

Computing

Νοέμβριος 2017

Έκδοση 1.0

Ενότητα Computing

Αυτή η ενότητα παραθέτει τις βασικές έννοιες και δεξιότητες που σχετίζονται με την ικανότητα χρήσης υπολογιστικής σκέψης (computational thinking) και τον προγραμματισμό για τη δημιουργία απλών προγραμμάτων/εφαρμογών για ηλεκτρονικούς υπολογιστές.

Στόχοι ενότητας

Οι υποψήφιοι θα είναι σε θέση να:

- Κατανοούν βασικές έννοιες σχετικά με την υπολογιστική/πληροφορική (computing) καθώς και τις τυπικές δραστηριότητες που σχετίζονται με τη δημιουργία ενός προγράμματος ή μιας εφαρμογής.
- Κατανοούν και χρησιμοποιούν τεχνικές υπολογιστικής σκέψης (computational thinking) όπως αποδόμηση προβλημάτων (decomposition), αναγνώριση μοτίβων (pattern recognition), αφαίρεση (abstraction) και δημιουργία αλγορίθμων (algorithms) για την ανάλυση ενός προβλήματος και την ανάπτυξη λύσεων.
- Γράφουν, δοκιμάζουν και τροποποιούν αλγορίθμους ενός προγράμματος, χρησιμοποιώντας διαγράμματα ροής (flow charts) και ψευδοκώδικα (pseudocode).
- Κατανοούν βασικές αρχές και όρους που σχετίζονται με τον προγραμματισμό και τη σημασία ενός καλά δομημένου και τεκμηριωμένου κώδικα.
- Κατανοούν και χρησιμοποιούν προγραμματιστικές δομές (constructs) όπως μεταβλητές (variables), τύπους δεδομένων (datatypes) και λογική ροή (logic) σε ένα πρόγραμμα.
- Βελτιώνουν την αποτελεσματικότητα και τη λειτουργικότητα ενός προγράμματος, χρησιμοποιώντας επαναλήψεις (iteration), συνθήκες (conditionals), διαδικασίες (procedures) και συναρτήσεις (functions), καθώς και συμβάντα (events) και εντολές (commands).
- Ελέγχουν και αποσφαλματώνουν (test & debug) ένα πρόγραμμα, εξασφαλίζοντας ότι πληροί τις απαιτήσεις πριν από τη δημοσίευσή του.

ΚΑΤΗΓΟΡΙΑ	ΓΝΩΣΤ. ΠΕΡΙΟΧΗ	ΑΝΑΦ.	ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΓΑΣΙΑΣ
1 Έννοιες Υπολογιστικής	<i>1.1 Βασικές Έννοιες</i>	1.1.1	Ορισμός του όρου computing (υπολογιστική/πληροφορική).
		1.1.2	Ορισμός του όρου υπολογιστική σκέψη (computational thinking).
		1.1.3	Ορισμός του όρου πρόγραμμα (program)
		1.1.4	Ορισμός του όρου κώδικας (code). Διαχωρισμός μεταξύ πηγαίου κώδικα (source code) και κώδικα μηχανής (machine code).
		1.1.5	Κατανόηση των όρων περιγραφή προγράμματος (program description) και προδιαγραφές (specification).
		1.1.6	Αναγνώριση συνήθων δραστηριοτήτων κατά τη δημιουργία ενός προγράμματος: ανάλυση, σχεδίαση, υλοποίηση/προγραμματισμός, δοκιμή, και βελτίωση.
		1.1.7	Κατανόηση της διαφοράς μεταξύ μιας τυπικής γλώσσας (formal) και μιας φυσικής (natural) γλώσσας.
2 Μέθοδοι Υπολογιστικής Σκέψης (Computational Thinking)	<i>2.1 Ανάλυση Προβλήματος</i>	2.1.1	Περιγραφή τυπικών μεθόδων που χρησιμοποιούνται στην υπολογιστική σκέψη (computational thinking): διάσπαση/αποδόμηση (decomposition), αναγνώριση μοτίβων (pattern recognition), αφαίρεση (abstraction), αλγόριθμοι (algorithms).
		2.1.2	Χρήση της αποδόμησης (decomposition) ενός προβλήματος για τη διάσπαση δεδομένων, διαδικασιών ή ενός περίπλοκου προβλήματος σε μικρότερα μέρη.
		2.1.3	Αναγνώριση μοτίβων (pattern recognition) μεταξύ μικρών, αποδομημένων προβλημάτων (decomposed problems).
		2.1.4	Χρήση αφαίρεσης (abstraction) για το φιλτράρισμα περιττών λεπτομερειών κατά την ανάλυση ενός προβλήματος.
		2.1.5	Κατανόηση τρόπου χρήσης αλγορίθμων (algorithms) στην υπολογιστική σκέψη (computational thinking).
	<i>2.2 Αλγόριθμοι</i>	2.2.1	Ορισμός της προγραμματιστικής δομής ακολουθία (sequence). Περιγραφή σκοπού ακολουθίας κατά το σχεδιασμό αλγορίθμων.
		2.2.2	Αναγνώριση πιθανών μεθόδων αναπαράστασης προβλημάτων όπως: διαγράμματα ροής (flow charts), ψευδοκώδικας (pseudocode).

ΚΑΤΗΓΟΡΙΑ	ΓΝΩΣΤ. ΠΕΡΙΟΧΗ	ΑΝΑΦ.	ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΓΑΣΙΑΣ
		2.2.3	Αναγνώριση συμβόλων διαγράμματος ροής (flow chart) όπως: αρχή / τέλος (start/stop), επεξεργασία/διαδικασία (process), απόφαση (decision), είσοδος/έξοδος (input/output), σύνδεσμος (connector), βέλος (arrow).
		2.2.4	Περιγραφή σειράς βημάτων/λειτουργιών ενός προγράμματος που περιγράφονται από ένα διάγραμμα ροής (flow chart), ψευδοκώδικα (pseudocode).
		2.2.5	Σύνταξη ακριβούς αλγόριθμου (algorithm) βάσει περιγραφής χρησιμοποιώντας μια τεχνική όπως: διάγραμμα ροής (flow chart), ψευδοκώδικα (pseudocode).
		2.2.6	Διόρθωση σφαλμάτων (errors) σε έναν αλγόριθμο (algorithm), όπως: έλλειψη βημάτων, λανθασμένη ακολουθία εντολών, λανθασμένη λήψη αποφάσεων.
3 Συγγραφή Κώδικα (Coding)	3.1 Εισαγωγή	3.1.1	Περιγραφή χαρακτηριστικών ενός καλά δομημένου και τεκμηριωμένου κώδικα όπως: εσοχές (indentation), κατάλληλα σχόλια (comments), περιγραφική ονομασία.
		3.1.2	Χρήση αριθμητικών τελεστών για την εκτέλεση πράξεων σε ένα πρόγραμμα: +, -, /, *.
		3.1.3	Κατανόηση προτεραιότητας τελεστών (operator precedence) και σειράς εκτέλεσης πράξεων σε σύνθετες εκφράσεις. Κατανόηση τρόπου χρήσης παρενθέσεων για τη δόμηση σύνθετων εκφράσεων.
		3.1.4	Κατανόηση του όρου παράμετρος (parameter). Περιγραφή σκοπού παραμέτρων (parameters) σε ένα πρόγραμμα.
		3.1.5	Ορισμός της προγραμματιστικής δομής σχόλιο (comment). Περιγραφή σκοπού ενός σχολίου (comment) σε ένα πρόγραμμα.
		3.1.6	Χρήση σχολίων (comments) σε ένα πρόγραμμα.
	3.2 Μεταβλητές (Variables) και Τύποι Δεδομένων (Data Types)	3.2.1	Ορισμός της προγραμματιστικής δομής μεταβλητή (variable). Περιγραφή σκοπού μιας μεταβλητής (variable) σε ένα πρόγραμμα.
		3.2.2	Ορισμός (define) και αρχικοποίηση (initialize) μεταβλητής.
		3.2.3	Ανάθεση τιμής σε μεταβλητή (variable).
		3.2.4	Χρήση κατάλληλων ονομάτων μεταβλητών (variables) σε ένα πρόγραμμα για υπολογισμούς, αποθήκευση τιμών.

ΚΑΤΗΓΟΡΙΑ	ΓΝΩΣΤ. ΠΕΡΙΟΧΗ	ΑΝΑΦ.	ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΓΑΣΙΑΣ
4 Προγραμματισμός με Κώδικα	4.1 Λογική (Logic)	3.2.5	Χρήση τύπων δεδομένων (data types) σε ένα πρόγραμμα: συμβολοσειρά (string), χαρακτήρας (character), ακέραιος (integer), κινητής υποδιαστολής (float), δυαδικής τιμής (Boolean).
		3.2.6	Χρήση συγκεντρωτικού τύπου δεδομένων (aggregate data type) σε ένα πρόγραμμα όπως: πίνακας (array), λίστα (list), πλειάδα (tuple).
		3.2.7	Εισαγωγή δεδομένων από τον χρήστη σε ένα πρόγραμμα.
		3.2.8	Εμφάνιση δεδομένων στην οθόνη από ένα πρόγραμμα.
	4.2 Επανάληψη (Iteration)	4.1.1	Ορισμός της προγραμματιστικής δομής λογικός έλεγχος (logic test). Περιγραφή σκοπού ενός λογικού ελέγχου σε ένα πρόγραμμα.
		4.1.2	Αναγνώριση τύπων δυαδικών λογικών εκφράσεων (Boolean logic expressions) για τη δημιουργία τιμών αληθής/ ψευδής (true/false) όπως: =, >, <, >=, <=, <>, !=, ==, AND, OR, NOT.
		4.1.3	Χρήση δυαδικών λογικών εκφράσεων (Boolean logic expressions) σε ένα πρόγραμμα.
	4.3 Συνθήκη (Conditionality)	4.2.1	Ορισμός της προγραμματιστικής δομής βρόγχος (loop). Περιγραφή σκοπού και πλεονεκτημάτων των βρόχων σε ένα πρόγραμμα.
		4.2.2	Αναγνώριση τύπων επαναληπτικών βρόχων (iterative loops): for, while, repeat.
		4.2.3	Χρήση επαναληπτικών βρόχων (iterative loops): for, while, repeat.
		4.2.4	Κατανόηση του όρου ατέρμον βρόχος (infinite loop).
		4.2.5	Κατανόηση του όρου αναδρομή (recursion).
	4.4 Διαδικασίες (Procedures) και Συναρτήσεις (Functions)	4.3.1	Ορισμός της προγραμματιστικής δομής έκφραση συνθήκης (conditional statement). Περιγραφή σκοπού εκφράσεων συνθήκης σε ένα πρόγραμμα.
		4.3.2	Χρήση έκφρασης συνθήκης Αν...Τότε...Αλλιώς (IF...THEN... ELSE) σε ένα πρόγραμμα.
		4.4.1	Κατανόηση του όρου διαδικασία (procedure). Περιγραφή σκοπού διαδικασίας σε ένα πρόγραμμα.
		4.4.2	Σύνταξη και ονομασία διαδικασίας (procedure) σε ένα πρόγραμμα.

ΚΑΤΗΓΟΡΙΑ	ΓΝΩΣΤ. ΠΕΡΙΟΧΗ	ΑΝΑΦ.	ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΓΑΣΙΑΣ
		4.4.3	Κατανόηση του όρου συνάρτηση (function). Περιγραφή σκοπού συνάρτησης σε ένα πρόγραμμα.
		4.4.4	Σύνταξη και ονομασία συνάρτησης (function) σε ένα πρόγραμμα.
	4.5 Συμβάντα (Events) και Εντολές (Commands)	4.5.1	Κατανόηση του όρου συμβάν/γεγονός (event). Περιγραφή σκοπού συμβάντων σε ένα πρόγραμμα.
		4.5.2	Χρήση χειριστών συμβάντων/γεγονότων (event handler) όπως: κλικ ποντικιού (mouse click), εισαγωγή από το πληκτρολόγιο (keyboard input), κλικ σε κουμπί (button click), χρονομετρητής (timer).
		4.5.3	Χρήση διαθέσιμων βασικών βιβλιοθηκών (generic libraries) όπως: math (μαθηματικές συναρτήσεις), random (τυχαίοι αριθμοί), time (συναρτήσεις χρόνου).
5 Έλεγχος, Αποσφαλμάτωση και Έκδοση (Test, Debug & Release)	5.1 Εκτέλεση (Run), Έλεγχος (Test) και Αποσφαλμάτωση (Debug)	5.1.1	Κατανόηση πλεονεκτημάτων ελέγχου και αποσφαλμάτωσης (debug) προγράμματος για επίλυση σφαλμάτων (errors).
		5.1.2	Κατανόηση τύπων σφαλμάτων (errors) σε ένα πρόγραμμα όπως: συντακτικά (syntax), λογικά (logic).
		5.1.3	Εκτέλεση προγράμματος.
		5.1.4	Αναγνώριση και επίλυση συντακτικού σφάλματος σε πρόγραμμα όπως: ορθογραφικά λάθη, παράλειψη σημείων στίξης.
		5.1.5	Αναγνώριση και επίλυση λογικού σφάλματος σε πρόγραμμα όπως: εσφαλμένη δυαδική έκφραση (Boolean), εσφαλμένος τύπος δεδομένων (data type).
	5.2 Έκδοση (Release)	5.2.1	Έλεγχος προγράμματος σύμφωνα με τις απαιτήσεις της αρχικής περιγραφής.
		5.2.2	Περιγραφή ολοκληρωμένου προγράμματος, επικοινωνώντας το σκοπό και την αξία του.
		5.2.3	Αναγνώριση προσθηκών, βελτιώσεων στο πρόγραμμα ίσως να ικανοποιούν πρόσθετες, παρεμφερείς ανάγκες.