τη διαδικασία της αφαίρεσης κατά την απλοποίηση ενός προβλήματος, για την ανάλυση ενός προβλήματος σε απλούστερα υποπροβλήματα, για την περιγραφή της λύσης του με αλγοριθμικό τρόπο εκφρασμένη με ψευδοκώδικα ή διάγραμμα ροής, για τις αλγοριθμικές δομές της ακολουθίας, της επιλογής, της επανάληψης και της αναδρομής.

Η σχεδίαση αλγορίθμων είναι μια βασική διεργασία της υπολογιστικής σκέψης. Σχετίζεται με τη διαδικασία επίλυσης του προβλήματος, με την ανάλυσή του σε απλούστερα και κυρίως με την κατασκευή και μορφοποίηση της λύσης του, ώστε να μπορεί να αναπαρασταθεί σε μορφή που μπορεί να υλοποιηθεί αποτελεσματικά από τον υπολογιστή. Η διαδικασία κατανόησης του προβλήματος, η ανάλυση της δομής του και η σχεδίαση της λύσης του με αλγοριθμικό τρόπο ως διεργασίες σκέψης δεν ακολουθούν πάντοτε μια ευθεία ακολουθιακή διαδικασία ιδιαίτερα στη περίπτωση σύνθετων προβλημάτων. Συχνά απαιτείται η σπειροειδής ανάπτυξη νοητικών διεργασιών με επιστροφή και ανατροφοδότηση σε προηγούμενα στάδια. Τα βασικά στάδια μιας τέτοιας ανάπτυξης περιλαμβάνουν:

- Κατανόηση του προβλήματος και ορισμός του σε απλοποιημένη μορφή σε σχέση με την αρχική διατύπωση του κρατώντας τη χρήσιμη πληροφορία
- ανάλυση του σε απλούστερα υποπροβλήματα
- αναδιατύπωση σκέψεων για τις γενικές και ειδικές περιπτώσεις για την επίλυση του
- ανίχνευση γνωστών προτύπων
- κατασκευή του αλγορίθμου περιγράφοντας τις ενέργειες για τη λύση του
- αξιολόγηση του αλγορίθμου και της λύσης που δόθηκε
- επιστροφή αν χρειαστεί σε προηγούμενα στάδια και ανακατασκευή του αλγορίθμου μέχρι την επίλυση του προβλήματος με αποτελεσματικό τρόπο
- γενίκευση της λύσης, ώστε να μπορεί να εφαρμοστεί σε παρόμοια προβλήματα.

Όσον αφορά τη διδασκαλία του αντικειμένου στη τάξη είναι σημαντικό να αναδειχθεί η συλλογιστική που αναπτύσσεται καθώς και οι διαδικασίες που πραγματοποιούνται. Μέσα από απλά στην αρχή προβλήματα, ο εκπαιδευτικός σε κάθε στάδιο καθοδηγεί τους μαθητές. Με κατάλληλα ερωτήματα τους ενθαρρύνει να καταθέτουν ελεύθερα τις σκέψεις τους και τις αποφάσεις που έλαβαν για να οδηγηθούν στη λύση τους προβλήματος. Οι μαθητές αναπτύσσουν τη συλλογιστική τους, προσεγγίζουν την επίλυση του προβλήματος και κατασκευάζουν αλγορίθμους.

Η κατανόηση της αλγοριθμικής λογικής από τους μαθητές στην ενότητα αυτή είναι ιδιαίτερα σημαντική για την προσέγγιση των εννοιών που ακολουθούν. Παρότι οι μαθητές έχουν προσεγγίσει το σχεδιασμό αλγορίθμων και σε προηγούμενες τάξεις, ο εκπαιδευτικός προτείνεται να συμπεριλάβει ποικίλες διδακτικές τεχνικές για τη διαδικασία ανάλυσης του προβλήματος και σχεδίασης της αλγοριθμικής λύσης του. Η κατανόηση των βασικών αλγοριθμικών δομών της ακολουθίας, της επιλογής, της επανάληψης και της αναδρομής είναι απαραίτητη για την κατασκευή αλγορίθμων και απαιτεί από τους μαθητές την ενεργοποίηση ιδιαίτερων νοητικών διεργασιών. Η προσέγγιση της αλγοριθμικής λογικής προϋποθέτει την ικανότητα των μαθητών να μπορούν να ερμηνεύουν αλλά και να εκφράζουν τη λύση σε διαφορετικά συστήματα αναπαράστασης: ψευδοκώδικα, διάγραμμα ροής και αργότερα με μορφή κώδικα μιας γλώσσας προγραμματισμού. Αξίζει να σημειωθεί ότι η οικοδόμηση της αλγοριθμικής λογικής δεν εξαντλείται στην ενότητα αυτή, αλλά λειτουργεί ως βασικό νοητικό σκαλοπάτι για την οικοδόμηση των εννοιών των επόμενων εννοιών.

Ανάμεσα στις διδακτικές τεχνικές που προτείνεται να συμπεριλάβει ο εκπαιδευτικός στον διδακτικό σχεδιασμό είναι:

- Η παρουσίαση χαρακτηριστικών παραδειγμάτων. Για παράδειγμα ο εκπαιδευτικός παρουσιάζει χαρακτηριστικά παραδείγματα που είναι οικεία στους μαθητές και επεξηγεί τη σημασία της αφαίρεσης για τη διαχείριση της πολυπλοκότητας ενός προβλήματος.
- Η σύντομη εισήγηση. Ο εκπαιδευτικός παρουσιάζει σύντομα τους βασικούς τρόπους αναπαράστασης αλγορίθμων που θα χρησιμοποιηθούν (ψευδοκώδικα και διάγραμμα ροής δεδομένων) και τις αλγοριθμικές δομές (ακολουθία, επιλογή, επανάληψη, αναδρομή).
- Παρουσίαση και διερεύνηση οπτικοποιημένων μοντέλων αλγορίθμων για γνωστά προβλήματα. Για παράδειγμα οι Πύργοι του Ανόι, αλγόριθμοι κρυπτογράφησης κ.ά. (δείτε προτεινόμενες πηγές-εκπαιδευτικούς πόρους).
- Οι ερωταποκρίσεις και ο καταιγισμός ιδεών, ώστε να ενισχυθεί η ενεργή συμμετοχή των μαθητών και η αυτοπεποίθηση τους καταθέτοντας σκέψεις, απόψεις σχετικά με την επίλυση προβλήματος και την κατασκευή αλγορίθμων.
- Η μελέτη περίπτωσης. Οι μαθητές σε ομάδες αναλαμβάνουν να μελετήσουν μια μελέτη περίπτωσης ενός σύνθετου προβλήματος που είναι κοντά στα ενδιαφέροντά τους και στα βιώματά τους. Οι μαθητές συμμετέχουν ενεργά στα στάδια καθορισμού του, στην απλούστευση του προβλήματος, στη διάσπαση του σε υποπροβλήματα, στο σχεδιασμό της λύσης ενός υποπροβλήματος (για οικονομία χρόνου) με αλγοριθμικό τρόπο, στις διαδικασίες γενίκευσης του.

Ιδιαίτερος χρόνος πρέπει να αφιερωθεί για την πρακτική εφαρμογή και την εμπάθυνση στις έννοιες. Οι μαθητές μέσα από προβλήματα προς επίλυση διερευνούν και κατασκευάζουν αλγορίθμους. Αρχικά οι μαθητές επιλύουν απλά κλασικά προβλήματα. Στη συνέχεια προτείνονται προς επίλυση ένα με δύο (ανάλογα το διαθέσιμο εκπαιδευτικό χρόνο) πιο σύνθετα προβλήματα.

Για παράδειγμα μπορεί να χρησιμοποιηθούν προβλήματα που σχετίζονται με διαδικασίες αυτοματοποίησης και ρομποτικής επιλέγοντας ελκυστικά και κατανοητά θέματα για τους μαθητές κοντά στα ενδιαφέροντα και στα βιώματά τους. Τα προβλήματα αυτά σχετίζονται με διάφορα γνωστικά πεδία από τα Μαθηματικά, τη Φυσική, τη Βιολογία, τη Μηχανική και αναδεικνύουν την

αξιοποίηση των αλγοριθμικών δομών της ακολουθίας, της επιλογής, της επανάληψης ή της αναδρομής. Στις ενδεικτικές δραστηριότητες (δραστηριότητα_1) προτείνεται ως πρόβλημα προς επίλυση η προσομοίωση της λειτουργίας μιας «έξυπνης» κατοικίας με αισθητήρες και αυτοματισμούς για να διατηρείται μια σταθερή θερμοκρασία με την ελάχιστη κατανάλωση ρεύματος.

Άλλη κατηγορία προβλημάτων που μπορεί να αξιοποιηθεί στην ενότητα είναι αυτή της κατασκευής εκπαιδευτικών παιχνιδιών και παιχνιδιών quiz με ερωτήσεις εγκυκλοπαιδικών γνώσεων. Τα προβλήματα αυτά σχετίζονται με μια μεγάλη ποικιλία γνωστικών αντικειμένων από τη φυσική, τη βιολογία, τα οικονομικά, την ιστορία, τη λογοτεχνία, την τέχνη, τα μαθηματικά κ.ά. Οι μαθητές καλούνται να αναλύσουν το πρόβλημα σχεδίασης και κατασκευής ενός εκπαιδευτικού παιχνιδιού και να επιλύσουν μια απλή υποπερίπτωσή του (δραστηριότητα 2).

5.5 Χώρος

Εργαστήριο Πληροφορικής - Αίθουσα.

5.6 Εποπτικά Μέσα-Διδακτικά Μέσα

Υπολογιστές με πρόσβαση στο Διαδίκτυο, Διαδραστικός ή πίνακας Μαρκαδόρου, Βιντεοπροβολέας ή Επιδιασκόπιο (Overhead Projector), Λογισμικό Υπολογιστικών Φύλλων, κατάλληλα διαμορφωμένα φύλλα εργασίας.

5.7 Ενδεικτικές Δραστηριότητες

5.7.1 Δραστηριότητα 1

Μία ομάδα μαθητών αποφάσισαν να δημιουργήσουν μια μικρογραφία σε μακέτα ενός σπιτιού για να προσομοιώσουν τη λειτουργία μιας «έξυπνης» κατοικίας. Για το σκοπό αυτό συνέδεσαν διάφορους αισθητήρες με ένα μικροεπεξεργαστή, που θα ελέγχει το εξωτερικό περιβάλλον και ανάλογα θα ενεργοποιούνται διάφοροι αυτοματισμοί (άνοιγμα παραθύρου, άνοιγμα κουρτινών, ενεργοποίηση ανεμιστήρα κ.ά.) για τη προσαρμογή του σπιτιού, ώστε να διατηρείται μία σταθερή θερμοκρασία γύρω στους 20 με 25 βαθμούς C° με το ελάχιστο δυνατό κόστος.

Να αναλύσετε το πρόβλημα σε απλούστερα υποπροβλήματα, ώστε να υποστηρίξετε το σχεδιασμό της διάταξης. Κρατήστε τις χρήσιμες πληροφορίες και στη συνέχεια καταγράψτε τις κατηγορίες και τις παραμέτρους που πρέπει να λάβετε υπόψη σας για την υλοποίηση της λύσης.

Στη συνέχεια υλοποιήστε αλγόριθμο εκφρασμένο σε λογικό διάγραμμα που να περιγράφει τα βασικά βήματα που πρέπει να εκτελέσει ο μικροεπεξεργαστής, ώστε αν η θερμοκρασία αυξηθεί να ανοίξει αυτόματα το παράθυρο της σκεπής για να μπει φρέσκος αέρας. Από ποιον αισθητήρα θα πάρει αυτήν την πληροφορία ως είσοδο; Ποιες συνθήκες πρέπει να ικανοποιηθούν ώστε να κλείσει το παράθυρο; Επέκταση: Ποια νέα προβλήματα μπορεί να δημιουργηθούν όταν ανοίξει το παράθυρο ώστε να ληφθούν υπόψη οι συνθήκες που πρέπει τελικά να ικανοποιηθούν για να δοθεί η αντίστοιχη εντολή για να ανοίξει το παράθυρο;

Σημείωση: Εναλλακτικά μπορεί να αποφασίσετε να επιλύσετε κάποιο άλλο υποπρόβλημα που συντελεί στην διατήρηση της επιθυμητής θερμοκρασίας.

5.7.2 Δραστηριότητα 2

Είστε μέλος μιας ομάδας για την κατασκευή ενός ηλεκτρονικού εκπαιδευτικού παιχνιδιού σε μορφή quiz με ερωτήσεις για διάφορα εγκυκλοπαιδικά θέματα. Αναλύστε το πρόβλημα της

δημιουργίας του παιχνιδιού σε απλούστερα υποπροβλήματα. Για ένα μικρό τμήμα του παιχνιδιού περιγράψτε έναν αλγόριθμο που για μια ερώτηση, αρχικά θα διαβάζει την απάντηση και θα ελέγχει την ορθότητά της. Στη συνέχεια θα επιστρέφει στον παίκτη του παιχνιδιού ανάλογο μήνυμα. Το παιχνίδι θα σταματάει αν ο παίκτης απαντήσει πάνω από τρεις φορές λάθος. Προσπαθήστε να γενικεύσετε τη λύση που περιγράψατε στον αλγόριθμο, ώστε να εφαρμοστεί για μια μεγαλύτερη ομάδα ερωτήσεων. Εμπλουτίστε το παιχνίδι με βαθμούς και ξεκινήστε.

Οδηγίες: Οι μαθητές με καταιγισμό ιδεών καταθέτουν τις απόψεις τους για τις κατηγορίες και τις παραμέτρους που πρέπει να λάβουν υπόψη τους για την υλοποίηση του παιχνιδιού. Ανάμεσα στις παραμέτρους που πρέπει να ληφθούν υπόψη ενδεικτικά μπορεί να αναφερθούν:

- οι κανόνες του παιχνιδιού,
- ο τρόπος βαθμολογίας,
- το είδος των ερωτήσεων (πολλαπλής επιλογής, απευθείας απάντησης κλπ),
- τα γνωστικά πεδία που θα αντληθούν οι ερωτήσεις και οι σωστές απαντήσεις τους,
- ο τρόπος ανανέωσης των ερωτήσεων και των αντίστοιχων σωστών απαντήσεων τους,
- η μορφή του παιχνιδιού,
- το γραφικό του περιβάλλον,
- ο τρόπος εισαγωγής των απαντήσεων,
- ο αριθμός των παιχτών που μπορεί να συμμετάσχουν.
- αν μπορεί να παίζουν οι παίχτες μεταξύ τους ή εναλλάξ;
- αν το παιχνίδι θα είναι Διαδικτυακό;

Στη συνέχεια οι μαθητές σε ομάδες συγκεντρώνουν και ομαδοποιούν σε κατηγορίες και σε υποκατηγορίες τις ιδέες από το καταιγισμό ιδεών. Για την οπτικοποίηση των σκέψεών τους μπορεί να χρησιμοποιηθεί ένα υπολογιστικό φύλλο ή ένα εργαλείο εννοιολογικής χαρτογράφησης (όπως το popplet <http://popplet.com/>). Παρουσιάζονται τα αποτελέσματα στη τάξη και αναπτύσσεται συζήτηση για τη διαδικασία και τα πλεονεκτήματα διάσπασης του αρχικού προβλήματος σε επιμέρους υποπροβλήματα.

Στη συνέχεια κάθε ομάδα αποφασίζει τη μορφή που θα έχει η ερώτηση και η απάντηση καθώς και το περιεχόμενό της. Αφού καταγράψει τις αποφάσεις της στο φύλλο εργασίας, σχεδιάζει τον αλγόριθμο. Ανάμεσα στα ερωτήματα που πρέπει να απαντήσουν οι μαθητές είναι:

- Τι μεταβλητές θα χρησιμοποιηθούν;
- Τι τύπος δεδομένων (αριθμός, αλφαριθμητικό) θα είναι η μεταβλητή για την αποθήκευση των απαντήσεων;
- Ποια ή ποιες αλγοριθμικές δομές θα πρέπει να επιλέξουν για να σχεδιάσουν τον αλγόριθμο;
- Ποιες συνθήκες πρέπει να ικανοποιούνται για να σταματήσει το παιχνίδι;
- Πως θα μετράνε τον αριθμό των λαθεμένων απαντήσεων;

Αναπτύσσεται σύντομη συζήτηση, ώστε οι μαθητές να κατανοήσουν ότι ανάλογα με τις αποφάσεις που θα πάρουν θα επηρεαστεί και ο τρόπος σχεδίασης του αλγορίθμου. Για παράδειγμα αν αποφασίσουν ότι οι απαντήσεις θα είναι αλφαριθμητικά τότε θα πρέπει να χειριστούν όλους τους δυνατούς τρόπους που μπορεί να απαντήσει ο παίχτης (με κεφαλαία, με μικρά χωρίς τόνους, με μικρά με τόνους κλπ.). Ένας εύκολος τρόπος είναι να επιλέξουν τις απαντήσεις πολλαπλής επιλογής αριθμημένες με ακεραίους. Στην περίπτωση αυτή για την απάντηση αρκεί να διαβαστεί ένας ακέραιος αριθμός. Για τη σχεδίαση του αλγορίθμου θα χρειαστεί να χρησιμοποιηθεί η δομή επανάληψης, αφού ο παίχτης έχει περισσότερες από μία ευκαιρίες να απαντήσει σωστά στην ερώτηση.

5.8 Δραστηριότητες-Ασκήσεις με απλά προβλήματα προς επίλυση

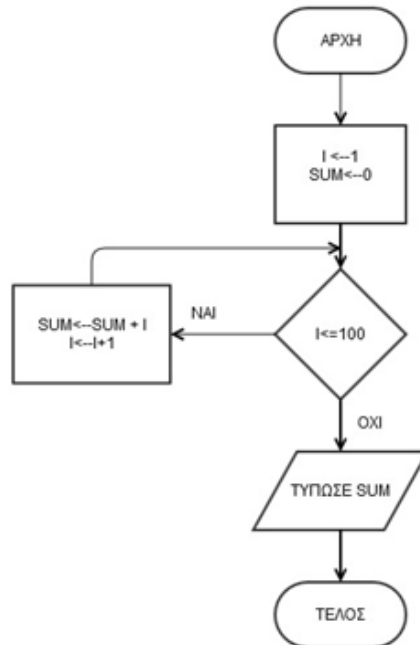
Ενδεικτικά αναφέρονται:

5.8.1 Δραστηριότητα 1η

Να γραφεί αλγόριθμος σε ψευδοκώδικα και σε διάγραμμα ροής που να διαβάσει τρεις ακεραίους αριθμούς και να υπολογίζει το μέσο όρο τους. Το αποτέλεσμα να εμφανίζεται στην οθόνη.

5.8.2 Δραστηριότητα 2^η

Να γραφεί αλγόριθμος με διάγραμμα ροής που να υπολογίζει το άθροισμα $1+2+3+\dots+100$



5.8.3 Δραστηριότητα 3η

Να γραφεί αλγόριθμος σε ψευδοκώδικα που να επιλύει την εξίσωση $Ax+B=0$, για A και B πραγματικούς αριθμούς.

5.8.4 Δραστηριότητα 4η

Να μετατρέψετε το παρακάτω τμήμα αλγορίθμου σε ισοδύναμο του χρησιμοποιώντας τις άλλες δυο εντολές επανάληψης (όσο_επανάλαβε και για..από...μέχρι):

$\alpha \leftarrow 5$

$\beta \leftarrow 7$

Επανάλαβε

Εμφάνισε β, α

$\beta \leftarrow \beta + 5$

Μέχρις_Ότου ($\beta > 40$)

5.8.5 Δραστηριότητα 5η

Να γραφεί αλγόριθμος σε ψευδοκώδικα που βρίσκει σε ποιον όρο το άθροισμα $1+2+3+4+\dots$ γίνεται μεγαλύτερο του 2000.

5.8.6 Δραστηριότητα 6η

Να γραφεί αλγόριθμος σε ψευδοκώδικα που δέχεται τριψήφιο θετικό ακέραιο και ελέγχει αν είναι παλίνδρομος (ο αριθμός παραμένει ο ίδιος όταν αναστρέψουμε τη σειρά των ψηφίων του π.χ. 131, 797).

5.8.7 Δραστηριότητα 7η

Να μετατραπεί η παρακάτω δομή σύνθετης επιλογής σε μία μόνο εντολή εκχώρησης που να έχει το ίδιο ακριβώς αλγοριθμικό αποτέλεσμα.

Αν $X*Y+Z > 200$ τότε

$k \leftarrow$ Ψευδής

αλλιώς

$k \leftarrow$ Αληθής

Τέλος_Αν

5.8.8 Δραστηριότητα 8η

Να γραφεί ισοδύναμα το παρακάτω τμήμα αλγορίθμου με μία απλή δομή επιλογής.

Αν $k > 0$ τότε

$f \leftarrow 140+z$

Αλλιώς

$f \leftarrow 100$

Τέλος_Αν

5.8.9 Δραστηριότητα 9η

Να γραφεί αλγόριθμος σε ψευδοκώδικα που υλοποιεί τον υπολογισμό της ακολουθίας Fibonacci.

5.8.10 Δραστηριότητα 10η

Να γραφεί αλγόριθμος σε ψευδοκώδικα που υλοποιεί τον υπολογισμό της δύναμης χωρίς χρήση του τελεστή « \wedge ».

5.8.11 Δραστηριότητα 11^η

Να γραφεί αλγόριθμος που να εξετάζει και να εμφανίζει πόσοι από τους αριθμούς που υπάρχουν μεταξύ του 50 και του 500 είναι πολλαπλάσια του 3.

5.8.12 Δραστηριότητα 12η

Να γραφεί αλγόριθμος ο οποίος διαβάζει αριθμούς μέχρι να δοθεί ο αριθμός 0 και εμφανίζει τον αριθμό που δόθηκε τις περισσότερες συνεχόμενες φορές.

5.8.13 Άσκηση Αυτοαξιολόγησης

Ερωτήσεις σωστού/λάθους

Χαρακτηρίστε τις παρακάτω προτάσεις ως σωστές ή λάθος βάζοντας (Σ) ή (Λ).

1. Η τιμή της μεταβλητής είναι το περιεχόμενο της αντίστοιχης θέσης μνήμης και δεν μεταβάλλεται στη διάρκεια εκτέλεσης του αλγορίθμου.

2. Σε ένα διάγραμμα ροής ο ρόμβος δηλώνει την είσοδο ή την έξοδο τιμών.
3. Ένας αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών.
4. Οι ενέργειες που ορίζει ένας αλγόριθμος είναι αυστηρά καθορισμένες.
5. Η έννοια του αλγορίθμου συνδέεται αποκλειστικά με την Πληροφορική.
6. Η λογική πράξη "ή" μεταξύ δύο προτάσεων είναι ψευδής, όταν οποιαδήποτε από τις δύο προτάσεις είναι ψευδής.
7. Στη δομή επιλογής μπορεί μία ή περισσότερες εντολές να μην εκτελεστούν.
8. Η δομή επιλογής επιτρέπει τη διενέργεια ελέγχων κατά τη διάρκεια εκτέλεσης των αλγορίθμων.
9. Αν έχουμε την εντολή $Y \leftarrow (X < 9)$, τότε η μεταβλητή Y είναι λογικού τύπου.
10. Η λογική παράσταση x ΚΑΙ $OXI(x)$ είναι πάντα ψευδής ανεξάρτητα από την τιμή του x .
11. Ο ρόμβος σε ένα διάγραμμα ροής έχει 2 εξόδους.
12. Κάθε εντολή ΟΣΟ μπορεί να μετατραπεί σε μια ισοδύναμη εντολή ΓΙΑ.
13. Αν η συνθήκη της ΟΣΟ είναι αληθής τότε ο βρόχος θα εκτελεστεί οπωσδήποτε πάνω από μια φορά.
14. Ο βρόχος Για I από 0 μέχρι 3 με_βήμα 3 εκτελείται τρεις φορές.
15. Με χρήση της εντολής ΟΣΟ επιτυγχάνεται η επανάληψη μίας διαδικασίας με βάση κάποια συνθήκη.
16. Οι εντολές της επαναληπτικής ΟΣΟ θα εκτελεστούν τουλάχιστον μια φορά.
17. Όταν το πλήθος των επαναλήψεων είναι γνωστό, δεν μπορεί να χρησιμοποιηθεί η εντολή επανάληψης ΟΣΟ.
18. Η εντολή επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ εκτελείται υποχρεωτικά τουλάχιστον μία φορά.
19. Ο βρόχος Για K από 7 μέχρι 7 δεν εκτελείται καμία φορά.
20. Ο βρόχος Για K από 4 μέχρι 1 εκτελείται 4 φορές.
21. Η εντολή επανάληψης ΓΙΑ μπορεί να χρησιμοποιηθεί όταν έχουμε άγνωστο αριθμό επαναλήψεων.
22. Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.
23. Όταν ένας βρόχος είναι εμφωλευμένος σε άλλο, ο βρόχος που ξεκινάει τελευταίος πρέπει να ολοκληρώνεται πρώτος.
24. Η επανάληψη που δεν τελειώνει ποτέ παραβιάζει το κριτήριο της περατότητας.
25. Μια δομή επιλογής είναι δυνατό να περιέχει μια δομή επανάληψης.

5.9 Βιβλιογραφία

- Kadesch, R. (1996). Problem Solving-Across the Disciplines, Prentice Hall Engineering.,
- Knuth, D., E., (1997). The Art of Computer Programming: Fundamental Algorithms vol1., Addison Wesley.
- Report of a Workshop on the Pedagogical Aspects of Computational Thinking, National Research Council, (2011) <http://www8.nationalacademies.org/cp/projectview.aspx?key=48969> (τελευταία προσπέλαση 09/01/2015).
- Selby, Cynthia, Dorling, Mark and Woollard, John (2014) Evidence of assessing computational thinking. Brookes eJournal of Learning and Teaching, 1-12. <http://eprints.soton.ac.uk/366152/> (τελευταία προσπέλαση 28/12/2014).

Sedgewick, P., Wayne, K. (2011) "Algorithms" , Addison-Wesley, 4th edition.

Wing, J. (2010). Research Notebook: Computational Thinking - What and Why? The Link. Pittsburgh, PA: Carneige Mellon. Διαθέσιμο:
<http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf> (τελευταία προσπέλαση 23/12/2014)

5.10 Πηγές - Πρόσθετο Υλικό - Χρήσιμοι Εκπαιδευτικοί Πόροι

Επίδειξη αλγορίθμων με οπτικοποίηση και δυνατότητα ανατροφοδότησης. Ο Δικτυακός Κόμβος αποτελεί το κόμβο διανομής του Wolfram Demonstrations Project, έμπνευση του Stephen Wolfram. Το πρόγραμμα έχει ως στόχο τη παραγωγή και διανομή ΕΛ/ΛΑΚ εκπαιδευτικών διαδραστικών και οπτικοποιημένων επιδείξεων για την επιστήμη, την τεχνολογία, τα μαθηματικά, τη τέχνη και για αρκετά άλλα επιστημονικά πεδία. Για την εκτέλεση των αρχείων επίδειξης στον υπολογιστή απαιτείται η εγκατάσταση του Wolfram CDF Player (παρέχεται δωρεάν). Στη συνέχεια μπορεί κανείς να κατεβάσει και να εκτελέσει το αρχείο επίδειξης ή να το εκτελέσει απευθείας (on-line) από το φυλλομετρητή-browser).

<http://demonstrations.wolfram.com/topics.html?Algorithms#123> (τελευταία προσπέλαση 05/01/2015)

Δικτυακός τόπος με ψηφιακό υλικό για την εισαγωγή στην αλγοριθμική σκέψη: <http://www.teaching-materials.org/algorithms/> (τελευταία προσπέλαση 05/01/2015).

Δικτυακός τόπος Computer Science Unplugged με διδακτικές προτάσεις, υλικό, και πηγές: <http://csunplugged.org/> (τελευταία προσπέλαση 05/01/2015).

Ψηφιακό Σχολείο-Αποθετήριο Φωτόμετρο. Περιέχει μεταξύ άλλων μαθησιακά αντικείμενα για την προσέγγιση της αλγοριθμικής σκέψης (τελευταία προσπέλαση 05/01/2015) <http://photodentro.edu.gr/lor/subject-search?locale=el>

Αποθετήριο εκπαιδευτικών πολυφασικών πόρων Merlot II. Περιέχει μεταξύ άλλων διαδραστικό εκπαιδευτικό υλικό για την προσέγγιση διαφόρων αλγορίθμων. <http://tinyurl.com/ldfme7a> (τελευταία προσπέλαση 05/01/2015).

Εργαλείο on-line για τη κατασκευή διαγραμμάτων ροής <https://www.glify.com> (τελευταία προσπέλαση 05/01/2015).

5.11 Σημειώσεις ενότητας εκπαιδευτικού

.....
.....
.....
.....
.....
.....
.....

6. Από τον αλγόριθμο στο πρόγραμμα

6.1 Λέξεις Κλειδιά

Διαδικασιακός προγραμματισμός, Αντικειμενοστρεφής προγραμματισμός, Συναρτησιακός προγραμματισμός, συγγραφή, μετάφραση και εκτέλεση προγράμματος, σύνταξη εντολών γλώσσας προγραμματισμού, Ακολουθία, Επιλογή, Επανάληψη, Συμβολοσειρές (Αλφαριθμητικά) (Strings), Λίστες.

6.2 Προτεινόμενος Χρόνος

Δεκατέσσερις (14) διδακτικές ώρες, Έξι (6) ώρες προόδου της ύλης και οκτώ (8) για εμπέδωση.

6.3 Σκοπός-Στόχοι

Μετά το τέλος της ενότητας οι μαθητές πρέπει να είναι σε θέση:

- Να εξηγούν τη λογική συγγραφής προγραμμάτων ανά είδος προγραμματισμού (Διαδικασιακός, Αντικειμενοστρεφής και Συναρτησιακός προγραμματισμός).
- Να υλοποιούν έναν πλήρη κύκλο ανάπτυξης προγράμματος σε περιβάλλον προγραμματισμού (συγγραφή κώδικα, αποσφαλμάτωση, μετάφραση/διερμηνεία, εκτέλεση).
- Να επιλέγουν και να χρησιμοποιούν βασικές αλγοριθμικές δομές και δομές δεδομένων, όπως η ακολουθία, επιλογή, επανάληψη και οι Συμβολοσειρές ή Λίστες στη γλώσσα προγραμματισμού.

6.4 Πλαίσιο

Η ενότητα έχει ως βασικό σκοπό τη γνωριμία με τα είδη προγραμματισμού και την υλοποίηση πλήρους κύκλου ανάπτυξης προγράμματος σε περιβάλλον προγραμματισμού, επιλέγοντας τις κατάλληλες αλγοριθμικές δομές αλλά και δομές δεδομένων, όπως οι συμβολοσειρές και οι λίστες.

Οι μαθητές, αφού ολοκλήρωσαν την προηγούμενη ενότητα και προσέγγισαν θέματα σχεδίασης αλγορίθμων για την επίλυση προβλήματος, έχουν την ευκαιρία να εξοικειωθούν με τη διαδικασία συγγραφής, μετάφρασης και εκτέλεσης προγράμματος χρησιμοποιώντας το προγραμματιστικό περιβάλλον IDLE της γλώσσας Python (μιας ευρέως γνωστής γλώσσας με ισχυρή κοινότητα υποστήριξης και πολλές υλοποιημένες εφαρμογές). Χωρίς να είναι αυτοσκοπός η εξάντληση όλων των εντολών της γλώσσας προγραμματισμού και των συντακτικών κανόνων που τις συνοδεύει, οι μαθητές μαθαίνουν τη σύνταξη των βασικών εντολών της γλώσσας προγραμματισμού για να υλοποιούν τις αλγοριθμικές δομές της ακολουθίας, της επιλογής και της επανάληψης.

Ο εκπαιδευτικός παρουσιάζει σταδιακά την υλοποίηση των δομών αυτών, μέσα από απλά στην αρχή παραδείγματα. Μέσα στο διδακτικό του σχεδιασμό για τη δομή επανάληψης προτείνεται να συμπεριλάβει την αναλυτική επεξήγηση για τις συνθήκες ελέγχου, το σώμα των εντολών που θα επαναληφθούν, την αρχική κατάσταση των μεταβλητών.

Επιπρόσθετα στην ενότητα αυτή γίνεται εισαγωγή στις βασικές δομές δεδομένων: συμβολοσειρές και λίστες. Για την αξιοποίηση της δυναμικής δομής της λίστας οι μαθητές μέσα από παραδείγματα και δραστηριότητες εξοικειώνονται με την επεξεργασία τους, τους τελεστές, τις μεθόδους.

6.5 Διδακτικές Προτάσεις

Ο εκπαιδευτικός με σύντομη εισήγηση συνδέει την ενότητα με τις έννοιες των αλγοριθμικών δομών και της επίλυσης προβλήματος που αναπτύχθηκαν στη προηγούμενη ενότητα. Με ερωτοαποκρίσεις κινητοποιεί το ενδιαφέρον των μαθητών για ενεργή συμμετοχή στη τάξη.

Στη συνέχεια με απλά παραδείγματα παρουσιάζει αναλυτικά στους μαθητές τη χρήση του περιβάλλοντος IDLE της Python για την συγγραφή του κώδικα του προγράμματος, την αποθήκευση του προγράμματος στους υπολογιστές του σχολικού εργαστηρίου Πληροφορικής, τη μετάφραση και εκτέλεσή του (ιδέες απλών παραδειγμάτων μπορείτε να βρείτε στις προτεινόμενες πηγές-πρόσθετο υλικό- χρήσιμοι εκπαιδευτικοί πόροι). Μέσα από παραδείγματα καλής και κακής γραφής αναλύει τους κανόνες που πρέπει να τηρούνται για τη συγγραφή ενός ευανάγνωστου και τεκμηριωμένου κώδικα (χρήση εσοχών, σχόλια, εισαγωγικά κλπ)

Για την επίτευξη των στόχων της ενότητας ιδιαίτερη βαρύτητα δίνεται στην υλοποίηση σημαντικού αριθμού δραστηριοτήτων διαφορετικού τύπου, ώστε οι μαθητές να έχουν τη δυνατότητα:

- Να πειραματίζονται με τη συγγραφή κώδικα μέσα από την κατασκευή απλών προγραμμάτων.
- Να υλοποιούν με το προγραμματιστικό περιβάλλον της γλώσσας Python αλγοριθμικές λύσεις εκφρασμένες σε ψευδοκώδικα, που περιέχουν διάφορες αλγοριθμικές δομές (ακολουθία, ελέγχου, επανάληψης).
- Να διερευνούν τι κάνει ένα έτοιμο πρόγραμμα και στη συνέχεια επαληθεύουν τις αρχικές τους υποθέσεις μέσα από το προγραμματιστικό περιβάλλον.
- Να πραγματοποιούν ασκήσεις διαβαθμισμένης δυσκολίας για την επεξεργασία συμβολοσειρών και λιστών.
- Να αναζητούν, συλλέγουν και οργανώνουν πληροφορίες από το Διαδίκτυο σχετικά με τα είδη προγραμματισμού, όπως ο Διαδικασιακός, Αντικειμενοστρεφής, Συναρτησιακός προγραμματισμός και του τρόπου λειτουργίας του καθενός.

Ο εκπαιδευτικός ενθαρρύνει και παρακινεί τους μαθητές να αναπτύσσουν πρωτοβουλίες, να αυτενεργούν, να ανακαλύπτουν μόνοι τους τη γνώση μέσα από τον πειραματισμό, προσπαθώντας να κάνουν τα προγράμματα να δουλέψουν έχοντας ανατροφοδότηση από το προγραμματιστικό περιβάλλον.

Μετά την υλοποίηση των δραστηριοτήτων, αναπτύσσεται συζήτηση. Οι μαθητές παρουσιάζουν τις απαντήσεις, καταθέτουν τις σκέψεις τους και συμμετέχουν στην εξαγωγή χρήσιμων συμπερασμάτων. Στον πίνακα της τάξης ή με την αξιοποίηση ενός βιντεοπροβολέα δίνονται οι ορθές απαντήσεις και ανακεφαλαιώνονται τα κυριότερα σημεία.

6.6 Χώρος

Αίθουσα-Εργαστήριο Πληροφορικής

6.7 Εποπτικά Μέσα-Διδακτικά Μέσα:

Υπολογιστές με πρόσβαση στο Διαδίκτυο, Διαδραστικός ή πίνακας μαρκαδόρου, Βιντεοπροβολέας ή Επιδιασκόπιο (Overhead Projector), Λογισμικό Υπολογιστικών Φύλλων, Προγραμματιστικό περιβάλλον IDLE Python 2.7.9, διαδικτυακό εργαλείο wiki της ηλεκτρονικής σχολικής τάξης.

6.8 Ενδεικτικές Δραστηριότητες

6.8.1 Δραστηριότητα: Τα βασικά είδη προγραμματισμού, αναζήτηση και οργάνωση πληροφοριών.

Οι μαθητές χωρίζονται σε μικρές ομάδες των τριών ατόμων. Με κατάλληλα διαμορφωμένο φύλλο εργασίας καλούνται να αναζητήσουν από το Διαδίκτυο, πληροφορίες για τα βασικά είδη προγραμματισμού όπως το Διαδικασιακό, Αντικειμενοστρεφή και Συναρτησιακό προγραμματισμό καθώς και για τον τρόπο λειτουργίας του καθενός. Για την καλύτερη οργάνωση των διαδικτυακών εκπαιδευτικών πόρων και την οικονομία του διδακτικού χρόνου ο εκπαιδευτικός μπορεί να δημιουργήσει μια ιστοεξερεύνηση (<http://webquest.org/>, <http://zunal.com/>) ή να αξιοποιήσει τα εργαλεία υποστήριξης συνεργατικής μάθησης (διαμοιρασμός πόρων, wiki, μονοπάτι μάθησης, συνομιλία κ.α) που παρέχει η ηλεκτρονική σχολική τάξη (<http://eclass.sch.gr/>).

Οι μαθητές με την υποστήριξη του εκπαιδευτικού, οργανώνουν το υλικό τους και δημιουργούν σε ένα Wiki ένα πολυτροπικό κείμενο για τα βασικά είδη προγραμματισμού. Το κείμενο μπορεί να περιέχει διαγράμματα, υπερσυνδέσεις για πρόσθετο, εκπαιδευτικό υλικό, παραδείγματα, σύντομα εκπαιδευτικά βίντεο. Αφού ολοκληρωθεί η οργάνωση των χρήσιμων πληροφοριών και η σύνταξη του κειμένου γίνεται σύντομη παρουσίαση στη τάξη. Αναπτύσσεται συζήτηση σχετικά με τις διαφορές ανάμεσα στα βασικά είδη προγραμματισμού, τα πλεονεκτήματα, τις σύγχρονες τάσεις. Αναφέρονται χαρακτηριστικά παραδείγματα. Με την ολοκλήρωση της δραστηριότητας ο εκπαιδευτικός ανακεφαλαιώνει τα κυριότερα σημεία στην τάξη.

6.8.2 Δραστηριότητα: Δομή ακολουθίας

Να γραφεί πρόγραμμα που να επιλύει το νόμο του Ωμ (Ohm) και υπολογίζει την ένταση του ρεύματος I για τάση V =220 Volt και R= 25 ohm.

```
# Πρόγραμμα εύρεσης έντασης του ηλεκτρικού ρεύματος
V = 220
R = 25
I = V / R
print (" Η ένταση του ρεύματος είναι:", I, " A")
```

Δομή Ελέγχου

Η εντολή ελέγχου AN_ΑΛΛΙΩΣ συντάσσεται ως:

```
if <συνθήκη ελέγχου>:
    #γράψε τις εντολές εδώ
else:
    #γράψε τις εντολές εδώ
```

Σημείωση!!!: Οι ομάδες εντολών που θα εκτελεστούν, αν ισχύει μια συνθήκη, ορίζονται ως ένα μπλοκ με εσοχή (κενά διαστήματα) βάζοντας τη μια εντολή κάτω από την άλλη!

6.8.3 Δραστηριότητα: Δομή ελέγχου - Ηλικία

Να γραφεί πρόγραμμα που να διαβάζει την ηλικία ενός ατόμου. Αν η ηλικία είναι μεγαλύτερη ή ίση των 18 ετών να εμφανίζει μήνυμα ότι επιτρέπεται να ψηφίζει, διαφορετικά να εμφανίζει μήνυμα ότι είναι αρκετά νέος ακόμα.

Ψευδοκώδικας	Πρόγραμμα σε Python
Αλγόριθμος ΗΛΙΚΙΑ Εκτύπωσε «Δώσε την ηλικία σου:» Διάβασε age Αν age >= 18 Τότε Εκτύπωσε «Είσαι αρκετά μεγάλος και μπορείς να ψηφίσεις στις εθνικές εκλογές» Αλλιώς Εκτύπωσε «Είσαι αρκετά μικρός ακόμα για να ψηφίσεις στις εθνικές εκλογές» Τέλος_Αν Τέλος ΗΛΙΚΙΑ	age = input("Δώσε την ηλικία σου: ") if age >= 18: print("Είσαι αρκετά μεγάλος και μπορείς να ψηφίσεις στις εθνικές εκλογές") else: print("Είσαι αρκετά μικρός ακόμα για να ψηφίσεις στις εθνικές εκλογές")

6.8.4 Δραστηριότητα Δομή ελέγχου - Εμφωλευμένο Αν –Αλλιώς

Το παρακάτω πρόγραμμα ανατροφοδοτεί τους μαθητές για ένα τεστ αυταξιολόγησης. Τι θα τυπώσει για τους βαθμούς 6, 10, 12, 15, 18, 20. Τι θα συμβεί αν κατά λάθος βάλουμε ως βαθμό το 25. Επέκταση: Πώς θα μπορούσε να τροποποιηθεί το πρόγραμμα, ώστε να περιέχει και την περίπτωση που ο χρήστης δώσει κατά λάθος ως βαθμό έναν αρνητικό αριθμό.

```
# Εισαγωγή βαθμού από το χρήστη
test = input('Παρακαλώ δώσε το βαθμό που πήρες από το τεστ:')

if test <= 20:
    # Εμφωλευμένο if
    if test >= 18:
        print 'Μπράβο τα πήγες πολύ καλά'
        # για 18 <= test <= 20
    else:
        if test >= 15:
            print 'Τα πήγες αρκετά καλά!'
            # για 15 <= test < 18`
        else:
            if test >= 10:
                print"Τα πήγες σχετικά καλά. Κοίταξε πιο προσεκτικά το κεφάλαιο"
```

```

        # για 10<=τεστ<15
    else:
        print "Θα πρέπει να διαβάσεις ξανά το κεφάλαιο. Μη διστάσεις να μου πεις τις
            απορίες σου"
        # για τεστ<10
else:
    # για test >20
    print "Ο βαθμός που έδωσες είναι μεγαλύτερος του είκοσι"

```

Η Python προσφέρει τη δυνατότητα για σύνταξη σύνθετων δομών ελέγχου με τη χρήση της εντολής `elif`. Η σύνταξη είναι ως εξής:

```

if <συνθήκη>:
    <εντολές>
elif <συνθήκη2>:
    <εντολές_2>
else:
    <εντολές_3>

```

6.8.5 Δραστηριότητα: Δομή ελέγχου - Έλεγχος θερμοστάτη για ψύξη θέρμανση

Να γραφεί πρόγραμμα που να διαβάζει την θερμοκρασία ενός δωματίου και αν η θερμοκρασία είναι ίση ή μεγαλύτερη από 32 βαθμούς να εμφανίζει μήνυμα για να ανοίξει η ψύξη του κλιματιστικού, αν είναι μικρότερη ή ίση από 16 να εμφανίζει μήνυμα για να ανοίξει η θέρμανση διαφορετικά να εμφανίζει μήνυμα ότι δεν χρειάζεται να ανοίξει το κλιματιστικό.

```

temperature = int(input("Ποια είναι η θερμοκρασία του δωματίου σε βαθμούς Κελσίου; "))
if temperature >= 32:
    print("Ανοιξε τη ψύξη του κλιματιστικού και ρύθμισε τη θερμοκρασία")
elif temperature <= 16:
    print("Ανοιξε τη θέρμανση του κλιματιστικού και ρύθμισε για οικονομία τη θερμοκρασία")
else:
    print("Η θερμοκρασία είναι καλή δε χρειάζεται ακόμη να ανοίξεις το κλιματιστικό")
print("τέλος")

```

6.8.6 Δραστηριότητα: Δομή ελέγχου

Βρείτε τι κάνει το παρακάτω πρόγραμμα και δώστε με μια πρόταση ένα χαρακτηριστικό τίτλο.

```
number = input("Δώστε έναν ακέραιο αριθμό: ")
if number < 0:
    print("Ο αριθμός είναι αρνητικός")
elif number > 0:
    print("ο αριθμός είναι θετικός")
else:
    print("ο αριθμός είναι 0")
```

6.8.7 Δραστηριότητα: Δομή Επανάληψης

Η Python υποστηρίζει δύο τύπους επανάληψης: Βρόχο For και βρόχο While.

6.8.7.1 Δραστηριότητα Δομή Επανάληψης 1η

Εξοικείωση με την επανάληψη.

Να γραφεί πρόγραμμα που να εκτυπώνει πέντε (5) φορές το μήνυμα «Θέλει αρετή και τόλμη η ελευθερία!»

```
for metritis in range (5):
    print "Θέλει αρετή και τόλμη η ελευθερία!"
```

6.8.7.2 Δραστηριότητα Δομή Επανάληψης 2η

Προσεγγίζοντας τη συνάρτηση range.

Η range() είναι μια ενσωματωμένη συνάρτηση της γλώσσας Python. Ανάμεσα σε άλλα χρησιμοποιείται για την υπόδειξη του αριθμού των επαναλήψεων που θα εκτελεστούν σε ένα βρόχο.

Η δομή της συνάρτησης range είναι της μορφής (αρχή, μέχρι, βήμα), όπου αρχή, μέχρι, βήμα ακέραιοι αριθμοί. Οι ενδείξεις της «αρχής» και του «βήματος» δεν είναι υποχρεωτικές, αλλά μπορεί να προστεθούν ώστε η επανάληψη να μοιάζει με τη σύνταξη της ψευδοκώδικα. Αντίθετα η ένδειξη «μέχρι» πρέπει πάντα να αναφέρεται.

Πειραματιστείτε με τη χρήση της συνάρτησης range στο προγραμματιστικό περιβάλλον της Python για να δείτε τι παράγει η συνάρτηση. Για παράδειγμα:

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

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

range(0, 35, 5) παράγει τη λίστα: [0,5,10,15,20,25, 30]

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

6.8.7.3 Δραστηριότητα Δομή Επανάληψης 3η

Να γραφεί αλγόριθμος και το αντίστοιχο πρόγραμμα σε γλώσσα Python που να διαβάζει πέντε τυχαίους ακεραίους αριθμούς και να υπολογίζει το άθροισμα τους.

Ψευδοκώδικας	Πρόγραμμα σε Python
Αλγόριθμος Άθροισμα Sum ← 0 Για i από 1 μέχρι 5 Διάβασε x Sum ← Sum+x Τέλος_επανάληψης Εκτύπωσε Sum Τέλος_Άθροισμα	<pre> sum = 0 for i in range(5): x = input("Δώσε έναν αριθμό: ") sum = x + sum print('Το αποτέλεσμα είναι ', sum) </pre>

6.8.7.4 Δραστηριότητα Δομή Επανάληψης 4^η

Να γραφεί αλγόριθμος και το αντίστοιχο πρόγραμμα σε γλώσσα Python που να υπολογίζει το άθροισμα 1+2+3+...+100

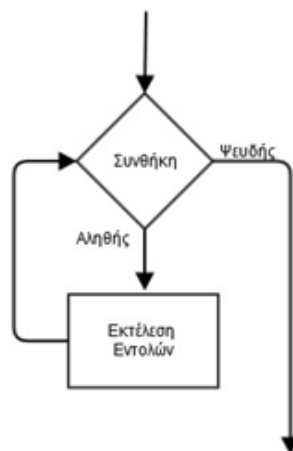
Ψευδοκώδικας	Πρόγραμμα σε Python
Αλγόριθμος αθροίζω Sum ← 0 Για i από 1 μέχρι 100 Sum ← Sum+i Τέλος_επανάληψης Εκτύπωσε Sum Τέλος αθροίζω	<pre> # Πρόγραμμα Αθροίζω sum = 0 for i in range(1, 101): sum = sum + i print(sum) </pre>

6.8.7.5 Δραστηριότητα Δομή Επανάληψης 5^η

Να γραφεί αλγόριθμος και το αντίστοιχο πρόγραμμα σε Python που να υπολογίζει την:

$$1+1*2+1*2*3+\dots+1*2*3*\dots*v$$

Δομή Επανάληψης με While βρόχο (ή Όσο <συνθήκη> επανάλαβε)



6.8.7.6 Δραστηριότητα Δομή Επανάληψης 6η

Να γραφεί, με χρήση του βρόχου While, ο αντίστοιχος αλγόριθμος και το αντίστοιχο πρόγραμμα της δραστηριότητας 4.

Ψευδοκώδικας	Πρόγραμμα σε Python
Αλγόριθμος αθροίζω Sum ← 0 I ← 1 Όσο I ≤ 100 επανάλαβε Sum ← Sum + i I ← i + 1 Τέλος_επανάληψης Εκτύπωσε Sum Τέλος αθροίζω	#Πρόγραμμα Αθροίζω sum = 0 i = 1 while i <= 100 : sum = sum + i i = i + 1 print(sum)

6.8.7.7 Δραστηριότητα Δομή Επανάληψης 7η

Μία συνήθης εφαρμογή του while βρόχου είναι να ελέγχει την εγκυρότητα των δεδομένων εισόδου από τον χρήστη. Συμπλήρωσε κατάλληλα την παρακάτω συνθήκη έτσι ώστε να εξασφαλίζεται ότι ο χρήστης θα εισάγει τελικά την τιμή 'ναι' ή 'όχι'.

```
choice = raw_input('Σας ενδιαφέρει ο προγραμματισμός? (ναι/όχι)')  
while _____:  
    choice = raw_input("Παρακαλώ δώστε έγκυρη τιμή ")
```

6.8.7.8 Δραστηριότητα Δομή Επανάληψης 8η: Δημιουργία quiz

Να γραφεί πρόγραμμα για την εύρεση της απάντησης σε μια ερώτηση πολλαπλής επιλογής (με τέσσερις απαντήσεις 1, 2, 3 και 4). Το πρόγραμμα θα δίνει αρχικά την ερώτηση και τις πιθανές απαντήσεις και θα ζητάει από το χρήστη να πληκτρολογήσει τον αριθμό της απάντησης που θεωρεί σωστή. Αν η απάντηση είναι λάθος θα εμφανίζει μήνυμα λάθους και θα ζητάει ξανά να δοθεί η απάντηση. Η διαδικασία θα συνεχίζεται όσο η απάντηση δεν είναι σωστή. Κάθε αποτυχημένη προσπάθεια θα αποθηκεύεται σε ένα μετρητή. Όταν απαντηθεί σωστά η ερώτηση θα εμφανίζεται ανάλογο μήνυμα μαζί με τις συνολικές προσπάθειες.

```
# Πρόγραμμα quiz  
print (" Που έζησε ο συγγραφέας και ποιητής Γιάννης Σκαρίμπας; ")  
print ("1. ΑΘΗΝΑ")  
print ("2. ΠΑΤΡΑ ")
```

```

print ("3. ΧΑΛΚΙΔΑ")
print ("4. ΙΩΑΝΝΙΝΑ")
metriti = 0
reply = 0
while reply != 3:
    reply = input ("Βρες τη σωστή απάντηση δίνοντας τον αντίστοιχο αριθμό που συνοδεύει τις
πιθανές απαντήσεις; ")
    if reply != 3:
        print("Πρέπει να ξαναδοκιμάσεις. Η απάντηση σου δεν είναι σωστή")
        metriti = metriti + 1

print (" ΜΠΡΑΒΟ ΑΠΑΝΤΗΣΕΣ ΣΩΣΤΑ ΜΕ ΤΗΝ", metriti, "η ΠΡΟΣΠΑΘΕΙΑ")

```

6.8.7.9 Δραστηριότητα Δομή Επανάληψης 9^η

Βρείτε τι κάνει το παρακάτω πρόγραμμα και δώστε με μία πρόταση έναν χαρακτηριστικό τίτλο.

```

thenum = random.randint(1,100)
print "Έχω σκεφτεί ένα αριθμό από το 1 μέχρι το 100."
print "Μπορείς να τον μαντέψεις;"
guess = 0
while guess != thenum:
    guess=input("Δώσε αριθμό: ")
    if guess>thenum:
        print "Έδωσες μεγαλύτερο αριθμό"
    if guess<thenum:
        print "Έδωσες μικρότερο αριθμό"
    if guess==thenum:
        print "Τον μάντεψες!"

```

6.8.8 Δραστηριότητα: Αλφαριθμητικά

Η Python δίνει πολλές δυνατότητες στη διαχείριση αλφαριθμητικών. Αλφαριθμητικά μπορούν να συνενωθούν, να επιλεγούν ανεξάρτητα μέρη τους ή να διαγραφούν.

6.8.8.1 Δραστηριότητα αλφαριθμητικά 1η

Εισαγωγή στους κανόνες σύνταξης της γλώσσας Python για τα αλφαριθμητικά. Να προσεγγιστούν οι έννοιες:

- Μεταβλητές (variables).
- Πράξεις με αλφαριθμητικά, τελεστές.
- Μέθοδοι.

Ο εκπαιδευτικός με κατάλληλα διαμορφωμένο φύλλο εργασίας αναθέτει στους μαθητές να εκτελέσουν διάφορες ενέργειες και να πειραματιστούν με το διερμηνευτή της γλώσσας Python, όπως για παράδειγμα παρουσιάζεται στην αριστερή στήλη του παρακάτω πίνακα:

Λίγη πρακτική εξάσκηση: Ενέργειες προς εκτέλεση στο διερμηνευτή επιλέγοντας τις κατάλληλες εντολές της γλώσσας Python	Αντίστοιχες εντολές και αποτελέσματα στον διερμηνευτή
<ul style="list-style-type: none">• Ορίστε σε μια μεταβλητή a να περιέχει τη τιμή «Καλημέρα»• Ορίστε σε μια μεταβλητή b να περιέχει τη τιμή «ηλιόλουστη μέρα»• Ορίστε σε μια μεταβλητή z το άθροισμα των a και b.• Ορίστε σε μια μεταβλητή d το γινόμενο της a * 2.• Να εμφανιστούν στην οθόνη οι τιμές των μεταβλητών a, b, z, d• Να βρεθεί με τη συνάρτηση len() το μέγεθος του αλφαριθμητικού b• Να επιλεγεί έναν συγκεκριμένος χαρακτήρας από το b για παράδειγμα b[1]• Να επιλεγεί ένα τμήμα από το αλφαριθμητικό b [11 : 15]• Να εφαρμοστούν ενδεικτικά μέθοδοι για την επεξεργασία των αλφαριθμητικών. Παραδείγματα μεθόδων: b.capitalize(), b.upper(), b.lower().	<pre>a = "Καλημέρα " b = "ηλιόλουστη μέρα" z = a + b d = a * 2 print a Καλημέρα print b ηλιόλουστη μέρα print z Καλημέρα ηλιόλουστη μέρα print d Καλημέρα Καλημέρα len(b) 15 b[1] λ b[11 : 15] μέρα</pre>

6.8.8.2 Δραστηριότητα αλφαριθμητικά 2η

Να γραφεί πρόγραμμα που να διαβάζει μια λέξη και να ελέγχει αν η λέξη είναι καρκινική (ονομάζονται συμμετρικές λέξεις οι οποίες μπορούν να διαβαστούν είτε από την αρχή είτε από το τέλος π.χ. ANNA, RADAR) εμφανίζοντας ανάλογο μήνυμα.

6.8.9 Δραστηριότητα: Λίστες

Η λίστα στη Python έχει δυναμικό μέγεθος και τα αντικείμενά της είναι οποιουδήποτε τύπου και έχουν διάταξη.

```
daysofweek= ["Δευτέρα", "Τρίτη", "Τετάρτη", "Πέμπτη", "Παρασκευή", "Σάββατο", "Κυριακή"]
```

Για κάθε αντικείμενο που εισάγεται στη λίστα δίνεται ένας αύξων αριθμός, που χρησιμοποιείται για την αναφορά του στο αντικείμενο. Μπορεί κανείς ανά πάσα στιγμή να προσθέσει, να διαγράψει ή να αλλάξει ένα αντικείμενο σε μία λίστα.

Ο εκπαιδευτικός με μικρή εισήγηση παρουσιάζει τη δομή της λίστας. Στη συνέχεια δίνει χαρακτηριστικά απλά παραδείγματα δημιουργίας και χειρισμού της. Με μικρές ασκήσεις καλεί τους μαθητές να πειραματιστούν περισσότερο με τη δομή.

6.8.9.1 Δραστηριότητα λίστες 1η

Δίνεται το παρακάτω πρόγραμμα. Τι πιστεύετε ότι θα εμφανιστεί στην οθόνη μετά την εκτέλεση του. Επαληθεύστε το αποτέλεσμα δοκιμάζοντας το πρόγραμμα στο προγραμματιστικό περιβάλλον της Python.

```
daysofweek= ["Δευτέρα", "Τρίτη", "Τετάρτη", "Πέμπτη", "Παρασκευή", "Σάββατο", "Κυριακή"]
>>> print "θα έχω πολύ ελεύθερο χρόνο την ", daysofweek[6]
```

Ερώτηση: Ποιο πιστεύετε ότι θα είναι το αποτέλεσμα μετά την εκτέλεση του προγράμματος;

Απάντηση: θα έχω πολύ ελεύθερο χρόνο την Κυριακή

Ερμηνεία: Οι λίστες ξεκινάνε από το στοιχείο 0

Οι μαθητές δημιουργούν μια απλή λίστα και στη συνέχεια μπορούν να γράψουν μικρά προγράμματα για να διαχειριστούν τα δεδομένα που περιέχει, για παράδειγμα ένα μικρό βρόχο επανάληψης για την εκτύπωση τους. Επίσης μπορούν να συνδυάσουν λίστες για να δημιουργήσουν νέες.

6.8.9.2 Δραστηριότητα λίστες 2η

Δίνονται τα παρακάτω προγράμματα. Τι πιστεύετε ότι θα εμφανιστεί στην οθόνη μετά την εκτέλεσή τους. Καταγράψτε τις απαντήσεις σας στο φύλλο εργασίας. Στη συνέχεια επαληθεύστε τις απαντήσεις σας εκτελώντας τα προγράμματα στο προγραμματιστικό περιβάλλον της Python.

```
x = [21, 23, 25, 27]
```

```
y = [5, 6, 7, 8]
```

```
z = x + y
```

```
print (z)
```

Απάντηση: Αποτέλεσμα στην οθόνη: [21, 23, 25, 27, 5, 6, 7, 8]

```
.....
```

```
print (z [1])
```

Απάντηση: Αποτέλεσμα στη οθόνη: 23

```
.....
```

```
z[0] = 45
```

```
print (z)
```

Απάντηση: Αποτέλεσμα στην οθόνη: [45, 23, 25, 27, 5, 6, 7, 8]

```
.....
```

```
a = [x,y]
print (a)
print (a [1][2])
```

Απάντηση: Αποτέλεσμα στην οθόνη: [[21, 23, 25, 27], [5, 6, 7, 8]] 7

.....

6.8.9.3 Δραστηριότητα λίστες 3η

Παρατηρήστε αυτό που θα εμφανιστεί στην οθόνη. Δοκιμάστε τις ίδιες εντολές χωρίς να ακολουθεί το , μετά την εντολή print. Τι παρατηρείτε;

```
>>> for a in [15,27,31]:
....     print a,
```

Απάντηση: Αποτέλεσμα στην οθόνη: 15, 27, 31

```
daysofweek = ["Δευτέρα", "Τρίτη", "Τετάρτη", "Πέμπτη", "Παρασκευή", "Σάββατο", "Κυριακή"]
for item in daysofweek:
    print (item),
```

Αποτέλεσμα στη οθόνη: Δευτέρα Τρίτη Τετάρτη Πέμπτη Παρασκευή Σάββατο Κυριακή

6.8.9.4 Δραστηριότητα λίστες 4η

Μελετήστε το παρακάτω πρόγραμμα. Τι πιστεύετε ότι εμφανίζεται στην οθόνη; Επαληθεύστε τις υποθέσεις σας εκτελώντας το πρόγραμμα στο προγραμματιστικό περιβάλλον. Τι παρατηρείτε;

```
listX = [1, 2, 3]
listY = listX
print 'listX =', listX, 'list =', listY
listX[1] = 5000
print 'listX =', listX, 'list =', listY
listY[2] = 100
print 'listX =', listX, 'list =', listY
```

Διαδοχικά αποτελέσματα στην οθόνη

```
listX = [1, 2, 3] list = [1, 2, 3]
listX = [1, 5000, 3] list = [1, 5000, 3]
listX = [1, 5000, 100] list = [1, 5000, 100]
```

6.8.9.5 Δραστηριότητα λίστες 5η

Δημιουργήστε μια λίστα με ζώα της θάλασσας. Στη συνέχεια με τις βασικές μεθόδους προσπαθήστε να προσθέσετε, να διαγράψετε και να επανεισάγετε στοιχεία.

Σημείωση: Για τη διαχείριση μίας λίστας εφαρμόζονται βασικές μέθοδοι όπως: η `append(X)` για τη προσθήκη ενός στοιχείου στο τέλος της λίστας, η `insert(i, x)` για την εισαγωγή ενός στοιχείου `x` στη λίστα πριν το `i` στοιχείο, η `pop(i)` για τη διαγραφή του `i` στοιχείου.

6.8.9.6 Δραστηριότητα λίστες 6η

Βρείτε τι κάνει το παρακάτω πρόγραμμα. Επαληθεύστε την υπόθεση σας εκτελώντας το στο προγραμματιστικό περιβάλλον της γλώσσας.

```
sqlist=[ ]
for x in range(1,11):
    sqlist.append(x*x)
print sqlist
```

6.8.9.7 Δραστηριότητα λίστες 7η

Να γράψετε μια συνάρτηση σε Python η οποία θα δέχεται μια λίστα από λέξεις και θα επιστρέφει την λέξη που σχηματίζεται από τα πρώτα γράμματα των λέξεων.

6.8.9.8 Δραστηριότητα λίστες 8η

Να γράψετε ένα πρόγραμμα σε Python το οποίο θα διαβάζει από το πληκτρολόγιο μια λίστα από λέξεις μέχρι να δοθεί η λέξη 'end' και

1. Θα εμφανίζει τη λέξη ή τις λέξεις με τα περισσότερα γράμματα. (Υπόδειξη: Κατασκευάστε μια λίστα με το πλήθος των γραμμάτων κάθε λέξης).
2. Θα εμφανίζει το πλήθος των λέξεων που τελειώνουν με το γράμμα 's'.

Δραστηριότητα με ενδεικτικό Φύλλο Εργασίας για τους τύπους δεδομένων και τις μεταβλητές.

Όνομα Επώνυμο Μαθητή:

Τμήμα.....Ημερομηνία:.....

Καταγράψετε στη δεξιά στήλη του πίνακα τι πιστεύετε ότι θα εμφανιστεί στην οθόνη μετά την εκτέλεση των παρακάτω προγραμμάτων. Στη συνέχεια επαληθεύστε τις απαντήσεις σας εκτελώντας τα προγράμματα μέσα από το περιβάλλον της γλώσσας Python.

ΠΡΟΓΡΑΜΜΑΤΑ	ΑΠΟΤΕΛΕΣΜΑΤΑ ΣΤΗΝ ΟΘΟΝΗ
<code>x = 45</code>	
<code>x = 45.5</code>	
<code>print x</code>	
<code>print x + x</code>	

<pre>x= "Μυρτώ" y = "Βασίλη" print "Καλημέρα", y, "και", x print "Καλημέρα", x + y</pre>	
<pre>x = 12 print x + 3</pre>	
<pre>x = 26 x = y = z = 23 print x, y, z x = y = "Ελευθερία" print x, y</pre>	
<pre>x , y, z = "Ελευθερία","Πέτρος", 2 print x, y, z print x, y * z</pre>	
<pre>x , y = 8, 12 print x, y</pre>	
<pre>a = 010 print a</pre>	
<pre>x = 6 +4j y = 8 -3j z = x + y print z</pre>	
<pre>a = 0xA print a</pre>	
<pre>X = [11, 22, 33] Y = X X[2] = 700 print X, Y</pre>	

Οδηγίες - Σημείωση για τον εκπαιδευτικό

Οι μαθητές ατομικά συμπληρώνουν στο Φύλλο Εργασίας, απαντώντας στο ερώτημα τι πιστεύουν ότι θα εμφανιστεί στη οθόνη. Στη συνέχεια ανοίγουν το περιβάλλον προγραμματισμού της Python και ελέγχουν τις απαντήσεις τους. Με αυτό το τρόπο έχουν άμεση ανατροφοδότηση με το δικό τους ρυθμό από τον υπολογιστή. Στη συνέχεια παρουσιάζονται τα αποτελέσματα στην τάξη και αναπτύσσεται συζήτηση ανατροφοδότησης για τις μεταβλητές και τον τρόπο διαχείρισης τους από τη γλώσσα Python (οι μαθητές θα έχουν την ευκαιρία να εμβαθύνουν και στην επόμενη ενότητα).

6.9 Πηγές-Πρόσθετο Υλικό- Χρήσιμοι Εκπαιδευτικοί Πόροι

Ο δικτυακός τόπος υποστήριξης της γλώσσας Python,

<https://www.Python.org/> (τελευταία προσπέλαση 3/01/2015)

Δικτυακός τόπος που περιγράφει μέσα από απλά παραδείγματα τη χρήση του προγραμματιστικού περιβάλλοντος IDLE της Python(τελευταία προσπέλαση 08/01/2015),

https://hkn.eecs.berkeley.edu/~dyoo/Python/idle_intro/IDLE_greek.htm

Tutorial από το δικτυακό τόπο της γλώσσας Python για την γνωριμία με τη γλώσσα Python – Δομή Ελέγχου και Δομές Επανάληψης (τελευταία προσπέλαση 03/01/2015),

<https://docs.Python.org/2/tutorial/controlflow.html>

Τύποι Δεδομένων Συμβολοσειρές, Λίστες

<https://docs.Python.org/2/tutorial/introduction.html> (τελευταία προσπέλαση 08/01/2015)

Δικτυακός Τόπος με Tutorial για την εκμάθηση της έκδοσης 2.7.x. της Python.

<http://www.Python-course.eu/course.php> (τελευταία προσπέλαση 02/01/2015)

Tutorial για την εισαγωγή στη γλώσσα Python: <http://www.codecademy.com/tracks/Python> (τελευταία προσπέλαση 4/01/2015)

Εκπαιδευτικό υλικό με δραστηριότητες, βίντεο, διδακτικές προτάσεις για τον εκπαιδευτικό για τη διδασκαλία της γλώσσας Python: <http://www.Pythonschool.net/category/basics.html> (τελευταία προσπέλαση 02/01/2015)

7. Πληροφοριακά Συστήματα

7.1 Λέξεις Κλειδιά

Πληροφοριακό Σύστημα (ΠΣ), τύποι πληροφοριακών συστημάτων, Κύκλος ζωής ανάπτυξης ΠΣ, Βάση δεδομένων, Ασφάλεια πληροφοριών.

7.2 Προτεινόμενος Χρόνος

Οκτώ (8) διδακτικές ώρες διδασκαλία ύλης

7.3 Σκοπός-Στόχοι

Μετά το τέλος της ενότητας οι μαθητές πρέπει να είναι σε θέση:

- Να περιγράψουν τι είναι ένα Πληροφοριακό Σύστημα.
- Να απαριθμούν τα συστατικά στοιχεία ενός ΠΣ.
- Να γνωρίσουν τις κατηγορίες εφαρμογής ενός ΠΣ και τις περιοχές που αξιοποιείται.
- Να κατονομάζουν και να αναλύουν τις φάσεις και τα στάδια ανάπτυξης ενός ΠΣ.
- Να υιοθετήσουν μια ευρύτερη οπτική για τα προβλήματα και τις λύσεις-προγράμματα χωρίς να εστιάζουν μόνο στο επίπεδο των τεχνικών χαρακτηριστικών, αλλά λαμβάνοντας υπόψη τον πραγματικό κόσμο (επιχειρήσεις, διοίκηση κλπ).

7.4 Πλαίσιο

Τα Πληροφοριακά Συστήματα (ΠΣ) είναι ιδιαίτερα σημαντικά για τη λειτουργία, την εσωτερική παραγωγικότητα και την επικοινωνία και συναλλαγή με το εξωτερικό περιβάλλον των σύγχρονων επιχειρήσεων και οργανισμών. Υπάρχουν πολλά επίπεδα χρήσης και αξιοποίησης των ΠΣ όπως:

- Εσωτερικά Πληροφοριακά Συστήματα (intranets).
- Αυτοματοποίηση εργασιών και διαδικασιών που οδηγεί στη μείωση του κόστους.
- Ανασχεδιασμός διαδικασιών και τρόπων εργασίας.
- Βελτίωση και δημιουργία νέων προϊόντων και υπηρεσιών προς τους πελάτες.
- Διαχείριση γνώσης.
- Υποστήριξη λήψης αποφάσεων.
- Εξωστρεφή Πληροφοριακά Συστήματα.
- Ηλεκτρονική διασύνδεση και συναλλαγές με τους πελάτες (Ηλεκτρονικό Εμπόριο).
- Ηλεκτρονική διασύνδεση και συναλλαγές με τους προμηθευτές (Ηλεκτρονικές Προμήθειες).
- Ηλεκτρονική συνεργασία για το σχεδιασμό νέων προϊόντων.
- Ηλεκτρονικές Αγορές (E – marketplaces).
- Ηλεκτρονική διασύνδεση και παροχή υπηρεσιών προς τους πολίτες (ηλεκτρονική διακυβέρνηση).

Σκοπός της ενότητας αυτής είναι να γνωρίσουν οι μαθητές την έννοια του Πληροφοριακού Συστήματος και τις περιοχές εφαρμογής του. Αυτή η οικοδόμηση γνώσης θα γίνει σταδιακά μέσα από παραδείγματα που θα παρουσιάζουν τα μέρη (συνιστώσες) ενός ΠΣ καθώς και τον κύκλο ζωής ανάπτυξής του. Στη συνέχεια, θα παρουσιαστούν και συζητηθούν θέματα διαχείρισης ενός ΠΣ, όπως η ασφάλεια πληροφοριών (εντοπισμός και χαρακτηρισμός ως εμπιστευτικών των πληροφοριών που πρόκειται να χρησιμοποιηθούν και να προστατευθούν, ακεραιότητα πληροφοριών, εμπιστευτικότητα, διαθεσιμότητα πληροφοριών, αλλά και αυθεντικότητα, εγκυρότητα,

μοναδικότητα και μη αποποίηση). Χρήση βάσεων δεδομένων (ΒΔ) (συλλογή και ανάλυση απαιτήσεων, εννοιολογικός σχεδιασμός ΒΔ, επιλογή συστήματος διαχείρισης βάσεων δεδομένων ΣΔΒΔ, λογικός σχεδιασμός ΒΔ, φυσικός σχεδιασμός ΒΔ, υλοποίηση συστήματος ΒΔ, επεξεργασία της ΒΔ). Τέλος, η κατανόηση των λειτουργικών και επιχειρηματικών διαδικασιών, αποτελεί την αιτία για τη βελτίωση και την αυτοματοποίησή τους μέσω τεχνολογιών Πληροφορικής. Θέματα που αφορούν στην ανάλυση των επιχειρηματικών διαδικασιών, την μοντελοποίηση και την αναβάθμισή τους θα αναλυθούν.

7.5 Διδακτικές Προτάσεις

Ο εκπαιδευτικός με σύντομη εισήγηση συνδέει την ενότητα με τις προηγούμενες ενότητες, ως εφαρμογή τους. Με ερωτοαποκρίσεις κινητοποιεί το ενδιαφέρον των μαθητών για ενεργό συμμετοχή στη τάξη και προωθεί την ανακάλυψη των βασικών χαρακτηριστικών των ΠΣ από τους ίδιους τους μαθητές.

Με τη χρήση παραδειγμάτων και ερωτήσεων εισάγει τους μαθητές στην έννοια του Πληροφοριακού Συστήματος (βασική σημειογραφία) και των δομικών συνιστωσών του (λογισμικό, υλικό, διαδικασίες και ανθρώπινο δυναμικό). Ιδιαίτερη βαρύτητα θα δοθεί στις επιχειρηματικές διαδικασίες καθώς η αυτοματοποίησή τους μέσω υπολογιστών αποτελεί τον βασικότερο λόγο υιοθέτησης ΠΣ από τους οργανισμούς.

Οι μαθητές μέσα απο ομαδοσυνεργατικές δραστηριότητες θα πρέπει να ανακαλύψουν περιοχές εφαρμογής ενός ΠΣ και να καταλήξουν στην κατηγοριοποίησή τους. Ο εκπαιδευτικός μπορεί να έχει το ρόλο του αρωγού και του διευκολυντή στη γνώση δίνοντας τη δυνατότητα στους μαθητές να την ανακαλύψουν όσο το δυνατόν μόνοι τους (βαθμοί ελευθερίας) συνεργαζόμενοι μεταξύ τους. Στη διαδικασία αυτή πολλές φορές βοηθάει η οπτικοποίηση της γνώσης (π.χ εννοιολογική χαρτογράφηση των κατηγοριών των ΠΣ και παραδειγμάτων τους). Για την αποδοτική συνεργασία των μαθητικών ομάδων μπορεί χρησιμοποιηθεί ένα Διαδικτυακό λογισμικό συνεργατικής γραφής (οργάνωση υλικού, κοινό έγγραφο με διαγράμματα, πληροφορίες και υπερσυνδέσμους σε ένα Wiki. Για παράδειγμα μπορεί να αξιοποιηθεί το εργαλείο wiki της ηλεκτρονικής σχολικής τάξης <http://eclass.sch.gr/>).

Με χρήση μελέτης περίπτωσης ο εκπαιδευτικός μπορεί να παρουσιάσει τα συστατικά μέρη ενός ΠΣ και τις σχέσεις μεταξύ τους. Στη συνέχεια μέσα από συζήτηση (ανάδραση) πρέπει να αναδειχθούν τα πλεονεκτήματα χρήσης ενός ΠΣ.

Με απλοποιημένα παραδείγματα να αναδειχθεί η διαφορά του Διαγράμματος Ροής δεδομένων ΔΡΔ (ροή δεδομένων) με το Διάγραμμα Ροής (ροή ελέγχου).

Με επίδειξη και πειραματισμό θα παρουσιαστεί λογισμικό CASE (Upper και Lower) έτσι ώστε οι μαθητές να εξοικειωθούν με μεθόδους μοντελοποίησης και επίλυσης προβλημάτων.

Με καταιγισμό ιδεών θα καλυφθούν τα θέματα ασφάλειας ενός ΠΣ, π.χ. πόροι (επιχειρηματικά σχέδια, απόρρητος κώδικας, ιδιωτικά κλειδιά κρυπτογράφησης, ιδέες, χρήματα, φήμη).

Με τις παραπάνω προτάσεις παρέχεται μαθησιακή υποστήριξη (scaffolding) και άρα προωθείται η συμμετοχή, η συνεργασία και η αυτόνομη ανάπτυξη όλων των μαθητών (ενεργά υποκείμενα μάθησης).

7.6 Χώρος

Εργαστήριο Πληροφορικής - Αίθουσα

7.7 Εποπτικά Μέσα-Διδακτικά Μέσα

Υπολογιστές με πρόσβαση στο Διαδίκτυο, Διαδραστικός ή πίνακας μαρκοδόρου, Βιντεοπροβολέας, Κατάλληλα διαμορφωμένα φύλλα εργασίας, Λογισμικό CASE, Λογισμικό εννοιολογικής χαρτογράφησης.

7.8 Ενδεικτικές Δραστηριότητες

7.8.1 Δραστηριότητα 1^η

Οι μαθητές χωρίζονται σε ομάδες των 2-3 ατόμων και αναλαμβάνουν να αναζητήσουν με τη χρήση του Διαδικτύου, παραδείγματα ΠΣ σε διαφορετικές θεματικές περιοχές (π.χ. συστήματα υποστήριξης αποφάσεων, διαχείρισης γνώσης, διαχείρισης συναλλαγών, υποστήριξης διοίκησης κλπ.). Στη συνέχεια όλες οι ομάδες μαζί οπτικοποιούν τα αποτελέσματα της έρευνάς τους με τη χρήση ενός συνεργατικού εργαλείου εννοιολογικής χαρτογράφησης (π.χ. Popplet, mindMeinder, Xmind κλπ.) και τα παρουσιάζουν στην τάξη.

7.8.2 Δραστηριότητα 2^η

Οι μαθητές εργαζόμενοι ομαδοσυνεργατικά, αφού μελετήσουν, να παρουσιάσουν τις κοινωνικές και οικονομικές επιδράσεις της Πληροφορικής και των ΠΣ καθώς επίσης και τον στρατηγικό τους ρόλο.

7.8.3 Δραστηριότητα 3^η

Οι αεροπορικές εταιρίες χρησιμοποιούν μεγάλα υπολογιστικά συστήματα και δίκτυα υπολογιστών για να μπορούν τα ταξιδιωτικά γραφεία και οι ιδιώτες να κάνουν κρατήσεις και να εκτυπώνουν το εισιτήριό τους. Στο συγκεκριμένο Πληροφοριακό Σύστημα να καταγράψετε τα βασικά του μέρη, την τεχνολογία που χρησιμοποιείται, το διαφορετικό ανθρώπινο δυναμικό και τους οργανισμούς που συμμετέχουν και το είδος της πληροφορίας που χρειάζονται (στοιχεία που απαιτούνται).

7.8.4 Δραστηριότητα 4^η

Να γίνει συζήτηση στην τάξη για τα παρακάτω:

α) Τι περιλαμβάνει η φάση συντήρησης ενός πληροφοριακού συστήματος.

β) Μετά την παράδοση ενός ΠΣ έχουν ανιχνευθεί λάθη. Έτσι ενημερώθηκε η εταιρεία ανάπτυξης του συστήματος ώστε να διορθωθούν. Να τεκμηριώσετε σε ποια φάση του κύκλου ζωής και ανάπτυξης βρίσκεται το ΠΣ.

(γ) Να εξηγηθεί σε ποια φάση του κύκλου ζωής και ανάπτυξης ενός ΠΣ ο αναλυτής δημιουργεί τα Διαγράμματα Ροής Δεδομένων (ΔΡΔ).

7.8.5 Δραστηριότητα 5^η

Να σχεδιαστεί διάγραμμα ροής δεδομένων (μοντελοποίηση συστήματος) για το εξής πρόβλημα: Ένα ξενοδοχείο θέλει να κατασκευάσει ένα πρόγραμμα το οποίο διαβάζει το αρχείο πελατών (π.χ. customer.dat) και εκτυπώνει το ενημερωμένο ποσό προσφοράς ανάλογα με τα ακόλουθα.

Δομή Αρχείου:

Κωδικός Πελάτη	Όνοματεπώνυμο	Ημέρες διαμονής	Ημερήσια χρέωση	Περίοδος*
----------------	---------------	-----------------	-----------------	-----------

*Το πεδίο περίοδος δέχεται τιμές “υψηλή”/ “χαμηλή”

Στη συνέχεια εφαρμόζει στις κρατήσεις υψηλής περιόδου οι οποίες ξεπερνούν τις πέντε μέρες διαμονής, έκπτωση 10%, αλλιώς 5%. Να εφαρμόζει στις κρατήσεις χαμηλής περιόδου οι οποίες ξεπερνούν τις πέντε μέρες έκπτωση 20%, αλλιώς 15%.

7.8.6 Δραστηριότητα 6^η

Τα παρακάτω καταγράφηκαν από αναλυτές συστημάτων στη φάση της προκαταρκτικής έρευνας:

- i) Τα τέλη κυκλοφορίας των οχημάτων θα εκτυπώνονται αποκλειστικά από το Διαδίκτυο μέσα από το Πληροφοριακό Σύστημα της Γενικής Γραμματείας Πληροφοριακών Συστημάτων. Το Πληροφοριακό Σύστημα πρέπει να τροποποιηθεί για να ακολουθήσει τα νέα δεδομένα.
- ii) Η σχεδίαση και λειτουργία ιστότοπου εκτιμάται ότι θα αυξήσει σημαντικά τους πελάτες και τις πωλήσεις της εταιρείας.
- iii) Λόγω του συνεχούς αυξανόμενου όγκου πληροφοριών, πολλές υπηρεσίες του συστήματος έγιναν αργές και δύσχρηστες, οπότε η επιχείρηση δέχεται πολλά παράπονα πελατών.
- iv) Ο κύκλος εργασιών μιας επιχείρησης έχει υπερδιπλασιαστεί και οι πελάτες σχηματίζουν ουρές στα ταμεία εξυπηρέτησης.

Να χαρακτηρίσετε ποια από τα παραπάνω αποτελούν Πρόβλημα, Ευκαιρία ή Εντολή. Τεκμηριώστε τις επιλογές σας.

7.8.7 Δραστηριότητα 7^η

Με έναν υπολογιστή χειρός, οι οδηγοί μιας εταιρείας ταχυμεταφορών εισάγουν αυτόματα τις υπογραφές των πελατών μαζί με πληροφορίες για την παραλαβή-παράδοση και τον χρόνο μεταξύ τους. Τα πληροφοριακά συστήματα της εταιρείας αξιοποιούν αυτά τα δεδομένα για να παρακολουθούν τα δέματα κατά τη μεταφορά.

Συζητήστε την παραπάνω μελέτη περίπτωσης (αναζητήστε στο Διαδίκτυο σχετικές πληροφορίες) και τεκμηριώστε τα ακόλουθα ερωτήματα:

- Ποιες είναι οι εισοδοι, η επεξεργασία και οι έξοδοι του συστήματος;
- Ποιες τεχνολογίες αξιοποιούνται; Τι σχέση έχουν αυτές οι τεχνολογίες με τον επιχειρηματικό σχεδιασμό της εταιρείας;
- Ποια προβλήματα αντιμετωπίζουν αυτές οι τεχνολογίες;

7.8.8 Δραστηριότητα 8^η

Στόχος αυτής της εργαστηριακής δραστηριότητας είναι η εξοικείωση των μαθητών με το λογισμικό ενός ΠΣ μιας επιχείρησης το οποίο υποστηρίζει: Πελάτες-Προμηθευτές, Διαχείριση Πωλήσεων, Αγορών, Αποθεμάτων, Πωλητών κλπ. Κατά τη διάρκεια αυτής της δραστηριότητας αρχικά παρουσιάζεται ο εντοπισμός και η δημιουργία των ομάδων δεδομένων πελατών, προμηθευτών και ειδών. Έπειτα οι μαθητές προσεγγίζουν να κατανοήσουν την αρχιτεκτονική του συστήματος και να αντιληφθούν τη σημασία της ασφαλούς διαχείρισης και διακίνησης της πληροφορίας σε ένα ΠΣ που λειτουργεί εντός μιας επιχείρησης.

Θέματα για συζήτηση:

1. Εντοπίστε τα δυνατά και αδύνατα σημεία του συστήματος.
2. Αξιολογήστε το σύστημα με τα εξής κριτήρια: Φιλικότητα διεπαφής - Ευχρηστία - Αξιοπιστία - Ταχύτητα - Περιεκτικότητα - Διαλειτουργικότητα - Ασφάλεια.
3. Να γραφούν τα αρχεία τεκμηρίωσης που αφορούν στα δεδομένα του ΠΣ.

7.8.9 Δραστηριότητα 9^η

Σκοπός της δραστηριότητας είναι η σύνταξη μιας μελέτης σκοπιμότητας για ένα ΠΣ ενός οργανισμού. Για παράδειγμα, ένα Σύστημα Ηλεκτρονικού Εμπορίου: Μία εταιρεία πώλησης ηλεκτρονικών προϊόντων ζητά την ανάπτυξη ΠΣ. Το πληροφοριακό αυτό σύστημα θα πρέπει να καλύπτει τις απαιτήσεις εσωτερικής λειτουργίας, όπως η διαχείριση αποθεμάτων και παραγγελιών της επιχείρησης, αλλά και της πώλησης ειδών μέσω Διαδικτύου. Η εταιρεία διαθέτει προϊόντα όπως ΗΥ, smartphones, ψηφιακές κάμερες κ.ά.

Οδηγία: μπορείτε να ακολουθήσετε τα εξής ενδεικτικά βήματα:

- Επιλογή προβλήματος που σχετίζεται με έναν οργανισμό (όπως τον παραπάνω) το οποίο πρέπει να είναι κατάλληλο για εφαρμογή των μεθόδων ανάλυσης και σχεδίασης ΠΣ.
- Ορισμός της εμβέλειας του προβλήματος.
- Ορισμός ενός συνόλου διάφορων λύσεων για την επίλυση του προβλήματος.
- Ορισμός των κριτηρίων αξιολόγησης των εναλλακτικών λύσεων, αξιολόγηση τους και επιλογή της καταλληλότερης.
- Για την λύση που επιλέξατε πρέπει να συντάξετε: Ένα σύνολο από βασικές περιπτώσεις χρήσης - διάγραμμα κλάσεων που να εξηγεί τις βασικές έννοιες του πεδίου εφαρμογής και τις σχέσεις τους - διαγράμματα δραστηριοτήτων - σύνταξη αναφοράς.

7.9 Βιβλιογραφία

- Avgerou, C., Cornford, T. (1993) "Developing Information Systems Concepts, Issues and Practice", Macmillan Information Systems Series.
- Laudon, C., K., Laudon, J. (2012) "Essentials of Management Information Systems" 10th edition, Prentice Hall.

7.10 Πηγές - Πρόσθετο Υλικό - Χρήσιμοι Εκπαιδευτικοί Πόροι

Μοντέλο πυραμίδας τεσσάρων επιπέδων για τους διάφορους τύπους Π.Σ.

http://en.wikipedia.org/wiki/Information_system#mediaviewer/File:Four-Level-Pramid-model.png

Εργαλεία εννοιολογικής χαρτογράφησης

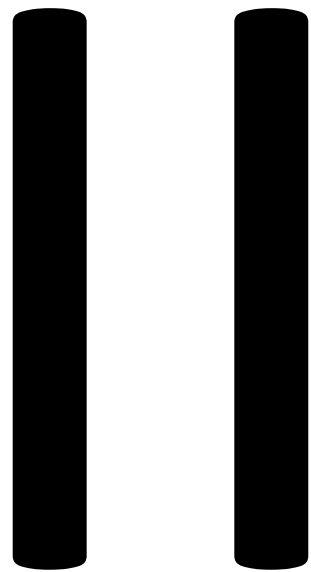
<http://popplet.com/>

<http://www.mindmeister.com/>

<https://www.xmind.net/>

Συνοδευτική ιστοσελίδα για το βιβλίο: Laudon, K.C. and J.P. Laudon, Essentials of Management Information Systems, 10η έκδοση με πολλά παραδείγματα (case studies): http://wps.prenhall.com/bp_laudon_essmis_10/219/56286/14409392.cw/index.html (τελευταία προσπέλαση 10/01/2015)

Μέρος



Αναλυτικές δραστηριότητες

Φιλοσοφία των δραστηριοτήτων

Οι δραστηριότητες που δίνονται στη συνέχεια ακολουθούν μια διερευνητική/ανακαλυπτική προσέγγιση σύμφωνα με την οποία οι μαθητές κατακτούν τους διδακτικούς στόχους του μαθήματος μέσα από τον πειραματισμό με το προγραμματιστικό περιβάλλον. Οι μαθητές είτε επιλύουν προβλήματα ως μελέτες περίπτωσης είτε τους δίνονται τμήματα κώδικα για να πειραματιστούν με αυτά και να εξάγουν συμπεράσματα που θα τους βοηθήσουν να σχηματίσουν τα θεωρητικά μοντέλα (αλγόριθμοι) που θέλουμε. Η μελέτη των αλγόριθμων με διερευνητικό τρόπο, χρειάζεται υπομονή, επιμονή και χρόνο. Για αυτό το λόγο οι ώρες που προτείνονται σε κάποιες δραστηριότητες είναι ενδεικτικές.

Τα σχέδια των δραστηριοτήτων που ακολουθούν είναι δομημένα σε τρία μέρη:

- Το περίγραμμα της δραστηριότητας, που περιέχει πληροφορίες όπως ο προτεινόμενος χρόνος, τα εποπτικά μέσα, τα προσδοκώμενα μαθησιακά αποτελέσματα κ.ά.
- Τις οδηγίες προς τους εκπαιδευτικούς όπου προτείνονται διάφορες διδακτικές στρατηγικές για την εισαγωγή των μαθητών στις νέες αυτές έννοιες της επιστήμης των υπολογιστών, με βάση το φύλλο εργασίας που ακολουθεί.
- Το προτεινόμενο φύλλο εργασίας που δίνεται στους μαθητές (σε έντυπη ή ηλεκτρονική μορφή).

Όλα τα προγράμματα σε Python έχουν αναπτυχθεί στην έκδοση 2.7.9 της γλώσσας, ενώ μπορούν να μεταφερθούν και στην Python 3 με ελάχιστες αλλαγές.

Προτείνεται ο διδάσκων να καταγράψει τις χρήσιμες συναρτήσεις που δίνονται στις δραστηριότητες, όπως οι `trace`, `random_array`, `timer` και να τις συγκεντρώσει σε ένα άρθρωμα (module) το οποίο θα επεκτείνεται σταδιακά και θα αποτελέσει τη βιβλιοθήκη προγραμμάτων του μαθήματος.

1. Σχέδιο Δραστηριότητας: Η έννοια του Αλγορίθμου

Προτεινόμενος χρόνος:	1 διδακτική ώρα
Περιγραφή:	Εισαγωγή στην έννοια του αλγορίθμου μέσα από την ενασχόληση με ένα αλγοριθμικό πρόβλημα, όπου οι μαθητές πρέπει να περιγράψουν τον αλγόριθμο επίλυσής του σε μια συγκεκριμένη γλώσσα
Ενότητα:	Από το πρόβλημα στον Αλγόριθμο
Εμπλεκόμενες Έννοιες:	Αλγόριθμος, Υπολογιστική Σκέψη, Πρόγραμμα

Διδακτική Προσέγγιση:	Ερωταποκρίσεις, ανακαλυπτική μάθηση, καταιγισμός ιδεών
Εποπτικά μέσα	Βιντεοπροβολέας Λογισμικό προσομοίωσης/οπτικοποίησης του προβλήματος από το Διαδίκτυο http://coolmath4kids.com/math_puzzles/Logic-waterjars/index.html
Χώρος	Αίθουσα (ή και εργαστήριο)

1.1 Προσδοκώμενα μαθησιακά αποτελέσματα

Οι μαθητές μετά το τέλος της δραστηριότητας αναμένεται:

- να αναλύουν το πρόβλημα σε άλλα απλούστερα
- να ανάγουν το πρόβλημα σε άλλα απλούστερα προβλήματα
- να διακρίνουν ποια είναι τα επιτρεπτά βήματα που μπορούν να τους οδηγήσουν στην επίλυση του προβλήματος
- να μπορούν να περιγράψουν τον αλγόριθμο επίλυσης του προβλήματος στη γλώσσα της νοητής μηχανής που τους δίνεται
- να αντιληφθούν την αναγκαιότητα των γλωσσών προγραμματισμού, ως γλώσσες αυστηρής αναπαράστασης των αλγορίθμων

1.2 Οδηγίες για τον εκπαιδευτικό

Η δραστηριότητα αυτή έχει διττό σκοπό. Καταρχήν να έρθουν σε επαφή οι μαθητές με τη διαδικασία ανάλυσης και επίλυσης ενός προβλήματος που επιδέχεται αλγοριθμική λύση. Αφού σκιαγραφήσουν τα βήματα του αλγορίθμου που οδηγούν στην επιθυμητή λύση του, στη συνέχεια θα πρέπει να γράψουν ένα πρόγραμμα στη γλώσσα της νοητής μηχανής που έχει οριστεί.

Αν οι μαθητές δυσκολεύονται να ξεκινήσουν την επίλυση του προβλήματος, πιθανόν επειδή δεν έχουν κατανοήσει την εκφώνηση, προτείνεται να γίνει επίδειξη 2-3 εντολών, ώστε να αντιληφθούν ποιες είναι οι επιτρεπτές εντολές. Εδώ ίσως να φανεί χρήσιμη η οπτικοποίηση μέσω κατάλληλου λογισμικού από το Διαδίκτυο.

Προτείνεται να τονιστεί από τον εκπαιδευτικό ότι κατά την επίλυση του προβλήματος θα πρέπει να χρησιμοποιηθούν μόνο οι τρεις εντολές και ότι τα δοχεία μπορούν να γεμίζουν μόνο

μέχρι πάνω. Επίσης να δοθεί προσοχή ώστε η σύνταξη των εντολών να είναι σύμφωνα με τις προδιαγραφές που δίνονται.

Στο τέλος να υπολογίσουν οι μαθητές το πλήθος των βημάτων του αλγορίθμου. Σε περίπτωση που κάποιοι μαθητές σχεδιάσουν και δεύτερο αλγόριθμο να γίνει σύγκριση των δυο αλγορίθμων όσον αφορά το πλήθος των βημάτων.

1.2.1 Φύλλο εργασίας: Εισαγωγή στην έννοια του αλγόριθμου

ΟΝΟΜΑ ΕΠΩΝΥΜΟ.....

ΤΑΞΗ..... ΤΜΗΜΑ..... ΗΜ/ΝΙΑ:

Μετρήστε ακριβώς 6 λίτρα, αν έχετε στην διάθεσή σας ένα δοχείο των 5, ένα των 7 λίτρων και μια πηγή με άφθονο νερό. Μπορείτε μόνο να γεμίζετε μέχρι πάνω και να αδειάζετε εντελώς τα δοχεία όσες φορές θέλετε.

Για να πειραματιστείτε με την οπτικοποίηση του προβλήματος, μπορείτε να μεταβείτε στην παρακάτω ιστοσελίδα: http://coolmath4kids.com/math_puzzles/Logic-waterjars/index.html

Σας δίνετε μια γλώσσα προγραμματισμού για το πρόβλημα των δοχείων η οποία έχει τις εξής τρεις εντολές:

Γέμιση(A) : Γεμίζει το δοχείο A από τη βρύση, π.χ. η εντολή **Γέμιση(5)** γεμίζει το 5λιτρο

Άδειασε(A): Αδειάζει το δοχείο A κάτω, π.χ. η εντολή **Άδειασε(5)** αδειάζει το 5λιτρο

Μετακίνηση(A, B): Μετακινεί το περιεχόμενο του δοχείου A στο δοχείο B. π.χ. η εντολή

Μετακίνηση(5 , 7) θα μπορούσε να έχει τα παρακάτω αποτελέσματα ανάλογα με το περιεχόμενο των δοχείων εκείνη τη στιγμή:

1. Το 5λιτρο είναι γεμάτο και το 7λιτρο είναι άδειο. Σε αυτή την περίπτωση το 5λιτρο θα αδειάσει εντελώς και τα 5 λίτρα θα μετακινηθούν στο 7λιτρο το οποίο θα θέλει ακόμα 2 λίτρα για να γεμίσει.
2. Το 5λιτρο είναι γεμάτο και το 7λιτρο έχει 4 λίτρα. Σε αυτή την περίπτωση αδειάζουμε μέχρι να γεμίσει το 7λιτρο. Άρα θα μετακινηθούν 3 λίτρα ($3+4 = 7$) στο 7λιτρο και θα μείνουν 2 λίτρα στο 5λιτρο.

Να αναπτύξετε αλγόριθμο στην παραπάνω γλώσσα προγραμματισμού, έτσι ώστε να μείνουν ακριβώς 6 λίτρα στο 7λιτρο.

Δραστηριότητες για το σπίτι

Δραστηριότητα Σ1

Αν είχατε στην διάθεσή σας ένα δοχείο των 3 και ένα των 5 λίτρων να δώσετε έναν αντίστοιχο αλγόριθμο έτσι ώστε να μείνουν ακριβώς 4 λίτρα στο 5λιτρο.

Δραστηριότητα Σ2 (ο γρίφος του νέου κόσμου)

Τέσσερις άνθρωποι θέλουν να διασχίσουν μια γέφυρα. Αρχικά, βρίσκονται όλοι στην ίδια πλευρά της. Υπάρχει ένα όριο των 17 λεπτών μετά το οποίο όλοι θα πρέπει να βρίσκονται στην άλλη άκρη της γέφυρας. Είναι νύχτα και διαθέτουν μόνο ένα φακό. Το πολύ δύο άτομα μπορούν να διασχίσουν ταυτόχρονα τη γέφυρα, σε κάθε χρονική στιγμή. Όποια ομάδα διασχίζει τη γέφυρα, είτε του ενός είτε των δυο ατόμων, θα πρέπει να έχει αυτή το φακό. Ο πρώτος χρειάζεται 1 λεπτό για να διασχίσει τη γέφυρα, ο δεύτερος 2 λεπτά, ο τρίτος 5 λεπτά και ο τέταρτος χρειάζεται 10 λεπτά. Δυο άνθρωποι μαζί, κινούνται με την ταχύτητα του βραδύτερου από αυτούς. Η νοητή μηχανή στην οποία θα εκτελεστεί ο αλγόριθμός σας έχει μόνο μια εντολή:

Διάσχιση_Γέφυρας(A, B): Ο A και ο B διασχίζουν τη γέφυρα στον χρόνο του βραδύτερου

Διάσχιση_Γέφυρας(X): Ο X διασχίζει τη γέφυρα στον δικό του χρόνο

2. Σχέδιο Δραστηριότητας: Αριθμομηχανή

Προτεινόμενος Χρόνος:	2 διδακτικές ώρες
Περιγραφή:	Χρήση του διερμηνευτή της γλώσσας σαν αριθμομηχανή για σύνθετους υπολογισμούς σε διάφορους τύπους δεδομένων, αλφαριθμητικά, αριθμούς (ακέραιους, κινητής υποδιαστολής),
Ενότητα:	Γνωριμία με τη γλώσσα προγραμματισμού Η γλώσσα Python ως αριθμομηχανή
Εμπλεκόμενες Έννοιες:	Τύποι δεδομένων, διερμηνευτής της Python
Διδακτική Προσέγγιση:	Πειραματισμός με το περιβάλλον προγραμματισμού, διερευνητική μάθηση
Εποπτικά μέσα:	Προβολέας, υπολογιστές, περιβάλλον προγραμματισμού
Χώρος:	Εργαστήριο Πληροφορικής

2.1 Προσδοκώμενα μαθησιακά αποτελέσματα

Οι μαθητές μετά το τέλος της δραστηριότητας αναμένεται:

- να εξοικειωθούν με το περιβάλλον του διερμηνευτή εντολών της γλώσσας προγραμματισμού
- να πειραματιστούν με τους αριθμητικούς και λογικούς τελεστές της γλώσσας
- να υλοποιήσουν πράξεις πολύ μεγάλων αριθμών
- να μελετήσουν τον τρόπο με τον οποίο η Python αναπαριστά μιγαδικούς αριθμούς

2.2 Οδηγίες για τον εκπαιδευτικό

Είναι αναγκαίο να έχουν ήδη αναφερθεί οι βασικοί τύποι της Python (int, float, long, bool, str) και να έχει δοθεί από ένα παράδειγμα για τον καθένα.

Αρχικά γίνεται μια σύντομη παρουσίαση των αριθμητικών, συγκριτικών και λογικών τελεστών της Python και δίνεται από ένα παράδειγμα για κάθε περίπτωση. Στη συνέχεια δίνεται στους μαθητές το φύλλο εργασίας ώστε να εξοικειωθούν μέσω πειραματισμού με το διερμηνευτή της γλώσσας.

Με βάση τα αποτελέσματα των εντολών προτείνεται να καθοδηγήσουμε τους μαθητές στην εξαγωγή συμπερασμάτων σχετικά με τη λειτουργία κάποιων τελεστών, όπως για παράδειγμα θα πρέπει να συζητηθεί η διαφορά μεταξύ του 3/2 και του 3.0/2.

Όσον αφορά τους μιγαδικούς αριθμούς οι μαθητές θα πρέπει να χρησιμοποιήσουν την εντολή `help(complex)` για να δουν τις διαθέσιμες μεθόδους του τύπου `complex`, αλλά και ξεχωριστά για κάθε μέθοδο, π.χ. `help(complex.conjugate)`.

Στο τέλος προτείνεται να δοθεί χρόνος στους μαθητές για να πειραματιστούν μόνοι τους με το διερμηνευτή της Python, υπολογίζοντας τις δικές τους παραστάσεις.

2.3 Φύλλο εργασίας: Αριθμομηχανή

ΟΝΟΜΑ ΕΠΩΝΥΜΟ.....

ΤΑΞΗ..... ΤΜΗΜΑ..... ΗΜ/ΝΙΑ:

Ανοίξτε το περιβάλλον προγραμματισμού IDLE

```

Python 2.7.9 Shell
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> z = 1j
>>> z*z
(-1+0j)
>>> 0b10000000
128
>>> bin(128)
'0b10000000'
>>> hex(128)
'0x80'
>>> 2**1000
10715086071862673209484250490600018105614048117055336074437503883703510511
24936122493198378815695858127594672917553146825187145285692314043598457757
46985748039345677748242309854210746050623711418779541821530464749835819412
67398767559165543946077062914571196477686542167660429831652624386837205668
069376L
>>>
  
```

Να εκτελέσετε τις παρακάτω εντολές στο διερμηνευτή της γλώσσας και να απαντήσετε στα ερωτήματα που ακολουθούν.

Αριθμοί και Τύποι

>>> 1+1	>>> 18+10	>>> type(28)	>>> 2**4
>>> '1' + '1'	>>> 18+10.0	>>> type(3.14)	>>> 4**0.5
>>> 16/3	>>> 7.0/2	>>> type("Python")	>>> 2**1000
>>> 16.0/3	>>> 7/2	>>> type(True)	>>> 2**100000
>>> 16%3	>>> 7%2	>>> type(1j)	>>> 2**(-1)

Αλφαριθμητικά

>>> "Python" + " 3"	>>> name[0]	>>> name[1:]
>>> 5 * "Python"	>>> name[1]	>>> name[3:]
>>> len("Python ")	>>> name[3]	>>> name[6:]
>>> len("monty Python")	>>> name[1:4]	>>> name[:5]
>>> name = 'monty Python'		>>> name[:]

Δυαδικοί και Δεκαεξαδικοί αριθμοί

>>> 0b100	>>> bin(4)	>>> 0x10	>>> hex(256)
>>> 0b10	>>> bin(6)	>>> 0x100	>>> hex(16)
>>> 0b11	>>> bin(128)	>>> 0xA	>>> 0b1 + 0x1
>>> 0b101	>>> bin(2**10)	>>> 0xF	>>> 0b10*0x10

>>> 0b100000	>>> bin(2**5)	>>> 0x1F	
--------------	---------------	----------	--

Μιγαδικοί αριθμοί

>>> 1j**2	>>> z = 1 - 1j	>>> z.conjugate()
>>> (1-2j)*(1+2j)	>>> z.real	>>> z**2
>>> abs(3+4j)	>>> z.imag	>>> z**16

ΕΡΩΤΗΜΑΤΑ

Τι συμπεράσματα βγάξετε από τα αποτελέσματα;

.....

Ποια από αυτά σας προκάλεσαν εντύπωση;

.....

Ποια είναι η λειτουργία των τελεστών που χρησιμοποιούνται;

.....

3. Σχέδιο Δραστηριότητας: Εισαγωγή στις συναρτήσεις της Python

Προτεινόμενος Χρόνος:	1 διδακτική ώρα
Περιγραφή:	Εισαγωγή στις συναρτήσεις της Python
Ενότητα:	Συναρτήσεις
Εμπλεκόμενες Έννοιες:	Συναρτήσεις, προγραμματισμός σε Python
Διδακτική Προσέγγιση:	Διερευνητική μάθηση, Πειραματισμός με το περιβάλλον προγραμματισμού
Εποπτικά μέσα:	Βιντεοπροβολέας, υπολογιστές, προγραμματιστικό περιβάλλον
Χώρος:	Εργαστήριο Πληροφορικής

3.1 Προσδοκώμενα μαθησιακά αποτελέσματα

Οι μαθητές

- να γενικεύουν στιγμιότυπα συναρτήσεων σε μια γενικευμένη συνάρτηση
- να ορίζουν τις δικές τους συναρτήσεις
- να διακρίνουν τα βασικά σημεία στον ορισμό μιας συνάρτησης
- να κατανοούν τη σημασία των παραμέτρων στις συναρτήσεις.

3.2 Οδηγίες για τον εκπαιδευτικό

Ο σκοπός του πρώτου βήματος της δραστηριότητας είναι να κατανοήσει ο μαθητής ότι μια συνάρτηση χωρίς ορίσματα δεν είναι τίποτα παραπάνω από τον χαρακτηρισμό ενός τμήματος κώδικα με κάποιο όνομα για να μην επαναλαμβάνεται.

Στη συνέχεια οι μαθητές χρησιμοποιούν διάφορα στιγμιότυπα της συνάρτησης `printPython(N)`, ώστε να φανεί η ανάγκη για γενίκευση όλων αυτών των στιγμιότυπων με τη χρήση της παραμέτρου `N`.

Η εκχώρηση της συνάρτησης `printPython` σε μια μεταβλητή θα προκαλέσει λάθος εκτέλεσης. Εδώ θα πρέπει να φανεί η ανάγκη χρήσης της εντολής `return` για επιστροφή αποτελέσματος. Αυτό φαίνεται καλύτερα για μαθηματικές συναρτήσεις, όπως είναι η *root* που δίνεται παρακάτω. Στο σημείο αυτό προτείνεται να τονιστεί η δυνατότητα των συναρτήσεων να χρησιμοποιηθούν σε σύνθετες εμφωλευμένες συναρτησιακές εκφράσεις. Ένα παράδειγμα για να φανεί αυτό είναι ο υπολογισμός της 4^{ης} της 6^{ης} ή της 8^{ης} τετραγωνικής ρίζας ενός αριθμού.

Στη συνέχεια οι μαθητές καλούνται να σχεδιάσουν τη δική τους συνάρτηση για τον υπολογισμό της διακρίνουσας μιας δευτεροβάθμιας εξίσωσης και να την χρησιμοποιήσουν για την επίλυσή της. Εδώ πρέπει να συζητηθεί το γεγονός της χρήσης της συνάρτησης δυο φορές, μια για κάθε ρίζα, ώστε να αποφύγουμε την επανάληψη του ίδιου τμήματος κώδικα.

3.3 Φύλλο εργασίας: Εισαγωγή στις Συναρτήσεις της Python

ΟΝΟΜΑ ΕΠΩΝΥΜΟ.....

ΤΑΞΗ..... ΤΜΗΜΑ..... ΗΜ/ΝΙΑ:

Δραστηριότητα

Μέρος 1^ο:

Ανοίξτε ένα νέο αρχείο στην Python στο οποίο θα προσθέσετε τους ορισμούς των συναρτήσεων που δίνονται παρακάτω.

Ορίστε τη συνάρτηση `printPython3`, ώστε να εμφανίζει τη λέξη Python τρεις (3) φορές. Στη συνέχεια ορίστε τη συνάρτηση `printPython9`, ώστε να εμφανίζει τη λέξη Python εννέα (9) φορές

<pre>def printPython3(): print('Python') print('Python') print('Python') >>> Python3() Python Python Python</pre>	<pre>def printPython9(): print('Python') print('Python') print('Python') print('Python') print('Python') print('Python') print('Python') print('Python') print('Python')</pre>
--	--

Ερωτήσεις:

Πώς μπορούμε να γράψουμε τη συνάρτηση `printPython9` χρησιμοποιώντας λιγότερες εντολές;

.....

Πώς μπορούμε να ορίσουμε μια συνάρτηση η οποία να εμφανίζει 21 φορές τη λέξη Python με αποκλειστική χρήση των συναρτήσεων `printPython3()` και `printPython9()`.

.....

Μέρος 2^ο . (η ανάγκη για γενίκευση)

Δίνονται οι παρακάτω συναρτήσεις σε Python

<pre>def printPython3(): for i in range(3): print('Python')</pre>	<pre>def printPython12(): for i in range(12): print('Python')</pre>	<pre>def printPython100(): for i in range(100): print('Python')</pre>
---	---	---

Συμπληρώστε κατάλληλα τον ορισμό της παρακάτω συνάρτησης `printPython` ώστε να εμφανίζει τη λέξη Python N φορές, και στη συνέχεια δώστε τις διπλανές κλήσεις στον διερμηνευτή:

<pre>>>> def printPython(____): for i in range(N): print('Python')</pre>	<pre>>>> printPython(3) >>> printPython(9) >>> printPython(21) >>> printPython(100)</pre>
---	---

Ερωτήσεις:

Ως προς τι διαφέρει η τελευταία συνάρτηση από όλες τις προηγούμενες;

.....

Ποια είναι η σχέση της με αυτές;

.....

Δώστε στον διερμηνευτή την εντολή `b = printPython(2)`. Τι παρατηρείτε; Προσπαθήστε να αιτιολογήσετε το αποτέλεσμα.

.....

Μέρος 3^ο

Δοκιμάστε τις παρακάτω εντολές στο διερμηνευτή της γλώσσας Python και στη συνέχεια προσπαθήστε να απαντήσετε τις ερωτήσεις της δραστηριότητας:

<pre>>>> import math >>> def root(number): >>> return math.sqrt(number) >>> root(16) >>> root(2)</pre>	<pre>>>> a = root(16) >>> print(a) >>> print(root(a)) >>> print(root(root(16)))</pre>
---	---

Ερωτήσεις:

Περιγράψτε τι παρατηρείτε;

.....

Τι πιστεύετε ότι κάνει η εντολή `return`;

.....

Εργασία (για το σπίτι): Να ορίσετε συνάρτηση για τον υπολογισμό της διακρίνουσας $\Delta = \beta^2 - 4\alpha\gamma$ της εξίσωσης $ax^2 + \beta x + \gamma = 0$, και στη συνέχεια να αναπτύξετε πρόγραμμα για την επίλυσή της.

4. Σχέδιο Δραστηριότητας: Παράμετροι και εμβέλεια συναρτήσεων

Προτεινόμενος Χρόνος:	2 διδακτικές ώρες
Περιγραφή:	Δραστηριότητα σχετικά με την μεταβίβαση παραμέτρων και την εμβέλεια των τοπικών μεταβλητών στις συναρτήσεις
Ενότητα:	Συναρτήσεις
Εμπλεκόμενες Έννοιες:	Συναρτήσεις, παράμετροι, ορίσματα, τοπικές μεταβλητές
Διδακτική Προσέγγιση:	Διερευνητική μάθηση, Πειραματισμός με το περιβάλλον προγραμματισμού
Εποπτικά μέσα:	Βιντεοπροβολέας, υπολογιστές, προγραμματιστικό περιβάλλον
Χώρος:	Εργαστήριο Πληροφορικής

4.1 Προσδοκώμενα μαθησιακά αποτελέσματα

Οι μαθητές

- να εξοικειωθούν με τον πολυμορφισμό συναρτήσεων και το δυναμικό σύστημα τύπων της Python
- να κατανοήσουν τον μηχανισμό μεταβίβασης παραμέτρων στις συναρτήσεις της Python
- να μελετήσουν τον τρόπο μεταβίβασης λιστών ως παραμέτρους σε συναρτήσεις
- να κατανοήσουν τους κανόνες εμβέλειας τοπικών και καθολικών μεταβλητών στην Python.

4.2 Οδηγίες για τον/την εκπαιδευτικό

Στην πρώτη δραστηριότητα τονίζεται στους μαθητές η υπερφόρτωση συναρτήσεων ή πολυμορφισμός λόγω του συστήματος δυναμικών τύπων της Python. Μια συνάρτηση αρκεί να οριστεί μία φορά και μετά μπορεί να χρησιμοποιηθεί για όλους τους τύπους. Η απόφαση για το ποια έκδοση της συνάρτησης ή του τελεστή θα κληθεί καθορίζεται δυναμικά σε χρόνο εκτέλεσης.

Να γίνει σαφές στους μαθητές ότι όταν περνάμε απλούς τύπους ως παραμέτρους σε συναρτήσεις, το πέρασμα γίνεται κατά τιμή, δηλαδή δημιουργείται αντίγραφο, οπότε οι αλλαγές μέσα στη συνάρτηση δεν επηρεάζουν τη μεταβλητή-όρισμα.

Αντίθετα οι λίστες που είναι πιο σύνθετα αντικείμενα δεν αντιγράφονται ολόκληρες, αλλά μόνο η αναφορά στη θέση μνήμης που είναι αποθηκευμένες. Οι τυχόν αλλαγές σε στοιχεία τους περνάνε πίσω στα ορίσματα με τα οποία έγινε η κλήση της συνάρτησης. Το αντίγραφο δηλαδή είναι επιφανειακό (shallow) και όχι λεπτομερές (deep).

Μέσω παραδειγμάτων να φανεί ότι δεν είναι καλή πρακτική να χρησιμοποιούμε μεταβλητές που δεν έχουν τοπική εμβέλεια.

Μια μεταβλητή δημιουργείται την πρώτη φορά που θα εμφανιστεί στο αριστερό μέρος μιας εκχώρησης. Μια τέτοια εντολή μέσα στη συνάρτηση δημιουργεί μια τοπική μεταβλητή η οποία επικαλύπτει την μεταβλητή που έχει οριστεί έξω από τη συνάρτηση.

4.3 Φύλλο εργασίας: Παράμετροι και Εμβέλεια Συναρτήσεων

ΟΝΟΜΑ ΕΠΩΝΥΜΟ.....
ΤΑΞΗ..... ΤΜΗΜΑ..... ΗΜ/ΝΙΑ:

4.3.1 Δραστηριότητα 1

Ανοίξτε ένα νέο αρχείο στην Python για να προσθέσετε τους ορισμούς των συναρτήσεων που δίνονται παρακάτω. Στη συνέχεια δώστε στον διερμηνευτή τις εντολές της 2^η στήλης:

Δημιουργία της συνάρτησης	Πειραματισμός στο περιβάλλον Python Shell με άμεση εκτέλεση εντολών
<pre>def add(arg1, arg2): result = arg1+arg2 return result def times3(arg): result = 3*arg return result</pre>	<pre>>>> add(10, 18) >>> add(10, 18.5) >>> add('Python', '3.4') >>> times3(100) >>> times3(2.5) >>> times3('Python')</pre>

Μελετήστε τα αποτελέσματα και καταγράψτε τα συμπεράσματά σας για τις συναρτήσεις και τους αριθμητικούς τύπους και τα αλφαριθμητικά.

.....
.....

4.3.2 Δραστηριότητα 2

Ερωτήματα:

Τι πιστεύετε ότι κάνει η παρακάτω συνάρτηση;

.....

Ποιες θα είναι οι τελικές τιμές των c και d;

.....

Πειραματιστείτε στο προγραμματιστικό περιβάλλον της Python ώστε να επαληθεύσετε τους ισχυρισμούς σας.

Δημιουργία της συνάρτησης	Πειραματισμός στο περιβάλλον Python Shell με άμεση εκτέλεση εντολών
<pre>def swap(a, b): temp = a a = b b = temp print(a, b)</pre>	<pre>>>> c = 2; d = 10 >>> swap(c, d) >>> print(c, d) >>> m = [1,2,3]; k = [4,5,6] >>> print(m); print(k) >>> swap(m, k)</pre>

Ερωτήσεις:

Πως εξηγείτε τα αποτελέσματα;

.....

Τι συμπέρασμα βγάζετε για τον μηχανισμό μεταβίβασης παραμέτρων στην Python;

.....

4.3.3 Δραστηριότητα 3

Δοκιμάστε να εκτελέσετε την παρακάτω συνάρτηση στον διερμηνευτή της Python για διάφορα δεδομένα εισόδου. Στη συνέχεια προσπαθήστε να απαντήσετε τις ερωτήσεις της δραστηριότητας:

Δημιουργία της συνάρτησης	Πειραματισμός στο περιβάλλον Python Shell με άμεση εκτέλεση εντολών
<pre>def swap0(a, b): temp = a[0] a[0] = b[0] b[0] = temp print(a, b)</pre>	<pre>>>> m = [1,2,3]; k = [4,5,6] >>> swap0(m, k) >>> print(m); print(k)</pre>

Ερωτήσεις: Ποιες διαφορές παρατηρείτε ως προς τα αποτελέσματα της δραστηριότητας 2. Προσπαθήστε να τεκμηριώσετε την άποψη σας.

.....

.....

4.3.4 Δραστηριότητα 4

Τι αναμένετε να εμφανιστεί στην οθόνη κατά την εκτέλεση των παρακάτω προγραμμάτων; Εκτελέστε τα παρακάτω τμήματα κώδικα και εξηγήστε τα αποτελέσματα. Ήταν αυτά που αναμένατε; Ποια ερμηνεία δίνετε;

<pre>>>> myglobal = 496 >>> def function(b): local = 28 print(myglobal) myglobal = 8128 print(myglobal) >>> print(myglobal) >>> print(local)</pre>	<pre>>>> myglobal = 496 >>> def function2(b): print(myglobal) myglobal = myglobal + 1 print(myglobal) >>> print(myglobal)</pre>
--	--

5. Σχέδιο Δραστηριότητας: Δομή Επιλογής

Προτεινόμενος χρόνος:	2 διδακτικές ώρες
Περιγραφή:	Έλεγχος αν το σημείο που επιλέγει με το ποντίκι ο χρήστης είναι εντός ενός ορθογωνίου παραλληλογράμμου.
Ενότητα:	Αλγοριθμικές Δομές / Δομή Επιλογής
Εμπλεκόμενες Έννοιες:	Δομή επιλογής, προγραμματισμός σε Python
Διδακτική Προσέγγιση:	Ανακαλυπτική/διερευνητική μάθηση, καταγισμός ιδεών, πειραματισμός με το προγραμματιστικό περιβάλλον
Εποπτικά μέσα	Η/Υ, Προβολέας, Περιβάλλον Προγραμματισμού IDLE
Χώρος	Εργαστήριο Πληροφορικής

5.1 Προσδοκώμενα μαθησιακά αποτελέσματα

Ο μαθητές

- να χρησιμοποιούν τη δομή επιλογής για επίλυση προβλημάτων με σύνθετες συνθήκες με λογικούς τελεστές
- να διακρίνουν περιπτώσεις και να τις κωδικοποιούν με την αντίστοιχη δομή επιλογής
- να τροποποιούν και να συμπληρώνουν κώδικα που τους δίνεται, με συγκεκριμένες προδιαγραφές
- να πειραματιστούν και να τροποποιήσουν τον κώδικα μιας απλής γραφικής διεπαφής στην Python και να εξοικειωθούν με τον προγραμματισμό οδηγούμενο από γεγονότα.

5.2 Οδηγίες για τον εκπαιδευτικό

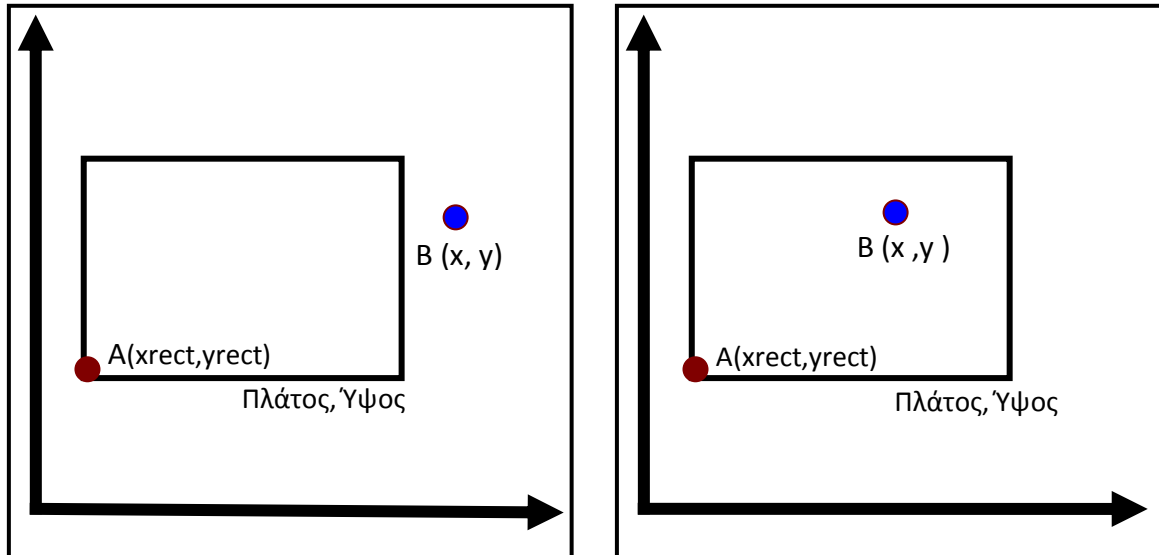
Οι μαθητές γνωρίζουν ήδη τη δομή επιλογής, από το μάθημα Εισαγωγή στις Αρχές της Επιστήμης Υπολογιστών της Β΄ Λυκείου. Ωστόσο, πριν από αυτή την δραστηριότητα πρέπει να προηγηθεί μια παρουσίαση της δομής `if..else` και της δομής `if..elif..else` της Python. Επίσης θα πρέπει να παρουσιαστούν με απλά παραδείγματα από τον καθηγητή οι λογικοί τελεστές **and**, **not**, **or** σε αντιστοιχία με τους τελεστές **και**, **όχι**, **ή** της ψευδογλώσσας που ήδη γνωρίζουν οι μαθητές.

Ιδιαίτερη προσοχή προτείνεται να δοθεί στο θέμα της εσοχής κώδικα που καθορίζει σε ποιον κλάδο της δομής επιλογής ανήκουν οι εντολές.

Τέλος οι μαθητές θα πρέπει να μελετήσουν το αρχείο κώδικα `testRectangle.py` και να εντοπίσουν το τμήμα του κώδικα που χειρίζεται το ποντίκι. Στη συνέχεια να πειραματιστούν με αυτό ώστε να αποκτήσουν μια εικόνα του προγραμματισμού οδηγούμενου από γεγονότα.

5.3 Φύλλο εργασίας: Δομή Επιλογής

Στα γραφικά υπολογιστών είναι πολλές φορές απαραίτητο να γνωρίζουμε αν ένα σημείο βρίσκεται εντός ή εκτός ενός ορθογωνίου παραλληλογράμμου. Να φτιάξετε έναν αλγόριθμο που θα εκτελεί αυτόν τον έλεγχο, όπως φαίνεται στο παρακάτω σχήμα:



Κάθε παραλληλόγραμμο αναπαρίσταται από 4 στοιχεία: Τις συντεταγμένες (x,y) της κάτω αριστερά γωνίας, το πλάτος (στον οριζόντιο άξονα) και το ύψος (στον κάθετο άξονα). Στο σχήμα βλέπουμε ότι στην πρώτη περίπτωση το σημείο B είναι εκτός του ορθογωνίου ενώ στη 2η εντός.

Βήμα 1: Ποια συνθήκη πρέπει να ικανοποιούν οι συντεταγμένες του σημείου $B(x,y)$ ώστε αυτό να βρίσκεται εντός του ορθογωνίου; Ποια είναι η αντίστοιχη συνθήκη ώστε να είναι ακριβώς πάνω στο σύνορο του ορθογωνίου;

Βήμα 2: Να ανοίξετε στο περιβάλλον προγραμματισμού IDLE το αρχείο **testRectangle.py** και να το εκτελέσετε. Τι παρατηρείτε;

Θεωρήστε ότι το μπλε ορθογώνιο ορίζεται από την κάτω αριστερή γωνία με συντεταγμένες $(xrect, yrect)$, το πλάτος $width$ στον οριζόντιο άξονα και το ύψος $height$ στον κάθετο άξονα. Επίσης όπου (x, y) είναι οι συντεταγμένες του σημείου στο οποίο έγινε κλικ με το ποντίκι.

Βήμα 3: Να υλοποιήσετε τη συνάρτηση

```
def isInside(x,y,xrect, yrect, width, height)
```

έτσι ώστε να επιστρέφει True όταν το σημείο όπου έγινε κλικ είναι εντός του ορθογωνίου, διαφορετικά να επιστρέφει False.

Στη συνέχεια να εκτελέσετε το πρόγραμμα και να δοκιμάσετε με το ποντίκι αν έχει την επιθυμητή λειτουργικότητα.

Βήμα 4: Να υλοποιήσετε τη συνάρτηση

```
def isOnBorder(x,y,xrect, yrect, width, height)
```

έτσι ώστε να επιστρέφει True όταν το σημείο όπου έγινε κλικ είναι ακριβώς πάνω στο σύνορο του ορθογωνίου, διαφορετικά να επιστρέφει False.

Να εκτελέσετε πάλι το πρόγραμμα και να ελέγξετε την λειτουργικότητά του κάνοντας κλικ πάνω στο σύνορο του ορθογωνίου.

```

# testRectangle.py
# Παράρτημα Κώδικα Δραστηριότητας:
import Tkinter as tk
import tkMessageBox

def isInside(x,y,xrect, yrect, width, height):

    #####
    # Γράψτε τον κώδικά σας εδώ
    #####

def isOnBorder(x,y,xrect, yrect, width, height):

    #####
    # Γράψτε τον κώδικά σας εδώ
    #####

# *****

def showxy(event):
    x = event.x
    y = h - event.y
    rwidth = x2-x1
    rheight = y2-y1
    if isOnBorder(x,y, x1, y1, rwidth, rheight):
        tkMessageBox.showinfo("Point test", "Point is on rectangle boundary")
    elif isInside(x,y, x1, y1, rwidth, rheight):
        tkMessageBox.showinfo("Point test","Point is inside rectangle")
    else:
        tkMessageBox.showinfo("Point test","Point is outside rectangle")

tkMessageBox.Dialog("test")

root = tk.Tk()
root.title("Point Inside Rectangle Check")

```



```
w = 500
h = 500
cv = tk.Canvas(root, width=w, height=h, bg='white')
cv.pack()

xoffset=100
yoffset=150
x1 = xoffset
y1 = yoffset
x2 = w-xoffset
y2 = h-yoffset
dif = 20
cv.create_rectangle(dif, dif, w-dif, h-dif, fill="blue", tag='rectangle2')
cv.create_rectangle(x1, y1, x2, y2, fill="cyan", tag='rectangle1')

cv.tag_bind('rectangle2', '<Button-1>', showxy)
cv.tag_bind('rectangle1', '<Button-1>', showxy)

root.mainloop()
```

6. Σχέδιο Δραστηριότητας: Πίνακες και Λίστες

Προτεινόμενος Χρόνος:	2 διδακτικές ώρες
Περιγραφή:	Οι πίνακες ως λίστες στην Python
Ενότητα:	Πίνακες
Εμπλεκόμενες Έννοιες:	Λίστες, μονοδιάστατοι πίνακες, πίνακες δυο διαστάσεων
Διδακτική Προσέγγιση:	Διερευνητική μάθηση, πειραματισμός στο προγραμματιστικό περιβάλλον
Εποπτικά μέσα	Η/Υ, Προβολέας, Περιβάλλον Προγραμματισμού IDLE
Χώρος	Εργαστήριο Πληροφορικής

6.1 Προσδοκώμενα μαθησιακά αποτελέσματα

Οι μαθητές:

- να κατανοήσουν την αναπαράσταση των πινάκων στην Python ως λίστες
- να δημιουργούν μονοδιάστατους και διδιάστατους πίνακες με δομικό στοιχείο τη δομή της λίστας
- να σχεδιάζουν και να υλοποιούν σε γλώσσα προγραμματισμού βασικούς αλγόριθμους επεξεργασίας πινάκων.

6.2 Οδηγίες για τον εκπαιδευτικό

Οι μαθητές έχουν ήδη έρθει σε επαφή με τις λίστες της Python.

Να τονιστεί στους μαθητές η δυναμική φύση των δομών δεδομένων στην Python. Να φανεί μέσα από τα παραδείγματα ότι μια λίστα είναι ταυτόχρονα και ένας μονοδιάστατος πίνακας, και ότι μια εμφωλευμένη λίστα 2 επιπέδων είναι ένας πίνακας δυο διαστάσεων.

Να εξοικειωθούν οι μαθητές και με τους δυο τρόπους σάρωσης μιας λίστας με τη δομή επανάληψης `for`. όπως φαίνεται παρακάτω για την περίπτωση ενός μονοδιάστατου πίνακα:

Χωρίς μεταβλητή-μετρητή	Με μεταβλητή-μετρητή
<pre>for item in array: print item</pre>	<pre>for i in range(len(array)): print array[i]</pre>

ή για πίνακα δυο διαστάσεων:

<pre>for row in A: for item in row: print item , print</pre>	<pre>for row in range(len(A)): for col in range(len(A[row])): print A[row][col] , print</pre>
--	---

ή στα πιο συνηθισμένα ιδιώματα για πίνακα array με N στοιχεία και πίνακα A με M γραμμές και N στήλες:

Κώδικας σε Python	Ψευδοκώδικας
<pre>for i in range(N): print array[i]</pre>	Για i από 0 μέχρι N-1 Εμφάνισε array[i] Τέλος_Επανάληψης
<pre>for i in range(M): for j in range(N): print A[i][j]</pre>	Για i από 0 μέχρι M-1 Για j από 0 μέχρι N-1 Εμφάνισε A[i][j] Τέλος_Επανάληψης Τέλος_Επανάληψης

Όπως φαίνεται παραπάνω, πρέπει να δοθεί ιδιαίτερη προσοχή στο εύρος των τιμών που παίρνουν οι δείκτες των πινάκων, διότι στην Python όπως και σε αρκετές γλώσσες προγραμματισμού (C++, Java, Ruby), η αρίθμηση στους πίνακες ξεκινάει από το 0 και όχι από το 1. Οπότε καλό θα ήταν και ο ψευδοκώδικας που χρησιμοποιούμε να ακολουθεί το ίδιο σχήμα δεικτοδότησης ώστε να υπάρχει μια πιστή αντιστοιχία με την Python για να μην δημιουργείται σύγχυση στους μαθητές.

Από τα παραπάνω είναι φανερό ότι δεν υπάρχει ανάγκη για εκτεταμένη χρήση ψευδοκώδικα για την διαχείριση πινάκων αφού οι δομές επανάληψης της Python βρίσκονται στο ίδιο επίπεδο αφαίρεσης με τον ψευδοκώδικα.

Επίσης, μπορούμε να ορίσουμε πίνακες δυο διαστάσεων όπου κάθε γραμμή μπορεί να έχει διαφορετικό πλήθος στοιχείων, προσθέτοντας σε κάθε γραμμή διαφορετικό πλήθος στοιχείων με την append.

6.3 Φύλλο Εργασίας: Πίνακες και Λίστες

ΟΝΟΜΑ ΕΠΩΝΥΜΟ.....

ΤΑΞΗ..... ΤΜΗΜΑ..... ΗΜ/ΝΙΑ:

6.3.1 Δραστηριότητα 1

Τι θα εμφανίσουν οι παρακάτω εντολές; Τι συμπεράσματα βγάξετε;

```
>>> mylist = [ [1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12] ]
>>> print( mylist[0] )
>>> print( mylist[3][1] )
>>> print( mylist[0] + mylist[1] + mylist[2] + mylist[3] )
>>> print( [ mylist[0][2], mylist[1][2], mylist[2][2], mylist[3][2] ] )
>>> print [ mylist[0][1], mylist[1][1], mylist[2][1], mylist[3][1] ]
```

6.3.2 Δραστηριότητα 2

Δώστε στον διερμηνευτή της Python τις παρακάτω εντολές.

```
>>> A = [ 1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 ]
>>> M = [ [1, 1, 1], [2, 2, 2], [3,3,3], [4, 4, 4] ]
>>> A[0]
>>> M[0]
>>> len(A)
>>> len(M)
>>> len(M[1])
```

Στη συνέχεια γράψτε το παρακάτω πρόγραμμα σε ένα αρχείο array.py, εκτελέστε το και μελετήστε τα αποτελέσματα:

```
for item in A:
    print item,
print
for i in range(len(A)):
    print A[i]

for row in M:
    for item in row:
        print item,
    print

for row in range(len(M)):
    for col in range(len(M[row])):
        print M[row][col] ,
    print
```

6.3.3 Δραστηριότητα 3

Να συμπληρώσετε τα κενά έτσι ώστε να εμφανίζονται τα στοιχεία της 2^{ης} γραμμής του πίνακα M.

```
for _____ in _____:  
    print _____,  
print
```

6.3.4 Δραστηριότητα 4

Να συμπληρώσετε τα κενά έτσι ώστε να εμφανίζονται τα στοιχεία της τελευταίας στήλης του πίνακα M.

```
for _____ in _____:  
    print _____,  
print
```

6.3.5 Δραστηριότητα 5

Να συμπληρώσετε τα κενά έτσι ώστε να εμφανίζονται τα στοιχεία του πίνακα M κατά στήλες και όχι κατά γραμμές.

```
for _____ in _____:  
    for _____ in _____:  
        print _____,  
print
```

7. Σχέδιο Δραστηριότητας: Αλγόριθμος συμπίεσης δεδομένων RLE

Προτεινόμενος Χρόνος:	2 διδακτικές ώρες
Περιγραφή:	Μια δραστηριότητα η οποία εκτός από επεξεργασία συμβολοσειρών παρουσιάζει και μια εφαρμογή του πολυμορφισμού στην Python
Ενότητα:	Αντικείμενα, Συμβολοσειρές, Λίστες
Εμπλεκόμενες Έννοιες:	Αντικείμενα, πολυμορφισμός, δυναμικό σύστημα τύπων
Προαπαιτούμενες γνώσεις:	Διαχείριση λιστών και συμβολοσειρών
Διδακτική Προσέγγιση:	Καταιγισμός ιδεών, πειραματισμός με το προγραμματιστικό περιβάλλον, σχεδιασμός και ανάπτυξη αλγορίθμων
Εποπτικά μέσα	Η/Υ, Προβολέας, Περιβάλλον Προγραμματισμού IDLE
Χώρος	Εργαστήριο Πληροφορικής

7.1 Προσδοκώμενα μαθησιακά αποτελέσματα

Οι μαθητές:

- να εντοπίζουν και να διορθώνουν τα λογικά λάθη σε έναν αλγόριθμο
- να χρησιμοποιούν την ιδιότητα του πολυμορφισμού τελεστών της Python
- να εξοικειωθούν με τους αλγορίθμους συμπίεσης δεδομένων
- να εμβαθύνουν στην επεξεργασία συμβολοσειρών
- να σχεδιάζουν συναρτήσεις στην Python για την υλοποίηση αλγορίθμων βάσει συγκεκριμένων προδιαγραφών.

7.2 Οδηγίες για τον εκπαιδευτικό

Για να πραγματοποιηθεί αποτελεσματικά η δραστηριότητα αυτή θα πρέπει οι μαθητές να έχουν εξοικειωθεί με τις λίστες και τις συμβολοσειρές.

7.2.1 Δραστηριότητα 1

Αφού οι μαθητές εκτελέσουν το πρόγραμμα που τους δίνεται για διάφορα σύνολα δεδομένων ο εκπαιδευτικός ρωτάει τους μαθητές για ποιον σκοπό πιστεύουν ότι σχεδιάστηκε αυτό το πρόγραμμα και αν τον πετυχαίνει.

Αφού μετά από καταιγισμό ιδεών και συζήτηση καταλήξουμε στο σκοπό του προγράμματος, δηλαδή στην καταμέτρηση των συνεχόμενων εμφανίσεων κάθε χαρακτήρα, ρωτάμε τους μαθητές αν τα αποτελέσματα του προγράμματος ήταν τα αναμενόμενα, ή αν έχουν εντοπίσει κάποιο λάθος. Το ένα λάθος στο οποίο προτείνεται να καθοδηγήσουμε τους μαθητές, αν δεν το βρουν άμεσα,

είναι ότι το πλήθος των εμφανίσεων των χαρακτήρων είναι κατά 1 μικρότερο. Άρα όταν συναντάμε νέο σύμβολο ο μετρητής times πρέπει να επανεκκινείται από 1.

Το επόμενο λάθος που είναι και το πιθανότερο που θα έκαναν οι μαθητές αν τους ζητούσαμε την εξ' ολοκλήρου ανάπτυξη του προγράμματος, είναι η εκτύπωση του πλήθους των εμφανίσεων του τελευταίου χαρακτήρα. Αυτό συμβαίνει γιατί ο τελικός υπολογισμός του πλήθους γίνεται όταν συναντήσουμε διαφορετικό χαρακτήρα, όμως μετά τον τελευταίο δεν υπάρχει άλλος χαρακτήρας. Οπότε η εκτύπωση του πλήθους των εμφανίσεων του πρέπει να γίνει αμέσως μετά το τέλος της επανάληψης.

7.2.2 Δραστηριότητα 2

Στη 2^η δραστηριότητα προτείνεται να συζητηθεί με τους μαθητές η πολυμορφική συμπεριφορά των τελεστών '+' και '*'. Ειδικά η συμπεριφορά του τελεστή '*' έχει εξαιρετικό ενδιαφέρον γιατί επαληθεύει την σχέση

"Python" + "Python" + "Python" = 3 * "Python".

Ο βασικός στόχος είναι να καταλάβουν οι μαθητές ότι στην Python όλες οι αποφάσεις λαμβάνονται σε χρόνο εκτέλεσης, δηλαδή την τελευταία στιγμή (lazy evaluation). Αυτό προσδίδει στην γλώσσα τη δυναμική της συμπεριφορά.

Αυτό φαίνεται πολύ καλά στον υπολογισμό του γινομένου των στοιχείων-ζευγών του πίνακα zipr, όπου είναι πλέον φανερό ότι η απόφαση για το είδος της πράξης που επιτελείται λαμβάνεται τη στιγμή της εκτέλεσης.

Εδώ για να προετοιμάσουμε τους μαθητές για την επόμενη δραστηριότητα τους ζητάμε να ξαναγράψουν τον υπολογισμό του text

text = zipr[0]*zipr[1] + zipr[2]*zipr[3] + zipr[4]*zipr[5] με χρήση δομής επανάληψης.

```
text = ""
for i in range(0,len(zipr),2):
    text = text + zipr[i]*zipr[i+1]
```

7.2.3 Δραστηριότητα 3

Αφού οι μαθητές εξοικειωθούν με τη λειτουργία του αλγορίθμου συμπίεσης RLE τους ζητάμε να συμπληρώσουν τον ορισμό της συνάρτησης συμπίεσης RLEzip, το μεγαλύτερο μέρος της οποίας υλοποίησαν στην πρώτη δραστηριότητα. Τώρα το ζεύγος συχνότητα-χαρακτήρας δεν εμφανίζεται στην οθόνη, αλλά προστίθεται σε μια λίστα (ziplist) η οποία επιστρέφεται από τη συνάρτηση. Στη συνέχεια κατά την αποσυμπίεση στη συνάρτηση RLEunzip με μια επανάληψη σαρώνουμε τα αντικείμενα της λίστας ανά δυο και με τον τελεστή '*' παράγουμε το αρχικό μήνυμα όπως δείξαμε στην προηγούμενη δραστηριότητα.

<pre>def RLEcompress(text): ziplist = [] times = 0 prev = text[0] for letter in text: if (letter==prev): times = times + 1 else: ziplist = ziplist + [times, prev] prev = letter times = 1 ziplist = ziplist + [times, prev] return ziplist</pre>	<pre>def RLEuncompress(ziplist): text = "" for i in range(0,len(ziplist),2): text = text + ziplist[i]*ziplist[i+1] return text</pre>
---	--

Υλοποιήστε τις παρακάτω συναρτήσεις συμπίεσης και αποσυμπίεσης με τον αλγόριθμο RLE και να τις χρησιμοποιήσετε για διάφορα δεδομένα.

```
# Συμπιέζει το κείμενο text στη λίστα ziplist και επιστρέφει τη λίστα αυτή
def RLEzip(text):
    ziplist = []
    # Ο κώδικάς σας γράφεται εδώ
    return ziplist
# Δημιουργεί το κείμενο text από την RLE-συμπιεσμένη λίστα ziplist
def RLEunzip(ziplist):
    text = ""
    # Ο κώδικάς σας γράφεται εδώ
    return text
```

8. Αναφορές Μέρους II

1. www.Python.org
2. <http://Pythontutor.com>
3. Philip J. Guo. *Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education*. In Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE), March 2013
4. Code Like a Pythonista: Idiomatic Python
<http://Python.net/~goodger/projects/pycon/2007/idiomatic/handout.html>
5. A Byte of Python (ελληνική μετάφραση)
http://dide.flo.sch.gr/Plinet/Meetings/Meeting23/A_Byte_of_Python-el.pdf
6. Downey, A. Think Python, How to think like a computer scientist,
<http://www.thinkPython.com>
7. <http://www.greenteapress.com/thinkPython/thinkPython.html>
8. <http://www.diveintoPython.net/>

Εναλλακτικά περιβάλλοντα προγραμματισμού

1. Περιβάλλον PyScripter <https://code.google.com/p/pyscripter/>
2. Eclipse με pydev <http://pydev.org/> , <http://eclipse.org/>
3. <https://www.jetbrains.com/pycharm/> (δωρεάν έκδοση)

Διαδικτυακά Περιβάλλοντα προγραμματισμού σε Python

1. <http://www.Pythontutor.com/>
2. <http://www.skulpt.org/>
3. <http://repl.it/languages/Python>
4. http://www.compileonline.com/execute_Python_online.php
5. <https://www.Pythonanywhere.com/try-iPython/>

Οδηγοί Βοήθειας

<https://docs.Python.org/3/>

<http://www.Python-course.eu/>

http://www.Python-course.eu/Python3_course.php

Διαδικτυακά μαθήματα με αρκετό υλικό

<http://interactivePython.org/runestone/static/thinkcspy/index.html>

<http://www-inst.eecs.berkeley.edu/~cs61a/fa14/>

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-189-a-gentle-introduction-to-programming-using-Python-january-iap-2011/>

MOOCS

<https://www.coursera.org/course/interactivePython>

<https://www.udacity.com/course/ud036>

<https://www.edx.org/course/introduction-computer-science-mitx-6-00-1x-0#.VLFu2dKsWaQ>

«ΝΕΟ ΣΧΟΛΕΙΟ (Σχολείο 21ου αιώνα) – Νέο Πρόγραμμα Σπουδών»

με κωδικό ΟΠΣ: 295450

Οριζόντια Πράξη στις 8 Π.Σ., 3 Π.Στ. Εξ., 2 Π.Στ. Εισ.

Υποέργο 9: «Εκπόνηση Προγραμμάτων Σπουδών Γενικού Λυκείου, Μουσικών και Καλλιτεχνικών Λυκείων»

Δράση: Επιστημονική Εποπτεία της εκπόνησης των Προγραμμάτων Σπουδών του Γενικού Λυκείου ανά γνωστικό αντικείμενο



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ