

Ατομική Διπλωματική Εργασία

**Σύστημα Υποστήριξης Φοιτητών Για Το Μάθημα
ΕΠΛ131 – Αρχές Προγραμματισμού**

Αντώνιος Κιουρτζίδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σύστημα Υποστήριξης Φοιτητών Για Το Μάθημα

ΕΠΑ131 – Αρχές Προγραμματισμού

Αντώνιος Κιουρτζίδης

Επιβλέπουσα Καθηγήτρια

Ελπίδα Κεραυνού – Παπαηλιού

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2024

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω το Πανεπιστήμιο Κύπρου που μου έδωσε την δυνατότητα να επεκτείνω τους γνωστικούς μου ορίζοντες στον κλάδο της Επιστήμης της Πληροφορικής. Επίσης θα ήθελα να ευχαριστήσω την Ελπίδα Κεραυνού – Παπαηλίου που επέβλεψε την παρούσα διπλωματική εργασία.

Περίληψη

Ο κύριος στόχος της παρούσας διπλωματικής εργασίας είναι η ομαλή ένταξη των νεοεισερχόμενων φοιτητών στο Τμήμα Πληροφορικής, οι οποίοι εκκινούν τις σπουδές τους με το μάθημα ΕΠΑ131 – Αρχές Προγραμματισμού το οποίο είναι υποχρεωτικό βάσει του προγράμματος σπουδών. Όπως αναφέρθηκε και στο κείμενο της Δρ. Κεραυνού «Figuring and Drawing A Visual Approach to Principled Programming» οι φοιτητές που εισέρχονται στο Τμήμα Πληροφορικής, προέρχονται από διαφορετικά γνωστικά υπόβαθρα, με αποτέλεσμα να καθίσταται πιο δύσκολη η ένταξη στην ροή του μαθήματος για κάποιους φοιτητές. Με τον όρο «υποστήριξη των φοιτητών» εννοείται η καλλιέργεια θεμελιωδών δεξιοτήτων στον χώρο της πληροφορικής, όπως η αλγοριθμική σκέψη, αφαιρετικότητα στον τρόπο επίλυσης προβλημάτων, επαναχρησιμοποίηση κώδικα, απλότητα στις λύσεις αλλά και οργάνωση στον τρόπο γραφής και σκέψης χρησιμοποιώντας την γλώσσα προγραμματισμού JAVA. Με την χρήση του συστήματος που αναπτύχθηκε στο πλαίσιο της διπλωματικής εργασίας, επιτυγχάνεται η ενδυνάμωση των παραπάνω δεξιοτήτων με απώτερο σκοπό την δημιουργία ενός κοινού παρονομαστή μεταξύ των φοιτητών του μαθήματος ΕΠΑ131.

Με την επιτυχή ολοκλήρωση του μαθήματος, οι φοιτητές θα είναι σε θέση να επιλύουν προγραμματιστικά προβλήματα, αφού θα έχουν κατανοήσει σε ικανοποιητικό βαθμό την έννοια του αλγορίθμου, της αφαιρετικότητας προγράμματος και δεδομένων, συναρτήσεων, κλάσεων και μεθόδων. Ειδικότερα θα έχουν εμβαθύνει στα θέματα της δόμησης και άρθρωσης κώδικα, επιλογής και επανάληψης σε συνδυασμό με λογικές εκφράσεις, συναρτήσεις και αναδρομής. Προς το τέλος του μαθήματος γίνεται μια εισαγωγή στα αντικείμενα τα οποία οι φοιτητές θα δουν εις βάθος στο επόμενο εξάμηνο των σπουδών τους στο μάθημα ΕΠΑ133 – Αντικειμενοστρεφής Προγραμματισμός, δεδομένης της επιτυχής ολοκλήρωσης του μαθήματος ΕΠΑ131.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Γενική Εισαγωγή	1
	1.2 Στόχος Διπλωματικής Εργασίας	2
	1.3 Εργαλεία Ανάπτυξης	2
	1.4 Δομή Διπλωματικής Εργασίας	3
Κεφάλαιο 2	Συστήματα Εκπαίδευσης και Διδασκαλίας	4
	2.1 Εισαγωγή	4
	2.2 Μαθησιακό Περιβάλλον	4
	2.3 Εκμάθηση Μέσω Διαδικτύου	5
Κεφάλαιο 3	Απαιτήσεις Συστήματος.....	6
	3.1 Εισαγωγή	6
	3.2 Συλλογή Απαιτήσεων Συστήματος	7
	3.2.1 Τρόποι Εξαγωγής Απαιτήσεων	7
	3.2.2 Ερωτηματολόγιο	7
	3.2.3 Συμπεράσματα Ερωτηματολογίου	10
	3.3 Περιγραφή Συστήματος	11
	3.4 Απαιτήσεις Υλικού και Λογισμικού	12
	3.4.1 Απαιτήσεις Χρήστη	12
	3.4.2 Απαιτήσεις Παροχέα	12
	3.5 Ιδιότητες και Χαρακτηριστικά Συστήματος	12
	3.5.1 Αξιοπιστία και Διαθεσιμότητα	12
	3.5.2 Φορητότητα	13
	3.5.3 Επεκτασιμότητα	13

Κεφάλαιο 4 Σχεδίαση και Υλοποίηση Συστήματος	14
4.1 Εισαγωγή	14
4.2 Σχεδίαση Ιστοσελίδας	14
4.2.1 Κυρίως Μενού	14
4.2.2 Ασκήσεις	15
4.3.3 Συμπληρωματικό Υλικό	15
4.3 Εργαλεία, Προγράμματα και Υπηρεσίες που Χρησιμοποιήθηκαν	16
4.3.1 HTML	16
4.3.2 CSS	16
4.3.3 Google Forms	16
4.3.4 Microsoft Word	16
4.4 Παρουσίαση Συστήματος	17

Κεφάλαιο 5 Συμπεράσματα	26
5.1 Γενικά Συμπεράσματα που Εξάγονται από την Υλοποίηση	26
5.2 Δυσκολίες	26
5.3 Επίλογος	26

Βιβλιογραφία	27
---------------------------	-----------

Παράρτημα Α.....	Α-1 έως Α-2
-------------------------	--------------------

Παράρτημα Β.....	Β-1 έως Β-85
-------------------------	---------------------

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Εισαγωγή	1
1.2 Στόχος Διπλωματικής Εργασίας	2
1.3 Εργαλεία Ανάπτυξης	2
1.4 Δομή Διπλωματικής Εργασίας	3

1.1 Γενική Εισαγωγή

Το υποχρεωτικό μάθημα ΕΠΛ131 – Αρχές Προγραμματισμού αποτελεί την βάση για την εκκίνηση των σπουδών των φοιτητών στο Τμήμα Πληροφορικής του Πανεπιστημίου Κύπρου, καθώς καλύπτει βασικές έννοιες με τις οποίες έρχονται σε επαφή οι φοιτητές καθ' όλη τη διάρκεια των σπουδών τους. Με την χρήση της γλώσσας προγραμματισμού Java, η οποία είναι ισορροπημένη από την άποψη της αφαιρετικότητας σε σημείο που είναι εύκολη για την ένταξη των φοιτητών στον κόσμο του προγραμματισμού, ενώ ταυτόχρονα είναι ικανή να διδάξει αρχές όπως αυτές εμφανίζονται στα κεφάλαια του μαθήματος: Βασικοί Τύποι Δεδομένων, Αλγοριθμικές Δομές Επιλογής και Επανάληψης, Πίνακες, Συμβολοσειρές, Είσοδος και Έξοδος, Αρχές Ορθού Προγραμματισμού, Συναρτήσεις, Αντικείμενα, Αναδρομή, Πίνακες Πολλαπλών Διαστάσεων. Με την ολοκλήρωση και κατά συνέπεια κατανόηση της διδακτέας ύλης του μαθήματος αυτού, ανοίγουν οι πόρτες για την παρακολούθηση των εξής μαθημάτων τα οποία έχουν ως απαραίτητη προϋπόθεση την επιτυχή παρακολούθηση του ΕΠΛ131. Τα μαθήματα αυτά είναι: ΕΠΛ133: Αντικειμενοστρεφής Προγραμματισμός, ΕΠΛ232: Προγραμματιστικές Τεχνικές και Εργαλεία, ΕΠΛ231: Δομές Δεδομένων και Αλγόριθμοι, ΕΠΛ222: Λειτουργικά Συστήματα, ΕΠΛ236: Αλγόριθμοι και Πολυπλοκότητα, ΕΠΛ326: Επικοινωνίες και Δίκτυα, ΕΠΛ326: Ασφάλεια Συστημάτων, ΕΠΛ341: Τεχνητή Νοημοσύνη, ΕΠΛ342: Βάσεις Δεδομένων, ΕΠΛ343: Τεχνολογία Λογισμικού.

1.2 Στόχος Διπλωματικής Εργασίας

Ο στόχος της Διπλωματικής Εργασίας είναι ο σχεδιασμός και υλοποίηση ενός συστήματος στον παγκόσμιο ιστό το οποίο μέσω διαδραστικών διαδικτυακών μαθημάτων θα καλλιεργεί στους φοιτητές θεμελιώδη δεξιότητες στον χώρο της πληροφορικής, όπως η αλγοριθμική σκέψη, αφαιρετικότητα στον τρόπο επίλυσης προβλημάτων, επαναχρησιμοποίηση κώδικα, απλότητα στις λύσεις αλλά και οργάνωση στον τρόπο γραφής και σκέψης χρησιμοποιώντας την γλώσσα προγραμματισμού JAVA, σε θεματικές ενότητες όπως αυτές εμφανίζονται στο μάθημα ΕΠΛ131. Σε αυτό το σημείο πρέπει να τονιστεί πως η παρακολούθηση μέσω της πλατφόρμας αυτής δεν αντικαθιστά, ούτε συναγωνίζεται την βιβλιογραφία και τις διαλέξεις του μαθήματος ΕΠΛ131, και θα πρέπει να αντιμετωπίζεται ως βοηθητικό υλικό για την περαιτέρω κατανόηση των εννοιών που διδάσκονται στο μάθημα. Ο στόχος αυτός εκπληρώνεται με την ύπαρξη των εξής δυνατοτήτων εντός κάθε θεματικής ενότητας:

1. Σύντομες σημειώσεις με τα κύρια σημεία της ενότητας
2. Διαδραστικές ασκήσεις:
 - Συμπλήρωσης κενών σε υφιστάμενα προγράμματα JAVA
 - Σύνταξη κώδικα JAVA
 - Απάντησης εξόδου υφιστάμενου προγράμματος JAVA
 - Εντοπισμός σφαλμάτων υφιστάμενου προγράμματος JAVA
 - Βελτιστοποίηση υφιστάμενου προγράμματος JAVA
 - Ερωτήσεις πολλαπλής επιλογής

Με την ολοκλήρωση των δραστηριοτήτων της κάθε ενότητας ο φοιτητής θα λαμβάνει ανατροφοδότηση για τα δυνατά του σημεία αλλά και τομείς στους οποίους επιδέχεται βελτίωσης.

1.3 Εργαλεία Ανάπτυξης

Το σύστημα θα δημιουργηθεί με την γνωστή υπηρεσία αξιολόγησης και ερωτηματολογίων Google Forms σε συνεργασία με έναν Web Server ο οποίος θα σερβίρει την ιστοσελίδα η οποία αναπτύχθηκε με HTML και CSS. Περισσότερες πληροφορίες για τα εργαλεία αυτά και την χρήση τους θα αναλυθούν περαιτέρω στην συνέχεια.

1.4 Δομή Διπλωματικής Εργασίας

Η δομή της Διπλωματικής Εργασίας έχει ως εξής (βάσει κεφαλαίων):

Κεφάλαιο 2 Συστήματα Εκπαίδευσης και Διδασκαλίας

Στο Κεφάλαιο 2, γίνεται αναφορά για τα συστήματα εκπαίδευσης, και ειδικότερα για τα διδικτυακά συστήματα εκπαίδευσης σε γενικές γραμμές αλλά και εξειδικευμένα για το μάθημα ΕΠΑ131.

Κεφάλαιο 3 Απαιτήσεις Συστήματος

Στο κεφάλαιο 3, γίνεται αναφορά για τις απαιτήσεις που έχει το σύστημα που αναπτύσσεται στα πλαίσια της Διπλωματικής Εργασίας. Ειδικότερα, αναλύεται το τι πρέπει να επιτυγχάνει το παρόν σύστημα καθώς και τι παρέχει.

Κεφάλαιο 4 Σχεδίαση και Υλοποίηση Συστήματος

Στο κεφάλαιο 5, γίνεται αναφορά στον τρόπο σχεδιασμού και υλοποίησης του συστήματος, πράγματα που λαμβάνονται υπ' όψιν για την καλύτερη δυνατή αλληλεπίδραση μεταξύ του μαθητευόμενου και του συστήματος αλλά και εργαλεία, τεχνικές που συνέβαλαν στην ανέγερση του συστήματος.

Κεφάλαιο 5 Συμπεράσματα

Στο κεφάλαιο 5, γίνεται αναφορά στα συμπεράσματα που εξάγονται από την υλοποίηση του παρόντος συστήματος. Ειδικότερα αναφέρονται κάποιες δυσκολίες, παρατηρήσεις και κάποια γενικά σχόλια.

Κεφάλαιο 2

Συστήματα Εκπαίδευσης και Διδασκαλίας

2.1 Εισαγωγή	4
2.2 Μαθησιακό Περιβάλλον	4
2.3 Εκμάθηση Μέσω Διαδικτύου	5

2.1 Εισαγωγή

Πολλοί φοιτητές με την είσοδό τους στο Πανεπιστήμιο Κύπρου, δυσκολεύονται να ενταχθούν στον κόσμο του προγραμματισμού, κυρίως λόγω διαφορετικού μαθησιακού υπόβαθρου. Σε αρκετές περιπτώσεις, φοιτητές που κατέκτησαν μια θέση στο Πανεπιστήμιο Κύπρου από μια τεχνική σχολή, δυσκολεύονται περισσότερο στον προγραμματισμό, καθώς δεν έχουν διδαχθεί επαρκώς τις βασικές γνώσεις που απαιτούνται στο μάθημα ΕΠΛ131. Οι μεν βασικές γνώσεις διδάσκονται από την αρχή σε όλους τους φοιτητές στο Πανεπιστήμιο, ωστόσο λόγω της προηγούμενης εξοικειώσής τους, οι υπόλοιποι φοιτητές προχωρούν με πιο γοργούς ρυθμούς με αποτέλεσμα να υπάρχει σημαντικό μερίδιο φοιτητών που υστερεί. Το σύστημα το οποίο αναπτύχθηκε στα πλαίσια αυτής της εργασίας, έχει ως απώτερο σκοπό να μειώσει το χάσμα αυτό και να ενισχύσει τις ικανότητες όλων των φοιτητών ανεξαιρέτως γνωστικού υπόβαθρου.

2.2 Μαθησιακό Περιβάλλον

Η διδασκαλία γίνεται σε προκαθορισμένο τόπο και χρόνο από το Πανεπιστήμιο. Για κάποιους φοιτητές ο χρόνος αυτός ενδείκνυται να μην είναι αρκετός για να αντιληφθούν τις νέες έννοιες και τον τρόπο σκέψης τον οποίο διδάσκονται. Σε άλλες περιπτώσεις, για οποιοδήποτε προσωπικό λόγο, μπορεί οι φοιτητές να μην ήταν ικανοί να αποδόσουν επαρκώς την ημέρα της διάλεξης. Πρωτεύων στόχος του συστήματος που αναπτύχθηκε είναι να εξαλείψει τους παραπάνω παράγοντες και να καθοδηγήσει τους νεοεισερχόμενους φοιτητές στον σωστό δρόμο του προγραμματισμού και της προγραμματιστικής σκέψης.

2.3 Εκμάθηση Μέσω Διαδικτύου

Η πλατφόρμα η οποία αναπτύχθηκε στα πλαίσια της ατομικής διπλωματικής εργασίας παρέχει στους φοιτητές το περιεχόμενο των διαλέξεων για μελέτη. Κάθε κεφάλαιο συνοδεύεται από ασκήσεις οι οποίες εξετάζουν τον φοιτητή και με την ολοκλήρωσή τους, ο φοιτητής θα ξέρει ποιά σημεία τον δυσκόλεψαν και που πρέπει να εστιάσει την προσοχή του. Οι ασκήσεις αποτελούνται από: θεωρητικές ερωτήσεις, ερωτήσεις συμπλήρωσης κώδικα, ερωτήσεις εύρεσης λάθους σε υπάρχον κώδικα, ερωτήσεις για το τι θα εκτυπώσει υπάρχον κώδικας. Οι ασκήσεις είναι δομημένες με τέτοιον τρόπο ώστε να εξετάζουν συχνά θέματα που μπερδεύουν και δυσκολεύουν τους φοιτητές, ενώ παράλληλα μαθαίνουν τους φοιτητές τα βασικά του προγραμματισμού.

Κεφάλαιο 3

Απαιτήσεις Συστήματος

3.1 Εισαγωγή	6
3.2 Συλλογή Απαιτήσεων Συστήματος	7
3.2.1 Τρόποι Εξαγωγής Απαιτήσεων	7
3.2.2 Ερωτηματολόγιο	7
3.2.3 Συμπεράσματα Ερωτηματολογίου	10
3.3 Περιγραφή Συστήματος	11
3.4 Απαιτήσεις Υλικού και Λογισμικού	12
3.4.1 Απαιτήσεις Χρήστη	12
3.4.2 Απαιτήσεις Παροχέα	12
3.5 Ιδιότητες και Χαρακτηριστικά Συστήματος	12
3.5.1 Αξιοπιστία και Διαθεσιμότητα	13
3.5.2 Φορητότητα	13
3.5.3 Επεκτασιμότητα	

3.1 Εισαγωγή

Στο κεφάλαιο αυτό περιγράφονται οι απαιτήσεις του συστήματος ως προς το περιεχόμενο που θα παρέχει σύμφωνα με έρευνα που διεξήχθη μέσω ερωτηματολογίου. Επίσης γίνεται μια περιγραφή του συστήματος και αναλύονται οι απαιτήσεις του ως προς το υλικό και το λογισμικό για τους τελικούς χρήστες, αλλά και τον παροχέα του συστήματος. Επιπρόσθετα, αναλύονται σημαντικά χαρακτηριστικά του συστήματος που εξασφαλίζουν την γρήγορη, ορθή και αξιόπιστη εξυπηρέτηση των χρηστών του συστήματος.

3.2 Συλλογή Απαιτήσεων Συστήματος

3.2.1 Τρόποι Εξαγωγής Απαιτήσεων

Το παρόν σύστημα προορίζεται για τους φοιτητές του μαθήματος ΕΠΛ131 του Πανεπιστημίου Κύπρου και ο καλύτερος τρόπος για την εξαγωγή των απαιτήσεων ενός τέτοιου συστήματος είναι να ρωτήσουμε φοιτητές του τμήματος Πληροφορικής. Ειδικότερα, ρωτήθηκαν μέσω ερωτηματολογίου 55 φοιτητές του τμήματος Πληροφορικής ανεξαρτήτως έτους φοίτησης, δηλαδή φοιτητές που ολοκλήρωσαν το μάθημα ΕΠΛ131 αλλά και φοιτητές που το διδάσκονταν αυτήν την χρονική περίοδο. Στο ερωτηματολόγιο που κυκλοφόρησε, σημαντικό ποσοστό των φοιτητών που απάντησε ήταν του 1^{ου} έτους πράγμα που δηλώνει πως μας ανέφεραν τις δυσκολίες που αντιμετωπίζουν στο μάθημα την στιγμή που το διδάσκονται.

Επιπρόσθετα, ρωτήθηκαν φοιτητές εκτός ερωτηματολογίου μέσω συζήτησης για τον καθορισμό των απαιτήσεων βάσει των δυσκολιών που οι ίδιοι αντιμετώπισαν κατά την παρακολούθησή τους του μαθήματος ΕΠΛ131.

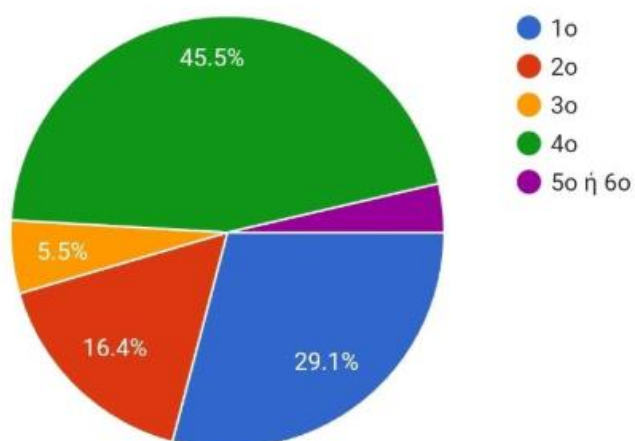
Από τις απαιτήσεις του συστήματος που περιγράφουν τις δυσκολίες των φοιτητών του μαθήματος του ΕΠΛ131, δεν θα μπορούσαν να λείπουν και προσωπικές μου δυσκολίες κατά την παρακολούθηση του μαθήματος λίγα χρόνια νωρίτερα, καθώς κι εγώ υπήρξα στην θέση του νεοεισερχόμενου φοιτητή αναζητώντας ένα σύστημα υποστήριξης.

3.2.2 Ερωτηματολόγιο

Για την καλύτερη εξαγωγή των απαιτήσεων, έχει διεξαχθεί το ακόλουθο ερωτηματολόγιο μέσω του οποίου έχουμε αριθμούς και στατιστικά των φοιτητών για τις δυσκολίες τους καθώς και για την ανάγκη ανέγερσης του συστήματος. Παρακάτω παρουσιάζονται τα αποτελέσματα του ερωτηματολογίου ενώ στο Παράρτημα Α θα βρείτε αυτούσιο το ερωτηματολόγιο που δόθηκε στους φοιτητές.

Ποιό είναι το έτος σπουδών σας;

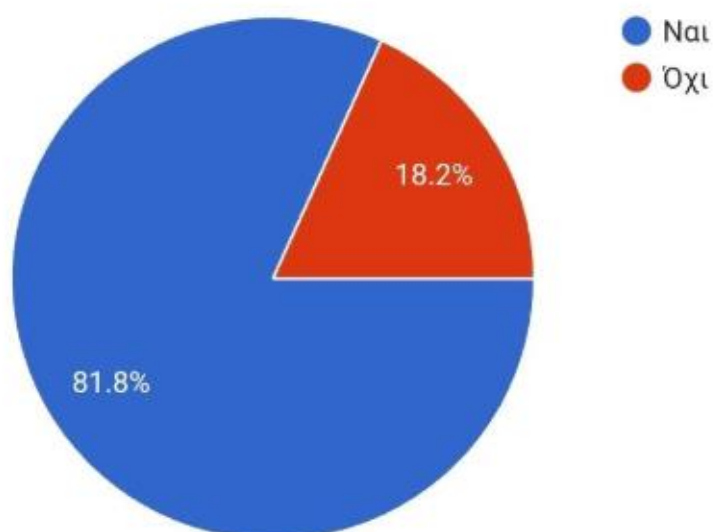
55 responses



Σχήμα 3.2.2.1: Διάγραμμα κατανομής έτους σπουδών

Πριν την έναρξη της φοίτησής σας είχατε προηγούμενη εμπειρία με τον προγραμματισμό;

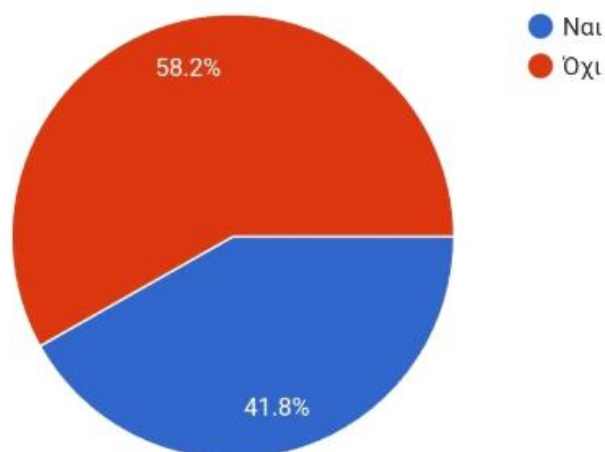
55 responses



Σχήμα 3.2.2.2: Διάγραμμα προηγούμενης εμπειρίας

Αντιμετωπίσατε/αντιμετωπίζετε δυσκολίες στο μάθημα ΕΠΛ131 - Αρχές Προγραμματισμού;

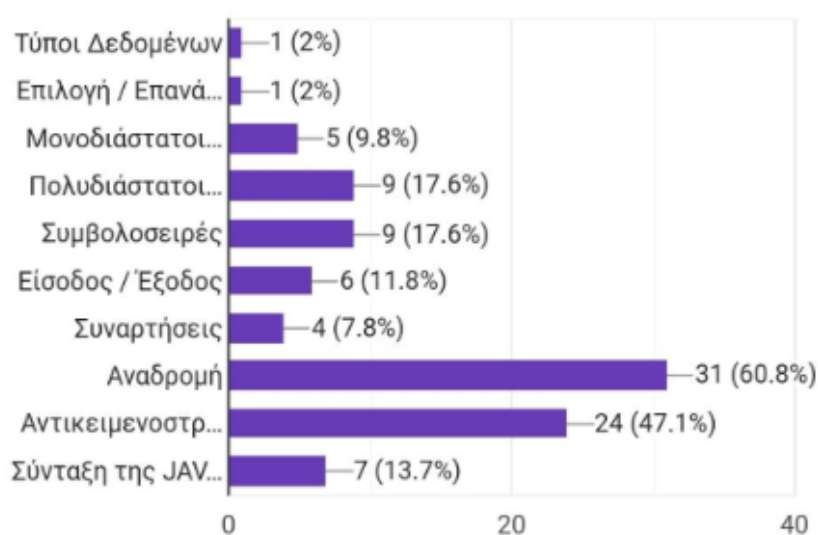
55 responses



Σχήμα 3.2.2.3: Διάγραμμα κατανομής δυσκολιών

Τι σας δυσκόλεψε/δυσκολεύει στο μάθημα ΕΠΛ131; (Μπορείτε να επιλέξετε περισσότερες από μία επιλογές)

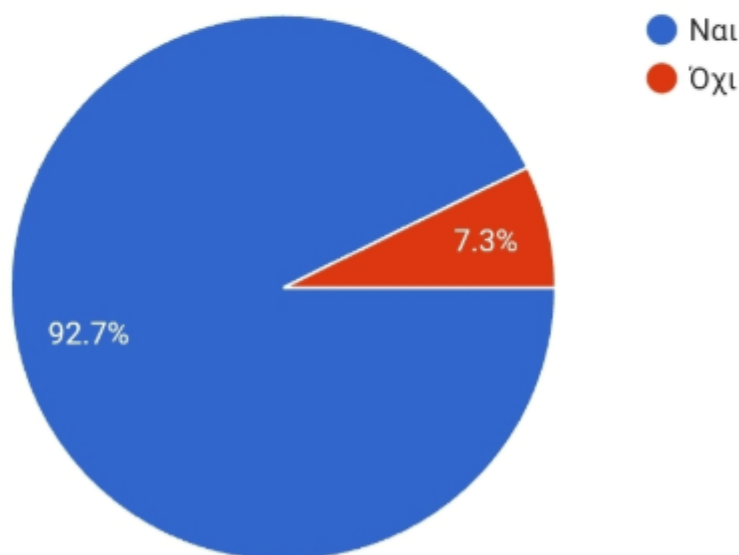
51 responses



Σχήμα 3.2.2.4: Διάγραμμα δυσκολιών

Θα χρησιμοποιούσατε κάποιο
διαδραστικό online σύστημα
εκπαίδευσης για την περαιτέρω
κατανόηση των πιο πάνω εννοιών;

55 responses



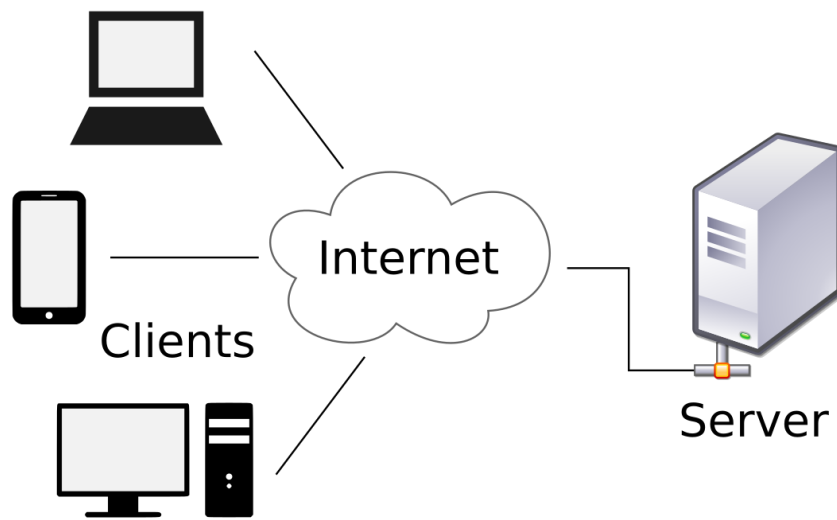
Σχήμα 3.2.2.5: Διάγραμμα πρόθεσης χρήσης του συστήματος

3.2.3 Συμπεράσματα Ερωτηματολογίου

Από τα αποτελέσματα του ερωτηματολογίου συμπεραίνεται ότι υπάρχει ποσοστό των φοιτητών σχεδόν 20% (18.2%) το οποίο εισέρχεται στο Πανεπιστήμιο Κύπρου χωρίς οποιαδήποτε προηγούμενη εμπειρία στον προγραμματισμό. Το ποσοστό αυτό δεν είναι η πλειοψηφία αλλά δεν είναι και ασήμαντο. Οι φοιτητές αυτοί μπορούν να επωφεληθούν του παρόντος συστήματος. Επίσης ένα όχι τόσο μικρό ποσοστό των φοιτητών (41.8%) το οποίο αντιμετώπισε δυσκολίες στο μάθημα ΕΠΛ131. Από τα αποτελέσματα όσον αφορά το τι δυσκόλεψε τους φοιτητές, το μεγαλύτερο ποσοστό αντιμετώπισε δυσκολίες στα κεφάλαια της Αναδρομής και στον Αντικειμενοστραφή Προγραμματισμό (60.8% και 47.1% αντίστοιχα). Στην ερώτηση για το αν θα χρησιμοποιούσαν οι φοιτητές κάποιο σύστημα για την περαιτέρω κατανόηση των εννοιών που δυσκολεύτηκαν, η συντριπτική πλειοψηφία απάντησε «ναι» (92.7%).

3.3 Περιγραφή Συστήματος

Το σύστημα βασίζεται στο μοντέλο Πελάτη – Διακομιστή ή αλλιώς Client – Server. Στον διακομιστή είναι αποθηκευμένα τα αρχεία HTML και CSS τα οποία είναι υπεύθυνα για την λειτουργικότητα και παρουσίαση του συστήματος στους χρήστες οι οποίοι μπορούν να έχουν πρόσβαση στην σελίδα του συστήματος μέσω οποιαδήποτε συσκευής συνδέεται στο διαδίκτυο και έχει φυλλομετρητή (browser).



Σχήμα 3.3.1: Μοντέλο Πελάτη - Διακομιστή

Το σύστημα στο οποίο θα έχουν πρόσβαση οι χρήστες αποτελείται από μια ιστοσελίδα πάνω στην οποία υπάρχουν οι διαλέξεις του μαθήματος αριθμημένες και με τον τίτλο τους, καθώς και υποστηρικτικό υλικό για κάθε ένα από τα κεφάλαια.

Το υποστηρικτικό υλικό για τους φοιτητές είναι ερωτήσεις κατανόησης της μορφής:

- Θεωρητικές Ερωτήσεις.
- Ερωτήσεις Συμπλήρωσης Κώδικα.
- Ερωτήσεις Εύρεσης Λάθους σε Κώδικα.
- Ερωτήσεις Εξόδου/Εκτύπωσης Κώδικα.

Επιπρόσθετα υπάρχει και υποστηρικτικό υλικό για τα συστήματα UNIX, για την πρόσβαση στους υπολογιστές του τμήματος Πληροφορικής, αλλά και γενικές οδηγίες των πληροφοριακών συστημάτων του Πανεπιστημίου και επιπρόσθετες ασκήσεις εξάσκησης, παραδείγματα. Επίσης παρέχεται η ιστοσελίδα της W3Schools ως εφεδρικό υλικό σε περίπτωση που το υπάρχον δεν καλύπτει επαρκώς τον φοιτητή.

3.4 Απαιτήσεις Υλικού και Λογισμικού

3.4.1 Απαιτήσεις Χρήστη

Οι απαιτήσεις λογισμικού για τον χρήστη είναι ελάχιστες, καθώς το μόνο που θα χρειαστεί είναι μια συσκευή με φυλλομετρητή (browser), και σύνδεση στο διαδίκτυο. Το σύστημα είναι απλό και ελαφρύ με αποτέλεσμα να μπορεί να λειτουργήσει σε όλες τις συσκευές με οποιοδήποτε υλικό που υποστηρίζει τα πιο πάνω.

3.4.2 Απαιτήσεις Παροχέα

Οι απαιτήσεις λογισμικού του παροχέα είναι και αυτές απλές. Αρκεί να μπορεί να φιλοξενήσει και να σερβίρει την ιστοσελίδα η οποία είναι απλή. Οι εκπαιδευτικές ασκήσεις παρέχονται από την Google μέσω της υπηρεσίας Google Forms, με αποτέλεσμα να μην επιβαρύνεται ο διακομιστής. Από τη μεριά του υλικού ισχύουν τα παραπάνω, ενώ αξίζει να σημειωθεί πως το μέγεθος του συστήματος μαζί με το υλικό των διαλέξεων και του επιπρόσθετου υλικού δεν υπερβαίνει τα 17MB!

3.5 Ιδιότητες και Χαρακτηριστικά Συστήματος

3.5.1 Αξιοπιστία και Διαθεσιμότητα

Το σύστημα είναι σχεδιασμένο με τρόπο ώστε να μην είναι δυνατή η λανθασμένη χρήση του. Ο τρόπος χρήσης του είναι ξεκάθαρος και το σύστημα λόγω του ότι είναι ιστοσελίδα συμπεριφέρεται σε κάθε κατάσταση σωστά. Καθώς οι ασκήσεις φιλοξενούνται στην αξιόπιστη και με κύρος πλατφόρμα Google Forms, η αξιοπιστία και διαθεσιμότητα του συστήματος είναι εγγυημένη. Το μόνο σημείο που μπορεί να επιφέρει ζητήματα διαθεσιμότητας είναι ο διακομιστής ο οποίος σερβίρει το περιεχόμενο στους χρήστες, αλλά αυτό δεν είναι πρόβλημα καθώς μπορούμε να επιλέξουμε έναν αξιόπιστο παροχέα για να φιλοξενήσουμε το σύστημά μας.

3.5.2 Φορητότητα

Λόγω της φύσης του συστήματος (ιστοσελίδα), δεν τίθεται κανένα ζήτημα φορητότητας καθώς οι ιστοσελίδες είναι το πιο φορητό μέσο που μπορεί να χρησιμοποιηθεί για ένα τέτοιο σύστημα. Όλοι οι χρήστες έχουν άμεση πρόσβαση από παντού (σταθερούς υπολογιστές, laptop, κινητά τηλέφωνα/smartphones, ταμπλέτες), ενώ ο διακομιστής για να αλλάξει, αρκεί να μεταφερθούν τα αρχεία του πηγαίου κώδικα της ιστοσελίδας.

3.5.3 Επεκτασιμότητα

Λόγω της φύσης του συστήματος (ιστοσελίδα) και πάλι, το σύστημα είναι άκρως επεκτάσιμο, αφού ο κώδικας HTML μπορεί εύκολα να επεκταθεί για να υποστηρίξει περισσότερο περιεχόμενο, και για τυχόν νέες ασκήσεις, με την πλατφόρμα Google Forms εύκολα μπορούν να δημιουργηθεί νέο περιεχόμενο.

Κεφάλαιο 4

Σχεδίαση και Υλοποίηση Συστήματος

4.1 Εισαγωγή	14
4.2 Σχεδίαση Ιστοσελίδας	14
4.2.1 Κυρίως Μενού	14
4.2.2 Ασκήσεις	15
4.2.3 Συμπληρωματικό Υλικό	15
4.3 Εργαλεία, Προγράμματα και Υπηρεσίες που Χρησιμοποιήθηκαν	16
4.3.1 HTML	16
4.3.2 CSS	16
4.3.3 Google Forms	16
4.3.4 Microsoft Word	16
4.4 Παρουσίαση Συστήματος	17

4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα γίνει αναφορά για τον τρόπο σχεδίασης και υλοποίησης του συστήματος. Ειδικότερα, θα αναλυθεί η ιστοσελίδα και το περιεχόμενό της, καθώς και τα εργαλεία, προγράμματα και οι υπηρεσίες που χρησιμοποιήθηκαν για την ανέγερση του συστήματος. Τέλος θα γίνει μια παρουσίαση του συστήματος.

4.2 Σχεδίαση Ιστοσελίδας

4.2.1 Κυρίως Μενού

Το κυρίως μενού σχεδιάστηκε με γνώμονα την απλότητα με σκοπό να είναι όσο το δυνατόν πιο λειτουργικό. Χρησιμοποιήθηκε μινιμαλιστική προσέγγιση στον σχεδιασμό του. Το κυρίως μενού περιέχει το λογότυπο του Πανεπιστημίου και της Java, έναν τίτλο, μισ σύντομη

περιγραφή για το περιεχόμενο της ιστοσελίδας και δύο πίνακες. Ο πρώτος πίνακας περιέχει κάθε κεφάλαιο των διαλέξεων του μαθήματος ΕΠΑ131 σε μορφή PDF και αντίστοιχα για κάθε κεφάλαιο τις ασκήσεις τους. Ο δεύτερος πίνακας περιέχει το συμπληρωματικό υλικό με διαφόρους οδηγούς χρήσης και επιπλέον υλικό για το μάθημα (παραδείγματα, επιπρόσθετες ασκήσεις, ιστοσελίδα W3Schools, ιστοσελίδα HackerRank, ιστοσελίδα του βιβλίου).

4.2.2 Ασκήσεις

Η καρδιά του συστήματος, οι ασκήσεις, δημιουργήθηκαν βάσει του περιεχομένου του κάθε κεφαλαίου του μαθήματος. Αφού μελετήθηκαν όλα τα κεφάλαια του μαθήματος, έγινε περισυλλογή των πιο σημαντικών εννοιών, προβλημάτων και θεμάτων που ενδεχομένως να μπερδεύουν και μετατράπηκαν σε ασκήσεις οι οποίες θα βοηθήσουν τους φοιτητές να καταλάβουν εις βάθος τις έννοιες αυτές. Οι ασκήσεις φιλοξενούνται στην πλατφόρμα της Google, την Google Forms, στην οποία οι φοιτητές μπορούν πολύ εύκολα και απλά να λύσουν τις ασκήσεις και σε πραγματικό χρόνο να λάβουν ανατροφοδότηση.

4.2.3 Συμπληρωματικό Υλικό

Ένα άλλο πολύ σημαντικό μέρος του συστήματος είναι το συμπληρωματικό υλικό. Στον πίνακα αυτόν υπάρχουν οι εξής οδηγοί οι οποίοι θα φανούν ιδιαίτερα χρήσιμοι στους φοιτητές του μαθήματος ΕΠΑ131, καθώς είναι όλοι μαζεμένοι σε ένα μέρος:

- Οδηγός Εισαγωγής UNIX
- Οδηγός X2GO & Remote Desktop
- Οδηγός Χρήσης VPN
- Γενικές Οδηγίες Πληροφοριακών Συστημάτων του Πανεπιστημίου Κύπρου
- Σύνδεσμος για την stdlib.jar

Υπάρχουν επίσης:

- Επιπρόσθετες Ασκήσεις Εξάσκησης
- Παραδείγματα Αντικειμένων
- Μαθήματα W3Schools
- Εξάσκηση στην ιστοσελίδα HackerRank
- Η ιστοσελίδα του βιβλίου του μαθήματος
- Μεγάλη άσκηση με όλες τις ασκήσεις σε μια.

4.3 Εργαλεία, Προγράμματα και Υπηρεσίες που Χρησιμοποιήθηκαν

4.3.1 HTML

Για τον σκελετό του συστήματος έγινε χρήση της ευρέως γνωστής γλώσσας σήμανσης (markup language) της HTML (Hyper Text Markup Language) στην έκδοσή της HTML5. Μέσω αυτής έγινε η ενσωμάτωση κειμένου, εικόνων, αρχείων και συνδέσμων στην ιστοσελίδα.

4.3.2 CSS

Για την καλύτερη παρουσίαση του περιεχομένου της ιστοσελίδας, έγινε χρήση της γλώσσας στυλ φύλλου (stylesheet language) της CSS (Cascading Style Sheets). Ειδικότερα, με την γλώσσα αυτή, πετυχαίνεται το responsiveness (σωστή εμφάνιση ιστοσελίδας σε κινητές συσκευές), η αλλαγή χρώματος του φόντου, αλλαγή γραμματοσειράς και μεγέθους, αλλαγή τοποθέτησης εικόνων και αλλαγή χρώματος των πινάκων.

4.3.3 Google Forms

Η υπηρεσία Google Forms είναι μια υπηρεσία η οποία μας επιτρέπει να δημιουργούμε ερωτηματολόγια και κουίζ. Η υπηρεσία αυτή χρησιμοποιήθηκε ευρέως για την ανάπτυξη του συστήματος, από την εξαγωγή απαιτήσεων, μέχρι και στον τρόπο παρουσίασης των ασκήσεων και τρόπο αξιολόγησης.

4.3.4 Microsoft Word

Η Microsoft Word είναι το πρόγραμμα δημιουργίας και επεξεργασίας κειμένου που χρησιμοποιήθηκε για την δημιουργία του παρόντος εγγράφου.

4.4 Παρουσίαση Συστήματος



**University
of Cyprus**

Υποστηρικτικό Υλικό ΕΠΛ131 - Αρχές Προγραμματισμού Ι



Σε αυτήν την σελίδα θα βρείτε υλικό για το μάθημα ΕΠΛ131 – Αρχές Προγραμματισμού Ι

Εδώ υπάρχουν πάνω από 340 ασκήσεις κατανόησης βασισμένες σε κάθε ένα από τα 11 κεφάλαια του μαθήματος.

Οι ασκήσεις αυτές θα σας βοηθήσουν να καταλάβετε περαιτέρω το κάθε κεφάλαιο, καλλιεργώντας τις σημαντικότερες έννοιες με:

- θεωρητικές ερωτήσεις
- ερωτήσεις συμπλήρωσης κώδικα
- ερωτήσεις εύρεσης λάθους σε κώδικα
- ερωτήσεις εξόδου κώδικα

Οι ασκήσεις θα σας εξετάσουν σε βασικές έννοιες, αλλά και σημαντικές λεπτομέρειες οι οποίες μπορεί να σας διέφυγαν ή να σας δυσκόλεψαν. Θα βρείτε επίσης και τις διαφάνειες από κάθε κεφάλαιο μαζί με εργαστηριακές ασκήσεις για δική σας ευκολία.

Κάθε κεφάλαιο συνοδεύεται από ένα ερωτηματολόγιο το οποίο περιέχει την αξιολόγησή σας. Μετά από κάθε άσκηση θα λαμβάνετε ανατροφοδότηση και θα μπορείτε να δείτε τι σας δυσκόλεψε και που πρέπει να δώσετε βάση.

Διαλέξεις	Ασκήσεις Βάσει Επιπέδου Δυσκολίας			Συμπληρωματικό Υλικό
1. Βασικοί Τύποι Δεδομένων	Εισαγωγικό	N/A	N/A	Οδηγός Εισαγωγής UNIX
2. Αλγοριθμικές Δομές Επιλογής και Επανάληψης	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Οδηγός X2GO & Remote Desktop
3. Πίνακες Μιας Διάστασης	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Οδηγός Χρήσης VPN
4. Σειμβολοσειρές	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Γενικές Οδηγίες Πληροφοριακών Συστημάτων
5. Εισόδος και Έξοδος	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Κατεβάστε την stdlib.jar
6. Αρχές Ορθού Προγραμματισμού	Εισαγωγικό	N/A	N/A	Επιπρόσθετες Ασκήσεις Εξάσκησης
7. Συναρτήσεις	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Παραδείγματα Αντικειμένων
8. Πίνακες Πολλαπλών Διαστάσεων	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Μαθήματα W3Schools
9. Αναδρομή	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Εξάσκηση στο HackerRank
10. Αξιοποίηση και Χρήση Αντικειμένων	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Όλες οι Ασκήσεις
11. Δημιουργία Αντικειμένων	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο	Βιβλίο του Μαθήματος

Για περισσότερες πληροφορίες, ερωτήσεις ή εισηγήσεις:
ep131diplomatic@gmail.com
Αντώνιος Κιουρτζίδης Μάιος 2024

Σχήμα 4.4.1: Αρχική Σελίδα / Κυρίως Μενού

Μπορείτε να δοκιμάσετε το σύστημα εδώ: <https://www.cs.ucy.ac.cy/~akiour01/>

Εδώ βλέπουμε την αρχική σελίδα του συστήματος. Πάνω υπάρχει μια σύντομη περιγραφή για το περιεχόμενο και από κάτω δύο πίνακες. Ο πρώτος πίνακας περιέχει τις διαλέξεις κάθε κεφαλαίου μαζί με τις αντίστοιχες ασκήσεις σε τρία επίπεδα δυσκολίας: Εισαγωγικό, Ενδιάμεσο, Προχωρημένο. Δίπλα βρίσκεται ο δεύτερος πίνακας με το συμπληρωματικό υλικό του μαθήματος ΕΠΛ131. Αξίζει να σημειωθεί πως με το πάτημα του συνδέσμου [Κατεβάστε την stdlib.jar](#), κατεβαίνει η βιβλιοθήκη στην συσκευή του χρήστη. Στο τέλος της σελίδας υπάρχει ένα ηλεκτρονικό ταχυδρομείο στο οποίο οι χρήστες μπορούν να καταβάλουν ερωτήσεις και εισηγήσεις. Τον πηγαίο κώδικα μπορείτε να τον βρείτε εδώ: <https://we.tl/t-kk8uni0QyA>

Υποστηρικτικό Υλικό ΕΠΛ131 - Αρχές Προγραμματισμού Ι

Σε αυτήν την σελίδα θα βρείτε υλικό για το μάθημα ΕΠΛ131 - Αρχές Προγραμματισμού Ι

Εδώ υπάρχουν πάνω από 340 ασκήσεις κατανόησης βασισμένες σε κάθε ένα από τα 11 κεφάλαια του μαθήματος.

Οι ασκήσεις αυτές θα σας βοηθήσουν να καταλάβετε περαιτέρω το κάθε κεφάλαιο, καλλιεργώντας τις σημαντικότερες έννοιες με:

- θεωρητικές ερωτήσεις
- ερωτήσεις συμπλήρωσης κώδικα
- ερωτήσεις εύρεσης λάθους σε κώδικα
- ερωτήσεις εξόδου κώδικα

Οι ασκήσεις θα σας εξετάσουν σε βασικές έννοιες, αλλά και σημαντικές λεπτομέρειες οι οποίες μπορεί να σας διέφυγαν ή να σας δυσκόλεψαν. Θα βρείτε επίσης και τις διαφάνειες από κάθε κεφάλαιο μαζί με εργαστηριακές ασκήσεις για δική σας ευκολία.

Κάθε κεφάλαιο συνοδεύεται από ένα ερωτηματολόγιο το οποίο περιέχει την αξιολόγησή σας. Μετά από κάθε άσκηση θα λαμβάνετε ανατροφοδότηση και θα μπορείτε να δείτε τι σας δυσκόλεψε και που πρέπει να δώσετε βάση.

Διαλέξεις	Ασκήσεις	Θέμα	Επίπεδο
1. Βασικές Έννοιες Προγραμματισμού	Εισαγωγικό	N/A	N/A
2. Αλγόριθμοί, Δομές Επικοινωνίας και Επικοινωνία	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο
3. Πίνακες Μιας Διάστασης	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο
4. Στοιχειώδεις Δομές	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο
5. Εισαγωγή και Εξέταση	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο
6. Αρχές Οργάνωσης Προγραμματισμού	Εισαγωγικό	N/A	N/A
7. Συναρτήσεις	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο
8. Πίνακες Πολυδιάστατων Δομών	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο
9. Αναδρομή	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο
10. Αλγόριθμοι και Χρήση Αλγορίθμων	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο
11. Δομημένος Προγραμματισμός	Εισαγωγικό	Ενδιάμεσο	Προχωρημένο

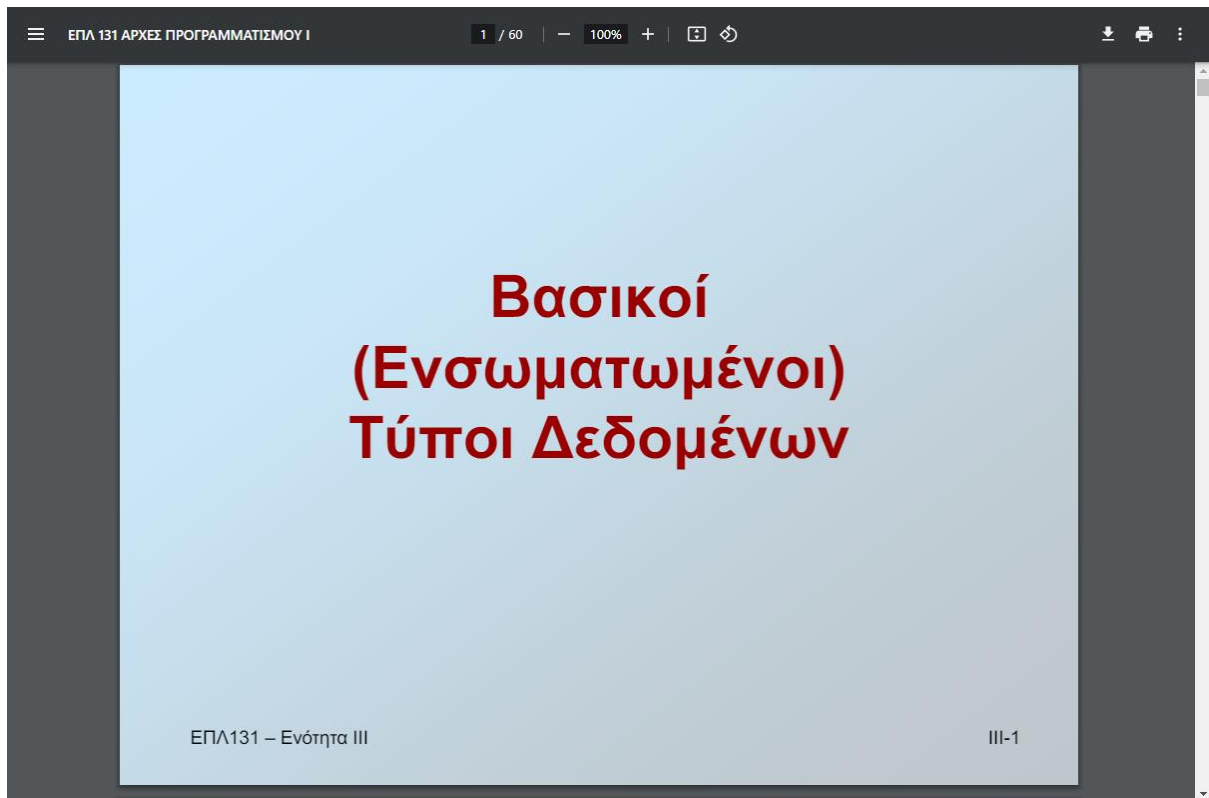
Συμπληρωματικό Υλικό
Οδηγός Εισαγωγής UNIX
Οδηγός K2SO & Remote Desktop
Οδηγός Κατάστασης VPN
Γενικές Οδηγίες Πληροφορικών Συστημάτων
Κατεβάστε την εγχειρίδα
Επιστρέψτε στην Εξέταση
Προβλήματα Αντικειμενικού
Μαθήματα Windows
Εξάσκηση στο Hackerrank
Οκεί οι Ασκήσεις
Βιβλίο του Μαθήματος

Για περισσότερες πληροφορίες, ερωτήσεις ή εισηγήσεις:

epl131diplomatie@gmail.com

Αντώνιος Κουματζής
Μάιος 2024

Σχήμα 4.4.2: Κυρίως Μενού από κινητή συσκευή - smartphone



Σχήμα 4.4.3: Διαφάνειες πρώτης διάλεξης

Με το πάτημα στον σύνδεσμο [1. Βασικοί Τύποι Δεδομένων](#), το σύστημα παραπέμπει τον χρήστη στην αντίστοιχη διάλεξη.

1α. Βασικοί (Ενσωματωμένοι) Τύποι Δεδομένων

Εισαγωγικό Επίπεδο

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

Οι βασικοί τύποι δεδομένων μπορούν να διασπαστούν σε απλούστερα στοιχεία.

1 βαθμός

- ☐ Ορθό
- ☐ Λάθος

Στην γλώσσα προγραμματισμού Java, υπάρχει η δυνατότητα δημιουργίας νέων τύπων δεδομένων από τον προγραμματιστή.

1 βαθμός

- ☐ Ορθό
- ☐ Λάθος

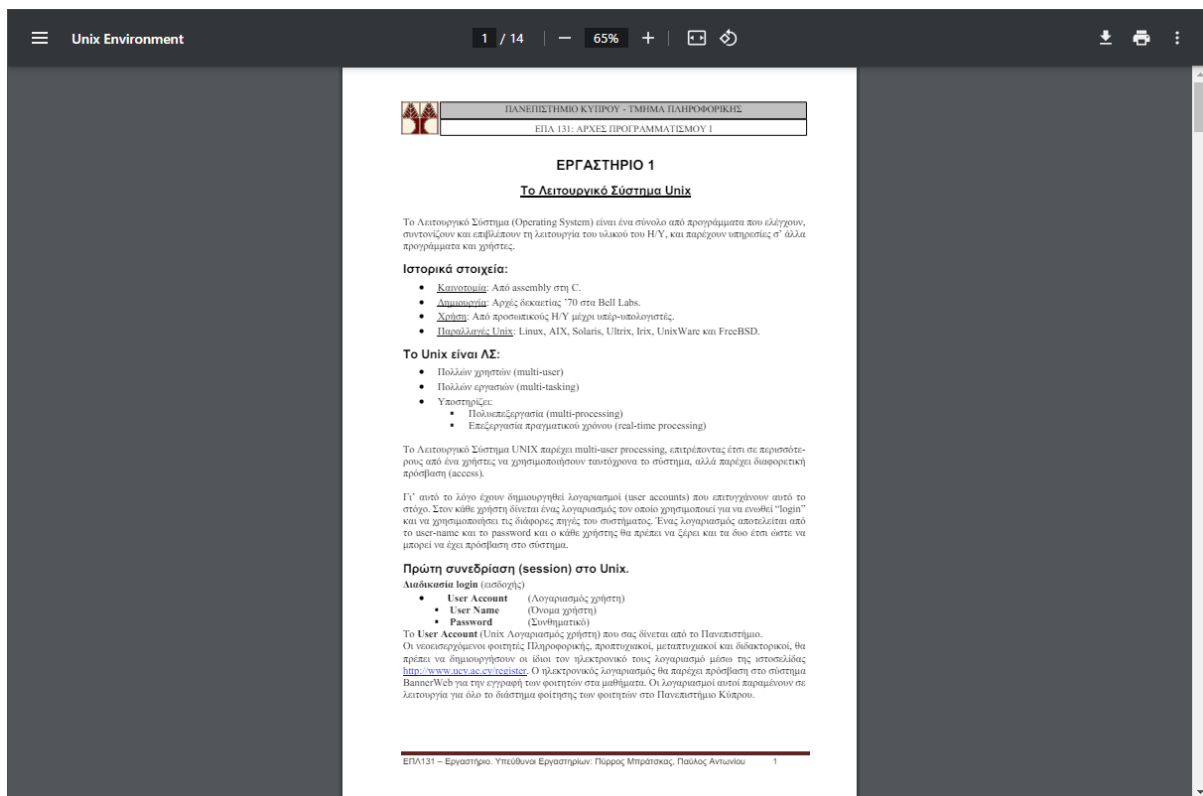
Το αποτέλεσμα της πράξης $5/3$ μεταξύ δύο ακεραίων (integers) στην Java είναι:

1 βαθμός

- ☐ 1.66666667
- ☐ 1
- ☐ Δεν ορίζεται

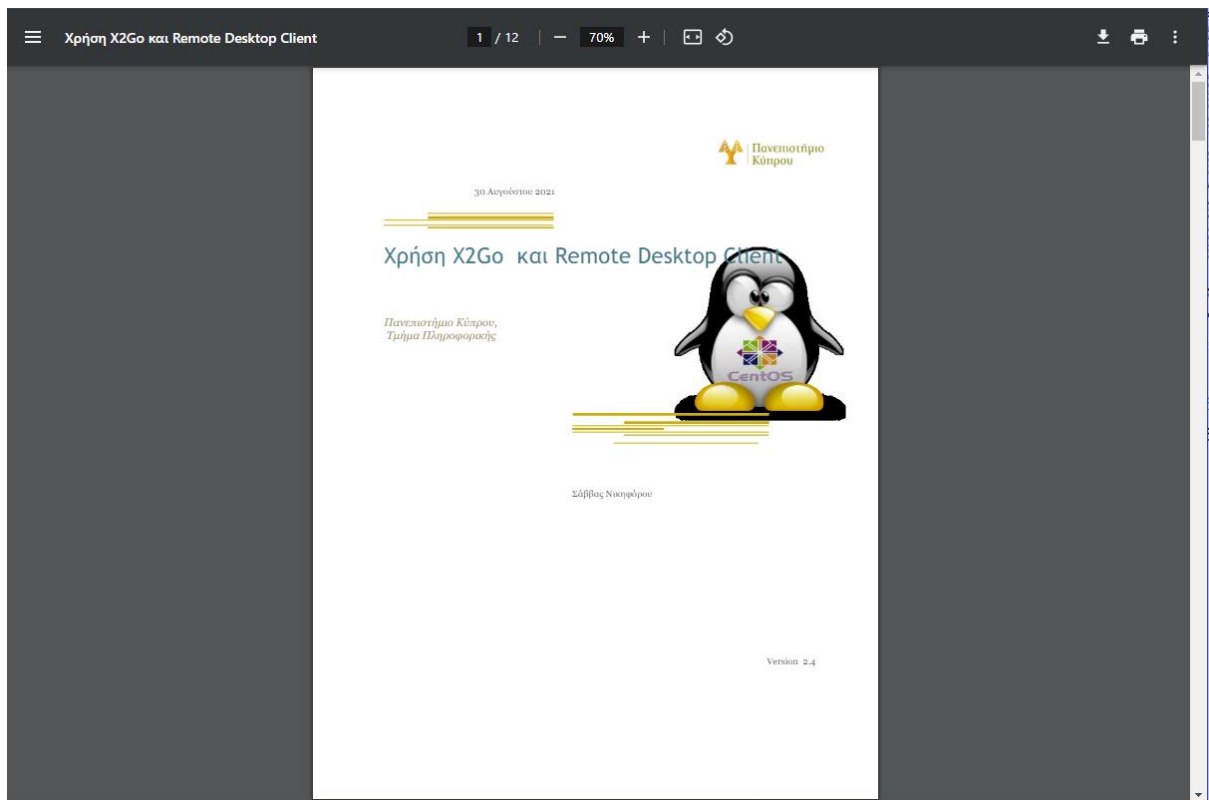
Σχήμα 4.4.4: Πρώτη άσκηση

Με το πάτημα στον σύνδεσμο [Εισαγωγικό](#) του πρώτου κεφαλαίου, το σύστημα παραπέμπει τον χρήστη στην σελίδα της Google Forms που περιέχει τις ασκήσεις του αντίστοιχου κεφαλαίου. Περισσότερα για τις ασκήσεις θα βρείτε στο Παράρτημα Β. Σημειώστε ότι στο Παράρτημα Β οι ασκήσεις είναι μαζεμένες ανά κεφάλαιο και όχι ανά επίπεδο δυσκολίας όπως είναι δομημένη η ιστοσελίδα. Επίσης σε κάθε ερωτηματολόγιο η σειρά των ασκήσεων είναι κάθε φορά διαφορετική.

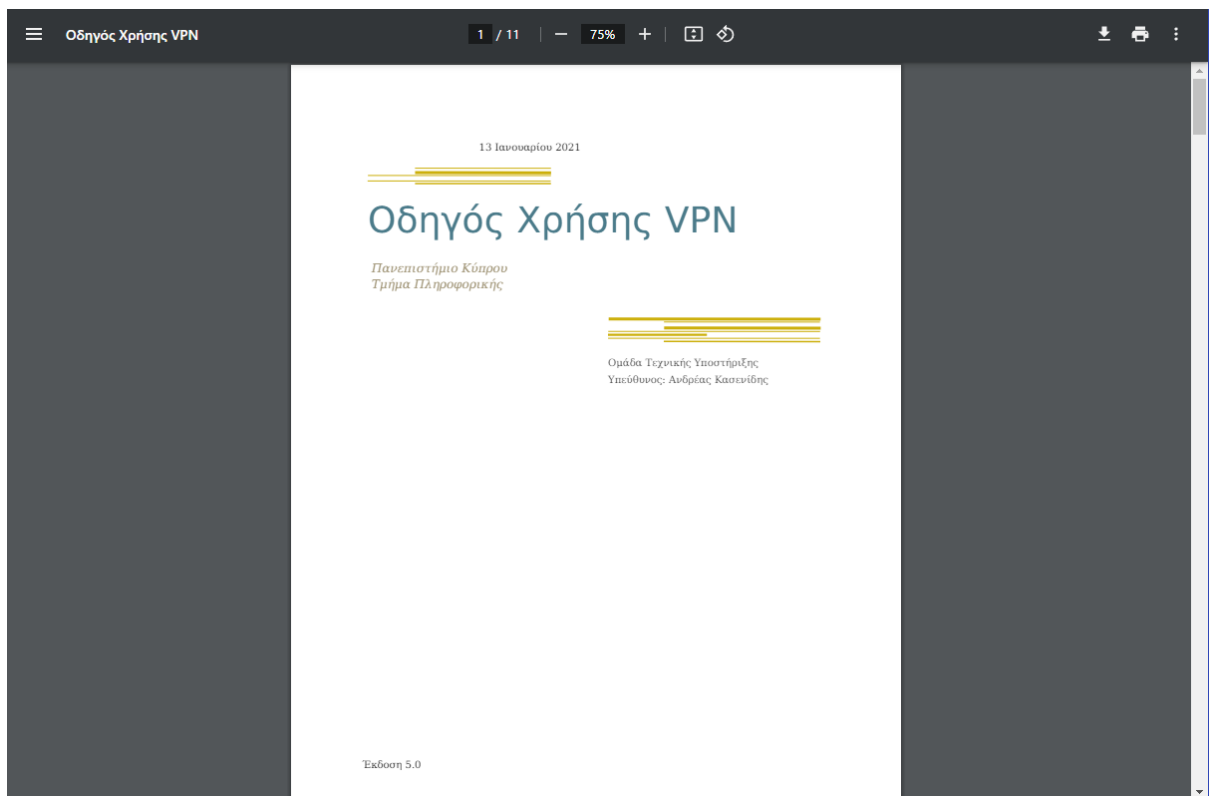


Σχήμα 4.4.5: Συμπληρωματικό Υλικό: Οδηγός Εισαγωγής UNIX

Με το πάτημα του συνδέσμου [Οδηγός Εισαγωγής UNIX](#), το σύστημα παραπέμπει στον χρήστη στον αντίστοιχο οδηγό. Το ίδιο ισχύει και για τα Σχήματα 4.4.6 έως 4.4.13



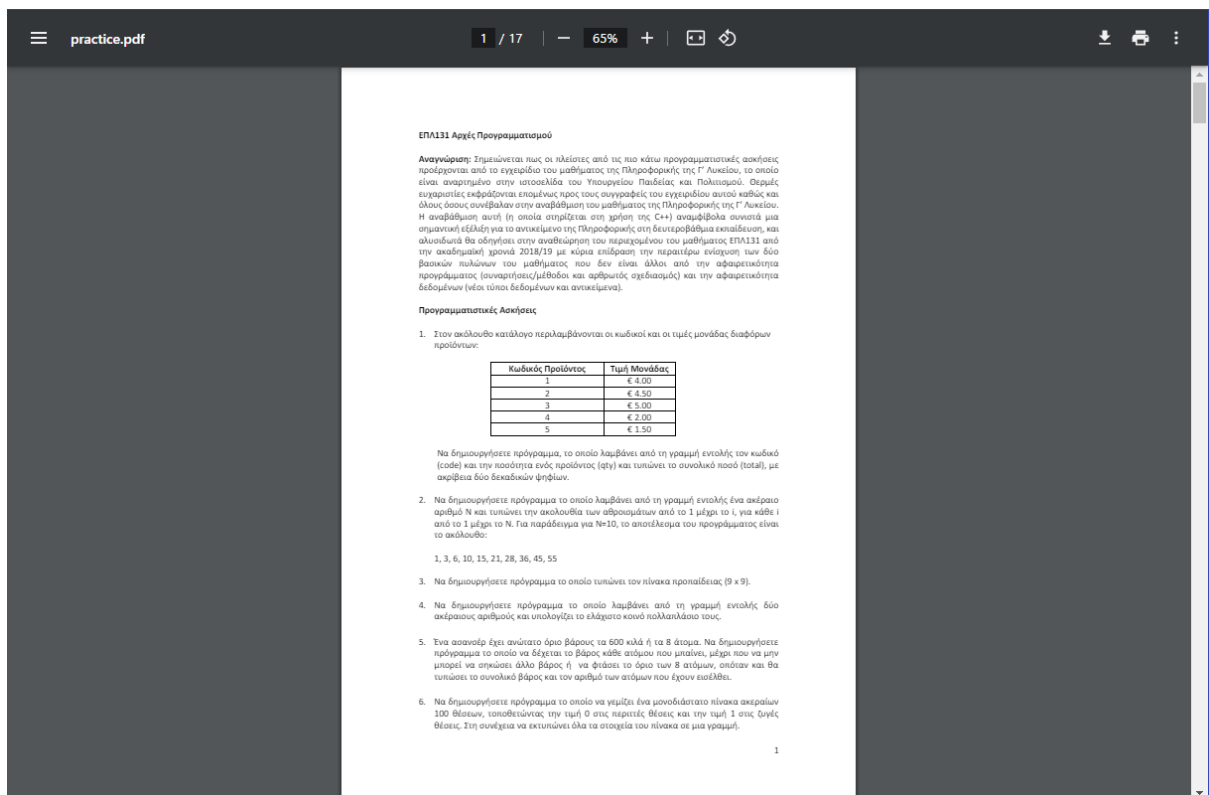
Σχήμα 4.4.6: Χρήση X2GO και Remote Desktop



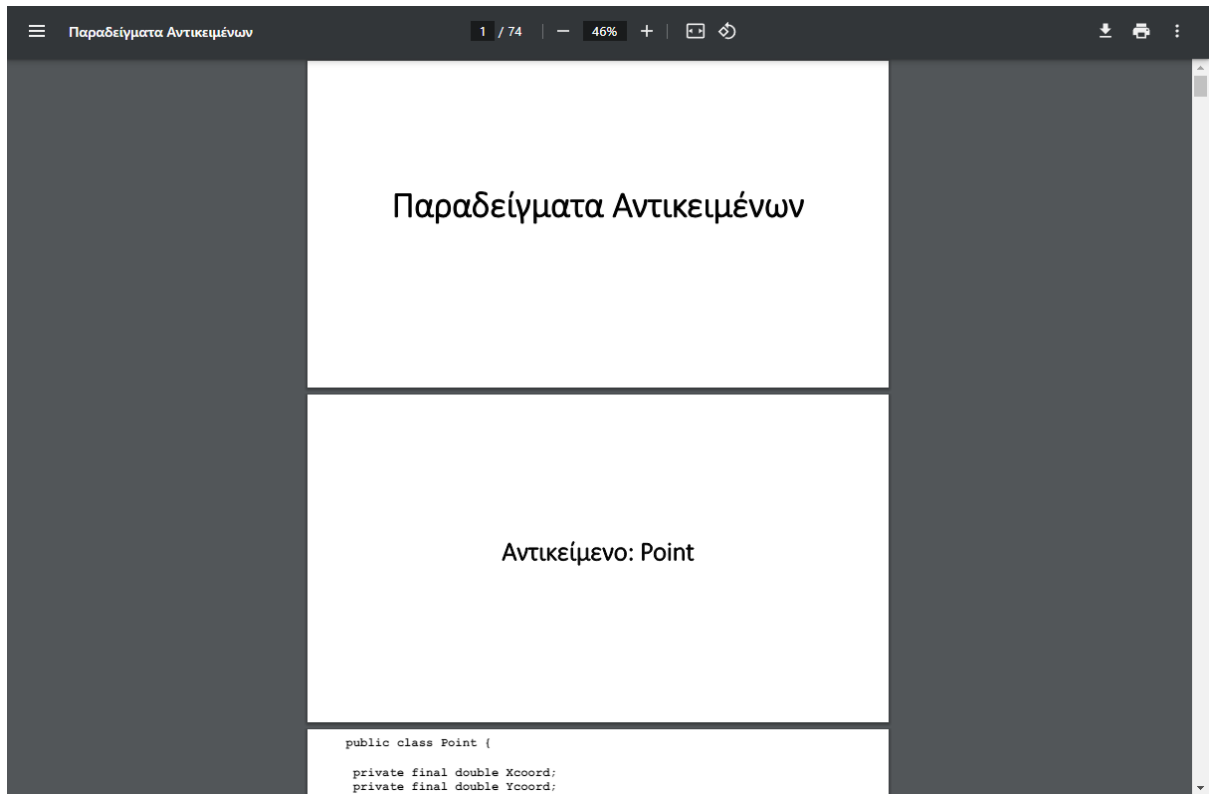
Σχήμα 4.4.7: Οδηγός Χρήσης VPN



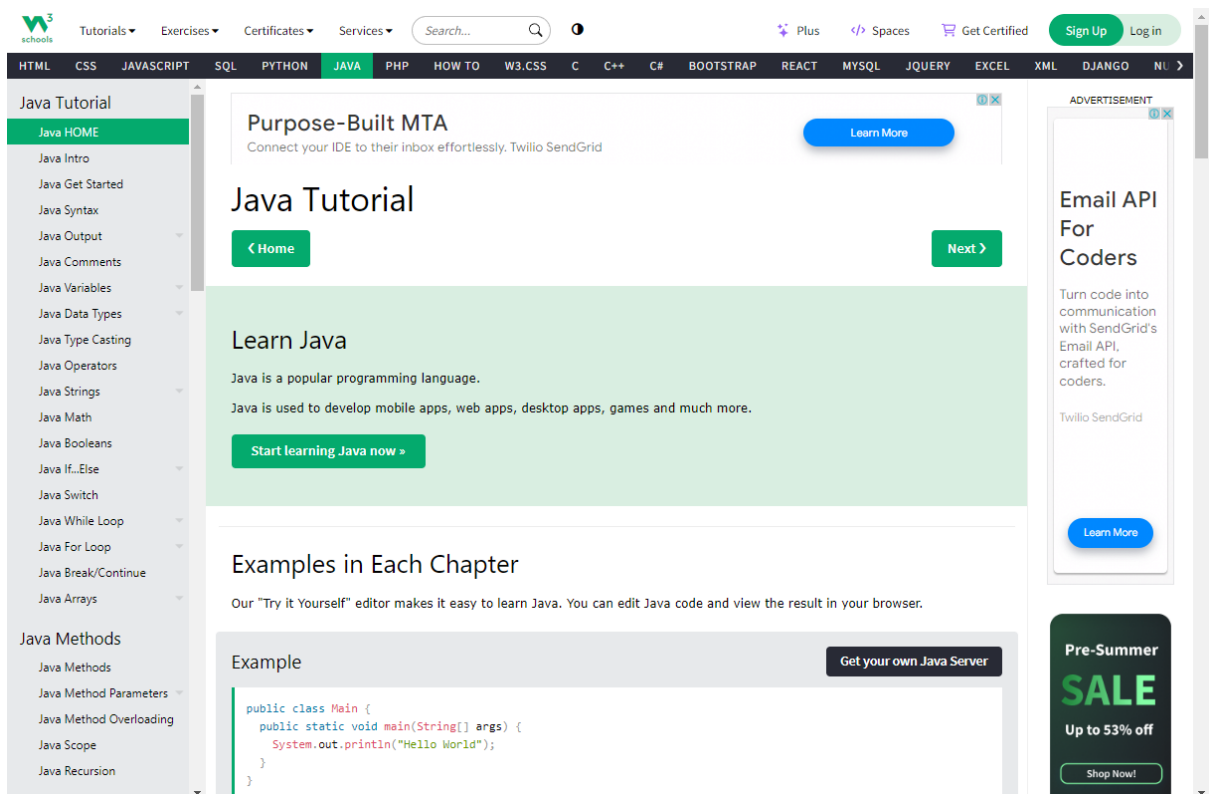
Σχήμα 4.4.8: Γενικές Οδηγίες Πληροφοριακών Συστημάτων



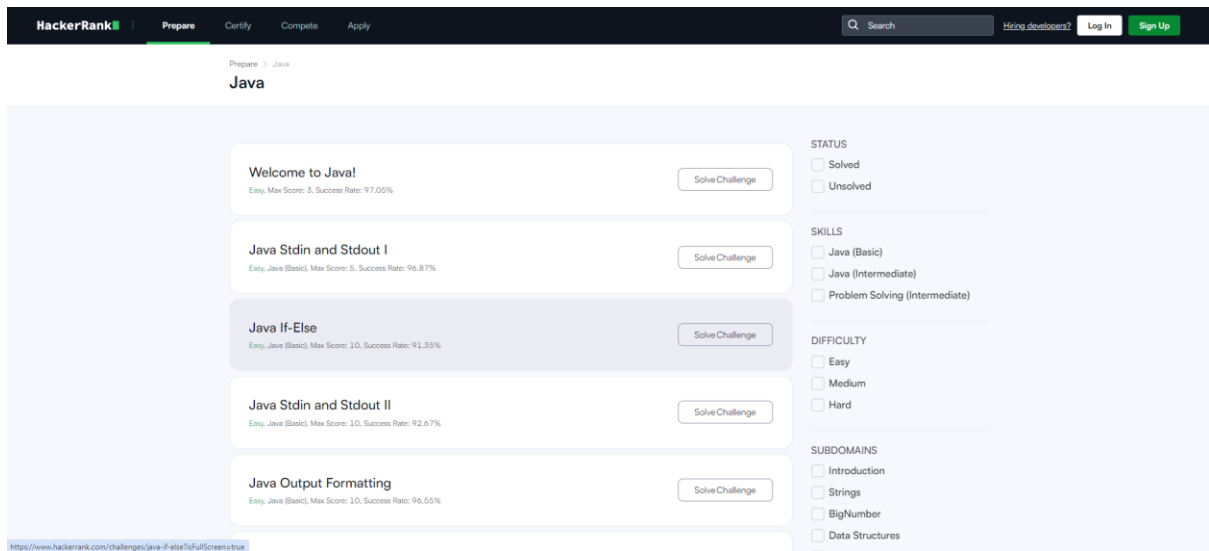
Σχήμα 4.4.9: Επιπρόσθετες Ασκήσεις Εξάσκησης



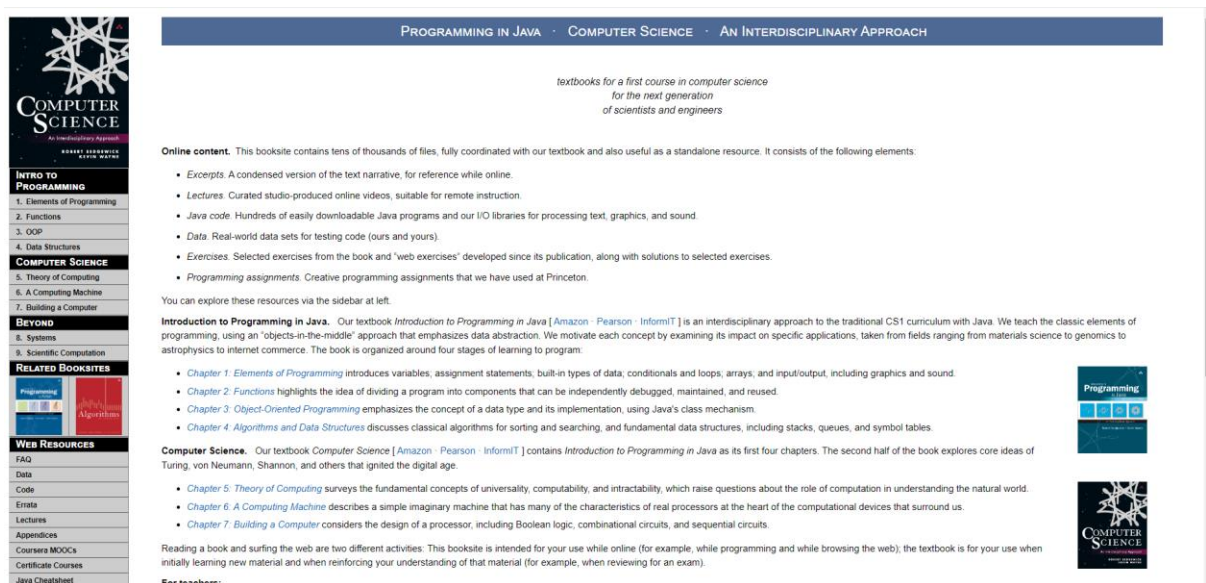
Σχήμα 4.4.10: Παραδείγματα Αντικειμένων



Σχήμα 4.4.11: Ιστοσελίδα W3Schools



Σχήμα 4.4.12: Ιστοσελίδα HackerRank



Σχήμα 4.4.13: Ιστοσελίδα του βιβλίου

Κεφάλαιο 5

Συμπεράσματα

5.1 Γενικά Συμπεράσματα που Εξάγονται από την Υλοποίηση	26
5.2 Δυσκολίες	26
5.3 Επίλογος	26

5.1 Γενικά Συμπεράσματα που Εξάγονται από την Υλοποίηση

Η υλοποίηση του συστήματος ήταν σχετικά εύκολη και απλή χάρη στην ευκολία χρήσης της υπηρεσίας Google Forms και της ευκολίας που παρέχουν οι γλώσσες HTML και CSS. Η ιστοσελίδα, αν και απλή οπτικά περιέχει χρήσιμο και εξειδικευμένο περιεχόμενο για του φοιτητές του μαθήματος ΕΠΛ131 – Αρχές Προγραμματισμού Ι. Η απλότητα στην χρήση σε συνδιασμό με το περιεχόμενο θα βοηθήσει πολλούς φοιτητές του τμήματος.

5.2 Δυσκολίες

Η κύρια δυσκολία που κλήθηκα να αντιμετωπίσω κατά την δημιουργία του συστήματος αυτού, δεν έχει να κάνει με την τεχνική υλοποίηση του, αλλά με το περιεχόμενο που παρέχει. Οι ερωτήσεις έχουν δημιουργηθεί με προσοχή και με γνώμονα το περιεχόμενο του κάθε κεφαλαίου και τις ανάγκες των φοιτητών, που είναι και οι τελικοί χρήστες που επωφελούνται από το σύστημα αυτό.

5.3 Επίλογος

Θεωρώ πως η υλοποίηση του συστήματος στέφθηκε με επιτυχία και πως το σύστημα μπορεί να χρησιμοποιηθεί από τους φοιτητές του μαθήματος ΕΠΛ131 – Αρχές Προγραμματισμού Ι οι οποίοι παρακολουθούν ή ακόμα και από φοιτητές που το έχουν παρακολουθήσει και θέλουν να αναπληρώσουν κάποια κενά και να ενισχύσουν τις γνώσεις τους.

Βιβλιογραφία

- [1] E. Keravnou-Papailiou, “Figuring and Drawing.”, Department of Computer Science, University of Cyprus, Cyprus

Παράρτημα Α

Το ερωτηματολόγιο το οποίο δόθηκε στους φοιτητές

Ερωτηματολόγιο Φοιτητών Τμήματος Πληροφορικής Πανεπιστημίου Κύπρου

Ένα σύντομο ερωτηματολόγιο για διευκρίνιση δυσκολιών φοιτητών του Τμήματος Πληροφορικής

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

Ποιό είναι το έτος σπουδών σας;

- ☐ 1ο
- ☐ 2ο
- ☐ 3ο
- ☐ 4ο
- ☐ 5ο ή 6ο

Πριν την έναρξη της φοίτησής σας είχατε προηγούμενη εμπειρία με τον προγραμματισμό;

- ☐ Ναι
- ☐ Όχι

Αντιμετωπίζετε/αντιμετωπίζετε δυσκολίες στο μάθημα ΕΠΛ131 - Αρχές Προγραμματισμού;

- ☐ Ναι
- ☐ Όχι

Τι σας δυσκόλεψε/δυσκολεύει στο μάθημα ΕΠΛ131; (Μπορείτε να επιλέξετε περισσότερες από μία επιλογές)

- ☐ Τύποι Δεδομένων
- ☐ Επιλογή / Επανάληψη
- ☐ Μονοδιάστατοι Πίνακες
- ☐ Πολυδιάστατοι Πίνακες
- ☐ Συμβολοσειρές
- ☐ Είσοδος / Έξοδος
- ☐ Συναρτήσεις
- ☐ Αναδρομή
- ☐ Αντικειμενοστρέφεια
- ☐ Σύνταξη της JAVA

Θα χρησιμοποιούσατε κάποιο διαδραστικό online σύστημα εκπαίδευσης για την περαιτέρω κατανόηση των πιο πάνω εννοιών;

- ☐ Ναι
- ☐ Όχι

Submit

Clear form

Παράρτημα Β

Οι ασκήσεις που περιέχει το σύστημα υποστήριξης φοιτητών του μαθήματος ΕΠΑ131 – Αρχές Προγραμματισμού Ι ανά κεφάλαιο.

Κεφάλαιο 1: Βασικοί (Ενσωματωμένοι) Τύποι Δεδομένων

1. Βασικοί (Ενσωματωμένοι) Τύποι Δεδομένων

Εισαγωγή στους απλούς Βασικούς Τύπους Δεδομένων

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

Οι βασικοί τύποι δεδομένων μπορούν να διασπαστούν σε απλούστερα στοιχεία. 1 βαθμός

- ☐ Ορθό
- ☐ Λάθος

Στην γλώσσα προγραμματισμού Java, υπάρχει η δυνατότητα δημιουργίας νέων τύπων δεδομένων από τον προγραμματιστή. 1 βαθμός

- ☐ Ορθό
- ☐ Λάθος

Το αποτέλεσμα της πράξης $5/3$ μεταξύ δύο ακεραίων (integers) στην Java είναι: 1 βαθμός

- ☐ 1.66666667
- ☐ 1
- ☐ Δεν ορίζεται

Ποιές από τις παρακάτω τιμές είναι ορθές αναθέσεις για τον τύπο κινητής υποδιαστολής μονής ακρίβειας (float); 5 βαθμοί

- ☐ 3.14159
- ☐ 0.04
- ☐ 12.5e.3
- ☐ 34,500.99
- ☐ 12e+5
- ☐ 17e-0.3
- ☐ 15.0e-04
- ☐ 5

Ποιός τύπος δεδομένων κινητής υποδιαστολής έχει μεγαλύτερη ακρίβεια με μέγεθος 8 byte; 1 βαθμός

- ☐ float
- ☐ double

Το αποτέλεσμα της πράξης 5/3 όταν οι αριθμοί είναι τύπου double στην Java είναι: 2 βαθμοί

- ☐ 1
- ☐ 1.6666667
- ☐ 1.0
- ☐ 1.6666666666666667

Εάν $a = \text{true}$ και $b = \text{false}$ τότε: 1 βαθμός

- ☐ $a \& b == \text{false}$, $a || b == \text{false}$
- ☐ $a \& b == \text{true}$, $a || b == \text{true}$
- ☐ $a \& b == \text{true}$, $a || b == \text{false}$
- ☐ $a \& b == \text{false}$, $a || b == \text{true}$

Ποιοί τελεστές έχουν την υψηλότερη προτεραιότητα

1 βαθμός

- ☐ Μοναδιαίοι (+,-)
- ☐ *, /, %
- ☐ Δυαδικοί +,-

Μπορούμε να χρησιμοποιούμε τον τύπο δεδομένων float για σύγκριση ισότητας (πχ. `a == 5.0`), καθώς οι σύγχρονοι υπολογιστές δεν κάνουν λάθη στην αναπαράσταση των αριθμών αυτών και είναι ακριβείς.

1 βαθμός

- ☐ Ορθό
- ☐ Λάθος

Υποβολή

Σελίδα 1 από 1

Εκκαθάριση φόρμας

Κεφάλαιο 2: Αλγοριθμικές Δομές Επιλογής και Επανάληψης

2. Αλγοριθμικές Δομές Επιλογής και Επανάληψης

Βασικές Έννοιες Δομών Επιλογής και Επανάληψης

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

Θεωρείστε τον παρακάτω κώδικα:

1 βαθμός

```
1. if(expression){  
2.   action_1;  
3.   action_2;  
4. }
```

οι γραμμές 2 και 3 εκτελούνται εάν:

- ☐ Η συνθήκη expression είναι αληθής
- ☐ Η συνθήκη expression είναι ψευδής

Ποία η προτεραιότητα των παρακάτω εκφράσεων;

5 βαθμοί

- 1) &&
- 2) κλήσεις μεθόδων
- 3) * / %
- 4) + -
- 5) ! + - & * (μοναδιαίοι τελεστές)
- 6) =
- 7) ||
- 8) == !=
- 9) < <= >= >

Δώστε την απάντησή σας με την μορφή
1,2,3,4,5,6,7,8,9 με φθίνουσα τιμή προτεραιότητας

Η απάντησή σας

Πώς ελέγχετε αν δύο μεταβλητές, a και b, είναι ίσες ως προς την τιμή και τον τύπο τους;

1 βαθμός

- ☐ if (a == b)
- ☐ if (a.equals(b))
- ☐ if (a == b && a.getClass() == b.getClass())
- ☐ if (a is b)

Ποια από τις παρακάτω είναι μια έγκυρη μέθοδος συνδυασμού πολλαπλών συνθηκών σε ένα if statement;

1 βαθμός

- ☐ if (condition1 && condition2)
- ☐ if (condition1 || condition2)
- ☐ if (condition1 & condition2)
- ☐ if (condition1 or condition2)

Τι παρέχει το switch statement στην Java που μια σειρά if-else δεν παρέχει;

1 βαθμός

- ☐ Απλότητα
- ☐ Ευελιξία
- ☐ Ευανάγνωστος κώδικας
- ☐ Αποτελεσματικός τρόπος για την χειρισμό πολλαπλών συνθηκών

Στην Java, ποια είναι η σκοπιμότητα του τρικυκλικού τελεστή "a?b:c" ;

2 βαθμοί

- ☐ Χρησιμοποιείται για δυαδικές λογικές πράξεις μεταξύ των a και b όπου το αποτέλεσμα αποθηκεύεται στην μεταβλητή c
- ☐ Παρέχει μια συντομευμένη μορφή για ένα if-else statement της μορφής if(a) then b else c
- ☐ Χρησιμοποιείται για μαθηματικούς υπολογισμούς μεταξύ των a και b όπου το αποτέλεσμα αποθηκεύεται στην μεταβλητή c
- ☐ Ορίζει μια δομή επανάληψης με συνθήκη a η b όπου εκτελείται ο κώδικας που εμπεριέχεται στο c

Τι συμβαίνει αν η συνθήκη σε ένα if statement δεν ικανοποιείται και δεν υπάρχει μπλοκ else;

1 βαθμός

- ☐ Το πρόγραμμα τερματίζεται
- ☐ Προκαλείται μια εξαίρεση κατά την εκτέλεση
- ☐ Ο κώδικας μέσα στο if block παραλείπεται και το πρόγραμμα συνεχίζεται με την επόμενη δήλωση
- ☐ Ο μεταγλωττιστής εκτυπώνει ένα σφάλμα που υποδεικνύει έλλειψη else block

Πώς ελέγχετε αν ένα δοθέν αριθμός (number) είναι ζυγός στην Java;

1 βαθμός

- ☐ if (number.mod(2) == 0)
- ☐ if (number % 2 == 0)
- ☐ if (number.even())
- ☐ if (isEven(number))

Πώς ελέγχετε αν ένα δοθέν αριθμός είναι περιττός στην Java;

1 βαθμός

- ☐ if (!(number.mod(2) == 0))
- ☐ if (!(number % 2 == 0))
- ☐ if (number.odd())
- ☐ if (isOdd(number))

Στην Java, τι κάνει η δήλωση break μέσα σε ένα if block;

1 βαθμός

- ☐ Βγαίνει από το πρόγραμμα
- ☐ Μεταβαίνει στην επόμενη επανάληψη του βρόγχου
- ☐ Μεταβαίνει στο τέλος της μεθόδου
- ☐ Τερματίζει την εσωτερική ενθυλάκωση if

Ποια είναι η σκοπιμότητα της δήλωσης continue σε έναν βρόγχο μέσα σε μια if δήλωση;

1 βαθμός

- ☐ Παραλείπει την τρέχουσα επανάληψη και πηγαίνει στην επόμενη επανάληψη του βρόγχου
- ☐ Εξέρχεται από τον βρόγχο άμεσα
- ☐ Συνεχίζει την εκτέλεση από την αρχή του βρόγχου
- ☐ Παραλείπει τον υπόλοιπο κώδικα μέσα στον βρόγχο και πηγαίνει στην επόμενη επανάληψη

Στην Java, τι σημαίνει η λέξη-κλειδί default σε ένα switch statement;

1 βαθμός

- ☐ Καθορίζει την προεπιλεγμένη περίπτωση σε μια if-else δήλωση
- ☐ Ορίζει μια προεπιλεγμένη τιμή για μια μεταβλητή
- ☐ Παρέχει τον κώδικα που θα εκτελεστεί όταν καμία περίπτωση δεν ταιριάζει με την έκφραση του switch
- ☐ Χρησιμοποιείται για να δηλώσει έναν προεπιλεγμένο κατασκευαστή

Τι ελέγχει ο τελεστής == στην Java;

1 βαθμός

- ☐ Ισότητα αναφοράς αντικειμένων
- ☐ Ισότητα τιμής πρωτογενούς τύπου δεδομένων
- ☐ Ισότητα περιεχομένου αντικειμένων
- ☐ Ισότητα τοποθεσίας μνήμης αναφοράς

Τι θα εμφανίσει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
int x = 7;
if (x % 2 == 0) {
    System.out.println("x is even");
} else {
    System.out.println("x is odd");
}
```

- ☐ "x is even"
- ☐ "x is odd"
- ☐ "x is not a number"
- ☐ Compilation error

Τι θα εμφανίσει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
int num = 0;
if (num > 0) {
    System.out.println("Positive number");
} else if (num < 0) {
    System.out.println("Negative number");
} else {
    System.out.println("Zero");
}
```

- ☐ "Positive number"
- ☐ "Negative number"
- ☐ "Zero"
- ☐ Compilation error

Τι θα εμφανίσει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
int a = 5;
int b = 5;
if (a == b) {
    System.out.println("a and b are equal");
} else if (a = b) {
    System.out.println("a is assigned the value of b");
} else {
    System.out.println("This won't be executed");
}
```

- ☐ "a and b are equal"
- ☐ "a is assigned the value of b"
- ☐ Compilation error
- ☐ "This won't be executed"

Τι θα εμφανίσει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
int marks = 75;
if (marks >= 90) {
    System.out.println("Excellent");
} else if (marks >= 75) {
    System.out.println("Very Good");
} else if (marks >= 50) {
    System.out.println("Good");
} else {
    System.out.println("Needs Improvement");
}
```

- ☐ "Very Good"
- ☐ "Excellent"
- ☐ "Good"
- ☐ "Needs Improvement"

Τι θα εμφανίσει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
int num = -8;
if (num < 0) {
    num = -num;
}
System.out.println(num);
```

- ☐ -8
- ☐ 8
- ☐ 0
- ☐ Compilation error

Τι θα εμφανίσει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
int num = 100;
String res = (num > 0) ? "Positive" : (num < 0) ? "Negative" : "Zero";
System.out.println(res);
```

- ☐ "Positive"
- ☐ "Negative"
- ☐ "Zero"
- ☐ Compilation error

Τι θα εμφανίσει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
int value = 9;
String message = "";
if (value > 10) {
    message = "Greater than 10";
} else if (value < 5) {
    message = "Less than 5";
}
System.out.println(message);
```

- ☐ "Greater than 10"
- ☐ "" (κενή συμβολοσειρά)
- ☐ "Less than 5"
- ☐ Compilation error

Ποιο τμήμα κώδικα λείπει για να ελέγχεται σωστά αν η μεταβλητή value ανήκει στο εύρος [10, 20);

1 βαθμός

```
boolean isValid = _____;
```

- ☐ value >= 10 && value < 20
- ☐ if (value.between(10, 20))
- ☐ isValid = value >= 10 && value < 20;
- ☐ isValid = (value >= 10) ? (value < 20) : false;

Ποια είναι η σωστή σύνταξη για ένα if statement που ελέγχει αν η συμβολοσειρά text δεν είναι κενή;

1 βαθμός

- ☐ if (!text.empty()) {
- ☐ if (!text.isEmpty()) {
- ☐ if (text.notNull()) {
- ☐ if (text == null) {

Τι τιμή επιστρέφει η έκφραση `int x = 5; int y = ++x;` στο τέλος της;

1 βαθμός

- ☐ x θα είναι 5, και y θα είναι 6.
- ☐ x θα είναι 6, και y θα είναι 6
- ☐ x θα είναι 6, και y θα είναι 5
- ☐ x θα είναι 5, και y θα είναι 5

Τι τιμή επιστρέφει η έκφραση `int x = 5; int y = x++;` στο τέλος της;

1 βαθμός

- ☐ x θα είναι 5, και y θα είναι 6.
- ☐ x θα είναι 6, και y θα είναι 6
- ☐ x θα είναι 6, και y θα είναι 5
- ☐ x θα είναι 5, και y θα είναι 5

Τι συμβαίνει αν χρησιμοποιηθεί ο τελεστής `++` σε έναν αριθμό που βρίσκεται σε μια δήλωση `System.out.println(++num);` ;

1 βαθμός

- ☐ Ο αριθμός αυξάνεται και η νέα τιμή εκτυπώνεται.
- ☐ Η τρέχουσα τιμή του αριθμού εκτυπώνεται, και μετά ο αριθμός αυξάνεται.
- ☐ Ο αριθμός παραμένει ίδιος, και μια εξαίρεση εκτυπώνεται.
- ☐ Η τρέχουσα τιμή του αριθμού εκτυπώνεται, αλλά ο αριθμός δεν αλλάζει.

Ποια είναι η έξοδος του παρακάτω κώδικα;

1 βαθμός

```
int num = 2;
while (num < 10) {
    System.out.print(num + " ");
    num *= 2;
}
```

- ☐ Εκτυπώνει "2 4 6 8"
- ☐ Εκτυπώνει "2 4 8 16"
- ☐ Εκτυπώνει "2 4 8 10"
- ☐ Εκτελεί έναν ατέλειωτο βρόγχο

Τι θα εκτυπώσει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
int c = 3;
while (c <= 12) {
    if (c % 2 == 0) {
        System.out.print(c + " ");
    }
    c++;
}
```

- ☐ Εκτυπώνει "4 6 8 10 12"
- ☐ Εκτυπώνει "3 5 7 9 11"
- ☐ Εκτυπώνει "3 4 5 6 7"
- ☐ Εκτυπώνει "4 8 12"

Ποια είναι η έξοδος του παρακάτω κώδικα;

1 βαθμός

```
for (int n = 2; n < 10; n *= 2) {
    System.out.print(n + " ");
}
```

- ☐ Εκτυπώνει "2 4 8"
- ☐ Εκτυπώνει "2 4 8 16"
- ☐ Εκτυπώνει "4 8"
- ☐ Εκτυπώνει "2 4 6 8"

Τι κάνει το παρακάτω κομμάτι κώδικα;

1 βαθμός

```
double c = 3.5;
c += 2.5;
```

- ☐ Προσθέτει την τιμή 2.5 στη μεταβλητή c
- ☐ Αφαιρεί την τιμή 2.5 από τη μεταβλητή c
- ☐ Αυξάνει την τιμή της μεταβλητής c κατά 2.5 φορές
- ☐ Δεν έχει έγκυρη σύνταξη

Ποια είναι η βασική διαφορά μεταξύ ενός "do-while" βρόγχου και ενός "while" βρόγχου;

1 βαθμός

- ☐ Ο "do-while" βρόγχος ελέγχει τη συνθήκη στην αρχή του βρόγχου, ενώ ο "while" βρόγχος στο τέλος.
- ☐ Ο "do-while" βρόγχος εκτελείται τουλάχιστον μια φορά, ανεξαρτήτως της συνθήκης.
- ☐ Ο "do-while" βρόγχος δεν μπορεί να χρησιμοποιηθεί για επανάληψη ενός κομματιού κώδικα.
- ☐ Δεν υπάρχει διαφορά μεταξύ τους.

Τι θα κάνει ο παρακάτω κώδικας;

1 βαθμός

```
int x = 10;  
do {  
    System.out.println(x);  
} while (x < 5);
```

- ☐ Εκτυπώνει "10" μια φορά
- ☐ Εκτυπώνει "10" πέντε φορές
- ☐ Δεν εκτυπώνει τίποτα
- ☐ Προκαλείται σφάλμα κατά την εκτέλεση

Ποιές από τις παρακάτω εκφράσεις ισούνται με την έκφραση: while(a){b};

3 βαθμοί

- ☐ if(a) do {b} while (a);
- ☐ do{b} while(a);
- ☐ for(;;){b}
- ☐ for(a){b}
- ☐ while(!a){b}
- ☐ for(!a){b}

Ποιές από τις παρακάτω εκφράσεις ισούνται με την έκφραση: do {b} while(a);

3 βαθμοί

- ☐ while(a) do {b}
- ☐ if(1==1) {do {b} while (a)}
- ☐ {b} while(a){b}
- ☐ {b} for(a) {b}
- ☐ {b} for(a){b}

Υποβολή

Σελίδα 1 από 1

Εκκαθάριση φόρμας

3. Πίνακες Μιας Διάστασης

Εισαγωγή στους Μονοδιάστατους Πίνακες

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Τι είναι ένας μονοδιάστατος πίνακας στην Java; *

1 βαθμός

- ☐ Μια δομή δεδομένων που αποθηκεύει πολλές τιμές διαφορετικών τύπων.
- ☐ Μια συλλογή μεταβλητών με τον ίδιο τύπο δεδομένων.
- ☐ Ένας τύπος βρόγχου στην Java.
- ☐ Ένας τρόπος εκτύπωσης τιμών σε μια μόνο γραμμή.

Πώς δηλώνεται ένας μονοδιάστατος πίνακας ακεραίων στην Java; *

1 βαθμός

- ☐ `int array = new int[];`
- ☐ `int[] array = new int();`
- ☐ `int[] array = new int[];`
- ☐ `int[] array = new int[5];`

Πώς μπορείτε να έχετε πρόσβαση στα στοιχεία ενός μονοδιάστατου πίνακα στην Java;

* 1 βαθμός

- ☐ Μόνο με χρήση βρόγχου.
- ☐ Δηλώνοντας το δείκτη περικλεισμένο σε αγκύλες.
- ☐ Χρησιμοποιώντας τη μέθοδο `get()`.
- ☐ Τα στοιχεία δεν μπορούν να προσπελαστούν ατομικά.

Πώς αρχικοποιείτε τα στοιχεία ενός μονοδιάστατου πίνακα στην Java; *

1 βαθμός

- ☐ Χρησιμοποιώντας την εντολή `initialize`.
- ☐ Με τον τελεστή `=` και στη συνέχεια τις τιμές των στοιχείων.
- ☐ Μόνο μέσω `constructor`.
- ☐ Με την εντολή `start`.

Ποιος είναι ο τρόπος ανάκτησης του μεγέθους (size) ενός μονοδιάστατου πίνακα στην Java; * 1 βαθμός

- ☐ Μέσω της μεθόδου length.
- ☐ Χρησιμοποιώντας τη μέθοδο size().
- ☐ NullPointerException
- ☐ Με τη μέθοδο getSize().

Πώς μπορείτε να προσθέσετε τιμή σε ένα συγκεκριμένο δείκτη (index) σε έναν μονοδιάστατο πίνακα; * 1 βαθμός

- ☐ Μόνο με τον τελεστή +=
- ☐ Με τη μέθοδο addValue()
- ☐ Χρησιμοποιώντας τον τελεστή + και στη συνέχεια την τιμή.
- ☐ Με τον τελεστή ++

Τι είδους τιμές μπορεί να περιέχει ένας μονοδιάστατος πίνακας στη Java; * 1 βαθμός

- ☐ Μόνο ακέραιους αριθμούς
- ☐ Μόνο δεκαδικούς αριθμούς
- ☐ Μόνο χαρακτήρες
- ☐ Οποιοδήποτε τύπο δεδομένων

Πώς μπορούμε να αλλάξουμε την τιμή ενός συγκεκριμένου στοιχείου σε έναν μονοδιάστατο πίνακα; * 1 βαθμός

- ☐ array[index] = newValue;
- ☐ changeValue(index, newValue)
- ☐ setValue(index, newValue)
- ☐ modify(index, newValue)

Ποια είναι η συνολική πολυπλοκότητα (complexity) της πρόσβασης σε ένα στοιχείο ενός μονοδιάστατου πίνακα; * 1 βαθμός

- ☐ $O(n)$
- ☐ $O(\log n)$
- ☐ $O(n \log n)$
- ☐ $O(1)$

Ποια μέθοδος χρησιμοποιείται για τον έλεγχο εάν ένας μονοδιάστατος πίνακας είναι άδειος;

* 1 βαθμός

- ☐ isEmpty()
- ☐ length == 0
- ☐ checkEmpty()
- ☐ arrayEmpty()

Δεδομένο το παρακάτω κομμάτι κώδικα, ποιος είναι ο τρόπος πρόσβασης στο τρίτο στοιχείο του πίνακα;

* 1 βαθμός

```
int[] array = {5, 8, 12, 6, 10};
```

- ☐ array[2]
- ☐ array(3)
- ☐ array[3]
- ☐ array.element(2)

Δεδομένο το παρακάτω κομμάτι κώδικα, ποια είναι η τιμή του μεταβλητής length μετά την εκτέλεση του κώδικα;

* 1 βαθμός

```
int[] array = {2, 4, 6, 8, 10};  
int length = array.length;
```

- ☐ 5
- ☐ 4
- ☐ 10
- ☐ 6

Σε ποιο σημείο του παρακάτω κώδικα γίνεται η αρχικοποίηση του πίνακα με τιμές 1 έως 5;

* 1 βαθμός

```
int[] numbers;  
for (int i = 1; i <= 5; i++) {  
    // Εδώ  
}
```

- ☐ numbers[i] = i;
- ☐ numbers[i-1] = i;
- ☐ numbers[i+1] = i;
- ☐ numbers[i] = i+1;

Ποιο είναι το αποτέλεσμα του παρακάτω κώδικα; *

1 βαθμός

```
int[] arr1 = {1, 2, 3};  
int[] arr2 = arr1;  
arr2[0] = 10;  
System.out.println(arr1[0]);
```

- ☐ 1
- ☐ 10
- ☐ 2
- ☐ 3

Ποια είναι η τιμή της μεταβλητής sum μετά την εκτέλεση του παρακάτω κώδικα;

* 1 βαθμός

```
int[] numbers = {3, 8, 2, 6, 1};  
int sum = 0;  
for (int num : numbers) {  
    sum += num;  
}
```

- ☐ 20
- ☐ 12
- ☐ 8
- ☐ 30

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
int[] values = new int[]{1, 2, 3, 4};  
for (int i = 0; i < values.length; i++) {  
    System.out.println(values[i]);  
}  
System.out.println(values[4]);
```

- ☐ Λάθος σύνταξης στη δήλωση του πίνακα.
- ☐ Προσπάθεια πρόσβασης σε έναν έγκυρο δείκτη του πίνακα.
- ☐ Προσπάθεια πρόσβασης σε έναν μη υπάρχοντα δείκτη του πίνακα.
- ☐ Λάθος τύπος δεδομένων για τον πίνακα.

Συμπληρώστε το κενό ώστε να ανατρίψετε τη σειρά των στοιχείων στον πίνακα numbers: * 1 βαθμός

```
int[] numbers = {4, 7, 2, 9};
for (int i = 0; i < numbers.length / 2; i++) {
    int temp = numbers[i];
    numbers[i] = numbers[/* Συμπληρώστε εδώ */];
    numbers[/* Συμπληρώστε εδώ */] = temp;
}
```

- ☐ i+1,i-1
- ☐ i+1,i
- ☐ i,i+1
- ☐ i-1,i

Συμπληρώστε το κενό ώστε να υπολογιστεί το άθροισμα των άρτιων αριθμών στον πίνακα data: * 1 βαθμός

```
int[] data = {3, 6, 8, 2, 9};
int sum = 0;
for (int num : data) {
    if (/* Συμπληρώστε εδώ */) {
        sum += num;
    }
}
```

- ☐ num % 2 != 1
- ☐ num % 2 != 0
- ☐ num % 2 == 1
- ☐ num % 2 == 0

Συμπληρώστε το κενό ώστε να ελέγξετε αν ο πίνακας array είναι αύξουσα ταξινομημένος: * 1 βαθμός

```
int[] array = {4, 7, 12, 15, 20};
boolean isSorted = true;
for (int i = 1; i < array.length; i++) {
    if (/* Συμπληρώστε εδώ */) {
        isSorted = false;
        break;
    }
}
```

- ☐ array[i] >= array[i - 1]
- ☐ array[i] < array[i - 1]
- ☐ array[i] <= array[i - 1]
- ☐ array[i] > array[i - 1]

Ποιο είναι το σωστό κομμάτι κώδικα για τη δημιουργία ενός τυχαίου αριθμού στο εύρος $[a, b]$ στη γλώσσα προγραμματισμού Java;

* 1 βαθμός

Σημείωση: Η `Math.random()` επιστρέφει τιμές $[0.0, 1.0)$

- ☐ `int randomNum = (int) (Math.random() * b + a);`
- ☐ `int randomNum = (int) (Math.random() * (b - a + 1) + a);`
- ☐ `int randomNum = (int) (Math.random() * (b - a) + a);`
- ☐ `int randomNum = (int) (Math.random() * (b + 1) + a);`

Ποια από τις παρακάτω δηλώσεις ισχύει για τους πίνακες στη Java? *

1 βαθμός

- ☐ Οι πίνακες μπορούν να αποθηκεύουν στοιχεία διαφορετικών τύπων δεδομένων.
- ☐ Οι πίνακες στη Java είναι πάντα δυναμικά αλλάζουμε μέγεθος.
- ☐ Το μήκος ενός πίνακα στη Java είναι σταθερό μόλις δημιουργηθεί.
- ☐ Οι πίνακες στη Java δεν μπορούν να ταξινομηθούν.

Ποια είναι η προεπιλεγμένη τιμή για ένα στοιχείο σε έναν πίνακα ακεραίων στη Java?

* 1 βαθμός

- ☐ a) 0
- ☐ b) -1
- ☐ c) null
- ☐ d) false

Πώς έχετε πρόσβαση στο τρίτο στοιχείο ενός πίνακα με όνομα "arr" στη Java? *

1 βαθμός

- ☐ `arr[2]`
- ☐ `arr[3]`
- ☐ `arr(3)`
- ☐ `arr.get(2)`

Τι συμβαίνει αν προσπαθήσετε να έχετε πρόσβαση σε ένα δείκτη εκτός των ορίων ενός πίνακα στη Java? *

1 βαθμός

- ☐ Επιστρέφει null.
- ☐ `NullPointerException`.
- ☐ `ArrayOutOfBoundsException`.
- ☐ Επιστρέφει -1.

Τι κάνει το παρακάτω τμήμα κώδικα; *

1 βαθμός

```
int[] array = {1, 2, 3, 4, 5};
int sum = 0;
for (int i = 0; i < array.length; i++) {
    sum += array[i] * i;
}
System.out.println(sum);
```

- ☐ Εκτυπώνει το άθροισμα όλων των στοιχείων του πίνακα.
- ☐ Εκτυπώνει το άθροισμα των δεικτών όλων των στοιχείων του πίνακα.
- ☐ Εκτυπώνει το άθροισμα των τετραγώνων όλων των στοιχείων του πίνακα.
- ☐ Εκτυπώνει το άθροισμα των γινομένων κάθε στοιχείου με το δείκτη του.

Ποιος είναι ο σωστός τρόπος αρχικοποίησης ενός διδιάστατου πίνακα σε Java; * 1 βαθμός

- ☐ `int[] array = new int[3][3];`
- ☐ `int[] array = new int[3, 3];`
- ☐ `int[] array = { {1, 2}, {3, 4}, {5, 6} };`
- ☐ `int[] array = new int[]{{1, 2}, {3, 4}, {5, 6}};`

Ποια είναι η έξοδος του παρακάτω κώδικα; *

1 βαθμός

```
int[] array = {1, 2, 3, 4, 5};
int sum = 0;
for (int i = 0; i < array.length; i++) {
    sum += array[i];
    if (i % 2 == 0) {
        i++;
    }
}
System.out.println(sum);
```

- ☐ 9
- ☐ 15
- ☐ 0
- ☐ 12

Ποιο είναι το μήνυμα λάθους που θα εμφανιστεί αν εκτελέσετε τον παρακάτω κώδικα; * 1 βαθμός

```
int[] array = new int[-5];
```

☐ `ArrayIndexOutOfBoundsException`
☐ `NegativeArraySizeException`
☐ `NullPointerException`
☐ Compile-time error

Ποιο είναι το σφάλμα σε αυτόν τον κώδικα; * 1 βαθμός

```
int[] array = new int[5];
System.out.println(array[5]);
```

☐ Δεν υπάρχει σφάλμα
☐ `NullPointerException`
☐ `ArrayIndexOutOfBoundsException`
☐ Σφάλμα κατά τη μεταγλώττιση

Υποβολή

Εκκαθάριση φόρμας

Κεφάλαιο 4: Σειμβολοσειρές – Αντικείμενο String

4. Σειμβολοσειρές - Αντικείμενο String

Εισαγωγή στις Σειμβολοσειρές και το αντικείμενο String

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Τι είναι το String στην Java; * 1 βαθμός

☐ Πρωτογενής τύπος δεδομένων
☐ Αντικείμενο
☐ Λειτουργία
☐ Σταθερά

Ποιο από τα παρακάτω είναι αληθές για τα String στην Java; * 1 βαθμός

☐ Τα String είναι μεταβλητά
☐ Τα String είναι αμετάβλητα
☐ Τα String δεν μπορούν να περιέχουν κενά
☐ Τα String είναι τύποι δεδομένων

Ποια μέθοδος συγκρίνει δύο String αντικείμενα για ισότητα περιεχομένου; * 1 βαθμός

- ☐ equals()
- ☐ compareTo()
- ☐ equalsIgnoreCase()
- ☐ contentEquals()

Τι επιστρέφει η μέθοδος length() ενός String; * 1 βαθμός

- ☐ Το μήκος του String συμπεριλαμβανομένων των κενών χαρακτήρων
- ☐ Τον αριθμό των λέξεων στο String
- ☐ Το μέγεθος του String σε bytes
- ☐ Το συνολικό μήκος χωρίς τα κενά

Ποιο από τα παρακάτω δημιουργεί ένα νέο String αντικείμενο με το περιεχόμενο "Java"; * 1 βαθμός

- ☐ String s = new "Java";
- ☐ String s = "Java";
- ☐ String s = String("Java");
- ☐ String s = Java;

Ποια από τις παρακάτω δηλώσεις δημιουργεί μια νέα συμβολοσειρά στην Java; * 1 βαθμός

- ☐ String s = "Hello";
- ☐ String s = new String("Hello");
- ☐ String s = "Hello" + " World";
- ☐ δ) Όλα τα παραπάνω

Ποιο από τα παρακάτω δημιουργεί ένα κενό String; * 1 βαθμός

- ☐ String s = "";
- ☐ String s = null;
- ☐ String s = " ";
- ☐ String s = new "";

Ποια μέθοδος χρησιμοποιείται για την μετατροπή όλων των χαρακτήρων ενός String σε πεζά γράμματα; * 1 βαθμός

- ☐ toLower()
- ☐ toLowerCase()
- ☐ lowerCase()
- ☐ toLowerString()

Πώς μπορούμε να αφαιρέσουμε τα κενά από την αρχή και το τέλος ενός String; * 1 βαθμός

- ☐ strip()
- ☐ trim()
- ☐ removeSpaces()
- ☐ cutSpaces()

Ποια μέθοδος χρησιμοποιείται για την αντικατάσταση όλων των εμφανίσεων μιας συμβολοσειράς μέσα σε ένα String; * 1 βαθμός

- ☐ change()
- ☐ replace()
- ☐ swap()
- ☐ substitute()

Ποια μέθοδος επιστρέφει έναν υποσυμβολοσειρά από ένα String; * 1 βαθμός

- ☐ subString()
- ☐ getString()
- ☐ substring()
- ☐ getSubstring()

Έστω ο κώδικας; *

1 βαθμός

```
String s1 = "Hello";  
String s2 = "World";  
s1=s2;
```

Τι κάνει ο παραπάνω κώδικας;

- ☐ Ελέγχει αν τα s1 και τα s2 ισούνται
- ☐ Σφάλμα
- ☐ Αλλάζει τον δείκτη του s2 να δείχνει στο s1
- ☐ Αλλάζει τον δείκτη του s1 να δείχνει στο s2

Ποιά η σωστή σύνταξη για τη δήλωση ενός πίνακα από `String` στην `Java`; * 1 βαθμός

- ☐ `String array[] = new String[];`
- ☐ `String[] array = new String[];`
- ☐ `String[] array = new String[10];`
- ☐ `String array = new String[10];`

Ποια είναι η προεπιλεγμένη τιμή για τα στοιχεία ενός πίνακα από `String` στην `Java` όταν ο πίνακας δηλώνεται αλλά δεν αρχικοποιείται; * 1 βαθμός

- ☐ `null`
- ☐ `"`
- ☐ `"default"`
- ☐ `0`

Πώς δημιουργούμε έναν πίνακα από `String` με αρχικές τιμές στην `Java`; * 1 βαθμός

- ☐ `String[] array = {"Hello", "World"};`
- ☐ `String array = {"Hello", "World"};`
- ☐ `String[] array = new String[2]({"Hello", "World"});`
- ☐ `String[] array = new String[]{"Hello", "World"};`

Πώς μπορούμε να αλλάξουμε την τιμή του δεύτερου στοιχείου σε έναν πίνακα από `String`; * 1 βαθμός

- ☐ `array[1] = "New Value";`
- ☐ `array[2] = "New Value";`
- ☐ `array.set(1, "New Value");`
- ☐ `array.set(2, "New Value");`

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; * 1 βαθμός
`String s = 'Hello';`

- ☐ Οι μονές εισαγωγικές πρέπει να είναι διπλές
- ☐ Το `String` δεν πρέπει να έχει μεταβλητή όνομα
- ☐ Το `String` δεν μπορεί να περιέχει κείμενο
- ☐ Το `String` πρέπει να είναι κεφαλαίο

Τι λείπει από τον παρακάτω κώδικα; *

1 βαθμός

```
String s;  
s = "Hello";  
System.out.println(s);
```

- ☐ Η δήλωση του String
- ☐ Η αρχικοποίηση του String
- ☐ Η κλήση της μεθόδου print
- ☐ Τίποτα, ο κώδικας είναι σωστός

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
String s1 = "Hello";  
String s2 = "World";  
String s3 = s1 + s2;  
System.out.println(s1 + " " + s3);
```

- ☐ Λείπει η μεταβλητή s2
- ☐ Το String δεν υποστηρίζει πράξεις πρόσθεσης
- ☐ Το s3 δεν είναι αρχικοποιημένο
- ☐ Τίποτα, ο κώδικας είναι σωστός

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
String s = "Hello";  
s = s.toUpperCase();
```

- ☐ Το String δεν έχει μέθοδο toUpperCase
- ☐ Το String πρέπει να είναι αρχικοποιημένο
- ☐ Το s δεν μπορεί να αλλάξει
- ☐ Λείπει το System.out.println

Έστω τα ακόλουθα Strings: *

1 βαθμός

```
String s1 = "Hello";  
String s2 = "World";
```

Η σύγκριση `s1 == s2` τι συγκρίνει;

- ☐ Το περιεχόμενο των s1 και s2
- ☐ Το μέγεθος των s1 και s2
- ☐ Η σύγκριση δεν είναι έγκυρη
- ☐ Τις διευθύνσεις στην μνήμη των s1 και s2

Ολοκληρώστε τον παρακάτω κώδικα για να συγκρίνετε δύο String: *

1 βαθμός

```
String s1 = "Hello";  
String s2 = "Hello";  
if (.....) {  
    System.out.println("Τα Strings είναι ίσα");  
}
```

- ☐ s1 == s2
- ☐ s1.equals(s2)
- ☐ s1.compareTo(s2)
- ☐ s1 = s2

Ολοκληρώστε τον παρακάτω κώδικα για να βρείτε το μήκος ενός String: *

1 βαθμός

```
String s = "Hello, World!";  
int length = .....
```

- ☐ s.size()
- ☐ s.length()
- ☐ s.len()
- ☐ s.getLength()

Ολοκληρώστε τον παρακάτω κώδικα για να αντικαταστήσετε χαρακτήρες σε ένα String: *

1 βαθμός

```
String s = "Hello";  
s = s. ....('l', 'p');
```

- ☐ change
- ☐ swap
- ☐ replace
- ☐ substitute

Ολοκληρώστε τον παρακάτω κώδικα για να βρείτε έναν υποσυμβολοσειρά από ένα String:

* 1 βαθμός

```
String s = "Hello, World!";  
String sub = s. ....(7, 12);
```

- ☐ subString
- ☐ getString
- ☐ substring
- ☐ getSubstring

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
String s = "Hello";  
System.out.println(s.charAt(1));
```

- ☐ H
- ☐ e
- ☐ l
- ☐ Σφάλμα

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
String s = "Java";  
System.out.println(s.substring(1, 3));
```

- ☐ Ja
- ☐ av
- ☐ va
- ☐ ava

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
String s = "Hello";  
System.out.println(s.indexOf('l'));
```

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
String s = "Hello";  
System.out.println(s.concat(" World"));
```

- ☐ Hello
- ☐ HelloWorld
- ☐ Hello World
- ☐ WorldHello

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
String s = "Hello";  
s = s.replace('l', 'p');  
System.out.println(s);
```

- ☐ Heppo
- ☐ Hpilo
- ☐ Helpo
- ☐ Heplo

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
String[] array = new String[3];  
array[3] = "Hello";
```

- ☐ Ο πίνακας δεν έχει αρχικοποιηθεί
- ☐ Η σύνταξη για την αρχικοποίηση του πίνακα είναι λανθασμένη
- ☐ Υπέρβαση των ορίων του πίνακα
- ☐ Δεν υπάρχει σφάλμα

*

1 βαθμός

- ☐ Option 1

Υποβολή

Εκκαθάριση φόρμας

Κεφάλαιο 5: Είσοδος και Έξοδος

5. Είσοδος και Έξοδος

Εισαγωγή στην Έισοδο και Έξοδο και στην βιβλιοθήκη StdDraw

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Ποια μέθοδος της κλάσης Scanner χρησιμοποιείται για να διαβάσετε μια ολόκληρη γραμμή από την είσοδο; *

1 βαθμός

- ☐ next()
- ☐ nextLine()
- ☐ readLine()
- ☐ read()

Ποιο είναι το αποτέλεσμα της κλήσης `System.out.println()`; χωρίς καμία παράμετρο; * 1 βαθμός

- ☐ Εκτυπώνει ένα κενό διάστημα
- ☐ Εκτυπώνει μια νέα γραμμή
- ☐ Εκτυπώνει null
- ☐ Παράγει λάθος

Ποια κλάση χρησιμοποιείται για να διαβάσετε δεδομένα από το χρήστη στη Java; * 1 βαθμός

- ☐ `BufferedReader`
- ☐ `Scanner`
- ☐ `InputStreamReader`
- ☐ `DataInputStream`

Ποια είναι η σωστή μέθοδος για να διαβάσετε έναν ακέραιο αριθμό από την είσοδο χρησιμοποιώντας την κλάση `Scanner`; * 1 βαθμός

`Scanner sc = new Scanner(System.in);`

- ☐ `int num = sc.readInt();`
- ☐ `int num = sc.nextInt();`
- ☐ `int num = sc.inputInt();`
- ☐ `int num = sc.getInt();`

Ποια είναι η σωστή μέθοδος για να διαβάσετε μια λέξη από την είσοδο χρησιμοποιώντας την κλάση `Scanner`; * 1 βαθμός

`Scanner sc = new Scanner(System.in);`

- ☐ `String word = sc.nextWord();`
- ☐ `String word = sc.readWord();`
- ☐ `String word = sc.getWord();`
- ☐ `String word = sc.next();`

Ποιο είναι το σωστός τρόπος για να διαβάσετε έναν αριθμό κινητής υποδιαστολής (double) από την είσοδο;

* 1 βαθμός

Scanner sc = new Scanner(System.in);

- ☐ double num = sc.nextDouble();
- ☐ double num = sc.readDouble();
- ☐ double num = sc.getDouble();
- ☐ double num = sc.scanDouble();

Ποια είναι η σωστή σειρά για να διαβάσετε μια γραμμή από την είσοδο και να την εκτυπώσετε;

* 1 βαθμός

Scanner sc = new Scanner(System.in);

- ☐ String line = sc.next(); System.out.println(line);
- ☐ String line = sc.nextLine(); System.out.println(line);
- ☐ String line = sc.read(); System.out.println(line);
- ☐ String line = sc.getLine(); System.out.println(line);

Τι κάνει η μέθοδος parseDouble() στη Java; *

1 βαθμός

- ☐ Μετατρέπει μια αλφαριθμητική τιμή σε ακέραιο αριθμό.
- ☐ Μετατρέπει μια αλφαριθμητική τιμή σε διπλό ακέραιο.
- ☐ Μετατρέπει μια αλφαριθμητική τιμή σε δεκαδικό αριθμό.
- ☐ Μετατρέπει μια αλφαριθμητική τιμή σε αριθμό κινητής υποδιαστολής.

Ποιος είναι ο σωστός τρόπος χρήσης της μεθόδου parseDouble() για να μετατρέψετε ένα αλφαριθμητικό σε διπλό αριθμό κινητής υποδιαστολής;

* 1 βαθμός

- ☐ Double.parseDouble(value);
- ☐ parseDouble(Double, value);
- ☐ Double.parse(value);
- ☐ value.parseDouble();

Τι κάνει η εντολή java Program > output.txt στη γραμμή εντολών; *

1 βαθμός

- ☐ Ανακατεύθυνει την έξοδο από το πρόγραμμα Program σε ένα αρχείο με το όνομα output.txt.
- ☐ Ανακατεύθυνει την είσοδο στο πρόγραμμα Program από το αρχείο output.txt.
- ☐ Αντιγράφει το περιεχόμενο του αρχείου output.txt στο πρόγραμμα Program.
- ☐ Εμφανίζει στην οθόνη το περιεχόμενο του αρχείου output.txt.

Τι είναι τα StdIn και StdOut στην Java; *

1 βαθμός

- ☐ Δύο αντικείμενα για την ανάγνωση από και την εγγραφή σε αρχεία.
- ☐ Δύο αντικείμενα που εκτελούν την είσοδο και έξοδο δεδομένων από και προς τη γραμμή εντολών.
- ☐ Δύο κλάσεις για τη διαχείριση της εισόδου και της εξόδου από και προς το πρόγραμμα.
- ☐ Δύο δομές δεδομένων για την αποθήκευση δεδομένων εισόδου και εξόδου σε ένα πίνακα.

Ποια μέθοδος χρησιμοποιείται για την ανάγνωση ενός ακραίου αριθμού από το StdIn στην Java; *

1 βαθμός

- ☐ readInt()
- ☐ nextInt()
- ☐ getInt()
- ☐ inputInt()

Ποια μέθοδος χρησιμοποιείται για την εγγραφή ενός αλφαριθμητικού στο StdOut στην Java; *

1 βαθμός

- ☐ writeString()
- ☐ printString()
- ☐ write()
- ☐ print()

Ποιο από τα παρακάτω είναι λάθος στον κώδικα που χρησιμοποιεί το StdIn στην Java;
`int num = StdIn.readInt();` *

1 βαθμός

- ☐ Η μέθοδος readInt() δεν υπάρχει στην κλάση StdIn.
- ☐ Το πακέτο java.util δεν έχει εισαχθεί.
- ☐ Η μέθοδος readInt() χρειάζεται παράμετρο.
- ☐ Δεν υπάρχει σφάλμα

Ποια είναι η σωστή σύνταξη για να ανακατευθύνετε την είσοδο από ένα αρχείο σε ένα πρόγραμμα στην Java; *

1 βαθμός

- ☐ `java Program < input.txt`
- ☐ `java Program > input.txt`
- ☐ `java Program >> input.txt`
- ☐ `java Program | input.txt`

Τι κάνει η εντολή `java Program > output.txt` στη γραμμή εντολών; *

1 βαθμός

- ☐ Ανακατευθύνει την έξοδο από το πρόγραμμα `Program` σε ένα αρχείο με το όνομα `output.txt`.
- ☐ Ανακατευθύνει την είσοδο στο πρόγραμμα `Program` από το αρχείο `output.txt`.
- ☐ Αντιγράφει το περιεχόμενο του αρχείου `output.txt` στο πρόγραμμα `Program`.
- ☐ Εμφανίζει στην οθόνη το περιεχόμενο του αρχείου `output.txt`.

Τι σημαίνει το προσάρτημα (append) δεδομένων σε ένα αρχείο; *

1 βαθμός

- ☐ Το να διαγράφουμε τα προηγούμενα δεδομένα και να αντικαθιστούμε με νέα.
- ☐ Το να επικολλούμε νέα δεδομένα στο τέλος του αρχείου χωρίς να διαγράφουμε τα υπάρχοντα.
- ☐ Το να αντιγράφουμε τα δεδομένα από ένα αρχείο σε ένα άλλο.
- ☐ Το να μετακινούμε τα δεδομένα από ένα τμήμα του αρχείου σε ένα άλλο.

Πώς κάνουμε append ένα αρχείο σε ένα άλλο; *

1 βαθμός

- ☐ `cat file1.txt >> file2.txt`
- ☐ `cat file1.txt > file2.txt`
- ☐ `cat file1.txt 2> file2.txt`
- ☐ `cat file1.txt < file2.txt`

Ποια εντολή χρησιμοποιείται για να ανακατευθύνετε τα σφάλματα ενός προγράμματος σε ένα αρχείο;

* 1 βαθμός

- ☐ `java Program > error.txt`
- ☐ `java Program 2> error.txt`
- ☐ `java Program >> error.txt`
- ☐ `java Program 2>> error.txt`

Τι είναι το EOF (End Of File) *

1 βαθμός

- ☐ Χαρακτήρας που σηματοδοτεί το τέλος του αρχείου
- ☐ Μέθοδος στην Java για ανάγνωση αρχείου έως το τέλος του
- ☐ Ειδική κατηγορία αρχείων
- ☐ Τύπος δεδομένων στην Java

Τι είναι λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
Scanner sc = new Scanner(System.in);  
String input = sc.nextLine();  
System.out.println("Εισαγωγή: " input);
```

- ☐ Λείπει το '+' στην τελευταία γραμμή
- ☐ Ο Scanner δεν έχει κλείσει
- ☐ Η μέθοδος nextLine() είναι λάθος
- ☐ Δεν υπάρχει λάθος

Τι είναι λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
Scanner sc = new Scanner(System.in);  
double d = sc.nextDouble();  
System.out.println("Διπλό: " + d);  
sc.close();  
int i = sc.nextInt();  
System.out.println("Ακέραιος: " + i);
```

- ☐ Το System.out.println είναι λάθος
- ☐ Δεν μπορείτε να χρησιμοποιήσετε το sc μετά το close()
- ☐ Η μέθοδος nextDouble() είναι λάθος
- ☐ Ο διπλός τύπος δεδομένων δεν είναι υποστηριζόμενος από τη Scanner

Τι είναι η StdDraw στη γλώσσα προγραμματισμού Java; *

1 βαθμός

- ☐ Μια κλάση για γραφικές εφαρμογές
- ☐ Ένας τύπος δεδομένων για ακέραιους αριθμούς
- ☐ Ένας τύπος δεδομένων για αλφαριθμητικά
- ☐ Ένας τύπος δεδομένων για δεκαδικούς αριθμούς

Ποια μέθοδος χρησιμοποιείται για να σχεδιάσετε μια ευθεία στην StdDraw;

* 1 βαθμός

- ☐ drawTriangle()
- ☐ drawLine()
- ☐ drawPolygon()
- ☐ drawShape()

Βρείτε το σφάλμα στον κώδικα: *

1 βαθμός

```
public class Test {  
    public static void main(String[] args) {  
        StdDraw.square(0.5, 0.5, 0.2);  
        StdDraw.setPenColor(StdDraw.RED);  
        StdDraw.filledSquare(0.5, 0.5, 0.1);  
    }  
}
```

- ☐ Δεν υπάρχει σφάλμα
- ☐ Η μέθοδος square() πρέπει να καλείται πριν το setPenColor()
- ☐ Η filledSquare() απαιτεί παραπάνω ορίσματα
- ☐ Η filledSquare() πρέπει να καλείται πριν το square()

Ολοκληρώστε τον κώδικα με μία από τις 4 επιλογές: *

1 βαθμός

```
public class Drawing {  
    public static void main(String[] args) {  
        StdDraw.setScale(-1, 1);  
        StdDraw.setPenColor(StdDraw.BLUE);  
        StdDraw. ....(0, 0, 0.5);  
    }  
}
```

- ☐ drawCircle()
- ☐ filledCircle()
- ☐ drawSquare()
- ☐ filledSquare()

Τι θα είναι το αποτέλεσμα του παρακάτω κώδικα: *

1 βαθμός

```
public class Example {  
    public static void main(String[] args) {  
        StdDraw.setScale(-1, 1);  
        StdDraw.setPenColor(StdDraw.RED);  
        StdDraw.square(0, 0, 0.5);  
        StdDraw.setPenColor(StdDraw.BLUE);  
        StdDraw.filledSquare(0, 0, 0.3);  
    }  
}
```

- ☐ Ένα κόκκινο τετράγωνο με ένα μπλε γεμισμένο τετράγωνο στο εσωτερικό του
- ☐ Ένα μπλε τετράγωνο με ένα κόκκινο γεμισμένο τετράγωνο στο εσωτερικό του
- ☐ Ένα κόκκινο τετράγωνο χωρίς εσωτερικό γεμισμένο τετράγωνο
- ☐ Ένα μπλε τετράγωνο χωρίς εσωτερικό γεμισμένο τετράγωνο

Υποβολή

Εκκαθάριση φόρμας

6. Αρχές Ορθού Προγραμματισμού

Εισαγωγή στις βασικές Αρχές Ορθού Προγραμματισμού

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Ποια είναι η κύρια ιδέα πίσω από την αρχή της αφαίρεσης; *

1 βαθμός

- ☐ Να επαναλαμβάνουμε κώδικα για καλύτερη απόδοση.
- ☐ Να κάνουμε τον κώδικα όσο το δυνατόν πιο απλό.
- ☐ Να κρύβουμε την πολυπλοκότητα και να εμφανίζουμε μόνο τις απαραίτητες λεπτομέρειες.
- ☐ Να βελτιστοποιούμε τον κώδικα όσο το δυνατόν νωρίτερα.

Ποιο από τα παρακάτω παραδείγματα παραβιάζει την αρχή DRY; *

1 βαθμός

- ☐ Δημιουργία μιας μεθόδου που επαναχρησιμοποιείται σε πολλαπλά σημεία του κώδικα.
- ☐ Χρήση μεταβλητών για αποθήκευση δεδομένων που επαναχρησιμοποιούνται.
- ☐ Επαναλαμβανόμενος κώδικας σε διαφορετικές κλάσεις για την επίλυση του ίδιου προβλήματος.
- ☐ Διαχωρισμός των ανησυχιών σε διαφορετικές κλάσεις.

Ποιο είναι το κύριο μήνυμα της αρχής KISS; *

1 βαθμός

- ☐ Να κάνουμε τον κώδικα όσο το δυνατόν πιο περίπλοκο.
- ☐ Να κρατάμε τον κώδικα όσο το δυνατόν πιο απλό.
- ☐ Να βελτιστοποιούμε τον κώδικα από την αρχή.
- ☐ Να χρησιμοποιούμε τεχνολογίες αιχμής σε κάθε έργο.

Γιατί είναι σημαντικό να κάνουμε το πιο απλό που μπορεί να δουλέψει; *

1 βαθμός

- ☐ Για να εντυπωσιάσουμε τους συναδέλφους μας.
- ☐ Για να βελτιστοποιήσουμε τον κώδικα όσο το δυνατόν νωρίτερα.
- ☐ Για να δημιουργήσουμε περισσότερες δυνατότητες.
- ☐ Για να μειώσουμε την πολυπλοκότητα και να βελτιώσουμε τη συντηρησιμότητα.

Ποιο είναι το μήνυμα της αρχής YAGNI; *

1 βαθμός

- ☐ Να προσθέτουμε μόνο τις λειτουργίες που είναι απαραίτητες τώρα.
- ☐ Να προσθέτουμε λειτουργίες που ίσως χρειαστούν στο μέλλον.
- ☐ Να σχεδιάζουμε για όλες τις πιθανές μελλοντικές επεκτάσεις.
- ☐ Να βελτιστοποιούμε τον κώδικα από την αρχή.

Ποιο από τα παρακάτω ευθυγραμμίζεται με την αρχή Don't make me think;

* 1 βαθμός

- ☐ Χρήση ασυνήθιστων ονομάτων μεταβλητών.
- ☐ Προσθήκη πολλών σχολίων στον κώδικα.
- ☐ Κατασκευή περίπλοκων διαγραμμάτων ροής.
- ☐ Δημιουργία διαισθητικών διεπαφών χρήστη.

Γιατί είναι σημαντικό να γράφουμε κώδικα για τον συντηρητή (maintainer); *

1 βαθμός

- ☐ Για να εντυπωσιάσουμε τους χρήστες.
- ☐ Για να διευκολύνουμε την κατανόηση, συντήρηση και την επέκταση του κώδικα.
- ☐ Για να κάνουμε τον κώδικα πιο περίπλοκο.
- ☐ Για να βελτιστοποιήσουμε τον κώδικα από την αρχή.

Ποιο από τα παρακάτω παραδείγματα ευθυγραμμίζεται με την αρχή Principle of least astonishment ;

* 1 βαθμός

- ☐ Χρήση μη αναμενόμενης συμπεριφοράς στη διεπαφή χρήστη.
- ☐ Χρήση αναμενόμενων και καθιερωμένων προτύπων σχεδιασμού.
- ☐ Εισαγωγή πολλών νέων καινοτομιών στον κώδικα.
- ☐ Χρήση ασυνήθιστων συντομεύσεων πληκτρολογίου.

Τι σημαίνει το Open/Closed Principle; *

1 βαθμός

- ☐ Οι κλάσεις πρέπει να είναι ανοιχτές για τροποποιήσεις και κλειστές για επέκταση.
- ☐ Οι κλάσεις πρέπει να είναι ανοιχτές για επέκταση και κλειστές για τροποποιήσεις.
- ☐ Οι κλάσεις πρέπει να είναι πάντα ανοιχτές για δημόσια χρήση.
- ☐ Οι κλάσεις πρέπει να είναι κλειστές για οποιαδήποτε τροποποίηση.

Ποια είναι η κύρια ιδέα πίσω από το Single Responsibility Principle ; *

1 βαθμός

- ☐ Μια κλάση πρέπει να έχει μόνο έναν λόγο να αλλάξει.
- ☐ Μια κλάση πρέπει να κάνει πολλά πράγματα για να είναι αποδοτική.
- ☐ Μια κλάση πρέπει να είναι ανοιχτή για πολλές τροποποιήσεις.
- ☐ Μια κλάση πρέπει να επεκτείνεται συνεχώς με νέες λειτουργίες.

Γιατί είναι σημαντική η ελαχιστοποίηση της συμπλοκής στον κώδικα (Minimize Coupling);

* 1 βαθμός

- ☐ Για να κάνουμε τον κώδικα πιο πολύπλοκο.
- ☐ Για να διευκολύνουμε την επαναχρησιμοποίηση και τη συντήρηση του κώδικα.
- ☐ Για να προσθέσουμε περισσότερες λειτουργίες γρήγορα.
- ☐ Για να αυξήσουμε την εξάρτηση μεταξύ των μονάδων.

Ποιο από τα παρακάτω παραδείγματα ευθυγραμμίζεται με τη μεγιστοποίηση της συνοχής (Maximize Cohesion);

* 1 βαθμός

- ☐ Κάθε κλάση να κάνει όσο το δυνατόν περισσότερα πράγματα.
- ☐ Να υπάρχει μεγάλη αλληλεξάρτηση μεταξύ των κλάσεων.
- ☐ Κάθε κλάση και μέθοδος να έχει συγκεκριμένο και περιορισμένο σκοπό.
- ☐ Να προσθέτουμε λειτουργίες ανεξάρτητα από την υπάρχουσα δομή του κώδικα.

Ποιο είναι το όφελος του διαχωρισμού των ανησυχιών (Separation of Concerns);

* 1 βαθμός

- ☐ Κάνει τον κώδικα πιο περίπλοκο.
- ☐ Επιτρέπει τη χρήση περισσότερων γλωσσών προγραμματισμού.
- ☐ Μειώνει την αναγκαιότητα τεκμηρίωσης.
- ☐ Βοηθά στη διατήρηση και επέκταση του κώδικα.

Γιατί είναι σημαντικό να αποκρύπτουμε τις λεπτομέρειες υλοποίησης (Hide Implementation Details);

* 1 βαθμός

- ☐ Για να προστατεύσουμε τις λεπτομέρειες του κώδικα από τους χρήστες.
- ☐ Για να κάνουμε τον κώδικα πιο περίπλοκο.
- ☐ Για να προσθέτουμε περισσότερες λειτουργίες γρήγορα.
- ☐ Για να μειώσουμε την εξάρτηση και να αυξήσουμε την ευελιξία.

Ποιο είναι το κύριο μήνυμα του νόμου του Demeter; *

1 βαθμός

- ☐ Μια μέθοδος πρέπει να γνωρίζει όσο το δυνατόν λιγότερα για άλλες κλάσεις και αντικείμενα.
- ☐ Μια μέθοδος πρέπει να έχει πλήρη γνώση όλων των άλλων μεθόδων.
- ☐ Μια μέθοδος πρέπει να αλληλεπιδρά με όσο το δυνατόν περισσότερες κλάσεις.
- ☐ Μια μέθοδος πρέπει να έχει πρόσβαση σε όλες τις ιδιότητες άλλων κλάσεων.

Γιατί πρέπει να αποφεύγουμε την πρόωρη βελτιστοποίηση (Avoid Premature Optimization);

* 1 βαθμός

- ☐ Επειδή δεν είναι ποτέ απαραίτητη.
- ☐ Επειδή βελτιώνει την απόδοση του κώδικα.
- ☐ Επειδή μπορεί να προσθέσει περιττή πολυπλοκότητα και να αποσπάσει από τα πραγματικά προβλήματα.
- ☐ Επειδή διευκολύνει την κατανόηση του κώδικα.

Ποιο από τα παρακάτω παραδείγματα προωθεί την επαναχρησιμοποίηση κώδικα (Code Reuse); *

1 βαθμός

- ☐ Γράψιμο της ίδιας λειτουργίας σε πολλές κλάσεις.
- ☐ Αποφυγή δημιουργίας γενικών μεθόδων.
- ☐ Χρήση βιβλιοθηκών και πλαισίων (frameworks) για κοινές λειτουργίες.
- ☐ Ανάπτυξη μοναδικών λύσεων για κάθε πρόβλημα.

Ποιο είναι το κύριο μήνυμα της αρχής "Αποδέξου την Αλλαγή" (Embrace Change); *

1 βαθμός

- ☐ Να δημιουργούμε κώδικα που μπορεί εύκολα να τροποποιηθεί και να επεκταθεί.
- ☐ Να αντιστεκόμαστε σε οποιεσδήποτε αλλαγές στον κώδικα.
- ☐ Να αγνοούμε τις νέες τεχνολογίες και πρακτικές.
- ☐ Να προσθέτουμε όσο το δυνατόν περισσότερες δυνατότητες από την αρχή.

Υποβολή

Εκκαθάριση φόρμας

Κεφάλαιο 7: Συναρτήσεις

7. Συναρτήσεις

Εισαγωγή στις Συναρτήσεις

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Τι είναι μια συνάρτηση στην Java; *

1 βαθμός

- ☐ Ένα σύνολο εντολών που εκτελεί μια συγκεκριμένη εργασία.
- ☐ Ένα δομικό στοιχείο δεδομένων.
- ☐ Μια κλάση.
- ☐ Ένα πακέτο.

Ποιός από τους παρακάτω τρόπους δήλωσης μιας συνάρτησης στην Java είναι ορθός;

* 1 βαθμός

- ☐ `function myFunction() {}`
- ☐ `void myFunction() {}`
- ☐ `myFunction void() {}`
- ☐ `function void myFunction() {}`

Ποιος είναι ο σωστός τρόπος κλήσης μιας συνάρτησης με όνομα `calculate` χωρίς παραμέτρους;

* 1 βαθμός

- ☐ `calculate;`
- ☐ `calculate[];`
- ☐ `calculate();`
- ☐ `calculate;()`

Ποιο είναι το όνομα της προεπιλεγμένης μεθόδου εκκίνησης σε μια Java εφαρμογή; *

1 βαθμός

- ☐ `start`
- ☐ `main`
- ☐ `init`
- ☐ `run`

Ποια είναι η σωστή επιστροφή τύπου μιας συνάρτησης που δεν επιστρέφει τίποτα;

* 1 βαθμός

- ☐ null
- ☐ empty
- ☐ void
- ☐ none

Πώς δηλώνεται μια συνάρτηση που δέχεται δύο ακέραιους ως παραμέτρους και επιστρέφει έναν ακέραιο;

* 1 βαθμός

- ☐ int add(int a, int b)
- ☐ void add(int a, int b)
- ☐ int add(void a, void b)
- ☐ int add(int, int)

Ποια είναι η έξοδος της παρακάτω συνάρτησης αν κληθεί με printHello(); *

1 βαθμός

```
void printHello() {  
    System.out.println("Hello");  
}
```

- ☐ Hello
- ☐ World
- ☐ Hello World
- ☐ Κανένα από τα παραπάνω

Ποιο από τα παρακάτω ισχύει για τις συναρτήσεις στην Java; *

1 βαθμός

- ☐ Δεν μπορούν να επιστρέψουν πίνακες.
- ☐ Δεν μπορούν να έχουν παραμέτρους.
- ☐ Μπορούν να κληθούν από άλλες συναρτήσεις.
- ☐ Δεν μπορούν να είναι ιδιωτικές (private).

Ποιο είναι το αποτέλεσμα της κλήσης multiply(2, 3) αν η συνάρτηση είναι η παρακάτω; *

1 βαθμός

```
int multiply(int x, int y) {  
    return x * y;  
}
```

- ☐ 5
- ☐ 6
- ☐ 8
- ☐ 9

Πώς μπορείτε να καλέσετε μια συνάρτηση με όνομα `calculateArea` που δέχεται δύο `double` παραμέτρους; * 1 βαθμός

- ☐ `calculateArea();`
- ☐ `calculateArea(double, double);`
- ☐ `calculateArea(5.0, 7.0);`
- ☐ `calculateArea[5.0, 7.0];`

Ποια από τις παρακάτω δηλώσεις συνάρτησης είναι σωστή; * 1 βαθμός

- ☐ `public void function(int a, b)`
- ☐ `void public function(int a, int b)`
- ☐ `public void function(int a, int b)`
- ☐ `function public void(int a, int b)`

Ποια λέξη-κλειδί χρησιμοποιείται για την επιστροφή μιας τιμής από μια συνάρτηση; * 1 βαθμός

- ☐ `break`
- ☐ `exit`
- ☐ `return`
- ☐ `stop`

Τι επιστρέφει η συνάρτηση `max` με τις παραμέτρους (10, 20) αν είναι η παρακάτω; * 1 βαθμός

```
int max(int a, int b) {  
    if (a > b) {  
        return a;  
    } else {  
        return b;  
    }  
}
```

- ☐ 10
- ☐ 20
- ☐ 30
- ☐ 0

Πώς δηλώνεται μια συνάρτηση που επιστρέφει έναν πίνακα ακέραιων; * 1 βαθμός

- ☐ int arrayFunction()
- ☐ int[] arrayFunction()
- ☐ int arrayFunction[]
- ☐ arrayFunction() int[]

Τι είναι μια συνάρτηση υπερφόρτωσης (overloading) στην Java; * 1 βαθμός

- ☐ Μια συνάρτηση που επιστρέφει περισσότερες από μία τιμές.
- ☐ Μια συνάρτηση με το ίδιο όνομα αλλά διαφορετικές παραμέτρους.
- ☐ Μια συνάρτηση που κληρονομείται από άλλη κλάση.
- ☐ Μια συνάρτηση που χρησιμοποιεί εξωτερικές βιβλιοθήκες.

Πώς δηλώνετε μια συνάρτηση που δεν δέχεται παραμέτρους και επιστρέφει μια συμβολοσειρά; * 1 βαθμός

- ☐ String function()
- ☐ String function[]
- ☐ String[] function()
- ☐ void function(String)

Ποιο είναι το αποτέλεσμα της κλήσης greet("John"); αν η συνάρτηση είναι * 1 βαθμός
η παρακάτω;

```
void greet(String name) {  
    System.out.println("Hello, " + name);  
}
```

- ☐ Hello, John
- ☐ Hi, John
- ☐ Hello, name
- ☐ Κανένα από τα παραπάνω

Ποιο από τα παρακάτω είναι σωστό για τη μετάδοση παραμέτρων ακέραιων τύπου int στην Java; * 1 βαθμός

- ☐ Μετάδοση με αναφορά
- ☐ Μετάδοση με τιμή
- ☐ Μετάδοση με αντιγραφή
- ☐ Μετάδοση με δείκτες

Πώς δηλώνετε μια συνάρτηση που δέχεται έναν πίνακα ακεραίων ως παράμετρο;

* 1 βαθμός

- ☐ a) void function(int[] array)
- ☐ b) void function(int array[])
- ☐ c) void function(int array)
- ☐ d) α και β

Ποια είναι η έξοδος της παρακάτω συνάρτησης όταν κληθεί με printNumber();

* 1 βαθμός

```
int number = 5;
void printNumber() {
    int number = 10;
    System.out.println(number);
}
```

- ☐ 5
- ☐ 10
- ☐ 0
- ☐ Κανένα από τα παραπάνω

Ποια είναι η έξοδος της παρακάτω συνάρτησης όταν κληθεί με increment();

* 1 βαθμός

```
int count = 0;
void increment() {
    count++;
    System.out.println(count);
}
```

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ Κανένα από τα παραπάνω

Ποιά η σωστή σύνταξη για την επιστροφή μιας τιμής από μια συνάρτηση; * 1 βαθμός

- ☐ return value;
- ☐ return;
- ☐ return (value);
- ☐ return();

Ποια από τις παρακάτω δηλώσεις είναι σωστή για τις συναρτήσεις με τύπο επιστροφής void;

* 1 βαθμός

- ☐ Μπορούν να επιστρέψουν έναν ακέραιο.
- ☐ Δεν επιστρέφουν καμία τιμή.
- ☐ Μπορούν να επιστρέψουν μια συμβολοσειρά.
- ☐ Πρέπει να επιστρέψουν ένα αντικείμενο.

Ποια είναι η σωστή χρήση της συνάρτησης square που επιστρέφει το τετράγωνο ενός αριθμού;

* 1 βαθμός

- ☐ int result = square(5);
- ☐ square(5);
- ☐ int result = square();
- ☐ int result = square[5];

Ποιο είναι το αποτέλεσμα της κλήσης concatStrings("Hello", "World"); αν η συνάρτηση είναι η παρακάτω;

* 1 βαθμός

```
String concatStrings(String a, String b) {  
    return a + " " + b;  
}
```

- ☐ Hello
- ☐ World
- ☐ Hello World
- ☐ a b

Τι είναι λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public void myFunction(int a, int b) {  
    return a + b;  
}
```

- ☐ Η συνάρτηση δεν πρέπει να είναι δημόσια.
- ☐ Η συνάρτηση δεν πρέπει να έχει παραμέτρους.
- ☐ Η συνάρτηση δεν πρέπει να επιστρέφει τιμή.
- ☐ Η συνάρτηση έχει τύπο επιστροφής void αλλά επιστρέφει μια τιμή.

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public int add(int x, int y) {  
    int sum = x + y;  
    System.out.println(sum);  
}
```

- ☐ Λείπει η λέξη-κλειδί return.
- ☐ Το άθροισμα δεν υπολογίζεται σωστά.
- ☐ Η συνάρτηση δεν είναι δημόσια.
- ☐ Το όνομα της συνάρτησης δεν είναι σωστό.

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public static void main(String[] args) {  
    System.out.println(multiply(5, 2));  
}  
  
public int multiply(int a, int b) {  
    return a * b;  
}
```

- ☐ Η συνάρτηση multiply πρέπει να είναι στατική.
- ☐ Η συνάρτηση main δεν πρέπει να είναι στατική.
- ☐ Η συνάρτηση multiply επιστρέφει λάθος τύπο δεδομένων.
- ☐ Η συνάρτηση main έχει λάθος όνομα.

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public void displayMessage() {  
    System.out.println("Hello, World!")  
}
```

- ☐ Λείπει το ερωτηματικό.
- ☐ Δεν λείπει τίποτα.
- ☐ Λείπει το ελληνικό θαυμαστικό.
- ☐ Λείπει το ελληνικό ερωτηματικό.

Συμπληρώστε τον παρακάτω κώδικα ώστε να επιστρέφει το άθροισμα δύο αριθμών: * 1 βαθμός

```
public int sum(int a, int b) {  
    -----;  
}
```

- ☐ return a - b;
- ☐ return a + b;
- ☐ return a * b;
- ☐ return a / b;

Συμπληρώστε τον παρακάτω κώδικα ώστε να επιστρέφει το τετράγωνο ενός αριθμού: * 1 βαθμός

```
public int square(int x) {  
    -----;  
}
```

- ☐ return x / x;
- ☐ return x - x;
- ☐ return x * x;
- ☐ return x + x;

Ποια θα είναι η έξοδος του παρακάτω κώδικα; *

1 βαθμός

```
public static void main(String[] args) {  
    System.out.println(add(2, 3));  
}  
  
public static int add(int a, int b) {  
    return a + b;  
}
```

- ☐ 2
- ☐ 3
- ☐ 5
- ☐ 6

Ποιο από τα παρακάτω είναι παράδειγμα υπερφόρτωσης συναρτήσεων (function overloading); * 1 βαθμός

- ☐ Δύο συναρτήσεις με το ίδιο όνομα και ίδιες παραμέτρους.
- ☐ Δύο συναρτήσεις με το ίδιο όνομα και διαφορετικές παραμέτρους.
- ☐ Δύο συναρτήσεις με διαφορετικά ονόματα και ίδιες παραμέτρους.
- ☐ Δύο συναρτήσεις με διαφορετικά ονόματα και διαφορετικές παραμέτρους.

Ποιο είναι το αποτέλεσμα της κλήσης `print(5)`; στον παρακάτω κώδικα; * 1 βαθμός

```
public static void print(int x) {  
    System.out.println("Integer: " + x);  
}  
  
public static void print(String x) {  
    System.out.println("String: " + x);  
}
```

- ☐ Integer: 5
- ☐ String: 5
- ☐ 5
- ☐ Κανένα από τα παραπάνω

Ποιο είναι το αποτέλεσμα της κλήσης `print(3.14)`; στον παρακάτω κώδικα; * 1 βαθμός

```
public static void print(int x) {  
    System.out.println("Integer: " + x);  
}  
  
public static void print(double x) {  
    System.out.println("Double: " + x);  
}
```

- ☐ Integer: 3.14
- ☐ Double: 3.14
- ☐ 3.14
- ☐ Κανένα από τα παραπάνω

Ποια είναι η έξοδος της κλήσης `calculate(5)`; στον παρακάτω κώδικα; * 1 βαθμός

```
public static void calculate(int a) {  
    System.out.println("Square: " + (a * a));  
}  
  
public static void calculate(int a, int b) {  
    System.out.println("Sum: " + (a + b));  
}
```

- ☐ Square: 5
- ☐ Square: 25
- ☐ Sum: 5
- ☐ Sum: 25

Υποβολή

Εκκαθάριση φόρμας

Κεφάλαιο 8: Πίνακες Πολλαπλών Διαστάσεων

8. Πίνακες Πολλαπλών Διαστάσεων

Εισαγωγή στο Πίνακες Πολλαπλών Διαστάσεων

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Ποιο από τα παρακάτω δηλώνει σωστά έναν 2D πίνακα στην Java; *

1 βαθμός

- ☐ `int[] array;`
- ☐ `int[] array[];`
- ☐ `int array[][];`
- ☐ Όλα τα παραπάνω

Πώς αναφέρεστε στο στοιχείο της 2ης σειράς και 3ης στήλης σε έναν 2D * 1 βαθμός
πίνακα με όνομα `matrix`

- ☐ `matrix[3][2]`
- ☐ `matrix[2][3]`
- ☐ `matrix[1][2]`
- ☐ `matrix[2][2]`

Ποια είναι η κύρια διαφορά μεταξύ 2D και 3D πινάκων στην Java; *

1 βαθμός

- ☐ Οι 2D πίνακες έχουν δύο διαστάσεις, ενώ οι 3D πίνακες έχουν τρεις διαστάσεις.
- ☐ Οι 2D πίνακες είναι πίνακες χαρακτήρων, ενώ οι 3D πίνακες είναι πίνακες ακεραίων.
- ☐ Οι 2D πίνακες μπορούν να έχουν διαφορετικά μήκη σειρών, ενώ οι 3D πίνακες δεν μπορούν.
- ☐ Οι 3D πίνακες είναι πίνακες διδιάστατων πινάκων.

Τι επιστρέφει η `length` ιδιότητα ενός 2D πίνακα; *

1 βαθμός

- ☐ Τον αριθμό των στηλών
- ☐ Τον συνολικό αριθμό στοιχείων
- ☐ Τον αριθμό των σειρών
- ☐ Τίποτα από τα παραπάνω

Ποιο από τα παρακάτω είναι σωστό για τη δήλωση 3D πίνακα; *

1 βαθμός

- ☐ `int[][] array = new int[2][2][2];`
- ☐ `int[] array = new int[2][2][2];`
- ☐ `int[][] array = new int[2][2];`
- ☐ `int[] array[] = new int[2][2][2];`

Πώς δηλώνετε και αρχικοποιείτε έναν 2D πίνακα 3x3 στην Java; *

1 βαθμός

- ☐ `int[] array = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };`
- ☐ `int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9};`
- ☐ `int[] array = new int[3][3];`
- ☐ `int[] array = new int[9];`

Ποια εντολή δημιουργεί έναν 2D πίνακα με 5 σειρές και 10 στήλες; *

1 βαθμός

- ☐ `int array[] = new int[10][5];`
- ☐ `int[][] array = new int[5][10];`
- ☐ `int[] array[] = new int[5][10];`
- ☐ `int[][] array = new int[10][5];`

Ποια είναι η σωστή σύνταξη για την αρχικοποίηση ενός 3D πίνακα στην Java; *

1 βαθμός

- ☐ `int[][][] array = {{{1,2},{3,4}},{5,6},{7,8}}};`
- ☐ `int[][] array = {{1,2,3},{4,5,6},{7,8,9}};`
- ☐ `int[] array = {1,2,3,4,5,6,7,8};`
- ☐ `int[][][] array = new int[2][2][2];`

Πώς αρχικοποιείτε έναν 2D πίνακα με την παρακάτω τιμή; *

1 βαθμός

1 2 3
4 5 6
7 8 9

- ☐ `int[][] array = {1, 2, 3, 4, 5, 6, 7, 8, 9};`
- ☐ `int[][] array = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};`
- ☐ `int[][] array = { {1, 2}, {3, 4}, {5, 6}, {7, 8, 9} };`
- ☐ `int[][] array = { {1}, {2, 3}, {4, 5, 6, 7, 8, 9} };`

Ποια από τις παρακάτω εντολές αρχικοποιεί σωστά έναν 3D πίνακα στην Java; *

1 βαθμός

- ☐ `int[][][] array = {{{1,2,3},{4,5,6}},{7,8,9},{10,11,12}}};`
- ☐ `int[][] array = {{1,2,3},{4,5,6},{7,8,9}};`
- ☐ `int[] array = {1,2,3,4,5,6,7,8,9};`
- ☐ `int[][][] array = new int[3][3][3];`

Ποιο είναι το αποτέλεσμα της εκτέλεσης της εντολής `int[][] array = new int[2][3];`; *

1 βαθμός

- ☐ Δημιουργεί έναν 2D πίνακα με 2 στήλες και 3 σειρές
- ☐ Δημιουργεί έναν 2D πίνακα με 3 στήλες και 2 σειρές
- ☐ Δημιουργεί έναν 2D πίνακα με 2 σειρές και 3 στήλες
- ☐ Δημιουργεί έναν 2D πίνακα με 3 σειρές και 3 στήλες

Ποιο από τα παρακάτω είναι σωστό για τη δήλωση 2D πίνακα χωρίς άμεση αρχικοποίηση;

* 1 βαθμός

- ☐ `int[] array = new int[];`
- ☐ `int[] array;`
- ☐ `int[] array = { {1, 2}, {3, 4} };`
- ☐ `int[] array[] = new int[];`

Πώς αρχικοποιείτε έναν 2D πίνακα με συγκεκριμένες τιμές σε κάθε στοιχείο;

* 1 βαθμός

- ☐ `int[] array = { {1, 2, 3}, {4, 5, 6} };`
- ☐ `int array[2][3] = { {1, 2, 3}, {4, 5, 6} };`
- ☐ `int array[][] = new int[2][3];`
- ☐ `int array[2, 3] = {1, 2, 3, 4, 5, 6};`

Ποια από τις παρακάτω εντολές αρχικοποιεί σωστά έναν 3D πίνακα στην Java με συγκεκριμένες τιμές;

* 1 βαθμός

- ☐ `int[][] array = {{{1, 2, 3}, {4, 5, 6}}, {{7, 8, 9}, {10, 11, 12}}};`
- ☐ `int[][] array = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};`
- ☐ `int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9};`
- ☐ `int[][] array = new int[2][2][3];`

Ποια εντολή τροποποιεί το στοιχείο της 2ης σειράς και 3ης στήλης σε έναν 2D πίνακα `matrix` με την τιμή 10;

* 1 βαθμός

- ☐ `matrix[3][2] = 10;`
- ☐ `matrix[2][3] = 10;`
- ☐ `matrix[1][2] = 10;`
- ☐ `matrix[2][2] = 10;`

Πώς μπορείτε να τροποποιήσετε το στοιχείο της 1ης διάστασης, 2ης σειράς και 2ης στήλης σε έναν 3D πίνακα `array` με την τιμή 15;

* 1 βαθμός

- ☐ `array[2][1][1] = 15;`
- ☐ `array[1][2][1] = 15;`
- ☐ `array[1][1][2] = 15;`
- ☐ `array[0][1][1] = 15;`

Πώς μπορείτε να αλλάξετε όλα τα στοιχεία της 3ης σειράς ενός 2D πίνακα `matrix` σε 0; * 1 βαθμός

- ☐ `for (int i = 0; i < matrix[3].length; i++) matrix[2][i] = 0;`
- ☐ `for (int i = 0; i < matrix[2].length; i++) matrix[3][i] = 0;`
- ☐ `for (int i = 0; i < matrix.length; i++) matrix[2][i] = 0;`
- ☐ `for (int i = 0; i < matrix.length; i++) matrix[i][2] = 0;`

Πώς μπορείτε να αλλάξετε το στοιχείο της 1ης διάστασης, 3ης σειράς και 2ης στήλης σε έναν 3D πίνακα `array` με την τιμή 5; * 1 βαθμός

- ☐ `array[1][2][2] = 5;`
- ☐ `array[1][3][2] = 5;`
- ☐ `array[0][2][1] = 5;`
- ☐ `array[0][2][3] = 5;`

Πώς τροποποιείτε όλα τα στοιχεία ενός 2D πίνακα `matrix` σε 1; * 1 βαθμός

- ☐ `for (int i = 0; i < matrix.length; i++) for (int j = 0; j < matrix[i].length; j++) matrix[i][j] = 1;`
- ☐ `for (int i = 0; i < matrix[0].length; i++) for (int j = 0; j < matrix.length; j++) matrix[i][j] = 1;`
- ☐ `for (int i = 0; i < matrix.length; i++) for (int j = 0; j < matrix.length; j++) matrix[i][j] = 1;`
- ☐ `for (int i = 0; i < matrix[0].length; i++) for (int j = 0; j < matrix[0].length; j++) matrix[i][j] = 1;`

Πώς αναφέρεστε στο πλήθος των σειρών ενός 2D πίνακα `matrix`; * 1 βαθμός

- ☐ `matrix[0].length`
- ☐ `matrix.length`
- ☐ `matrix.length()`
- ☐ `matrix.size()`

Πώς αναφέρεστε στο πλήθος των στηλών της 2ης σειράς σε έναν 2D πίνακα `matrix`; * 1 βαθμός

- ☐ `matrix.length`
- ☐ `matrix[1].size()`
- ☐ `matrix[1].length`
- ☐ `matrix[0].length`

Ποια είναι η σωστή δήλωση για να βρείτε τον αριθμό των σειρών σε έναν 3D πίνακα array; * 1 βαθμός

- ☐ array.length
- ☐ array[0].length
- ☐ array[0][0].length
- ☐ array.size()

Πώς βρίσκετε τον αριθμό των στηλών της 1ης σειράς στη 2η διάσταση ενός 3D πίνακα array; * 1 βαθμός

- ☐ array[1].length
- ☐ array[0][1].length
- ☐ array[1][0].length
- ☐ array[0][0].length

Πώς αναφέρεστε στο πλήθος των στηλών της 3ης σειράς σε έναν 2D πίνακα matrix; * 1 βαθμός

- ☐ matrix.length
- ☐ matrix[3].length
- ☐ matrix[2].length
- ☐ matrix[2].size()

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; * 1 βαθμός

```
int[][] matrix = new int[3][3];  
matrix[3][0] = 1;
```

- ☐ Σύνταξη
- ☐ Εκτός ορίων πίνακα
- ☐ Τύπος δεδομένων
- ☐ Καμία από τις παραπάνω

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; * 1 βαθμός

```
int[][] matrix = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
System.out.println(matrix[1][3]);
```

- ☐ Σύνταξη
- ☐ Εκτός ορίων πίνακα
- ☐ Τύπος δεδομένων
- ☐ Καμία από τις παραπάνω

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
int[][] matrix = {{1, 2}, {3, 4}};
for (int i = 0; i <= matrix.length; i++) {
    for (int j = 0; j <= matrix[i].length; j++) {
        System.out.print(matrix[i][j] + " ");
    }
}
```

- ☐ Σύνταξη
- ☐ Εκτός ορίων πίνακα
- ☐ Τύπος δεδομένων
- ☐ Λογικό λάθος

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
int[][] matrix = new int[3][3];
for (int i = 0; i < matrix.length; i++) {
    for (int j = 0; j < matrix.length; j++) {
        matrix[i][j] = i + j;
    }
}
```

- ☐ Σύνταξη
- ☐ Εκτός ορίων πίνακα
- ☐ Τύπος δεδομένων
- ☐ Καμία από τις παραπάνω

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
int[][] matrix = new int[2][2];
matrix[0][0] = 1;
matrix[0][1] = 2;
matrix[1][0] = 3;
matrix[1][1] = "4";
```

- ☐ Σύνταξη
- ☐ Εκτός ορίων πίνακα
- ☐ Τύπος δεδομένων
- ☐ Καμία από τις παραπάνω

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
int[][] matrix = {(1, 2), (3, 4)};
for (int i = 0; i < matrix.length; i++) {
    for (int j = 0; j < matrix[i].size(); j++) {
        System.out.print(matrix[i][j] + " ");
    }
}
```

- ☐ Σύνταξη
- ☐ Εκτός ορίων πίνακα
- ☐ Μέθοδος που δεν υπάρχει
- ☐ Καμία από τις παραπάνω

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
int[][] matrix = new int[3][];
matrix[0] = new int[3];
matrix[1] = new int[3];
matrix[2] = new int[3];
matrix[2][3] = 10;
```

- ☐ Σύνταξη
- ☐ Εκτός ορίων πίνακα
- ☐ Μη αρχικοποιημένη σειρά
- ☐ Καμία από τις παραπάνω

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
int[][] matrix = new int[3][3];
for (int i = 0; i < matrix.length; i++) {
    for (int j = 0; j < matrix[i].length; j++) {
        matrix[i, j] = i * j;
    }
}
```

- ☐ α) Σύνταξη
- ☐ β) Εκτός ορίων πίνακα
- ☐ γ) Τύπος δεδομένων
- ☐ δ) Καμία από τις παραπάνω

Ποιο είναι το σφάλμα στον παρακάτω κώδικα; *

1 βαθμός

```
int[][] matrix = {{1, 2, 3}, {4, 5, 6}};
for (int i = 0; i < matrix.length; i++) {
    for (int j = 0; j < matrix[0].length; j++) {
        matrix[i][j] = matrix[i][j] * 2;
    }
}
matrix[2][1] = 10;
```

- ☐ Σύνταξη
- ☐ Εκτός ορίων πίνακα
- ☐ Τύπος δεδομένων
- ☐ Καμία από τις παραπάνω

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
int[][] array = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
System.out.println(array[2][1]);
```

- ☐ 2
- ☐ 5
- ☐ 8
- ☐ 7

Ποια είναι η τιμή της έκφρασης array.length για τον ακόλουθο πίνακα; *

1 βαθμός

```
int[][] array = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
```

- ☐ 3
- ☐ 9
- ☐ 6
- ☐ 2

Ποια είναι η τιμή της έκφρασης array[1].length για τον ακόλουθο πίνακα; *

1 βαθμός

```
int[][] array = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
```

- ☐ 3
- ☐ 9
- ☐ 6
- ☐ 2

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
int[][] array = new int[3][3];  
System.out.println(array[1][1]);
```

- ☐ 0
- ☐ 1
- ☐ Σφάλμα εκτέλεσης (ArrayIndexOutOfBoundsException)
- ☐ null

Τι θα είναι η τιμή της array.length στον ακόλουθο πίνακα; *

1 βαθμός

```
int[][][] array = new int[2][3][4];
```

- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 9

Ποια είναι η δήλωση πίνακα για έναν τρισδιάστατο πίνακα με διαστάσεις 2x3x4; *

1 βαθμός

- ☐ int[][] array = new int[2][3][4];
- ☐ int[][] array = new int[3][4][2];
- ☐ int[][] array = new int[4][3][2];
- ☐ int[][] array = new int[3][2][4];

Ποια είναι η τιμή του array[1][2][0] για τον ακόλουθο πίνακα; *

1 βαθμός

```
int[][][] array = {  
    {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},  
    {{10, 11, 12}, {13, 14, 15}, {16, 17, 18}}  
};
```

- ☐ 7
- ☐ 16
- ☐ 15
- ☐ 6

Υποβολή

Εκκαθάριση φόρμας

9. Αναδρομή

Εισαγωγή στην Αναδρομή

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Τι είναι η αναδρομή; *

1 βαθμός

- ☐ Μια δομή δεδομένων
- ☐ Μια τεχνική προγραμματισμού
- ☐ Ένας τύπος μεταβλητής
- ☐ Ένας τύπος σύνθετου αριθμού

Τι είναι η βάση μιας αναδρομικής συνάρτησης; *

1 βαθμός

- ☐ Ένας ειδικός τύπος δεδομένων
- ☐ Ένας τύπος αρχείου
- ☐ Η περίπτωση που η συνάρτηση σταματάει την αναδρομή
- ☐ Η αρχή ενός προγράμματος

Ποια είναι μια κοινή χρήση της αναδρομής; *

1 βαθμός

- ☐ Προσθέτει συνθήκες στον κώδικα
- ☐ Συνήθως χρησιμοποιείται για επαναληπτικούς αλγορίθμους
- ☐ Καθιστά τον κώδικα πιο αποδοτικό
- ☐ Μπορεί να αντικαταστήσει τις επαναληπτικές δομές ελέγχου

Ποιος είναι ο κίνδυνος της αναδρομής; *

1 βαθμός

- ☐ Να προκύψει ατέρμονος κύκλος
- ☐ Να είναι πιο αργή από την επανάληψη
- ☐ Να μην λειτουργεί σωστά με μεγάλα δεδομένα
- ☐ Να είναι περίπλοκος στην υλοποίηση

Ποιος είναι ένας τρόπος για να αποφευχθεί ο ατέρμονος κύκλος στην αναδρομή;

* 1 βαθμός

- ☐ Χρησιμοποιώντας μόνο επαναληπτικές δομές
- ☐ Με τη χρήση ενός μετρητή
- ☐ Ελέγχοντας την συνθήκη τερματισμού / βάση αναδρομής
- ☐ Αυξάνοντας το μέγεθος της μνήμης

Ποια είναι η βασική ιδέα πίσω από την αναδρομή; *

1 βαθμός

- ☐ Η χρήση του στοιχείου
- ☐ Η επανάληψη της καλούσας συνάρτησης από τον εαυτό της
- ☐ Η χρήση στοίβας
- ☐ Η επανάληψη της βασικής συνάρτησης

Τι συμβαίνει κατά τη διάρκεια μιας αναδρομικής κλήσης; *

1 βαθμός

- ☐ Η μνήμη αποθηκεύεται σε μια στοίβα
- ☐ Ο κώδικας εκτελείται στον ανώνυμο χώρο μνήμης
- ☐ Ο κώδικας εκτελείται στον ίδιο χώρο μνήμης με τη βασική συνάρτηση
- ☐ Ο κώδικας δεν εκτελείται

Ποιο είναι το κριτήριο τερματισμού σε μια αναδρομική συνάρτηση; *

1 βαθμός

- ☐ Η ύπαρξη συνθήκης εξόδου
- ☐ Η χρήση εξωτερικού μετρητή
- ☐ Η εκτέλεση ενός συγκεκριμένου αριθμού επαναλήψεων
- ☐ Η τυχαία επιλογή από τον προγραμματιστή

Ποιος είναι ο ρόλος της βασικής περίπτωσης στην αναδρομή; *

1 βαθμός

- ☐ Να ορίζει την αρχική κατάσταση ή την περίπτωση τερματισμού
- ☐ Να αποθηκεύει τα δεδομένα στη μνήμη
- ☐ Να αναδρομικά καλεί άλλες συναρτήσεις
- ☐ Να ελέγχει τη σωστή σύνταξη της κύριας συνάρτησης

Ποιο από τα παρακάτω είναι ένα παράδειγμα αναδρομικής συνάρτησης; *

1 βαθμός

- ☐ Υπολογισμός του αθροίσματος ενός πίνακα
- ☐ Υπολογισμός του παραγοντικού ενός αριθμού
- ☐ Ταξινόμηση ενός πίνακα
- ☐ Εύρεση του μέγιστου στοιχείου σε μια λίστα

Ποιο από τα παρακάτω είναι ένα παράδειγμα τυπικής αναδρομής στην Java; * 1 βαθμός

- ☐ for loop
- ☐ switch statement
- ☐ if-else statement
- ☐ recursive method

Ποιο από τα παρακάτω είναι ένα κοινό πρόβλημα που επιλύεται χρησιμοποιώντας αναδρομή; * 1 βαθμός

- ☐ Αντιγραφή ενός πίνακα
- ☐ Ταξινόμηση ενός πίνακα
- ☐ Υπολογισμός του αθροίσματος ενός πίνακα
- ☐ Αναζήτηση ενός στοιχείου σε έναν ταξινομημένο πίνακα

Τι θα συμβεί αν μια αναδρομική μέθοδος δεν έχει μια βάση, δηλαδή ένα σημείο που σταματάει η αναδρομή; * 1 βαθμός

- ☐ Θα προκύψει σφάλμα επανάληψης (StackOverflowError)
- ☐ Η μέθοδος θα επιστρέψει null
- ☐ Η μέθοδος θα επιστρέψει το 0
- ☐ Η μέθοδος θα συνεχίσει να εκτελείται μέχρι να εξαντληθεί η μνήμη

Ποια είναι μία από τις κύριες περιπτώσεις όπου η αναδρομή είναι χρήσιμη; * 1 βαθμός

- ☐ Υλοποίηση επαναληπτικών δομών δεδομένων
- ☐ Συγχρονισμός νημάτων (threads)
- ☐ Αποσφαλμάτωση κώδικα
- ☐ Υλοποίηση γραφικών περιβαλλόντων χρήστη

Πώς μπορεί να αποφευχθεί η άπειρη αναδρομή σε μια αναδρομική μέθοδο; * 1 βαθμός

- ☐ Χρησιμοποιώντας ένα βρόχο επανάληψης
- ☐ Ελέγχοντας για έναν κατάλληλο όρο τερματισμού
- ☐ Χρησιμοποιώντας μόνο αναδρομικές κλήσεις
- ☐ Αυξάνοντας το μέγεθος της στοίβας (stack)

Ποια από τις παρακάτω δομές δεδομένων είναι πιο κατάλληλη για την αποθήκευση των ενεργών κλήσεων μεθόδων κατά την αναδρομή σε μια αναδρομική συνάρτηση;

* 1 βαθμός

- ☐ Λίστα (List)
- ☐ Ουρά (Queue)
- ☐ Στοιβά (Stack)
- ☐ Δέντρο (Tree)

Ποια είναι η σωστή αναδρομική υλοποίηση του προβλήματος του μέγιστου κοινού διαιρέτη (GCD) στην Java;

* 1 βαθμός

- ☐ $\text{gcd}(a, b) = \text{gcd}(b, a \% b)$
- ☐ $\text{gcd}(a, b) = \text{gcd}(a \% b, b)$
- ☐ $\text{gcd}(a, b) = \text{gcd}(a - b, b)$
- ☐ $\text{gcd}(a, b) = \text{gcd}(a, b - a)$

Ποια είναι η αναδρομική υλοποίηση του προβλήματος του παραγοντικού στην Java;

* 1 βαθμός

- ☐ $\text{factorial}(n) = n * \text{factorial}(n - 1)$
- ☐ $\text{factorial}(n) = \text{factorial}(n - 1) / n$
- ☐ $\text{factorial}(n) = n + \text{factorial}(n - 1)$
- ☐ $\text{factorial}(n) = \text{factorial}(n - 1) * (n - 1)$

Ποια είναι η σωστή αναδρομική υλοποίηση της ακολουθίας Fibonacci στην Java;

* 1 βαθμός

- ☐ $\text{fibonacci}(n) = \text{fibonacci}(n - 1) + \text{fibonacci}(n - 2)$
- ☐ $\text{fibonacci}(n) = \text{fibonacci}(n + 1) + \text{fibonacci}(n + 2)$
- ☐ $\text{fibonacci}(n) = \text{fibonacci}(n - 1) * \text{fibonacci}(n - 2)$
- ☐ $\text{fibonacci}(n) = \text{fibonacci}(n) + \text{fibonacci}(n - 1)$

Ποιο από τα παρακάτω μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων όπως η λύση του προβλήματος του παλίνδρομου;

* 1 βαθμός

- ☐ Αναδρομικές κλήσεις
- ☐ Δομές επανάληψης, όπως το for loop
- ☐ Δομές δεδομένων, όπως οι λίστες

Συμπληρώστε τον παρακάτω κώδικα ώστε να υπολογιστεί το παραγοντικό ενός αριθμού:

* 1 βαθμός

```
public int factorial(int n) {  
    if (n ___) {  
        return 1;  
    } else {  
        return n * factorial(___);  
    }  
}
```

- ☐ == 1, n-1
- ☐ <= 1, n-1
- ☐ == 0, n+1
- ☐ <= 0, n+1

Τι θα είναι το αποτέλεσμα του παρακάτω κώδικα; *

1 βαθμός

```
public class Main {  
    public static void main(String[] args) {  
        int result = sum(5);  
        System.out.println(result);  
    }  
  
    public static int sum(int n) {  
        if (n <= 0) {  
            return 0;  
        } else {  
            return n + sum(n - 1);  
        }  
    }  
}
```

- ☐ 15
- ☐ 10
- ☐ 5
- ☐ Προκαλεί σφάλμα εκτέλεσης (runtime error)

Συμπληρώστε τον παρακάτω κώδικα ώστε να υπολογιστεί ο n-οστός όρος της ακολουθίας Fibonacci:

* 1 βαθμός

```
public int fibonacci(int n) {  
    if (n ___) {  
        return 0;  
    } else if (n ___) {  
        return 1;  
    } else {  
        return fibonacci(___ - 1) + fibonacci(___ - 2);  
    }  
}
```

- ☐ == 0, == 1, n
- ☐ <= 0, == 1, n
- ☐ <= 0, == 0, n
- ☐ == 1, == 0, n

Συμπληρώστε τον παρακάτω κώδικα ώστε να υπολογιστεί ο μέγιστος κοινός διαιρέτης δύο αριθμών:

* 1 βαθμός

```
public int gcd(int a, int b) {  
    if (___) {  
        return a;  
    } else {  
        return gcd(____, ____);  
    }  
}
```

- ☐ b == 0, b, a % b
- ☐ a == 0, a, b % a
- ☐ b <= 0, b, a - b
- ☐ a <= 0, a, b - a

Βρείτε το σφάλμα στον παρακάτω κώδικα: *

1 βαθμός

```
public class Main {  
    public static void main(String[] args) {  
        int result = calculateSum(5);  
        System.out.println("The sum is: " + result);  
    }  
  
    public static int calculateSum(int n) {  
        if (n == 0) {  
            return 0;  
        } else {  
            return n + calculateSum(n);  
        }  
    }  
}
```

- ☐ Η συνάρτηση calculateSum δεν καλείται με τον σωστό τρόπο.
- ☐ Η συνθήκη στο if πρέπει να είναι n <= 0.
- ☐ Η συνάρτηση δεν έχει σφάλματα, θα λειτουργήσει σωστά.
- ☐ Η συνάρτηση πρέπει να επιστρέφει 1 αντί για 0 στην περίπτωση n == 0.

Βρείτε το σφάλμα στον παρακάτω κώδικα: *

1 βαθμός

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Factorial of 5 is: " + findFactorial(5));  
    }  
  
    public static int findFactorial(int n) {  
        if (n == 1) {  
            return 1;  
        } else {  
            return n * findFactorial(n - 1);  
        }  
    }  
}
```

- ☐ Η συνθήκη στο if πρέπει να είναι $n \leq 1$.
- ☐ Η συνάρτηση δεν ελέγχει για αρνητικές τιμές του n .
- ☐ Το πρόγραμμα δεν θα δώσει σωστό αποτέλεσμα για μεγάλες τιμές του n .
- ☐ Το πρόγραμμα θα προκαλέσει σφάλμα υπερχείλισης (overflow) για μεγάλες τιμές του n .

Βρείτε το σφάλμα στον παρακάτω κώδικα: *

1 βαθμός

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Fibonacci series up to 5:");
        for (int i = 0; i <= 5; i++) {
            System.out.print(fibonacci(i) + " ");
        }
    }

    public static int fibonacci(int n) {
        if (n <= 1) {
            return n;
        } else {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }
}
```

- ☐ Η μεταβλητή i πρέπει να αρχίζει από 1 αντί για 0.
- ☐ Η συνάρτηση fibonacci πρέπει να επιστρέφει 1 αντί για n για n == 1.
- ☐ Η συνθήκη στο for loop πρέπει να είναι i < 5 αντί για i <= 5.
- ☐ Η συνάρτηση fibonacci δεν κάνει αναδρομική κλήση με τον σωστό τρόπο.

Σε ποιά γραμμή βρίσκεται η αναδρομική κλήση; *

1 βαθμός

```
1 public class Main {
2     public static void main(String[] args) {
3         System.out.println("Fibonacci series up to 5:");
4         for (int i = 0; i <= 5; i++) {
5             System.out.print(fibonacci(i) + " ");
6         }
7     }
8
9     public static int fibonacci(int n) {
10        if (n <= 1) {
11            return n;
12        } else {
13            return fibonacci(n - 1) + fibonacci(n - 2);
14        }
15    }
16 }
```

Η απάντησή σας

Βρείτε το σφάλμα στον παρακάτω κώδικα: *

1 βαθμός

```
public class Main {
    public static void main(String[] args) {
        System.out.println("GCD of 15 and 25 is: " + calculateGCD(15, 25));
    }

    public static int calculateGCD(int a, int b) {
        if (a == 0) {
            return b;
        } else if (b == 0) {
            return a;
        } else if (a > b) {
            return calculateGCD(a - b, b);
        } else {
            return calculateGCD(a, b - a);
        }
    }
}
```

- ☐ Οι συνθήκες στο if και στο else if πρέπει να είναι $a \leq 0$ και $b \leq 0$ αντίστοιχα.
- ☐ Η συνθήκη $a > b$ πρέπει να είναι $a \geq b$.
- ☐ Ο κώδικας δεν είναι σωστός για την υπολογισμό του μέγιστου κοινού διαιρέτη.
- ☐ Ο κώδικας δεν κάνει αναδρομική κλήση με τον σωστό τρόπο.

Βρείτε το σφάλμα στον παρακάτω κώδικα: *

1 βαθμός

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Result: " + calculateFibonacci(6));
    }

    public static int calculateFibonacci(int n) {
        if (n == 0 || n == 1) {
            return n;
        } else {
            return calculateFibonacci(n + 1) + calculateFibonacci(n + 2);
        }
    }
}
```

- ☐ Η συνθήκη στο if πρέπει να είναι $n \leq 0 \parallel n \leq 1$.
- ☐ Η συνάρτηση πρέπει να καλεί τον εαυτό της ως $\text{calculateFibonacci}(n) + \text{calculateFibonacci}(n - 1)$.
- ☐ Η συνθήκη στο if πρέπει να είναι $n \geq 0 \ \&\& \ n \leq 1$.
- ☐ Η συνάρτηση πρέπει να καλεί τον εαυτό της ως $\text{calculateFibonacci}(n - 1) + \text{calculateFibonacci}(n - 2)$.

Υποβολή

Εκκαθάριση φόρμας

10. Αξιοποίηση και Χρήση Αντικειμένων

Εισαγωγή στην Αξιοποίηση και Χρήση Αντικειμένων

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Τι είναι ένα αντικείμενο στην Java; *

1 βαθμός

- ☐ Μια μεταβλητή
- ☐ Μια συνάρτηση
- ☐ Μια περίπτωση κλάσης
- ☐ Ένας τύπος δεδομένων

Πώς καλούμε μια μέθοδο ενός αντικειμένου στην Java; *

1 βαθμός

- ☐ object.method()
- ☐ method(object)
- ☐ object:method()
- ☐ object->method()

Ποιο από τα παρακάτω ΔΕΝ είναι αληθές για τα αντικείμενα στην Java; *

1 βαθμός

- ☐ Μπορούν να έχουν ιδιότητες (μεταβλητές κλάσης)
- ☐ Μπορούν να έχουν μεθόδους
- ☐ Δεν μπορούν να δημιουργηθούν δυναμικά
- ☐ Είναι περιπτώσεις κλάσεων

Ποια είναι η βασική διαφορά μεταξύ μιας κλάσης και ενός αντικειμένου; *

1 βαθμός

- ☐ Μια κλάση είναι ένα συγκεκριμένο παράδειγμα ενός αντικειμένου
- ☐ Ένα αντικείμενο είναι ένα συγκεκριμένο παράδειγμα μιας κλάσης
- ☐ Δεν υπάρχει διαφορά
- ☐ Ένα αντικείμενο περιέχει μόνο στατικές μεθόδους

Ποιο είναι το αποτέλεσμα της `str.length()` αν το `str` είναι ένα αντικείμενο της κλάσης `String` με τιμή `"Hello"`;

* 1 βαθμός

- ☐ `'Hello'.length`
- ☐ 5
- ☐ `str.length`
- ☐ 6

Πώς ελέγχουμε αν δύο αντικείμενα της κλάσης String είναι ίσα ως προς το περιεχόμενό τους; * 1 βαθμός

- ☐ ==
- ☐ equals()
- ☐ isEqual()
- ☐ compare()

Ποιο είναι το αποτέλεσμα της παρακάτω εντολής; Integer.parseInt("123") * 1 βαθμός

- ☐ "123"
- ☐ 123
- ☐ parse("123")
- ☐ NumberFormatException

Τι θα εκτυπώσει η παρακάτω εντολή; System.out.println(new String("Java").equals("Java")); * 1 βαθμός

- ☐ true
- ☐ false
- ☐ Java
- ☐ Exception

Ποια είναι η προφανής διαφορά μεταξύ των βασικών τύπων και των αντικειμένων στην Java; * 1 βαθμός

- ☐ Οι βασικοί τύποι είναι απλούστεροι από τα αντικείμενα
- ☐ Οι βασικοί τύποι είναι πάντα null
- ☐ Τα αντικείμενα δεν έχουν τύπο
- ☐ Τα αντικείμενα δεν μπορούν να περιέχουν δεδομένα

Ποιο από τα παρακάτω είναι βασικός τύπος δεδομένων στην Java; * 1 βαθμός

- ☐ String
- ☐ Integer
- ☐ int
- ☐ ArrayList

Ποια είναι η διαφορά μεταξύ `int` και `Integer`; *

1 βαθμός

- ☐ `int` είναι αντικείμενο, `Integer` είναι βασικός τύπος
- ☐ `int` είναι βασικός τύπος, `Integer` είναι αντικείμενο
- ☐ Και τα δύο είναι βασικοί τύποι
- ☐ Και τα δύο είναι αντικείμενα

Πώς μετατρέπουμε έναν βασικό τύπο `int` σε ένα αντικείμενο `Integer`; *

1 βαθμός

- ☐ `Integer.parseInt(int)`
- ☐ `Integer.valueOf(int)`
- ☐ `int.toInteger()`
- ☐ `new Integer(int)`

Ποια είναι η προεπιλεγμένη τιμή για ένα αντικείμενο που δεν έχει αρχικοποιηθεί;

* 1 βαθμός

- ☐ 0
- ☐ null
- ☐ false
- ☐ undefined

Ποια είναι η προεπιλεγμένη τιμή για έναν βασικό τύπο `boolean` που δεν έχει αρχικοποιηθεί;

* 1 βαθμός

- ☐ 0
- ☐ null
- ☐ false
- ☐ true

Ποιο από τα παρακάτω είναι αντικείμενο στην Java; *

1 βαθμός

- ☐ `double`
- ☐ `char`
- ☐ `Boolean`
- ☐ `long`

Πώς ελέγχουμε αν ένα αντικείμενο `Integer` είναι `null`; *

1 βαθμός

- ☐ `object == null`
- ☐ `object.equals(null)`
- ☐ `object.isNull()`
- ☐ `object == 0`

Τι είναι ένας κατασκευαστής (constructor) στην Java; *

1 βαθμός

- ☐ Μια μέθοδος που καλείται όταν δημιουργείται ένα αντικείμενο
- ☐ Μια μέθοδος που επιστρέφει τιμές από το αντικείμενο
- ☐ Μια μέθοδος που αλλάζει τις τιμές του αντικειμένου
- ☐ Ένας τύπος δεδομένων

Ποια είναι η διαφορά μεταξύ ενός κατασκευαστή και μιας κανονικής μεθόδου;

* 1 βαθμός

- ☐ Ένας κατασκευαστής επιστρέφει πάντα μια τιμή
- ☐ Ένας κατασκευαστής έχει το ίδιο όνομα με την κλάση
- ☐ Μια κανονική μέθοδος δεν μπορεί να έχει παραμέτρους
- ☐ Μια κανονική μέθοδος καλείται αυτόματα

Ποιος είναι ο σκοπός μιας μεθόδου ανάκτησης (getter) στην Java; *

1 βαθμός

- ☐ Να αλλάζει την τιμή μιας μεταβλητής
- ☐ Να διαγράφει μια μεταβλητή
- ☐ Να επιστρέφει την τιμή μιας μεταβλητής
- ☐ Να δημιουργεί ένα νέο αντικείμενο

Ποιος είναι ο σκοπός μιας μεθόδου μετατροπής (setter) στην Java; *

1 βαθμός

- ☐ Να αλλάζει την τιμή μιας μεταβλητής
- ☐ Να διαγράφει μια μεταβλητή
- ☐ Να επιστρέφει την τιμή μιας μεταβλητής
- ☐ Να δημιουργεί ένα νέο αντικείμενο

Ποια είναι η σωστή ονομασία για μια μέθοδο ανάκτησης μιας ιδιότητας `age` στην Java;

* 1 βαθμός

- ☐ `setAge()`
- ☐ `getAge()`
- ☐ `changeAge()`
- ☐ `retrieveAge()`

Ποια είναι η σωστή ονομασία για μια μέθοδο μετατροπής μιας ιδιότητας `name` στην Java;

* 1 βαθμός

- ☐ `getName()`
- ☐ `retrieveName()`
- ☐ `setName()`
- ☐ `changeName()`

Τι σημαίνει όταν μια μέθοδος στην Java είναι `public`; *

1 βαθμός

- ☐ Μπορεί να καλείται μόνο από άλλες μεθόδους της ίδιας κλάσης
- ☐ Μπορεί να καλείται από οποιαδήποτε κλάση
- ☐ Μπορεί να καλείται μόνο από υποκλάσεις
- ☐ Δεν μπορεί να καλείται από πουθενά

Τι σημαίνει όταν μια μεταβλητή στην Java είναι `private`; *

1 βαθμός

- ☐ Μπορεί να προσπελαστεί από οποιαδήποτε κλάση
- ☐ Μπορεί να προσπελαστεί μόνο από την ίδια κλάση
- ☐ Μπορεί να προσπελαστεί μόνο από υποκλάσεις
- ☐ Δεν μπορεί να προσπελαστεί από πουθενά

Ποιο είναι το κύριο πλεονέκτημα της χρήσης `private` μεταβλητών; *

1 βαθμός

- ☐ Είναι πιο γρήγορες
- ☐ Είναι ασφαλέστερες και προστατεύουν τα δεδομένα
- ☐ Είναι ευκολότερες στη χρήση
- ☐ Καταλαμβάνουν λιγότερη μνήμη

Πώς μπορούμε να προσπελάσουμε μια `private` μεταβλητή από έξω από την κλάση; * 1 βαθμός

- ☐ Απευθείας
- ☐ Χρησιμοποιώντας μια `public` μέθοδο ανάκτησης (getter)
- ☐ Δεν μπορούμε να την προσπελάσουμε
- ☐ Χρησιμοποιώντας τη λέξη-κλειδί `protected`

Ποιο είναι το αποτέλεσμα του παρακάτω κώδικα; * 1 βαθμός

```
public class MyClass { private int value;  
public void setValue(int value) { this.value = value; }  
}
```

- ☐ Η μεταβλητή `value` δεν μπορεί να αλλάξει
- ☐ Η μεταβλητή `value` μπορεί να αλλάξει μέσω της μεθόδου `setValue`
- ☐ Η μέθοδος `setValue` δεν έχει πρόσβαση στη μεταβλητή `value`
- ☐ Δεν συμβαίνει τίποτα

Ποια είναι η διαφορά μεταξύ `public` και `private` μεθόδων; * 1 βαθμός

- ☐ Οι `public` μέθοδοι μπορούν να καλούνται από οπουδήποτε, οι `private` μόνο από την ίδια κλάση
- ☐ Οι `private` μέθοδοι είναι πιο γρήγορες
- ☐ Οι `public` μέθοδοι δεν μπορούν να επιστρέψουν τιμές
- ☐ Δεν υπάρχει διαφορά

Τι σημαίνει όταν μια μέθοδος είναι `static` στην `Java`; * 1 βαθμός

- ☐ Μπορεί να καλείται χωρίς να δημιουργηθεί αντικείμενο της κλάσης
- ☐ Μπορεί να καλείται μόνο από αντικείμενα της κλάσης
- ☐ Δεν μπορεί να προσπελαστεί από άλλες κλάσεις
- ☐ Δεν μπορεί να επιστρέψει τιμές

Ποιο είναι το αποτέλεσμα του παρακάτω κώδικα; * 1 βαθμός

```
public static void myMethod() { System.out.println("Hello"); }
```

- ☐ Η μέθοδος δεν μπορεί να εκτυπώσει τίποτα
- ☐ Η μέθοδος μπορεί να εκτυπώσει "Hello" χωρίς να δημιουργηθεί αντικείμενο
- ☐ Η μέθοδος μπορεί να εκτυπώσει "Hello" μόνο αν δημιουργηθεί αντικείμενο
- ☐ Η μέθοδος δεν μπορεί να καλείται

Ποια είναι η διαφορά μεταξύ μιας static και μιας μη στατικής μεταβλητής; * 1 βαθμός

- ☐ Μια static μεταβλητή ανήκει στην κλάση, μια μη στατική ανήκει στο αντικείμενο
- ☐ Μια μη στατική μεταβλητή ανήκει στην κλάση, μια static ανήκει στο αντικείμενο
- ☐ Και οι δύο ανήκουν στο αντικείμενο
- ☐ Δεν υπάρχει διαφορά

Πώς καλούμε μια static μέθοδο myStaticMethod της κλάσης MyClass; * 1 βαθμός

- ☐ MyClass.myStaticMethod()
- ☐ new MyClass().myStaticMethod()
- ☐ myStaticMethod()
- ☐ this.myStaticMethod()

Μπορεί μια static μέθοδος να προσπελάσει μη στατικές μεταβλητές της κλάσης; * 1 βαθμός

- ☐ Ναι
- ☐ Όχι
- ☐ Μόνο αν είναι public
- ☐ Μόνο αν είναι private

Ποιο από τα παρακάτω είναι σωστό για μη στατικές μεθόδους; * 1 βαθμός

- ☐ Μπορούν να καλούνται χωρίς να δημιουργηθεί αντικείμενο
- ☐ Πρέπει να καλούνται μέσω ενός αντικειμένου της κλάσης
- ☐ Δεν μπορούν να προσπελάσουν μη στατικές μεταβλητές
- ☐ Είναι πάντα private

Τι κάνει η μέθοδος toString() σε ένα αντικείμενο στην Java; * 1 βαθμός

- ☐ Καταγράφει το μέγεθος του αντικειμένου.
- ☐ Μετατρέπει το αντικείμενο σε δείκτη μνήμης.
- ☐ Επιστρέφει μια συμβολοσειρά που αντιπροσωπεύει το αντικείμενο.
- ☐ Εκτυπώνει τα στοιχεία του αντικειμένου στον κονσόλα.

Πατί είναι σημαντικό να υλοποιείται η μέθοδος `toString()` σε μια κλάση στην Java; * 1 βαθμός

- ☐ Για να αυξηθεί το μέγεθος του αντικειμένου.
- ☐ Για να επιτρέψει στο αντικείμενο να αλληλεπιδρά με άλλα αντικείμενα.
- ☐ Προσφέρει μια συνοπτική αναπαράσταση του αντικειμένου ως συμβολοσειρά.
- ☐ Προκαλεί μια εξαίρεση `NullPointerException`

Ποιος τύπος επιστροφής έχει η μέθοδος `toString()` στην Java; * 1 βαθμός

- ☐ `String`
- ☐ `int`
- ☐ `void`
- ☐ `boolean`

Ποια είναι η προεπιλεγμένη υλοποίηση της μεθόδου `toString()` στην κλάση `Object` της Java; * 1 βαθμός

- ☐ Επιστρέφει την τιμή `null`.
- ☐ Επιστρέφει τον τύπο δεδομένων της κλάσης.
- ☐ Επιστρέφει μια συμβολοσειρά που περιέχει το όνομα της κλάσης και ένα μοναδικό αναγνωριστικό του αντικειμένου.
- ☐ Δεν υπάρχει προεπιλεγμένη υλοποίηση.

Τι γίνεται αν μια κλάση δεν υλοποιεί τη μέθοδο `equals()`; * 1 βαθμός

- ☐ Ο μεταγλωττιστής παράγει έναν κώδικα λάθους.
- ☐ Η μέθοδος `equals()` επιστρέφει πάντα `false`.
- ☐ Η μέθοδος `equals()` επιστρέφει πάντα `true`.
- ☐ Το αντικείμενο χρησιμοποιεί την προεπιλεγμένη υλοποίηση της κλάσης `Object`.

Ποια είναι η προεπιλεγμένη υλοποίηση της μεθόδου `equals()` στην κλάση `Object` της Java; * 1 βαθμός

- ☐ Ελέγχει αν τα αντικείμενα δείχνουν στην ίδια διεύθυνση μνήμης.
- ☐ Συγκρίνει τα περιεχόμενα των αντικειμένων χρησιμοποιώντας τον τελεστή `==`.
- ☐ Επιστρέφει πάντα `false`.
- ☐ Δεν υπάρχει προεπιλεγμένη υλοποίηση.

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public class Test {  
    public void main(String[] args) {  
        System.out.println("Hello");  
    }  
}
```

- ☐ Λείπει το static στην μέθοδο main
- ☐ Λείπει η αγκύλη κλεισίματος της κλάσης
- ☐ Το println είναι γραμμένο λάθος
- ☐ Η μέθοδος main πρέπει να επιστρέφει int

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public class Example {  
    private int value;  
  
    public void setValue(int newValue) {  
        value = newValue;  
    }  
  
    public int getValue() {  
        return Value;  
    }  
}
```

- ☐ Η μέθοδος setValue πρέπει να επιστρέφει int
- ☐ Η μεταβλητή value πρέπει να είναι public
- ☐ Το Value πρέπει να είναι value στη μέθοδο getValue
- ☐ Λείπει ο κατασκευαστής της κλάσης

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public class MyClass {  
    public static void main(String[] args) {  
        String text = null;  
        System.out.println(text.length());  
    }  
}
```

- ☐ Το text πρέπει να είναι new String()
- ☐ Το text πρέπει να έχει τιμή
- ☐ Η μέθοδος println δεν είναι σωστή
- ☐ Η μεταβλητή text πρέπει να είναι static

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public class Test {  
    public static void main(String[] args) {  
        int[] numbers = new int[5];  
        numbers[5] = 10;  
    }  
}
```

- ☐ Δεν υπάρχει λάθος
- ☐ Το numbers[5] πρέπει να είναι numbers[4]
- ☐ Το numbers πρέπει να είναι static
- ☐ Η δήλωση int[] δεν είναι σωστή

Ποιο είναι το λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public class Test {  
    public void display() {  
        System.out.println("Hello");  
    }  
  
    public static void main(String[] args) {  
        display();  
    }  
}
```

- ☐ Η μέθοδος display πρέπει να είναι static
- ☐ Η μέθοδος main πρέπει να επιστρέφει int
- ☐ Η μεταβλητή args πρέπει να είναι String[]
- ☐ Το println είναι γραμμένο λάθος

Συμπληρώστε τον κώδικα για να θέσετε την τιμή της age: *

1 βαθμός

```
public class Person {  
    private int age;  
  
    public void setAge(int age) {  
        this. .... = age;  
    }  
}
```

- ☐ age
- ☐ thisAge
- ☐ setAge
- ☐ ageValue

Συμπληρώστε τον κώδικα για να καλέσετε τη μέθοδο display: *

1 βαθμός

```
public class Test {  
    public void display() {  
        System.out.println("Hello");  
    }  
  
    public static void main(String[] args) {  
        Test test = new Test();  
        test. ....;  
    }  
}
```

- ☐ α) display()
- ☐ β) test.display
- ☐ γ) display
- ☐ δ) test.display()

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
public class Test {  
    public static void main(String[] args) {  
        String str = "Java";  
        System.out.println(str.length());  
    }  
}
```

- ☐ 3
- ☐ 4
- ☐ 5
- ☐ 0

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
public class Test {  
    public static void main(String[] args) {  
        String str = "Java";  
        System.out.println(str.toUpperCase());  
    }  
}
```

- ☐ JAVA
- ☐ java
- ☐ Java
- ☐ JaVa

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
class Animal {  
    public void makeSound() {  
        System.out.println("Some sound");  
    }  
}  
  
public class Dog extends Animal {  
    public void makeSound() {  
        System.out.println("Bark");  
    }  
  
    public static void main(String[] args) {  
        Animal myDog = new Dog();  
        myDog.makeSound();  
    }  
}
```

- ☐ Some sound
- ☐ Bark
- ☐ Error
- ☐ Dog sound

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
class Person {  
    String name;  
  
    public Person(String name) {  
        this.name = name;  
    }  
  
    public void display() {  
        System.out.println("Name: " + name);  
    }  
  
    public static void main(String[] args) {  
        Person p = new Person("John");  
        p.display();  
    }  
}
```

- ☐ Name:
- ☐ Name: John
- ☐ Name: null
- ☐ Error

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
class Calculator {  
    public int add(int a, int b) {  
        return a + b;  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        Calculator calc = new Calculator();  
        System.out.println(calc.add(5, 10));  
    }  
}
```

- ☐ 15
- ☐ 510
- ☐ Error
- ☐ 5

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
class MyClass {  
    int x;  
  
    public MyClass(int val) {  
        x = val;  
    }  
  
    public static void main(String[] args) {  
        MyClass obj1 = new MyClass(5);  
        MyClass obj2 = new MyClass(10);  
        obj1 = obj2;  
        System.out.println(obj1.x);  
    }  
}
```

- ☐ 5
- ☐ 10
- ☐ 0
- ☐ Error

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
class MyClass {  
    static int count = 0;  
  
    public MyClass() {  
        count++;  
    }  
  
    public static void main(String[] args) {  
        MyClass obj1 = new MyClass();  
        MyClass obj2 = new MyClass();  
        MyClass obj3 = new MyClass();  
        System.out.println(MyClass.count);  
    }  
}
```

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 0

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
class Person {  
    String name;  
    int age;  
  
    public Person(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
  
    @Override  
    public String toString() {  
        return "Person(name='" + name + "', age=" + age + ")";  
    }  
  
    public static void main(String[] args) {  
        Person p = new Person("Alice", 30);  
        System.out.println(p.toString());  
    }  
}
```

- ☐ Person
- ☐ Person@15db9742
- ☐ Person{name='Alice', age=30}
- ☐ Alice, 30

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
class Book {
    String title;
    String author;

    public Book(String title, String author) {
        this.title = title;
        this.author = author;
    }

    @Override
    public String toString() {
        return title + " by " + author;
    }

    public static void main(String[] args) {
        Book b = new Book("Java Programming", "John Doe");
        System.out.println(b);
    }
}
```

- ☐ Book
- ☐ Book@6d06d69c
- ☐ Java Programming by John Doe
- ☐ John Doe

Υποβολή

Εκκαθάριση φόρμας

Κεφάλαιο 11: Δημιουργία Αντικειμένων

11. Δημιουργία Αντικειμένων

Εισαγωγή στην Δημιουργία Αντικειμένων

[Συνδεθείτε στο Google](#), για να αποθηκεύσετε την πρόοδό σου. [Μάθετε περισσότερα](#)

* Υποδεικνύει απαιτούμενη ερώτηση

Ποιο είδος δεδομένων χρησιμοποιείται για τη δημιουργία ενός αντικειμένου σε Java;

* 1 βαθμός

- ☐ int
- ☐ object
- ☐ String
- ☐ double

Ποια είναι η σωστή σύνταξη για τη δημιουργία ενός αντικειμένου της κλάσης "Car" στη Java;

* 1 βαθμός

- ☐ Car myCar = new Car();
- ☐ new Car = myCar;
- ☐ Car.create();
- ☐ Car myCar = create();

Τι γίνεται όταν καλείται ο κατασκευαστής (constructor) ενός αντικειμένου σε Java;

* 1 βαθμός

- ☐ Το αντικείμενο καταστρέφεται.
- ☐ Το αντικείμενο αρχικοποιείται.
- ☐ Το αντικείμενο αποθηκεύεται στη μνήμη.
- ☐ Το αντικείμενο γίνεται αθάνατο.

Τι ισχύει για το όνομα ενός κατασκευαστή σε Java; *

1 βαθμός

- ☐ Πρέπει να είναι ίδιο με το όνομα της κλάσης.
- ☐ Πρέπει να ξεκινά με κεφαλαίο γράμμα.
- ☐ Μπορεί να είναι οποιοδήποτε όνομα.
- ☐ Πρέπει να είναι μικρότερο από 5 χαρακτήρες.

Τι είναι το "this" στη Java; *

1 βαθμός

- ☐ Ένα αντικείμενο κλάσης.
- ☐ Ένας τελεστής που χρησιμοποιείται για σύγκριση αντικειμένων.
- ☐ Ένα σύμβολο που αναφέρεται στην τρέχουσα κλάση ή αντικείμενο.
- ☐ Ένας κατασκευαστής που χρησιμοποιείται για τη δημιουργία νέων αντικειμένων.

Πώς μπορείτε να καλέσετε τον κατασκευαστή ενός υπερκλάσης (superclass) στη Java;

* 1 βαθμός

- ☐ superclass();
- ☐ super();
- ☐ callSuperclass();
- ☐ this();

Ποιος κατασκευαστής καλείται αν δεν δηλωθεί κανένας κατασκευαστής σε * 1 βαθμός
για μια κλάση Java;

- ☐ Ο προκαθορισμένος (default) κατασκευαστής.
- ☐ Το πρόγραμμα δεν μεταγλωττίζεται.
- ☐ Κανένας κατασκευαστής δεν καλείται αυτόματα.
- ☐ Το πρόγραμμα τερματίζει.

Ποια είναι η διαφορά μεταξύ μεθόδου και κατασκευαστή σε μια κλάση Java; * 1 βαθμός

- ☐ Καμία διαφορά, είναι συνώνυμα.
- ☐ Οι μέθοδοι επιστρέφουν τιμές, ενώ οι κατασκευαστές δεν.
- ☐ Οι κατασκευαστές δημιουργούν νέα αντικείμενα, ενώ οι μέθοδοι δεν το κάνουν.
- ☐ Οι μέθοδοι καλούνται με το "new", ενώ οι κατασκευαστές όχι.

Ποια είναι η διαφορά μεταξύ του "this" και του "super" στη Java; * 1 βαθμός

- ☐ a) Το "this" αναφέρεται στο τρέχον αντικείμενο ενώ το "super" αναφέρεται στο αντικείμενο της υπερκλάσης.
- ☐ b) Το "this" αναφέρεται στο αντικείμενο της υπερκλάσης ενώ το "super" αναφέρεται στο τρέχον αντικείμενο.
- ☐ c) Το "this" χρησιμοποιείται για να καλεί τον κατασκευαστή της υπερκλάσης, ενώ το "super" χρησιμοποιείται για να αναφέρεται στο τρέχον αντικείμενο.
- ☐ d) Το "this" και το "super" είναι αντίστοιχα και μπορούν να χρησιμοποιηθούν ανταλλάξιμα σε κάθε περίπτωση.

Ποιο είναι το κύριο πλεονέκτημα της υπερφόρτωσης (function overloading) στη Java; * 1 βαθμός

- ☐ Επιτρέπει τη δημιουργία νέων αντικειμένων.
- ☐ Επιτρέπει την κλήση μιας μεθόδου με διαφορετικά ορίσματα.
- ☐ Μειώνει την πολυπλοκότητα του κώδικα.
- ☐ Αυξάνει το μέγεθος της μνήμης που χρησιμοποιείται.

Ποια δήλωση χρησιμοποιείται για τη δημιουργία νέου αντικειμένου στη Java; * 1 βαθμός

- ☐ create
- ☐ generate
- ☐ initialize
- ☐ new

Ποια είναι η λειτουργία του "new" operator στη Java; *

1 βαθμός

- ☐ Αποθηκεύει το αντικείμενο στη μνήμη.
- ☐ Διαγράφει το αντικείμενο από τη μνήμη.
- ☐ Επιστρέφει την τιμή του αντικειμένου.
- ☐ Καλεί τον constructor της κλάσης για τη δημιουργία του αντικειμένου.

Ποιος κατασκευαστής εκτελείται όταν χρησιμοποιείται η δήλωση "new" χωρίς παραμέτρους σε μια κλάση;

* 1 βαθμός

- ☐ Ο κατασκευαστής με τα περισσότερα ορίσματα.
- ☐ Ο κατασκευαστής που έχει οριστεί ρητά με παραμέτρους.
- ☐ Ο προκαθορισμένος (default) κατασκευαστής.
- ☐ Κανένας κατασκευαστής δεν εκτελείται.

Ποια είναι η σωστή δήλωση για τη δημιουργία ενός νέου αντικειμένου της κλάσης "Book" με τίτλο "Java Programming" και συγγραφέα "John Doe"; *

1 βαθμός

- ☐ Book book = new Book("Java Programming", "John Doe");
- ☐ Book book = create Book("Java Programming", "John Doe");
- ☐ Book book = new Book();
- ☐ Book book = Book.create("Java Programming", "John Doe");

Ποια δήλωση επιτρέπει στην κύρια μέθοδο (main) να καλείται απευθείας χωρίς τη δημιουργία ενός αντικειμένου;

* 1 βαθμός

- ☐ static
- ☐ non-static
- ☐ encapsulation
- ☐ this

Ποια λέξη-κλειδί δηλώνει ένα μέλος κλάσης ως σταθερό (constant) στο Java;

* 1 βαθμός

- ☐ static
- ☐ const
- ☐ final
- ☐ constant

Ποια δήλωση ορίζει ένα μέλος κλάσης ως προσβάσιμο μόνο από την ίδια * 1 βαθμός
την κλάση;

- ☐ public
- ☐ private
- ☐ protected
- ☐ static

Τι σημαίνει όταν ένα μέλος κλάσης είναι δηλωμένο ως static; * 1 βαθμός

- ☐ Ανήκει στην κλάση αντί για ένα συγκεκριμένο αντικείμενο
- ☐ Μπορεί να τροποποιηθεί από κάθε αντικείμενο της κλάσης
- ☐ Απαιτείται η ύπαρξη ενός αντικειμένου για να προσπελαστεί
- ☐ Είναι προσβάσιμο μόνο από υποκλάσεις της κύριας κλάσης

Τι γίνεται όταν ένα μέλος κλάσης είναι δηλωμένο ως non-static; * 1 βαθμός

- ☐ Απαιτείται η ύπαρξη ενός αντικειμένου για να προσπελαστεί
- ☐ Μπορεί να τροποποιηθεί από κάθε αντικείμενο της κλάσης
- ☐ Είναι προσβάσιμο μόνο από υποκλάσεις της κύριας κλάσης
- ☐ Ανήκει στην κλάση αντί για ένα συγκεκριμένο αντικείμενο

Ποια δήλωση διασφαλίζει ότι μια μεταβλητή της κλάσης αναφέρεται στο * 1 βαθμός
τρέχον αντικείμενο;

- ☐ static
- ☐ non-static
- ☐ this
- ☐ super

Ποια είναι η βασική αρχή της ενθυλάκωσης (encapsulation); * 1 βαθμός

- ☐ Η κλάση πρέπει να είναι κλειστή για τις τροποποιήσεις
- ☐ Η πρόσβαση στα μέλη της κλάσης πρέπει να γίνεται μόνο μέσω μεθόδων
- ☐ Τα μέλη της κλάσης πρέπει να είναι public
- ☐ Η κλάση πρέπει να περιλαμβάνει στατικά μέλη μόνο

Ποιο από τα παρακάτω είναι ένα λάθος στον παρακάτω κώδικα; *

1 βαθμός

`MyClass obj = new MyClass;`

- ☐ Η δήλωση του αντικειμένου είναι σωστή
- ☐ Το είδος δεδομένων του αντικειμένου είναι λανθασμένο
- ☐ Χρειάζεται να χρησιμοποιηθεί η λέξη-κλειδί "create" αντί για "new"
- ☐ Λείπει η παρένθεση μετά την κλάση "MyClass"

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
class Dog {  
    String breed;  
    int age;  
}  
  
public class MyClass {  
    public static void main(String[] args) {  
        Dog myDog;  
        System.out.println(myDog.breed);  
    }  
}
```

- ☐ Δεν έχει δημιουργηθεί αντικείμενο της κλάσης Dog
- ☐ Το πεδίο "breed" δεν έχει αρχικοποιηθεί
- ☐ Η κλάση Dog δεν έχει constructor
- ☐ Τίποτα, ο κώδικας είναι σωστός

Ποιο από τα παρακάτω είναι ένα λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
class Circle {  
    double radius;  
    Circle(double r) {  
        radius = r;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Circle myCircle = Circle(5.0);  
        System.out.println(myCircle.radius);  
    }  
}
```

- ☐ Λείπει η δήλωση του τύπου δεδομένων στην δημιουργία του αντικειμένου
- ☐ Λείπει η λέξη-κλειδί "new" στην δημιουργία του αντικειμένου
- ☐ Χρειάζεται να χρησιμοποιηθεί η λέξη-κλειδί "create" αντί για "new"
- ☐ Λείπει η παρένθεση μετά το "Circle" στην δημιουργία του αντικειμένου

Ποιο είναι το αποτέλεσμα της εξής εντολής; *

1 βαθμός

```
String str = new String("Hello");  
System.out.println(str == "Hello");
```

- ☐ Τυπώνει true
- ☐ Τυπώνει false
- ☐ Προκαλεί σφάλμα εκτέλεσης
- ☐ Είναι μη ορισμένο

Τι θα εκτυπώσει ο παρακάτω κώδικας; *

1 βαθμός

```
public class Test {  
    int x = 10;  
    public static void main(String[] args) {  
        Test myObj = new Test();  
        System.out.println(++myObj.x);  
    }  
}
```

- ☐ 10
- ☐ 11
- ☐ 12
- ☐ Σφάλμα μεταγλώττισης

Ποιο από τα παρακάτω είναι λάθος στον παρακάτω κώδικα; *

1 βαθμός

```
public class MyClass {  
    int x;  
    public static void main(String[] args) {  
        MyClass myObj = new MyClass();  
        System.out.println(myObj.y);  
    }  
}
```

- ☐ Η μεταβλητή "y" δεν έχει αρχικοποιηθεί
- ☐ Ο κώδικας είναι σωστός
- ☐ Η μεταβλητή "y" δεν έχει οριστεί
- ☐ Ο κώδικας δεν μπορεί να μεταγλωττιστεί

Υποβολή

Εκκαθάριση φόρμας