

Ατομική Διπλωματική Εργασία

**ΣΥΣΤΗΜΑ ΥΠΟΣΤΗΡΙΞΗΣ ΦΟΙΤΗΤΩΝ ΓΙΑ ΤΟ ΜΑΘΗΜΑ
ΕΠΑ 131 ΑΡΧΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ**

Κακουλλή Μαρία

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σύστημα υποστήριξης φοιτητών για το μάθημα ΕΠΛ 131 Αρχές Προγραμματισμού

Κακουλλή Μαρία

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 2019

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας, θα ήταν παράλειψή μου να μην ευχαριστήσω όλους όσους συνέβαλαν τόσο σ' αυτήν την εργασία, όσο και στην ολοκλήρωση των σπουδών μου.

Αρχικά, το μεγαλύτερο ευχαριστώ οφείλω στην κυρία Ελπίδα Κεραυνού – Παπαηλιού, που παρ' όλες τις ευθύνες ως Πρόεδρος του Τμήματος Πληροφορικής του Πανεπιστήμιου Κύπρου, έχει αναλάβει και την ευθύνη να είναι η επιβλέπων καθηγήτρια της διπλωματικής μου εργασίας. Με την βοήθεια της, με την αμέριστη συμπαράστασή της, με σωστό προγραμματισμό και σκληρή δουλειά καταφέραμε να φέρουμε εις πέρας αυτή την εργασία.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω και το υπόλοιπο ακαδημαϊκό προσωπικό για τις γνώσεις που μου παρείχε αυτά τα τέσσερα χρόνια στον κλάδο της Πληροφορικής, και που με προετοίμασε κατάλληλα ώστε να μπορώ να αντεπεξέλθω στις προκλήσεις της μετέπειτα καριέρας μου.

Ταυτόχρονα, θα ήθελα να πω ένα μεγάλο ευχαριστώ στην οικογένεια μου για την συνεχή ενθάρρυνση, την υποστήριξη, την κατανόηση και την οικονομική στήριξη που μου δείχνουν όλα αυτά τα 22 χρόνια της ζωής μου. Χαίρονταν με τις επιτυχίες μου, αλλά πιο πολύ με γέμιζαν πίστη να φτάσω στους στόχους μου όταν συναντούσα εμπόδια.

Περίληψη

Ο στόχος της διπλωματικής μου εργασίας είναι η ανάπτυξη εκπαιδευτικού υλικού για το μάθημα ΕΠΛ 131 Αρχές Προγραμματισμού. Το μάθημα αυτό είναι υποχρεωτικό με βάση το πρόγραμμα παρακολούθησης μαθημάτων του Τμήματος και διδάσκεται το πρώτο εξάμηνο εισδοχής του φοιτητή στο Πανεπιστήμιο. Σκοπός του μαθήματος αυτού είναι η ανάπτυξη δεξιοτήτων του φοιτητή στην επίλυση προβλημάτων, με αλγοριθμικό τρόπο, και η θεμελίωση της αλγοριθμικής σκέψης, τόσο σε επίπεδο αρχών προγραμματισμού, όσο και σε επίπεδο μιας υψηλής επιπέδου γλώσσας προγραμματισμού, την Java.

Τελειώνοντας αυτό το μάθημα, ο φοιτητής θα είναι σε θέση να γνωρίζει τις θεμελιώδεις έννοιες του δομημένου προγραμματισμού, τις βασικές αλγοριθμικές δομές που μπορούν να χρησιμοποιηθούν στην «κατασκευή» προγραμμάτων, να διακρίνει τις βασικές αρχές προγραμματισμού, αλγοριθμικές τεχνικές και δομές προγραμμάτων. Να αναγνωρίζει τις θεμελιώδεις αρχές της επαναχρησιμοποίησης, άρθρωσης και ιεραρχικής δόμησης, να επιλύει προβλήματα με χρήση διαδικαστικού και αντικειμενοστραφούς μοντέλου προγραμματισμού, να σχεδιάζει, να υλοποιεί, και να δοκιμάζει με οργανωμένο και συστηματικό τρόπο, να αποσφαλματώνει και να τεκμηριώνει αρθρωτά προγράμματα, εφαρμόζοντας τη μεθοδολογία της σταδιακής ή προοδευτικής εκλέπτυνσης. Επίσης, θα πρέπει να είναι σε θέση να επιχειρηματολογεί αναφορικά με την ορθότητα ενός προγράμματος, να αξιολογεί βάσει ποιοτικών και άλλων κριτηρίων εναλλακτικές (αλγοριθμικές) λύσεις για το ίδιο πρόβλημα, να αξιολογεί τη δυνατότητα επέκτασης (scaling up) ενός προγράμματος, με στόχο τη διεύρυνση των προβλημάτων που μπορεί να καλύψει, να επιδεικνύει την ικανότητα αλγοριθμικής σκέψης, ανεξαρτήτως οποιασδήποτε γλώσσας προγραμματισμού, και τέλος να επιδεικνύει ικανότητα στη χρήση της υψηλού επιπέδου γλώσσας προγραμματισμού Java.

Η διπλωματική αυτή εργασία εκπονείται για να βοηθήσει τους φοιτητές που παρακολουθούν το μάθημα ΕΠΛ131 ώστε να κατανοήσουν καλύτερα τις διάφορες έννοιες και τεχνικές του αντικειμενοστραφούς προγραμματισμού μέσα από διαλέξεις, παραδείγματα, βίντεο και παιχνίδια.

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	4
Κεφάλαιο 2	Συστήματα Εκπαίδευσης και Διδασκαλίας.....	7
	2.1 Εισαγωγή	7
	2.2 Μαθησιακό Περιβάλλον	8
	2.3 Εκμάθηση Μέσω Διαδικτύου	8
	2.4 Αφαιρετικότητα στον Προγραμματισμό	9
	2.5 Εκπαιδευτικά εξατομικευμένα συστήματα για το μάθημα Αρχές Προγραμματισμού I	10
Κεφάλαιο 3	Απαιτήσεις Συστήματος.....	10
	3.1 Εισαγωγή	10
	3.2 Συλλογή Απαιτήσεων	10
	3.2.1 Ερωματολόγιο	11
	3.2.2 Συμπεράσματα Ερωματολογίου	18
	3.3 Γενική Περιγραφή Συστήματος	19
	3.3.1 Ερωματολόγιο	19
	3.3.2 Προοπτική του συστήματος	20
	3.3.2.1 Διαπροσωπείες Συστήματος	20
	3.3.2.2 Διαπροσωπείες Χρηστών	21
	3.3.2.3 Διαπροσωπείες Υλικού (Hardware)	21
	3.3.2.4 Διαπροσωπείες Λογισμικού (Software)	22
	3.3.2.5 Διαπροσωπείες Επικοινωνίας	23
	3.3.2.6 Περιορισμοί μνήμης	23
	3.3.2.7 Απαιτήσεις Προσαρμογής Περιοχών	23
	3.3.3 Λειτουργίες Προϊόντος	24
	3.3.4 Χαρακτηριστικά Χρηστών	24

3.3.5 Περιορισμοί	24
3.3.6 Υποθέσεις και εξαρτήσεις	24
3.3.7 Διανομή απαιτήσεων	25
3.4 Συγκεκριμένες Απαιτήσεις	25
3.4.1 Εξωτερικές Απαιτήσεις Διεπαφών	25
3.4.1.1 Διεπαφές Χρηστών	25
3.4.1.2 Διεπαφές Υλικού (Hardware)	25
3.4.1.3 Διεπαφές Λογισμικού (Software)	25
3.4.1.4 Διεπαφές Επικοινωνίας	25
3.4.1.5 Διεπαφές Συστήματος	26
3.4.1.6 Απαιτήσεις Απόδοσης	26
3.4.1.7 Περιορισμοί Σχεδιασμού	26
3.5 Ιδιότητες Συστήματος Λογισμικού	26
3.5.1 Αξιοπιστία (reliability)	26
3.5.2 Διαθεσιμότητα	27
3.5.3 Ερωτηματολογίου	27
3.5.4 Ευελιξία – Συντηρησιμότητα (maintainability)	27
3.5.5 Φορητότητα (portability)	28
3.6 Λογικές Απαιτήσεις Βάσης Δεδομένων	28
Κεφάλαιο 4 Προδιαγραφές Συστήματος.....	29
4.1 Εισαγωγή	29
4.1.1 Σκοπός	29
4.2 Διάγραμμα Ροής Δεδομένων (Data Flow Diagram – DFD)	30
4.2.1 Εισαγωγή στο Διάγραμμα Ροής Δεδομένων (Data Flow Diagram – DFD)	30
4.2.2 Διάγραμμα Ροής Δεδομένων – Επίπεδο 0	31
4.2.3 Διάγραμμα Ροής Δεδομένων – Επίπεδο 1	32
4.3 Αντικειμενοστραφής Ανάλυση	34
4.3.1 Use Case Diagrams (Διαγράμματα Σεναρίου Χρήσης)	34
4.3.1.1 Διάγραμμα Σεναρίου Φοιτητή	36
4.3.1.2 Διάγραμμα Σεναρίου Διαχειριστή	37
4.3.2 Entity Relationship Diagram	39

Κεφάλαιο 5	Σχεδίαση Συστήματος.....	40
5.1	Εισαγωγή	40
5.1.1	Σκοπός	40
5.1.2	Αποδέκτες	40
5.2	Αντικειμενοστραφής σχεδίαση	41
5.2.1	Διαγράμματα Αλληλεπίδρασης (Interaction Diagrams)	41
5.2.1.1	Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για επιτυχή είσοδο φοιτητή στο σύστημα	44
5.2.1.2	Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για αποτυχή είσοδο φοιτητή στο σύστημα	45
5.2.1.3	Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για επιτυχή είσοδο διαχειριστή στο σύστημα	46
5.2.1.4	Διάγραμμα Αλληλεπίδρασης (Interaction Diagrams) για αποτυχή είσοδο διαχειριστή στο σύστημα	47
 ΚΕΦΑΛΑΙΟ 6	 Υλοποίηση Συστήματος.....	 48
6.1	Εισαγωγή	48
6.2	Παρουσίαση εργαλείων, γλωσσών και βοηθητικών προγραμμάτων που χρησιμοποιήθηκαν	49
6.2.1	Java	49
6.2.2	Hyper Text Markup Language – HTML	50
6.2.3	Cascading Styling Sheet - CSS	50
6.2.4	JavaScript	51
6.2.5	JSP	51
6.2.6	Bootstrap	51
6.2.7	MySQL	52
6.2.8	Draw.io	52
6.2.9	PHP Hypertext Preprocessor	52
6.2.10	Eclipse IDE for Java Developers	53
6.2.11	Microsoft Word	53
6.2.12	Φόρμες Google	54
6.2.13	Windows Live Movie Maker	54

6.3 Υλοποίηση Διαδικτυακής Σελίδας	54
6.3.1 Διαδικασία υλοποίησης	54
6.3.2 Προβλήματα που αντιμετωπίστηκαν	55
6.4 Παρουσίαση Συστήματος	57
 ΚΕΦΑΛΑΙΟ 7 Συμπεράσματα	 82
7.1 Εισαγωγή	82
7.2 Συμπεράσματα	82
7.3 Μελλοντική Επέκταση	83
 Βιβλιογραφία	 85
 Παράρτημα Α	 A-86
 Παράρτημα Β	 B-91
 Παράρτημα Γ	 Γ-99
 Παράρτημα Δ	 Δ-103

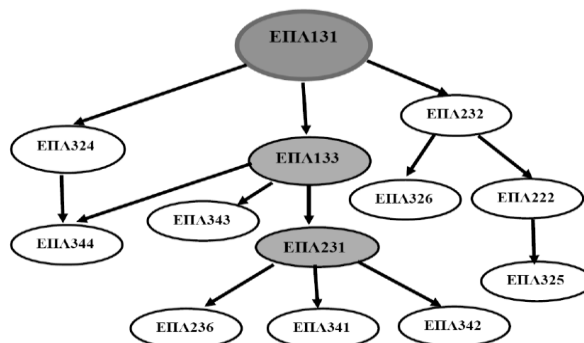
Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Εισαγωγή	1
1.2 Στόχος Διπλωματικής Εργασίας	2
1.3 Εργαλεία Ανάπτυξης	2
1.4 Δομή Διπλωματικής Εργασίας	4

1.1 Γενική Εισαγωγή

Όπως όλοι γνωρίζουμε, ο προγραμματισμός αποτελεί αναπόσπαστο κομμάτι της επιστήμης της Πληροφορικής. Το μάθημα ΕΠΑ131, αποτελεί τον ακρογωνιαίο λίθο για την περαιτέρω εξέλιξη της πορείας ενός φοιτητή κατά τη διάρκεια των σπουδών του, αφού σ' αυτό το μάθημα θα κατανοήσει τις βασικές αρχές προγραμματισμού, κάνοντας χρήση της γλώσσας προγραμματισμού Java, καθώς επίσης θα αντιληφθεί βασικές έννοιες σχετικά με την αρχιτεκτονική και τη λειτουργία των ηλεκτρονικών υπολογιστών. Λόγω αυτού, το συγκεκριμένο μάθημα, αποτελεί τον αρχικό κόμβο σε οκτώ αλυσίδες υποχρεωτικών μαθημάτων. Οι εξαρτήσεις αυτού του μαθήματος με άλλα υποχρεωτικά μαθήματα του προγράμματος σπουδών στην Πληροφορική, δίνονται στο σχεδιάγραμμα που ακολουθεί στο σχήμα 1. [1]



Σχήμα 1: Αλυσίδες υποχρεωτικών μαθημάτων που έχουν το μάθημα ΕΠΑ 131 ως αρχικό κόμβο

1.2 Στόχος Διπλωματικής Εργασίας

Στόχος αυτής της διπλωματικής εργασίας είναι η ανάπτυξη εκπαιδευτικού υλικού για το μάθημα ΕΠΛ131 Αρχές Προγραμματισμού, το οποίο προσφέρεται ως υποχρεωτικό κατά το πρώτο εξάμηνο εισδοχής του φοιτητή στο Πανεπιστήμιο Κύπρου. Στο εκπαιδευτικό αυτό υλικό, για κάθε κεφάλαιο, περιλαμβάνονται τόσο σημειώσεις με παραδείγματα και βίντεο για καλύτερη κατανόηση της ύλης του μαθήματος, καθώς επίσης και διάφορα παιχνίδια για καλύτερη εμπέδωση και εφαρμογή των εννοιών και τεχνικών που αναφέρονται στο συγκεκριμένο κεφάλαιο.

Πιο αναλυτικά, η πλατφόρμα για κάθε ενότητα περιέχει [1]:

- Σημειώσεις επεξήγησης της ύλης μαζί με παραδείγματα και μικρά βίντεο
- Παιχνίδι με ερωτήσεις πολλαπλής επιλογής μαζί με όσα επίπεδα βοήθειας χρειάζονται για κάθε ερώτηση, και εμφάνιση λύσης με επεξήγηση όταν ο χρήστης απαντήσει λάθος
- Παιχνίδι συμπλήρωσης μικρών τμημάτων κώδικα, βάσει κάποιου προβλήματος. Το παιχνίδι αυτό είναι υλοποιημένο με αύξουσα βαθμίδα δυσκολίας, καθώς στα πρώτα προβλήματα ζητούνται πολύ απλές – βασικές γνώσεις που στη συνέχεια εμβαθύνουν ακόμη περισσότερο. Υπάρχει η δυνατότητα καθαρισμού κώδικα σε περίπτωση που ο χρήστης επιθυμεί να επιστρέψει στον αρχικά δοσμένο κώδικα. Επίσης, ο χρήστης έχει την επιλογή να παραβλέψει το συγκεκριμένο πρόβλημα και να μετακινηθεί στο επόμενο ή προηγούμενο πρόβλημα. Τέλος, μπορεί να δει και τη λύση του προβλήματος
- Παιχνίδι με κώδικες πολλαπλής επιλογής, στο οποίο ο χρήστης, δοθέντος κάποιου αποτελέσματος, θα πρέπει να βρει ποιος από τους κώδικες δίνει το συγκεκριμένο αποτέλεσμα

1.3 Εργαλεία Ανάπτυξης

Τα εργαλεία τα οποία χρησιμοποιήθηκαν για την ανάπτυξη της πλατφόρμας, είναι τα ακόλουθα [3]:

- HTML: είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και να τα συνθέτει σε σελίδες που μπορεί κανείς να

διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας

- JSP: είναι μια τεχνολογία που βοηθά τους προγραμματιστές λογισμικού να δημιουργούν δυναμικά δημιουργούμενες ιστοσελίδες με βάση HTML, XML ή άλλους τύπους εγγράφων και χρησιμοποιεί τη γλώσσα προγραμματισμού Java
- CSS: χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστότοπου
- Bootstrap: είναι μια συλλογή εργαλείων ανοιχτού κώδικα (ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript
- JavaScript: είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα, είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η επεξεργασία του κώδικα JavaScript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML
- MySQL: είναι μια γλώσσα για το συντονισμό της βάσης δεδομένων
- PHP Hypertext Preprocessor: είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή (server) του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο
- Eclipse JEE: περιέχει εργαλεία για προγραμματιστές Java που δημιουργούν εφαρμογές Java EE και Web, συμπεριλαμβανομένου ενός Java IDE
- Draw.io: πρόγραμμα για τη δημιουργία διαγραμμάτων
- Microsoft Word: χρησιμοποιήθηκε για την εγγραφή του εγγράφου της ατομικής διπλωματικής εργασίας
- Φόρμες Google: χρησιμοποιήθηκε για τη δημιουργία ερωτηματολογίου
- Windows Live Movie Maker: χρησιμοποιήθηκε για την επεξεργασία των βίντεο

Όλα τα εργαλεία θα αναλυθούν με περισσότερη λεπτομέρεια στη συνέχεια.

1.4 Δομή Διπλωματικής Εργασίας

Η δομή της παρούσας Διπλωματικής Εργασίας έχει ως εξής:

Κεφάλαιο 2: Συστήματα Εκπαίδευσης και Διδασκαλίας

Σε αυτό το κεφάλαιο γίνεται αναφορά στο μαθησιακό περιβάλλον, στην εκμάθηση μέσω διαδικτύου, στην αφαιρετικότητα στον προγραμματισμό και στα εκπαιδευτικά εξατομικευμένα συστήματα για Αρχές Προγραμματισμού Ι.

Κεφάλαιο 3: Απαιτήσεις Συστήματος

Σε αυτό το κεφάλαιο γίνεται μια ανάλυση της κατάστασης για να βρεθούν οι πραγματικές ανάγκες – απαιτήσεις του συστήματος. Το στάδιο της ανάλυσης αυτής αφορά την συλλογή πληροφοριών, κυρίως από έρευνα μέσα από ερωτηματολόγιο, την επεξεργασία των δεδομένων και την εξαγωγή συμπερασμάτων.

Κεφάλαιο 4: Προδιαγραφές Συστήματος

Σε αυτό το κεφάλαιο γίνεται μια ανάλυση των προδιαγραφών του συστήματος και ακολούθως των λειτουργιών που θα περιλαμβάνονται στο σύστημα. Η ανάλυση των προδιαγραφών έγινε μέσα από την χρήση διαγραμμάτων Ροής Δεδομένων, διαφόρων test cases καθώς επίσης και διαγραμμάτων ERD.

Κεφάλαιο 5: Σχεδίαση Συστήματος

Σε αυτό το κεφάλαιο αφού καθορίσουμε τις απαιτήσεις και τις προδιαγραφές του συστήματος, σχεδιάζουμε τις διάφορες λειτουργίες που θα περιλαμβάνονται στο σύστημα. Σε αυτό το κεφάλαιο, οι προδιαγραφές του συστήματος και οι απαιτήσεις, θα καθοριστούν με διαγράμματα αλληλεπίδρασης για σενάρια διαφόρων περιπτώσεων.

Κεφάλαιο 6: Υλοποίηση Συστήματος

Σε αυτό το κεφάλαιο περιγράφεται η διαδικασία υλοποίησης του συστήματος, καθώς επίσης και οι διάφορες τεχνολογίες και εργαλεία που χρησιμοποιήθηκαν. Στο τέλος, παρουσιάζονται κάποια screenshots που αφορούν την μορφή και την παρουσίαση του συστήματος – πλατφόρμας, καθώς επίσης μέσω των διαφόρων εικόνων και επεξηγηματικών σχολίων, περιγράφονται και οι διάφορες λειτουργίες του.

Κεφάλαιο 7: Συμπεράσματα

Σε αυτό το κεφάλαιο δίνεται ο επίλογος και προτείνονται κάποιες εισηγήσεις για μελλοντική εργασία.

Παράρτημα Α:

Το παράρτημα Α περιλαμβάνει το ερωτηματολόγιο που δόθηκε προς τους φοιτητές.

Παράρτημα Β:

Το παράρτημα Β περιλαμβάνει μερικές ερωτήσεις πολλαπλής επιλογής, την λύση και την απάντηση κάθε ερώτησης.

Παράρτημα Γ:

Το παράρτημα Γ περιλαμβάνει το αποτέλεσμα και τους πιθανούς κώδικες που έδωσαν το συγκεκριμένο αποτέλεσμα, καθώς επίσης και την λύση.

Παράρτημα Δ:

Το παράρτημα Δ περιλαμβάνει μερικά προβλήματα και τμήματα κώδικα για συμπλήρωση μέσω της Java, καθώς επίσης και την λύση του προβλήματος.

Κεφάλαιο 2

Συστήματα Εκπαίδευσης και Διδασκαλίας

2.1 Εισαγωγή	6
2.2 Μαθησιακό Περιβάλλον	7
2.3 Εκμάθηση Μέσω Διαδικτύου	7
2.4 Αφαιρετικότητα στον Προγραμματισμό	8
2.5 Εκπαιδευτικά εξατομικευμένα συστήματα για το μάθημα Αρχές Προγραμματισμού I	8

2.1 Εισαγωγή

Στις μέρες μας, παρατηρείται το φαινόμενο ότι ένα ποσοστό φοιτητών αντιμετωπίζει δυσκολίες για να κατανοήσει τις βασικές αρχές Προγραμματισμού, σε σχέση με κάποιους άλλους φοιτητές, οι οποίοι αφομοιώνουν πιο εύκολα γνώσεις σε θέματα προγραμματισμού. Αυτό οφείλεται στο ότι κάποιοι φοιτητές είναι πιο έξυπνοι από τους άλλους; Είναι δυνατό αυτοί οι φοιτητές, οι οποίοι δυσκολεύονται στον προγραμματισμό, να αποκτήσουν το ίδιο επίπεδο γνώσεων μέσω κάποιας εκπαίδευσης; Τα συστήματα εκπαίδευσης και διδασκαλίας, τα οποία θα περιγραφούν σε μετέπειτα στάδιο, έχουν σκοπό να βοηθήσουν αυτούς τους φοιτητές ώστε να καταφέρουν να κατανοήσουν αυτές τις βασικές αρχές και μέσω της πρακτικής, να φτάσουν στο ίδιο επίπεδο με τους υπόλοιπους φοιτητές. Επιπλέον, έχοντας αφομοιώσει, κατανοήσει και εφαρμόσει τις προγραμματιστικές γνώσεις που παρέχει το μάθημα αυτό, θα μπορέσει να αντεπεξέλθει και στα πιο δύσκολα μαθήματα προγραμματισμού που ακολουθούν και εμβαθύνουν περισσότερο. Συνεπώς, προτού περιγράψουμε λεπτομερώς ποια είναι τα συστήματα αυτά και πως μπορούν να βοηθήσουν το φοιτητή, πρέπει πρώτα να αναφερθούμε σε κάποιες πιο γενικές πληροφορίες που σχετίζονται με τα συστήματα αυτά [2].

2.2 Μαθησιακό Περιβάλλον

Για να επιτευχθούν οι στόχοι του διδάσκοντα, δηλαδή ο εκπαιδευόμενος να πάρει τις απαραίτητες γνώσεις του μαθήματος που διδάσκεται, η ανάπτυξη περιβαλλόντων μάθησης είναι εξαιρετικά σημαντική. Πιο συγκεκριμένα, η ανάπτυξη του περιβάλλοντος αυτού, θα στηρίζεται στη δημιουργία και ανάπτυξη νέων και καινοτόμων προηγμένων συστημάτων, αξιοποιώντας όλα τα διαθέσιμα τεχνολογικά μέσα, τις μεθόδους μετάδοσης της πληροφορίας μέσω υπολογιστικών συστημάτων και την αλληλεπίδραση του χρήστη με το σύστημα διαθέτοντας όλη την απαραίτητη γνώση με σκοπό να επιτευχθεί ο στόχος του συστήματος αυτού, δηλαδή την εκπαίδευση του φοιτητή. Η ποιότητα και η λειτουργικότητα που θα παρέχεται από το περιβάλλον θα πρέπει να κρατούν το ενδιαφέρον του εκπαιδευόμενου, να τον κρατούν σε εγρήγορση, να του παρέχουν ευχάριστο περιβάλλον εκμάθησης, καθώς επίσης να στοχεύουν στην μετάδοση της γνώσης, όχι απλά της πληροφορίας.

2.3 Εκμάθηση Μέσω Διαδικτύου

Οι τεχνολογίες διαδικτύου και ο παγκόσμιος ιστός έχει γίνει μέρος της καθημερινότητας μας. Συνεπώς, το μαθησιακό σύστημα εκπαίδευσης στο διαδίκτυο είναι επιτακτική ανάγκη. Μέσω του Παγκόσμιου Ιστού (WWW) μπορούμε να ικανοποιήσουμε επαρκώς τους στόχους του συστήματος, εφ' όσον οι διάφορες μορφές εκπαίδευσης – πολυμέσα (για παράδειγμα υλικό διδασκαλίας, βίντεο, κλπ) μπορούν να ζητηθούν από το χρήστη (Get request) μέσω του HTTP πρωτοκόλλου. Οι εκπαιδευτικοί μπορούν εύκολα και γρήγορα να παράξουν ή να αναβαθμίσουν το υλικό εκπαίδευσης από απόσταση, χρησιμοποιώντας HTML, PHP, JavaScript, CSS αρχεία. Πιο συνοπτικά, οι πόροι του συστήματος θα βρίσκονται σε HTTP εξυπηρετητές (servers) που θα έχουν το ρόλο του εκπαιδευτικού, ενώ οι φοιτητές θα έχουν πρόσβαση στους πόρους αυτούς μέσω του διαδικτύου. Επίσης, ο εκπαιδευτικός μπορεί να κρίνει αν οι μέθοδοι και οι πόροι εκπαίδευσης είναι αποδοτικοί, παρέχοντας την απαραίτητη γνώση στους φοιτητές, ενώ παράλληλα έχει τη δυνατότητα να κάνει οποιεσδήποτε τροποποιήσεις στο σύστημα ή να προσθέσει νέα πληροφορία [3].

2.4 Αφαιρετικότητα στον Προγραμματισμό

Στα συστήματα εκπαίδευσης που αφορούν τον Προγραμματισμό, μια σημαντική δεξιότητα που πρέπει να καλλιεργηθεί είναι και η αφαιρετικότητα. Η αφαιρετικότητα, έχει να κάνει με το να κρύβει τις λεπτομέρειες από το χρήστη ώστε να κατανοήσει τη γενική εικόνα, δηλαδή να μπορεί να λύσει πολύπλοκα προβλήματα, χωρίς να τον ενδιαφέρει πως υλοποιείται κάτι. Η δεξιότητα της αφαιρετικότητας είναι απαραίτητη για την κατασκευή μοντέλων, σχεδίων και εφαρμογών που είναι κατάλληλα για ένα συγκεκριμένο σκοπό. Η αφηρημένη σκέψη είναι απαραίτητη για την επίλυση προβλημάτων σχετικά με τον προγραμματισμό. Η γενίκευση της αφαιρετικότητας μπορεί να φανεί σαφώς στον προγραμματισμό, με τη χρήση των αφαιρετικών δεδομένων και των κλάσεων σε αντικειμενοστραφή προγραμματισμό. Η αφηρημένη ερμηνεία για την ανάλυση του προγράμματος είναι ένα άλλο παράδειγμα γενίκευσης, όπου ο συγκεκριμένος τομέας του προγράμματος αναφέρεται σε έναν αφηρημένο τομέα για να συλλάβει τη σημασιολογία του υπολογισμού για την ανάλυση του προγράμματος [4].

2.5 Εκπαιδευτικά εξατομικευμένα συστήματα για το μάθημα Αρχές Προγραμματισμού

Για την ατομική διπλωματική εργασία, επιλέχθηκε σαν εκπαιδευτικό σύστημα μάθησης, μια ηλεκτρονική πλατφόρμα στο διαδίκτυο, η οποία θα παρέχει βασικές γνώσεις για κάποιες συγκεκριμένες ενότητες του μαθήματος αυτού. Οι μέθοδοι εκπαίδευσης και διδασκαλίας που διαθέτει κάθε ενότητα είναι κάποιες βασικές γνώσεις και θεωρίες της εκάστοτε ενότητας μαζί με παραδείγματα και κατατοπιστικά βίντεο, ερωτήσεις πολλαπλής επιλογής με επίπεδα βοήθειας, και εμφάνισης λύσης με μια μικρή επεξήγηση καθώς επίσης και δυνατότητα εγγραφής κώδικα Java. Οι πιο πάνω μέθοδοι εκπαίδευσης θα αναλυθούν με περισσότερη λεπτομέρεια στα επόμενα κεφάλαια.

Κεφάλαιο 3

Απαιτήσεις Συστήματος

3.1 Εισαγωγή	10
3.2 Συλλογή Απαιτήσεων	10
3.2.1 Ερωτηματολόγιο	11
3.2.2 Συμπεράσματα Ερωτηματολογίου	18
3.3 Γενική Περιγραφή Συστήματος	19
3.3.1 Ερωτηματολόγιο	19
3.3.2 Προοπτική του συστήματος	20
3.3.2.1 Διαπροσωπείες Συστήματος	20
3.3.2.2 Διαπροσωπείες Χρηστών	21
3.3.2.3 Διαπροσωπείες Υλικού (Hardware)	21
3.3.2.4 Διαπροσωπείες Λογισμικού (Software)	22
3.3.2.5 Διαπροσωπείες Επικοινωνίας	23
3.3.2.6 Περιορισμοί μνήμης	23
3.3.2.7 Απαιτήσεις Προσαρμογής Περιοχών	23
3.3.3 Λειτουργίες Προϊόντος	24
3.3.4 Χαρακτηριστικά Χρηστών	24
3.3.5 Περιορισμοί	24
3.3.6 Υποθέσεις και εξαρτήσεις	24
3.3.7 Διανομή απαιτήσεων	25
3.4 Συγκεκριμένες Απαιτήσεις	25
3.4.1 Εξωτερικές Απαιτήσεις Διεπαφών	25
3.4.1.1 Διεπαφές Χρηστών	25
3.4.1.2 Διεπαφές Υλικού (Hardware)	25
3.4.1.3 Διεπαφές Λογισμικού (Software)	25
3.4.1.4 Διεπαφές Επικοινωνίας	25
3.4.1.5 Διεπαφές Συστήματος	26
3.4.1.6 Απαιτήσεις Απόδοσης	26
3.4.1.7 Περιορισμοί Σχεδιασμού	26
3.5 Ιδιότητες Συστήματος Λογισμικού	26

3.5.1 Αξιοπιστία (reliability)	26
3.5.2 Διαθεσιμότητα	27
3.5.3 Ερωτηματολογίου	27
3.5.4 Ευελιξία – Συντηρησιμότητα (maintainability)	27
3.5.5 Φορητότητα (portability)	28
3.6 Λογικές Απαιτήσεις Βάσης Δεδομένων	28

3.1 Εισαγωγή

Το κεφάλαιο αυτό, έχει ως κύριο στόχο την καταγραφή των απαιτήσεων του συστήματος μέσω έρευνας και ανάλυσης των στατιστικών στοιχείων για ξεκάθαρη εξακρίβωση των απαιτήσεων. Πιο συγκεκριμένα, το στάδιο της έρευνας έγινε με χρήση κυρίως του ερωτηματολογίου, με σκοπό να γίνουν πιο ξεκάθαρες οι απαιτήσεις – ανάγκες του υπό μελέτη συστήματος. Επίσης, ακόμη ένας λόγος σημαντικότητας αυτού του σταδίου, είναι για να γίνει μια αρχική εκτίμηση των περιορισμών και δυσκολιών που πιθανόν να αντιμετωπίσουμε για την δημιουργία του συστήματος αυτού.

3.2 Συλλογή απαιτήσεων

Οι κύριοι ενδιαφερόμενοι του συστήματος θα είναι οι ίδιοι οι φοιτητές του Πανεπιστημίου Κύπρου του Τμήματος της Πληροφορικής και η έρευνα πραγματοποιήθηκε με τη συμπλήρωση ερωτηματολογίων. Η έρευνα έγινε τόσο σε προπτυχιακούς, όσο και μεταπτυχιακούς φοιτητές, ανεξαρτήτως έτος φοίτησης, για μια πιο ολοκληρωμένη εικόνα.

Ένας άλλος τρόπος συλλογής των απαιτήσεων του συστήματος, πραγματοποιήθηκε μέσω της φυσικής γλώσσας, ρωτώντας τόσο τους διδάσκοντες, όσο και τους φοιτητές για τις ανάγκες και δυσκολίες που υπάρχουν κατά τη διδασκαλία των μαθημάτων προγραμματισμού.

Εκτός από αυτές τις μεθόδους, βοήθησαν και οι δικές μου εμπειρίες ως φοιτήτρια Πληροφορικής. Μέσα από τις δυσκολίες που αντιμετώπισα σε αυτό το μάθημα ως πρωτοετής φοιτήτρια της Πληροφορικής, έδωσα περισσότερη έμφαση σε κεφάλαια τα οποία είχαν ένα επίπεδο δυσκολίας και πιθανόν να είναι δυσνόητα σε κάποιους από τους φοιτητές. Συνεπώς, η δημιουργία μιας τέτοιας πλατφόρμας, θα βοηθήσει αρκετούς φοιτητές οι οποίοι βρίσκονται στην ίδια θέση που βρισκόμουν και εγώ πριν από τέσσερα χρόνια. Η πλατφόρμα

αυτή, θα τους παρέχει την απαραίτητη τριβή και πρακτική με τις βασικές έννοιες σε θέματα προγραμματισμού για την πλήρη κατανόηση και αντίληψη των διάφορων εννοιών.

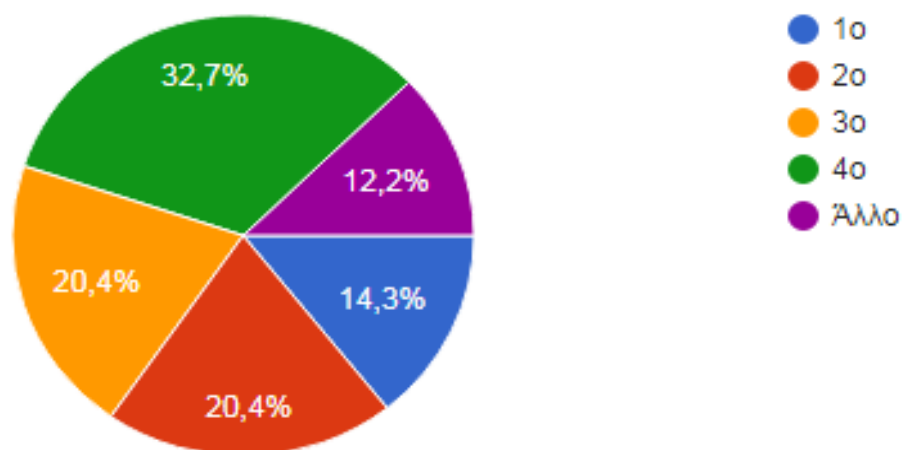
Έχοντας ολοκληρώσει το στάδιο της έρευνας, έχουμε επεξεργαστεί τα δεδομένα από τα αποτελέσματα της έρευνας αυτής, και πήραμε κάποιες γενικές πληροφορίες που αφορούν το σύστημα υπό ανάπτυξη. Αυτές οι πληροφορίες περιέχουν τις λειτουργίες που θα παρέχει το σύστημα, όπως επίσης, θα μας δώσει μια αρχική εικόνα, για τυχόν περιορισμούς και προβλήματα. Οι λειτουργίες και η μορφή του συστήματος θα αναλυθούν σε μετέπειτα στάδια.

3.2.1 Ερωτηματολόγιο

Σκοπός των αποτελεσμάτων του είναι να δείξει την άποψη των φοιτητών – μελλοντικών χρηστών του συστήματος για το πόσο χρήσιμη θα ήταν για αυτούς η ιδέα για υλοποίηση ενός τέτοιου συστήματος. Επίσης, μέσω του ερωτηματολογίου, οι φοιτητές μπορούσαν να εισηγηθούν οτιδήποτε πιστεύουν πως θα ήταν χρήσιμο – βοηθητικό γι' αυτούς. Στην έρευνα αυτή πήραν μέρος 49 φοιτητές της Πληροφορικής από το Πανεπιστήμιο Κύπρου. Ακολουθεί αναπαράσταση των αποτελεσμάτων σε διαγράμματα. Στο παράρτημα Α βρίσκεται το ερωτηματολόγιο που δόθηκε στους φοιτητές.

Έτος φοίτησης

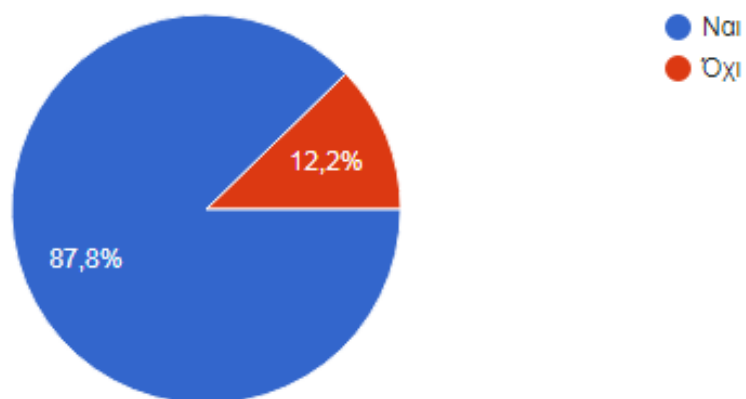
49 απαντήσεις



Σχήμα 3.2.1.1: Διάγραμμα έτος φοίτησης φοιτητή

Πριν αρχίσεις τις σπουδές σου στο Πανεπιστήμιο Κύπρου, είχες παρακολουθήσει κάποιο μάθημα Πληροφορικής;

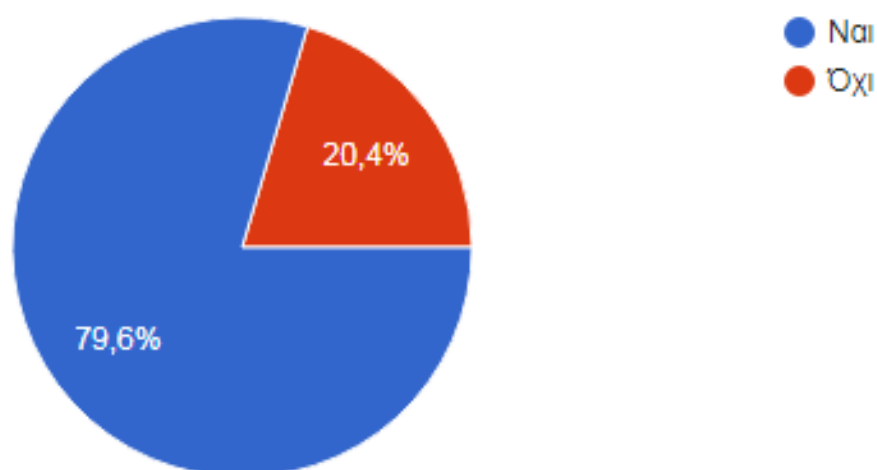
49 απαντήσεις



Σχήμα 3.2.1.2: Διάγραμμα παρακολούθησης μαθήματος Πληροφορικής πριν το Πανεπιστήμιο

Αντιμετώπισες δυσκολίες στο μάθημα ΕΠΛ 131;

49 απαντήσεις



Σχήμα 3.2.1.3: Διάγραμμα αντιμετώπισης στο μάθημα ΕΠΛ 131

Αν Ναι, τι σε δυσκόλεψε περισσότερο;

15 απαντήσεις

αντικειμενοστρέφια
Ηταν το πρωτο μαθημα προγραμματισμου, με πολλές νέες εννοιες
Η προγραμματιστική λογική
Antikeimena
επιλογή επανάληψη πίνακες αντικείμενα string
όλα
αντικειμενοστρέφεια
switch, case, do-while, break, continue, μονοδιάστατοι, συναρτήσεις, αντικειμενοστρεφής προγραμματισμός, java
ergasies
μεθοδοι και αναδρομη
συμβολοσειρες
εξετασεις

Σχήμα 3.2.1.4α: Εισηγήσεις αντιμετώπισης δυσκολίας

Αν Ναι, τι σε δυσκόλεψε περισσότερο;

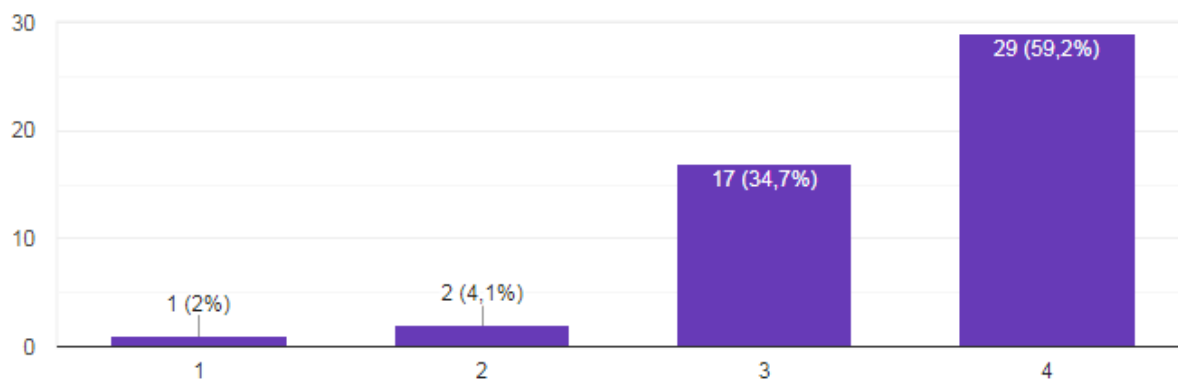
15 απαντήσεις

Antikeimena
επιλογή επανάληψη πίνακες αντικείμενα string
όλα
αντικειμενοστρέφεια
switch, case, do-while, break, continue, μονοδιάστατοι, συναρτήσεις, αντικειμενοστρεφής προγραμματισμός, java
ergasies
μεθοδοι και αναδρομη
συμβολοσειρες
εξετασεις
αντικειμενοστρεφης προγραμματισμός
η ύλη μετά το λύκειο
όλα επειδή δεν ασχολήθηκα πιο πριν με προγραμματισμό

Σχήμα 3.2.1.4β: Εισηγήσεις αντιμετώπισης δυσκολίας

Αν είχες τη δυνατότητα να χρησιμοποιούσες ένα σύστημα για καλύτερη εμπέδωση της ύλης του μαθήματος το οποίο να περιείχε διαλέξεις και διάφορα παιχνίδια, θα το χρησιμοποιούσες;

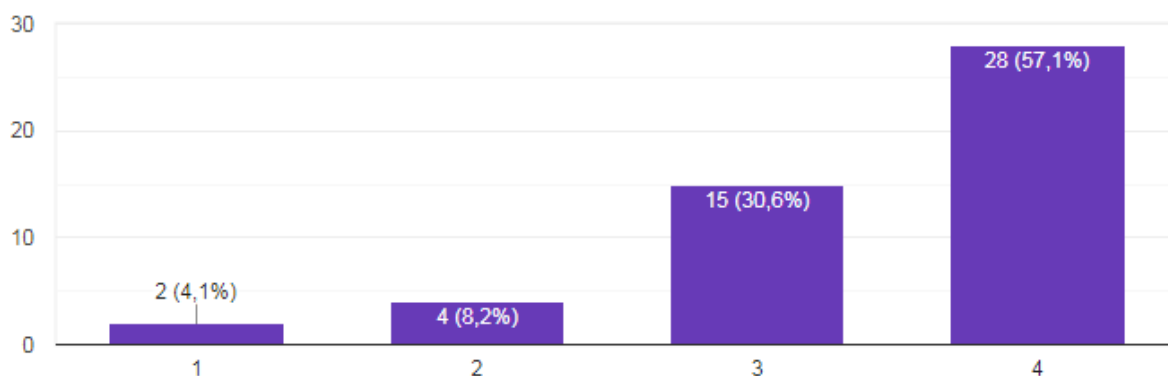
49 απαντήσεις



Σχήμα 3.2.1.5: Διάγραμμα για χρήση κάποιου ενισχυτικού συστήματος μάθησης για καλύτερη κατανόηση

Αν το συγκεκριμένο σύστημα παρείχε σημειώσεις για τη θεωρία του κάθε κεφαλαίου με παραδείγματα και μικρά επεξηγηματικά βίντεο, πόσο χρήσιμο θα σου φαινόταν;

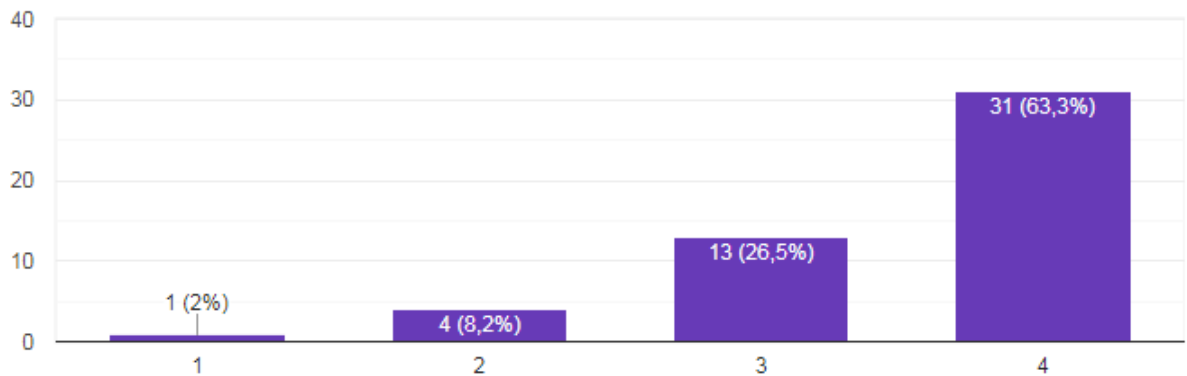
49 απαντήσεις



Σχήμα 3.2.1.6: Διάγραμμα για βοήθεια φοιτητή μέσω βίντεο και θεωρίας

Αν το συγκεκριμένο σύστημα παρείχε ερωτήσεις πολλαπλής επιλογής για την ύλη κάθε κεφαλαίου, πόσο χρήσιμο θα σου φαινόταν;

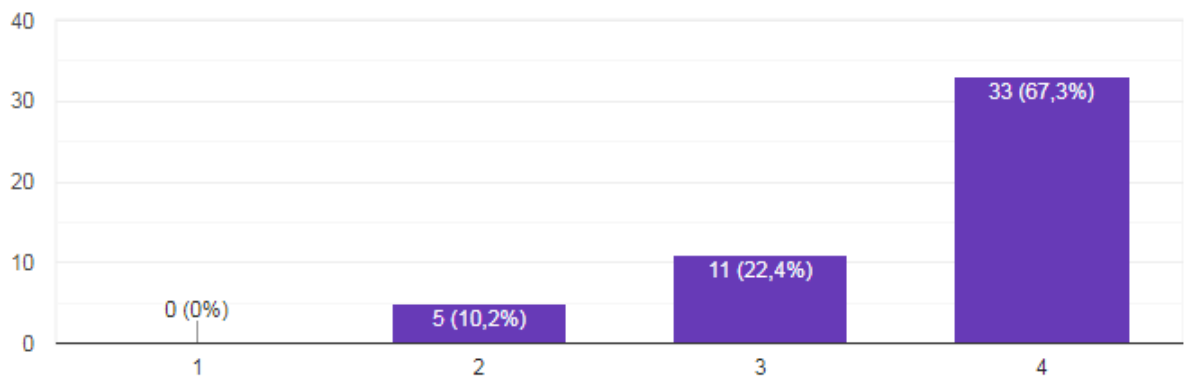
49 απαντήσεις



Σχήμα 3.2.1.7: Διάγραμμα βοήθειας φοιτητή μέσω ερωτήσεων πολλαπλής επιλογής

Αν το συγκεκριμένο σύστημα παρείχε ασκήσεις συμπλήρωσης κώδικα βαθμωτής δυσκολίας (δηλαδή αρχίζοντας από τις πολύ απλές ασκήσεις και στη συνέχεια στις πιο δύσκολες) βάση κάποιου προβλήματος, πόσο χρήσιμο θα σου φαινόταν;

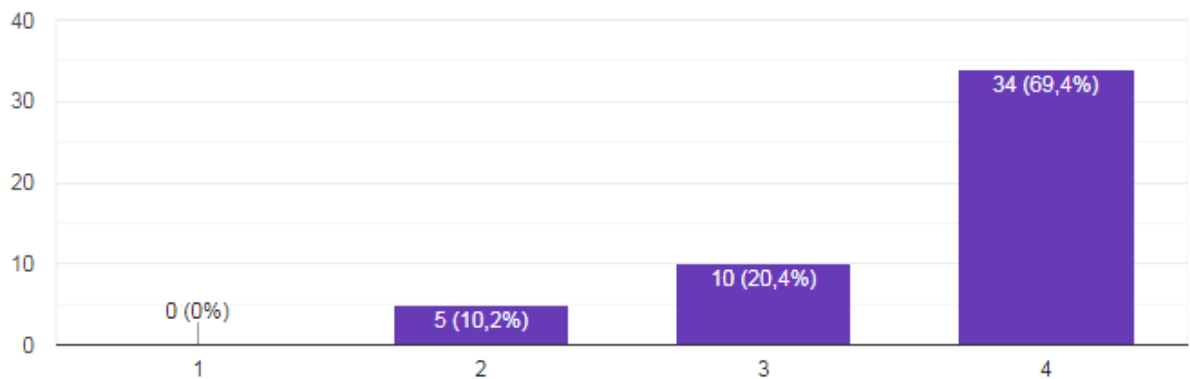
49 απαντήσεις



Σχήμα 3.2.1.8: Διάγραμμα βοήθειας φοιτητή μέσω ασκήσεων συμπλήρωσης κώδικα βαθμωτής δυσκολίας

Αν το συγκεκριμένο σύστημα παρείχε ασκήσεις πολλαπλής επιλογής του τύπου δοθέντος ενός αποτελέσματος, να επιλέξεις τον σωστό κώδικα που θα έδινε το συγκεκριμένο αποτέλεσμα, πόσο χρήσιμο θα σου φαινόταν;

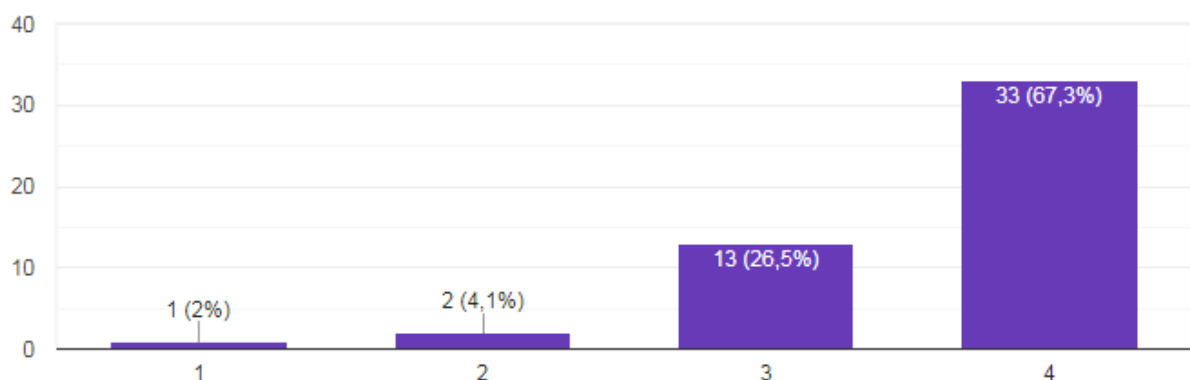
49 απαντήσεις



Σχήμα 3.2.1.9: Διάγραμμα βοήθειας φοιτητή μέσω ασκήσεων πολλαπλής επιλογής με κώδικα

Στα παιχνίδια που προαναφέρθηκαν πόσο χρήσιμο πιστεύεις θα ήταν αν υπήρχε η δυνατότητα παροχής επιπέδων βοήθειας (δηλαδή το επίπεδο βοήθειας 1 να παρέχει μια πιο γενική βοήθεια και όσο αυξάνεται το επίπεδο βοήθειας να σε κατευθύνει πιο συγκεκριμένα προς την λύση);

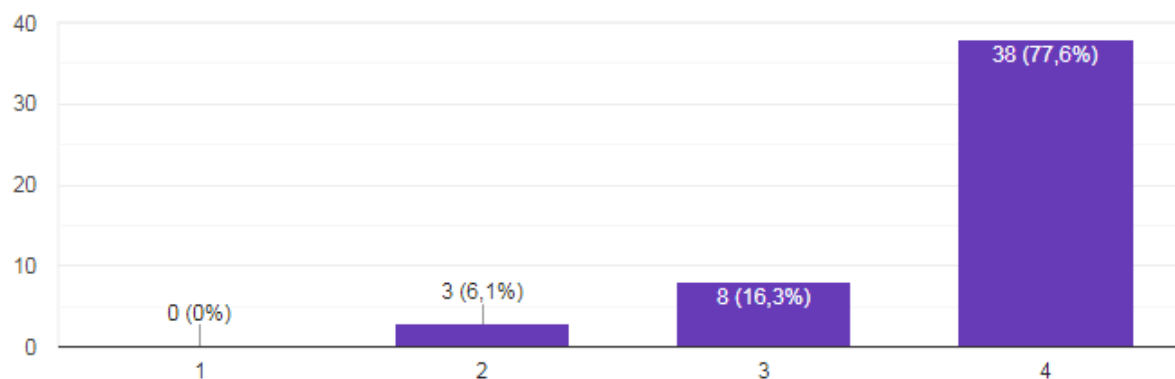
49 απαντήσεις



Σχήμα 3.2.1.10: Διάγραμμα βοήθειας φοιτητή μέσω παροχής επιπέδων βοήθειας

Στα παιχνίδια που προαναφέρθηκαν πόσο χρήσιμο πιστεύεις θα ήταν αν υπήρχε η δυνατότητα παροχής λύσης μαζί με μια σύντομη επεξήγηση της λύσης;

49 απαντήσεις



Σχήμα 3.2.1.11: Διάγραμμα βοήθειας φοιτητή μέσω παροχής λύσης με επεξήγηση

Θα μπορούσες να εισηγηθείς άλλα παιχνίδια που πιστεύεις θα βοηθούσαν για καλύτερη εμπέδωση της ύλης του μαθήματος;

12 απαντήσεις

paixnidi pou na perixe tis askiseis pou tha mpoun stin eksetasi

βιντεοσκοπηση τις διαλέξεις και τα εργαστήρια ούτως ώστε όταν δεν πάμε στο μάθημα να μπορούμε να τα δούμε από εκεί

λυμένες ασκήσεις, να μην τις λύνουμε εμείς επειδή δεν έχουμε χρόνο με τα άλλα μαθήματα

πιστεύω δεν χρειάζεται και πολύ ένα τέτοιο σύστημα αφού τα πάντα είναι στο internet

να βρίσκαμε το λάθος στην java, αλλά το λάθος να ήταν σημαντικό για καλύτερη κατανόηση των εντολών java

ta παιχνidia na einai omadika gia na paizoume me tous siffitites mas

όχι

παιχνίδια με πιθανες ερωτησεις της εξετασεις

ναι, θα μπορούσαμε να υποβάλλουμε ερωτήσεις που να τις βλέπουν όλοι και όποιος γνωρίζει να απαντά

να έχει πρόσβαση ο καθηγητής στο σύστημα, να έχει βαθμολογία και να βλέπει ο καθηγητής τις επιδόσεις μας και να μετρά ως bonus στον βαθμό μας

ρίντεο βλέπουμε και μόνοι μας

Σχήμα 3.2.1.12α: Εισηγήσεις παιχνιδιών για καλύτερη εμπέδωση της ύλης του μαθήματος

Θα μπορούσες να εισηγηθείς άλλα παιχνίδια που πιστεύεις θα βοηθούσαν για καλύτερη εμπέδωση της ύλης του μαθήματος;

12 απαντήσεις

λυμένες ασκήσεις, να μην τις λύνουμε εμείς επειδή δεν έχουμε χρόνο με τα άλλα μαθήματα

πιστεύω δεν χρειάζεται και πολύ ένα τέτοιο σύστημα αφού τα πάντα είναι στο internet

να βρίσκαμε το λάθος στην java, αλλά το λάθος να ήταν σημαντικό για καλύτερη κατανόηση των εντολών java

ta paixnidia na einai omadika gia na paizoume me tous siffittites mas

όχι

παιχνίδια με πιθανες ερωτήσεις της εξετασης

ναι, θα μπορούσαμε να υποβάλλουμε ερωτήσεις που να τις βλέπουν όλοι και όποιος γνωρίζει να απαντά

να έχει πρόσβαση ο καθηγητής στο σύστημα, να έχει βαθμολογία και να βλέπει ο καθηγητής τις επιδόσεις μας και να μετρά ως bonus στον βαθμό μας

βίντεο βλέπουμε και μόνοι μας

Τα εργαστήρια να μπορούσαμε να τα παρακολουθήσουμε και από το σύστημα. Μας λένε πολλές πληροφορίες σε λίγο χρόνο και δεν προλαμβάνουμε να τα καταλάβουμε. Ασχέτως που αν χάσεις ένα εργαστήριο, μετά δεν προλαβαίνεις να τα καλύψεις. Ακόμα λόγω του προγράμματος μας πάμε κουρασμένοι και δεν μπορούμε να συγκεντρωθούμε.

Σχήμα 3.2.1.12β: Εισηγήσεις παιχνιδιών για καλύτερη εμπέδωση της ύλης του μαθήματος

3.2.2 Συμπεράσματα Ερωματολογίου

Πιο κάτω θα αναλυθούν τα δεδομένα τα οποία συλλέχτηκαν από το ερωτηματολόγιο:

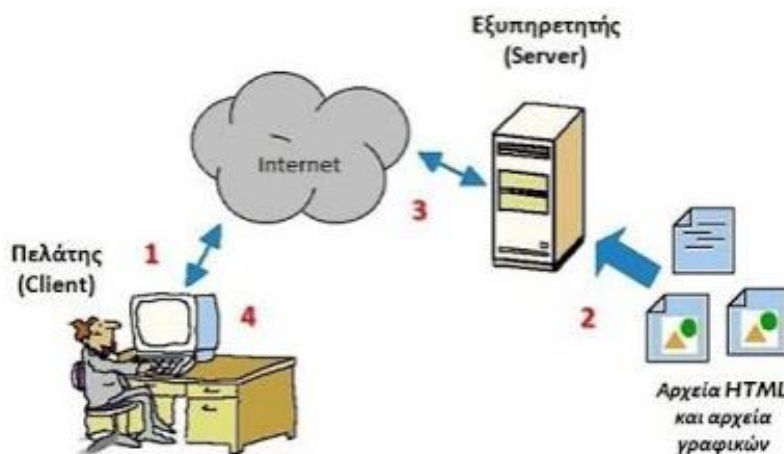
- Σύμφωνα με το διάγραμμα για χρήση κάποιου ενισχυτικού συστήματος μάθησης για καλύτερη κατανόηση (σχήμα 3.2.1.5) το ποσοστό των φοιτητών που απάντησαν ότι θα το χρησιμοποιούσαν «πάρα πολύ» είναι 59,2% και «πολύ» είναι 34,7%. Συνεπώς με ποσοστό περισσότερο από 90%, οι φοιτητές θα χρησιμοποιούσαν το σύστημα αυτό
- Σύμφωνα με το διάγραμμα για βοήθεια φοιτητή μέσω βίντεο και θεωρίας (σχήμα 3.2.1.6) το ποσοστό των φοιτητών που απάντησαν «πάρα πολύ» είναι 30,6% και «πολύ» είναι 57,1%. Με ένα συνολικό ποσοστό σχεδόν στο 90%, οι φοιτητές θα προτιμούσαν το σύστημα να περιέχει τόσο θεωρία, όσο και επεξηγηματικά βίντεο που αφορούν κάποιες έννοιες
- Σύμφωνα με τα διαγράμματα βοήθειας φοιτητή μέσω ασκήσεων πολλαπλής επιλογής και επιλογής σωστού κώδικα (σχήματα 3.2.1.7, 3.2.1.9), συμπεράνουμε ότι ποσοστό γύρω στο 85-88% των φοιτητών θα προτιμούσαν να έχουν τέτοιου είδους παιχνίδια

- Σύμφωνα με το διάγραμμα βοήθειας φοιτητή μέσω παροχής λύσης με επεξήγηση (σχήμα 3.2.1.11), με ποσοστό γύρω στο 95%, οι φοιτητές πιστεύουν ότι θα τους βοηθήσει
- Σύμφωνα με το διάγραμμα βοήθειας φοιτητή μέσω ασκήσεων συμπλήρωσης κώδικα βαθμωτής δυσκολίας (σχήμα 3.2.1.8), με ποσοστό γύρω στο 90%, θεωρούν ότι θα τους βοηθήσει να καταλάβουν καλύτερα τις βασικές έννοιες προγραμματισμού

3.3 Γενική Περιγραφή Συστήματος

3.3.1 Αρχιτεκτονική Πελάτη – Διακομιστή (Εξυπηρετητή)

Το μοντέλο αρχιτεκτονικής πελάτη – διακομιστή (σχήμα 3.3.1) αποτελεί μια μέθοδο ανάπτυξης λογισμικού, στην οποία ο πελάτης (ένα τμήμα λογισμικού) επικοινωνεί και ανταλλάζει πληροφορίες με ένα άλλο τμήμα λογισμικού, τον διακομιστή. Κάθε διακομιστής μπορεί να εξυπηρετεί πολλούς πελάτες. Ο πελάτης θεωρείται ο περιηγητής ιστού (π.χ Google Chrome) και ο εξυπηρετητής είναι αυτός που φιλοξενεί την ιστοσελίδα (π.χ Apache). Για την περίπτωση μας, οι χρήστες (φοιτητές) θα αποστέλλουν δεδομένα στον διακομιστή και αυτός με τη σειρά του θα επεξεργαστεί τα δεδομένα αυτά και θα επιστρέψει την ανάλογη απάντηση.



Σχήμα 3.3.1: Αρχιτεκτονική Πελάτη – Διακομιστή

Λόγω του ότι στην δική μας περίπτωση ο Πελάτης και ο Διακομιστής τρέχουν ταυτόχρονα στον ίδιο υπολογιστή, η αρχιτεκτονική αυτή μπορεί να χαρακτηριστεί πιο συγκεκριμένα Αρχιτεκτονική Πελάτη – Διακομιστή τριών επιπέδων (Client –Server Three Tier). Αυτό το μοντέλο έχει τρία στρώματα υλικού, όπου το ένα αφορά το πρόγραμμα περιήγησης για την παρουσίαση, το δεύτερο είναι ο διακομιστής εφαρμογών και το τρίτο είναι η Βάση Δεδομένων [14].

Υπάρχουν ορισμένα πλεονεκτήματα σε μια three-tier client – server προσέγγιση. Αυτά περιλαμβάνουν:

1. Κλιμάκωση
2. Γενικότερα πιο χαμηλά προβλήματα στα δίκτυα διανομής
3. Ευλυγισία

Η κλιμάκωση βελτιώθηκε, διότι ο κώδικας του server και η Βάση Δεδομένων είναι χωρισμένα, μπορούν να ξεκινήσουν από ένα απλό «υπολογιστή – οικοδεσπότη» και αργότερα να χωριστούν. Πολλαπλές εφαρμογές server μπορούν να επικοινωνήσουν με μια κεντρική βάση δεδομένων ή μια εφαρμογή server μπορεί ακόμα να εξυπηρετήσει τους clients, ενώ προσπελάζονται οι πολλαπλές βάσεις δεδομένων όσο το σύστημα αυξομειώνεται.

Τα χαμηλότερα προβλήματα στα δίκτυα απορρέουν από το πέρασμα μικρών μηνυμάτων στην εφαρμογή παρά από ολόκληρες εγγραφές πληροφοριών.

Η ευλυγισία κερδίζεται, διότι ο client, ο server και τα συστήματα βάσεων δεδομένων μπορούν το καθένα να αντικαθιστούν χωρίς να επηρεάζουν τα άλλα κομμάτια, δεδομένου ότι η διασύνδεση επίσης δεν αλλάζει.

3.3.2 Προοπτική του συστήματος

Το σύστημα θα χρησιμοποιείται από τους φοιτητές του Τμήματος Πληροφορικής, του Πανεπιστημίου Κύπρου. Ως εκ τούτου, για να έχει κάποιος φοιτητής πρόσβαση σε αυτό, θα πρέπει να έχει λογαριασμό, και λόγω του ότι η πλατφόρμα αυτή θα βρίσκεται στους server του Πανεπιστημίου Κύπρου, ο φοιτητής θα πρέπει να βρίσκεται στο δίκτυο του Πανεπιστημίου αυτού.

3.3.2.1 Διαπροσωπείες Συστήματος

Στο παρόν στάδιο, το υπάρχων σύστημα είναι ανεξάρτητο και δεν συνεργάζεται με οποιοδήποτε άλλο σύστημα. Στόχος είναι να υπάρξει πρόσβαση μέσω του διαδικτύου, εντός

του δικτύου του Πανεπιστημίου Κύπρου. Ως λογισμικό, δεν θα είναι πλήρως ανεξάρτητο, αφού θα πρέπει να υπάρχει μια βάση δεδομένων, στην οποία να αποθηκεύονται δεδομένα και να επαναχρησιμοποιούνται οποιαδήποτε στιγμή χρειαστεί. Στο μέλλον ίσως χρησιμοποιηθεί για σκοπούς ενισχυτικής υποστήριξης των φοιτητών του μαθήματος ΕΠΛ131 του Τμήματος Πληροφορικής, του Πανεπιστημίου Κύπρου.

3.3.2.2 Διαπροσωπείες Χρηστών

Το σύστημα θα αλληλεπιδρά με το χρήστη μέσω διαφόρων οθονών. Για να έχει πρόσβαση ένας χρήστης στην πλατφόρμα θα πρέπει να κάνει είσοδο (login) χρησιμοποιώντας το username και τον κωδικό ο οποίος θα δίνεται από το διδάσκοντα του μαθήματος. Η είσοδος στο σύστημα είναι απαραίτητη για να αρχίσει η αλληλεπίδραση του χρήστη με το σύστημα. Στην αρχική οθόνη θα υπάρχει το κυρίως μενού, μέσω του οποίου ο χρήστης θα έχει πρόσβαση στις διάφορες λειτουργίες του συστήματος και για κάθε λειτουργία (διάλεξη, παιχνίδι) θα υπάρχει μια μικρή εξήγηση του πώς χρησιμοποιείται.

3.3.2.3 Διαπροσωπείες Υλικού (Hardware)

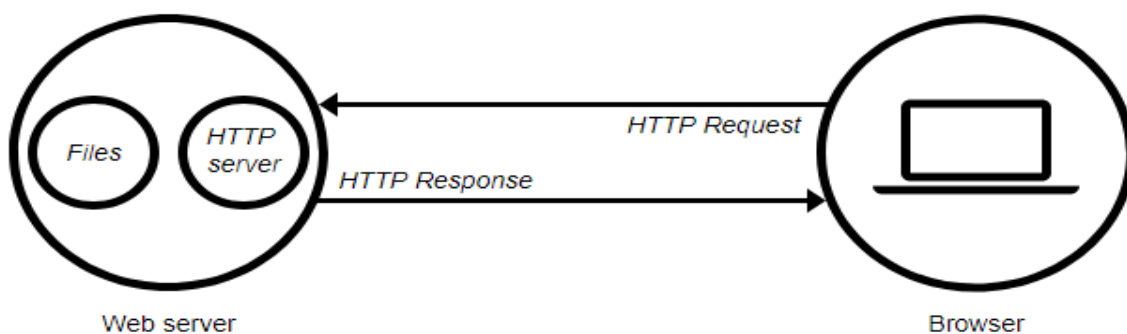
Για να λειτουργήσει το σύστημα πρέπει να υπάρχει ένας web server, στον οποίο θα βρίσκονται ο κώδικας του συστήματος και η βάση δεδομένων, με τη χρήση του οποίου θα μπορεί κάθε υπολογιστής να χρησιμοποιεί την πλατφόρμα.

Ο server ή αλλιώς εξυπηρετητής ή διακομιστής, στην πιο απλή του μορφή είναι ένας ηλεκτρονικός υπολογιστής που τρέχει κατάλληλο λογισμικό ώστε να εξυπηρετεί τους χρήστες που συνδέονται με αυτόν για κάποιο σκοπό. Ανάλογα με το σκοπό, ο server τρέχει και τις κατάλληλες υπηρεσίες και έχει και κατάλληλη ονομασία.

Για παράδειγμα, αν ο server εξυπηρετεί ιστοσελίδες, ονομάζεται web server και τρέχει λογισμικό κατάλληλο για να εξυπηρετεί τα αρχεία με τα οποία είναι κατασκευασμένη η ιστοσελίδα. Αν ο server εξυπηρετεί e-mail λογαριασμούς, ώστε οι χρήστες να έχουν το e-mail τους σε αυτόν, τότε θα είναι e-mail server και θα τρέχει κατάλληλο λογισμικό για αποστολή και λήψη e-mail μηνυμάτων.

Όπως είπαμε, στην πιο απλή του μορφή ένας server είναι ένας ηλεκτρονικός υπολογιστής ο οποίος τρέχει κατάλληλο λογισμικό. Στην πράξη βέβαια, δεν είναι απλοί υπολογιστές, αλλά είναι ηλεκτρονικοί υπολογιστές αυξημένων δυνατοτήτων και είναι εγκατεστημένοι σε ειδικούς χώρους που λέγονται datacenters. Ακριβώς επειδή η δουλειά ενός server είναι να εξυπηρετεί τους χρήστες που τον επισκέπτονται, ανάλογα με το πόσο μεγάλη είναι αυτή η επισκεψιμότητα (πόσοι χρήστες τον επισκέπτονται), αλλά και το τι εργασίες πρέπει να εκτελέσει ο server για κάθε επισκέπτη που συνδέεται με αυτόν, οι servers πρέπει να έχουν αυξημένες δυνατότητες σε:

- χωρητικότητα και ταχύτητα δίσκων
- μνήμη RAM
- επεξεργαστική ισχύ (CPU)
- χωρητικότητα δικτύου
- αξιοπιστία υλικών



Σχήμα 3.3.2.3: Αρχιτεκτονική Web server - Browser

3.3.2.4 Διαπροσωπείες Λογισμικού (Software)

Το σύστημα θα λειτουργεί στο διαδίκτυο, εντός του δικτύου του Πανεπιστημίου Κύπρου. Επομένως απαιτείται η χρήση περιηγητή ιστού (web browser). Η πλατφόρμα θα τρέχει στον web server και οι φοιτητές μέσω των browsers τους θα μπορούν να την χρησιμοποιούν.

Ένας περιηγητής ιστού (ή αλλιώς φυλλομετρητής σελίδων) είναι ένα λογισμικό που επιτρέπει στον χρήστη να προβάλλει, και να αλληλεπιδρά με κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες, συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Εντοπίζει αρχεία με τη χρήση του URL και εμφανίζει το περιεχόμενο τους στην οθόνη. Οι περιηγητές ιστού χρησιμοποιούν τη

γλώσσα HTML για την προβολή των ιστοσελίδων, γι' αυτό η εμφάνιση μιας ιστοσελίδας μπορεί να διαφέρει ανάλογα με τον browser.

Οι web browser ουσιαστικά αποτελούν λογισμικό πελάτη του δικτυακού πρωτοκόλλου επιπέδου εφαρμογών HTTP. Για κάθε web browser διατίθενται αρκετά πρόσθετα στοιχεία («add – ons» ή «plug – ins»), με στόχο την επαύξηση των δυνατοτήτων τους και την προστασία του χρήστη σε θέματα ασφάλειας.

Οι περισσότερο χρησιμοποιούμενοι περιηγητές ιστού είναι οι Google Chrome, Mozilla Firefox, Windows Internet Explorer, Safari και Opera.

Επιπρόσθετα, η βάση δεδομένων θα υλοποιηθεί με MySQL και το σύστημα με PHP, ενώ για την επεξεργασία κώδικα θα γίνει χρήση του TomCat server.

3.3.2.5 Διαπροσωπικές Επικοινωνίας

Όπως αναφέρθηκε και πιο πάνω, το σύστημα έχει τη μορφή διαδικτυακής εφαρμογής (web application), έτσι απαιτεί τη χρήση περιηγητή ιστού (web browser). Το σύστημα θα τρέχει στον web server, και οι χρήστες, μέσω των browsers τους θα μπορούν να το χρησιμοποιήσουν, εντός του δικτύου του Πανεπιστημίου Κύπρου.

3.3.2.6 Περιορισμοί μνήμης

Το σύστημα, δεν απαιτεί την ύπαρξη κάποιας συγκεκριμένης ποσότητας μνήμης στους υπολογιστές.

3.3.2.7 Απαιτήσεις Προσαρμογής Περιοχών

Ο υπολογιστής στον οποίο θα τρέχει το πρόγραμμα θα είναι ο υπολογιστής μου, αφού ο server και η βάση δεδομένων θα είναι τοπικά. Σε επόμενο στάδιο, οι υπολογιστές στους οποίους θα τρέχει το σύστημα θα είναι οι υπολογιστές που παρέχονται στα εργαστήρια του Τμήματος Πληροφορικής. Η βάση θα εγκατασταθεί στον server Electra του Τμήματος Πληροφορικής.

3.3.3 Λειτουργίες Προϊόντος

Με βάση τις απαιτήσεις, καθορίζονται οι πιο κάτω λειτουργίες προς υλοποίηση:

Λειτουργός – διδάσκοντας:

- Είσοδος στην πλατφόρμα
- Δυνατότητα εισαγωγής / τροποποίησης / διαγραφής λογαριασμών χρηστών (φοιτητών)
- Χρήση της πλατφόρμας

Χρήστης – φοιτητής:

- Είσοδος στην πλατφόρμα
- Χρήση της πλατφόρμας

3.3.4 Χαρακτηριστικά Χρηστών

Οι χρήστες που θα χρησιμοποιούν το συγκεκριμένο σύστημα, θα πρέπει να παρακολουθούν το μάθημα ΕΠΛ131 Αρχές Προγραμματισμού. Οι χρήστες θα έχουν πρόσβαση στο σύστημα, εφ' όσον πρώτα τους καταχωρήσει ο διδάσκοντας του μαθήματος και δεδομένου ότι είναι φοιτητές του Πανεπιστημίου Κύπρου.

3.3.5 Περιορισμοί

Για τη λειτουργία του παιχνιδιού συμπλήρωσης κώδικα Java, το λογισμικό που θα χρησιμοποιηθεί είναι ένα web application που τρέχει σε Tomcat server. Για τη λειτουργία της εισόδου, προσθήκης και διαγραφής μελών από τη βάση της πλατφόρμας, θα χρησιμοποιηθεί ο Apache server.

3.3.6 Υποθέσεις και εξαρτήσεις

Οι απαιτήσεις του συστήματος μπορούν να επηρεαστούν μερικώς και να χρειαστεί να διαφοροποιηθούν ελαφρώς ανάλογα με το πώς θα εξελιχθεί η υλοποίηση του συστήματος στην πορεία.

3.3.7 Διανομή απαιτήσεων

Μερικές από τις απαιτήσεις αναβάλλονται για το παρόν στάδιο. Σκοπός είναι να υλοποιηθεί ό,τι ζητήθηκε. Δύναται όμως, αστάθμητοι παράγοντες να εμποδίσουν ή να καθυστερήσουν, στην πορεία, την υλοποίηση κάποιων απαιτήσεων.

3.4 Συγκεκριμένες Απαιτήσεις

3.4.1 Εξωτερικές Απαιτήσεις Διεπαφών

3.4.1.1 Διεπαφές Χρηστών

Η διεπαφή χρήστη είναι υπεύθυνη για την αλληλεπίδραση του χρήστη με την πλατφόρμα. Ο χρήστης πλοηγείται στην πλατφόρμα εκπαίδευσης και αλληλεπιδρά με το σύστημα.

3.4.1.2 Διεπαφές Υλικού (Hardware)

Ο υπολογιστής στον οποίο θα τρέχει το σύστημα θα είναι HP Pavilion dv7, 2.27GHz με 4GB RAM. Επίσης θα χρησιμοποιηθεί server (TomCat, Apache) ο οποίος θα βρίσκεται εγκατεστημένος στον υπολογιστή μου.

3.4.1.3 Διεπαφές Λογισμικού (Software)

Η πλατφόρμα τρέχει κάτω από Windows 7. Απαραίτητη είναι και η ύπαρξη περιηγητή ιστού (internet browser) αφού η πλατφόρμα είναι web application.

3.4.1.4 Διεπαφές Επικοινωνίας

Το σύστημα είναι εφαρμογή πελάτη – διακομιστή, αφού ο υπολογιστής θα πρέπει να επικοινωνεί με τον web server όπου θα είναι εγκατεστημένο το πρόγραμμα για εκτέλεση. Το σύστημα θα επικοινωνεί με τη βάση δεδομένων στον server για επεξεργασία δεδομένων.

3.4.1.5 Διεπαφές Συστήματος

Μετά από την είσοδο (login) του διαχειριστή στο σύστημα, αυτός θα μπορεί να διαγράφει, προσθέτει μέλη από την πλατφόρμα και να πλοηγείται στην πλατφόρμα χρησιμοποιώντας όλες τις δυνατές λειτουργίες (διαλέξεις και παιχνίδια). Μετά από την είσοδο (login) του χρήστη στο σύστημα, αυτός θα μπορεί να πλοηγείται στην πλατφόρμα χρησιμοποιώντας όλες τις δυνατές λειτουργίες (διαλέξεις και παιχνίδια).

3.4.1.6 Απαιτήσεις Απόδοσης

Το σύστημα, αφού πρόκειται για web application, πρέπει να είναι ικανό να παρέχει την δυνατότητα σε περισσότερους από ένα χρήστες να το χρησιμοποιούν ταυτόχρονα. Ο web server, πρέπει να είναι σε θέση να παρέχει πρόσβαση σε περισσότερους από έναν χρήστες για να εξεταστούν την ίδια στιγμή. Θα πρέπει επίσης να λάβουμε υπόψη μας το διαθέσιμο χώρο μνήμης, το είδος του software και του hardware που έχουμε στη διάθεσή μας για να το χειριστούμε με τον καλύτερο τρόπο και να έχουμε τη βέλτιστη απόδοση που μπορούμε να έχουμε από την αλληλεπίδραση των πιο πάνω.

3.4.1.7 Περιορισμοί Σχεδιασμού

Το σύστημα θα αναπτυχθεί χρησιμοποιώντας τα εργαλεία Notepad++ για τη συγγραφή του κώδικα σε HTML, PHP, CSS. Το σύστημα θα αναπτυχθεί χρησιμοποιώντας το εργαλείο Eclipse Jee. Το σύστημα θα χρησιμοποιεί Apache server για τη λειτουργία του login, ενώ η προσθήκη και διαγραφή μελών από την μεριά του server αυτού, θα γίνεται με την βοήθεια της PHP και της MySQL. Το σύστημα θα χρησιμοποιεί τον TomCat server v8.5, για εκτέλεση Java servlets και επεξεργασία κώδικα Java. Το σύστημα πρέπει να έχει τη μορφή διαδικτυακής εφαρμογής με αποτέλεσμα να απαιτείται χρήση λειτουργικού συστήματος Windows 7 με internet browser.

3.5 Ιδιότητες Συστήματος Λογισμικού

3.5.1 Αξιοπιστία (reliability)

Η αξιοπιστία του συστήματος έγκειται στον αυστηρό έλεγχο του συστήματος με διάφορες περιπτώσεις ελέγχου (test cases), ώστε πιθανά λάθη ή παραβλέψεις από τον χρήστη, να αντιμετωπιστούν εγκαίρως. Καθορίζουν τη συμπεριφορά του λογισμικού σε καταστάσεις ενδογενών ή εξωγενών σφαλμάτων, τη διαδικασία αποκατάστασης, την πρόβλεψη τέτοιων καταστάσεων, καθώς και την επιθυμητή διαθεσιμότητα του λογισμικού. Επιπλέον, όσον αφορά την επεξεργασία δεδομένων (διαγραφή, προσθήκη) στην βάση δεδομένων, θα την έχει αποκλειστικά ο διαχειριστής (διδάσκοντας), ώστε τα δεδομένα – πληροφορίες του συστήματος να μην μπορούν να τροποποιηθούν από τους απλούς χρήστες (φοιτητές). Επίσης θα πρέπει να δίνονται τα σωστά μηνύματα ελέγχου, ώστε να καθοδηγούν το χρήστη ανά πάσα στιγμή.

3.5.2 Διαθεσιμότητα

Το σύστημα θα πρέπει να είναι διαθέσιμο 24 ώρες την μέρα και 365 μέρες το χρόνο. Σε περίπτωση υπερβολικού bursty traffic, δίνουμε μια πιθανότητα να «πέσει» ο server αν ξεπεραστεί ένα συγκεκριμένο όριο, για λόγους προστασίας του υλικού (point of failure). Ο μέσος χρόνος μεταξύ σφαλμάτων θα πρέπει να είναι κάθε δύο χρόνια. Επίσης, θα φυλάγονται προηγούμενες εκδόσεις του συστήματος, έτσι ώστε αν σε κάποια περίπτωση ανανέωσης του συστήματος έχουμε αποτυχία, να μπορούμε να μεταβούμε σε προηγούμενες εκδόσεις.

3.5.3 Ασφάλεια

Η ασφάλεια του συστήματος επιτυγχάνεται με το να προστατεύουμε το σύστημά μας από μη εξουσιοδοτημένους χρήστες. Ο διαχειριστής του συστήματος (καθηγητής του μαθήματος ΕΠΛ131), θα έχει πρόσβαση στο σύστημα με κάποια δικαιώματα, σε αντίθεση με τους απλούς χρήστες (φοιτητές) που θα μπορούν μόνο να πλοηγούνται στην πλατφόρμα. Με τον διαχωρισμό αυτό σε διαχειριστές και απλούς χρήστες, επιτυγχάνεται η ασφάλεια διότι η πρόσβαση στα ευαίσθητα δεδομένα του συστήματος, θα γίνεται μόνο από τους διαχειριστές.

3.5.4 Ευελιξία – Συντηρησιμότητα (maintainability)

Η συντηρησιμότητα του συστήματος αφορά την διάρκεια της λειτουργικότητας και της ποιότητας που παρέχει το σύστημα μας στους χρήστες σε σχέση με τις νέες τεχνολογικές εξελίξεις, καθώς και η δυνατότητα της συνεχούς αναβάθμισης και προσαρμογής αυτού σε νέα τεχνολογικά δεδομένα που αφορούν τόσο το Hardware, όσο και το Software. Δηλαδή το σύστημα θα πρέπει να είναι αρκετά εύελικτο για να επιτρέπει την διόρθωση των σφαλμάτων, να είναι αρκετά εύελικτο για να επιτρέπει μελλοντικές βελτιώσεις, να μπορεί να προστεθεί λειτουργικότητα με την πάροδο του χρόνου και να λειτουργεί για τουλάχιστον ένα με δύο χρόνια χωρίς να γίνει κάποια βελτίωση. Επίσης, ο κώδικας του συστήματος θα πρέπει να είναι ευανάγνωστος και εύκολα κατανοητός ως προς τον προγραμματιστή, ώστε όταν θα χρειαστεί να τον αναβαθμίσει, να γίνει εύκολα, γρήγορα και με το ελάχιστο κόστος. Για καλύτερη συντήρηση του κώδικα μπορεί να γίνει και χρήση κάποιων αφαιρετικών μονάδων ή επαναχρησιμοποίηση λογισμικών ενοτήτων.

3.5.5 Φορητότητα (portability)

Το σύστημα για να λειτουργήσει απαιτεί κάποιο λειτουργικό σύστημα, ένα συγκεκριμένο browser, server και μίαν βάση δεδομένων. Το δικό μας το σύστημα υλοποιήθηκε πάνω στο λειτουργικό σύστημα Windows 7, σε browser Google Chrome και τρέχει πάνω στον TomCat server (ή Apache για login).

Για να μεταφερθεί σε άλλον υπολογιστή με διαφορετικό λειτουργικό σύστημα, θα πρέπει να γίνουν κάποιες μικρό - αλλαγές όσον αφορά κάποιες από τις λειτουργίες που αφορούν την υλοποίηση του συστήματος. Στο παρόν στάδιο, το σύστημα δεν περιέχει ολοκληρωμένη φορητότητα, εφ' όσον δεν προβλέφθηκε από την αρχή. Εν τούτοις, για μελλοντική χρήση, θα γίνουν οι κατάλληλες αλλαγές ώστε να γίνει φορητό το σύστημα αυτό.

3.6 Λογικές Απαιτήσεις Βάσης Δεδομένων

Οι προσβάσεις στο σύστημα θα γίνονται, τόσο από τους φοιτητές (απλοί χρήστες), όσο και από τους καθηγητές (διαχειριστές). Τα δύο αυτά είδη προσβάσεων διαφέρουν, επειδή ο κάθε ένας έχει διαφορετικά δικαιώματα σε ό,τι αφορά το σύστημα. Ο απλός χρήστης, έχει μόνο το δικαίωμα να πλοηγηθεί στην πλατφόρμα, χωρίς να μπορεί να πειράξει τα δεδομένα που βρίσκονται στη βάση. Ο διαχειριστής, έχει το δικαίωμα να εισάγει και να διαγράφει μέλη από τη βάση, όπως επίσης και το δικαίωμα πλοήγησης στην πλατφόρμα. Με λίγα λόγια, μόνο ο διαχειριστής του συστήματος μπορεί να έχει πρόσβαση στις πληροφορίες της βάσης δεδομένων.

Κεφάλαιο 4

Προδιαγραφές Συστήματος

4.1 Εισαγωγή	29
4.1.1 Σκοπός	29
4.2 Διάγραμμα Ροής Δεδομένων (Data Flow Diagram – DFD)	30
4.2.1 Εισαγωγή στο Διάγραμμα Ροής Δεδομένων (Data Flow Diagram – DFD)	30
4.2.2 Διάγραμμα Ροής Δεδομένων – Επίπεδο 0	31
4.2.3 Διάγραμμα Ροής Δεδομένων – Επίπεδο 1	32
4.3 Αντικειμενοστραφής Ανάλυση	34
4.3.1 Use Case Diagrams (Διαγράμματα Σεναρίου Χρήσης)	34
4.3.1.1 Διάγραμμα Σεναρίου Φοιτητή	36
4.3.1.2 Διάγραμμα Σεναρίου Διαχειριστή	37
4.3.2 Entity Relationship Diagram	39

4.1 Εισαγωγή

4.1.1 Σκοπός

Σκοπός του κεφαλαίου αυτού είναι η εξακρίβωση των απαιτήσεων και των πραγματικών αναγκών του συστήματος. Μέσα από τη δημιουργία διαφόρων διαγραμμάτων, θα γίνουν πιο ξεκάθαρες οι απαιτήσεις, καθώς θα μπορούν να διευκρινιστούν λεπτομέρειες που στο προηγούμενο κεφάλαιο δεν είχαν καθοριστεί. Όλα τα πιο πάνω συνθέτουν το έγγραφο των τεχνικών προδιαγραφών του συστήματος που πρόκειται να αναπτυχθεί.

Η ανάλυση των απαιτήσεων που έχουν καταγραφεί στο προηγούμενο κεφάλαιο διευκολύνει τη δημιουργία σημαντικών διαγραμμάτων τα οποία θα ακολουθήσουν στο κεφάλαιο αυτό, και στη συνέχεια θα είναι πολύ χρήσιμα για τη σχεδίαση του συστήματος, αφού συγκεκριμενοποιούνται οι ανάγκες του χρήστη και καταγράφονται με τρόπο που να είναι πιο σαφείς, γιατί με τη χρήση της φυσικής γλώσσας που ήδη χρησιμοποιήσαμε δεν

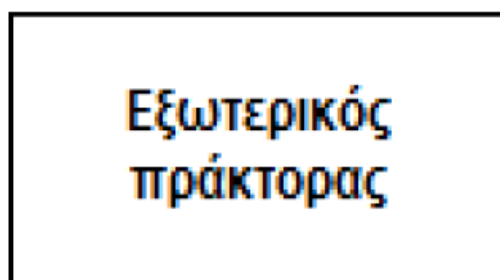
περιγράφονται τα χαρακτηριστικά του συστήματος με μεγάλη σαφήνεια. Με αυτό τον τρόπο, οι ανάγκες του συστήματος θα είναι πιο ξεκάθαρες, το οποίο είναι απαραίτητη προϋπόθεση για την ορθή ανάπτυξη του τελικού συστήματος.

4.2 Διάγραμμα Ροής Δεδομένων (Data Flow Diagram – DFD)

4.2.1 Εισαγωγή στο Διάγραμμα Ροής Δεδομένων (Data Flow Diagram – DFD)

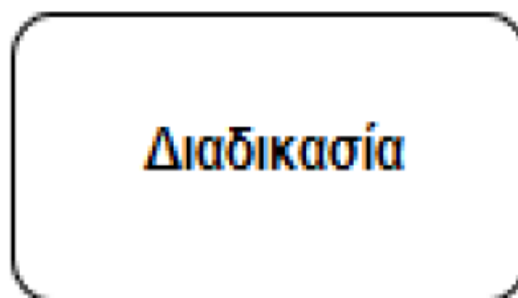
Τα διαγράμματα Ροής Δεδομένων μπορούν με μια λογική αναπαράσταση, να περιγράψουν τις λειτουργίες και τις απαιτήσεις που αφορούν το σύστημα – λογισμικό. Πιο κάτω αναλύονται τα διαγράμματα Ροής Δεδομένων που αφορούν το σύστημα υποστήριξης των φοιτητών του μαθήματος ΕΠΛ131, καθώς επίσης γίνεται η κατάλληλη επεξήγηση για την πλήρη κατανόηση του εκάστοτε διαγράμματος [16].

Εξωτερικός Πράκτορας: είναι πρόσωπα, μονάδες του οργανισμού, κάποιο άλλο σύστημα ή άλλος οργανισμός. Αλληλεπιδρούν με το σύστημα ανταλλάσσοντας δεδομένα.



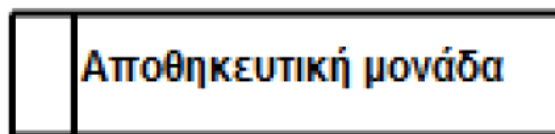
Σχήμα 4.2.1.1: Συμβολισμός για Εξωτερικό Πράκτορα

Διαδικασία: είναι επεξεργασίες που δέχονται δεδομένα και παράγουν στην έξοδο άλλα δεδομένα. Κάθε διαδικασία πρέπει να έχει ένα βέλος εισόδου και ένα βέλος εξόδου.



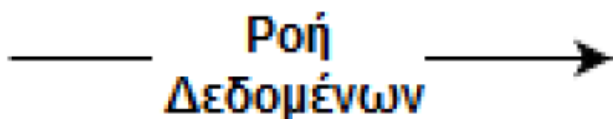
Σχήμα 4.2.1.2: Συμβολισμός για Διαδικασία

Αποθηκευτική Μονάδα: είναι χώρος όπου φυλάγονται μόνιμα δεδομένα σε βάση. Περιγράφουν δεδομένα που θέλει το σύστημα να αποθηκεύσει. Περιγράφουν βάσεις δεδομένων ή αρχεία.



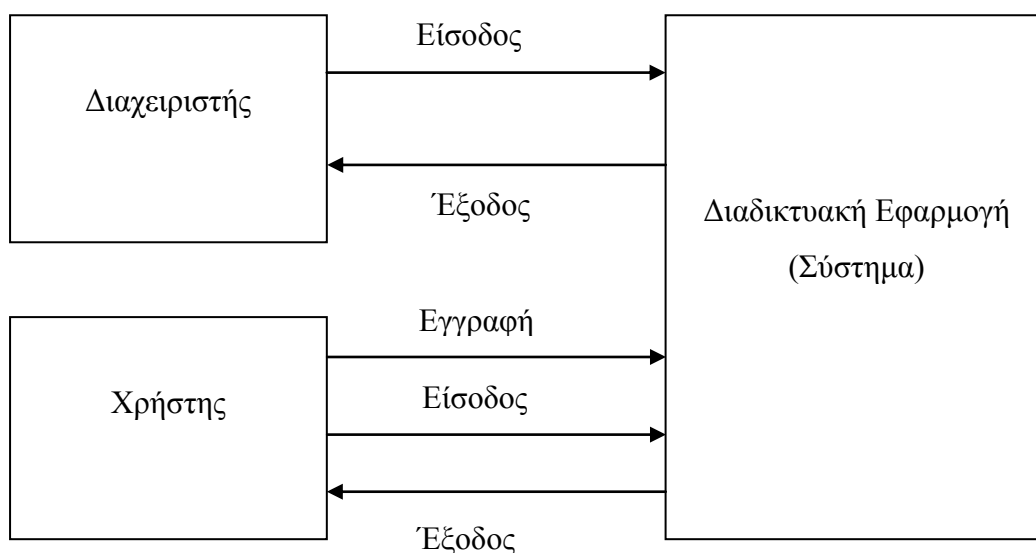
Σχήμα 4.2.1.3: Συμβολισμός για Αποθηκευτική Μονάδα

Ροή Δεδομένων: είναι είσοδοι και έξοδοι δεδομένων από μια διαδικασία σε μία άλλη διαδικασία. Χρησιμοποιούνται σε εξωτερικούς πράκτορες, αποθηκευτικές μονάδες και διαδικασίες.



Σχήμα 4.2.1.4: Συμβολισμός για Ροή Δεδομένων

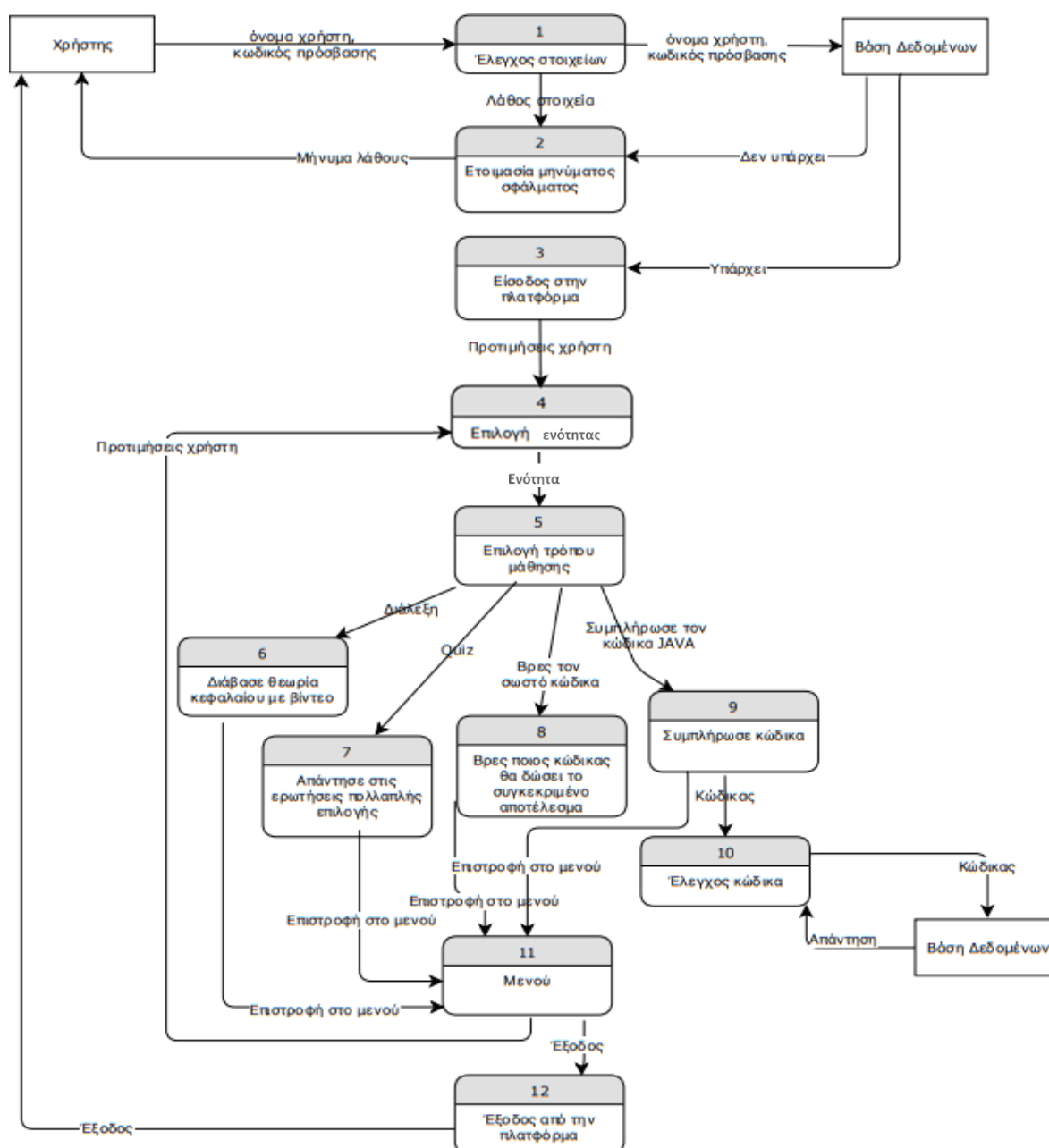
4.2.2 Διάγραμμα Ροής Δεδομένων – Επίπεδο 0



Σχήμα 4.2.2: Διάγραμμα Ροής Δεδομένων – Επίπεδο 0

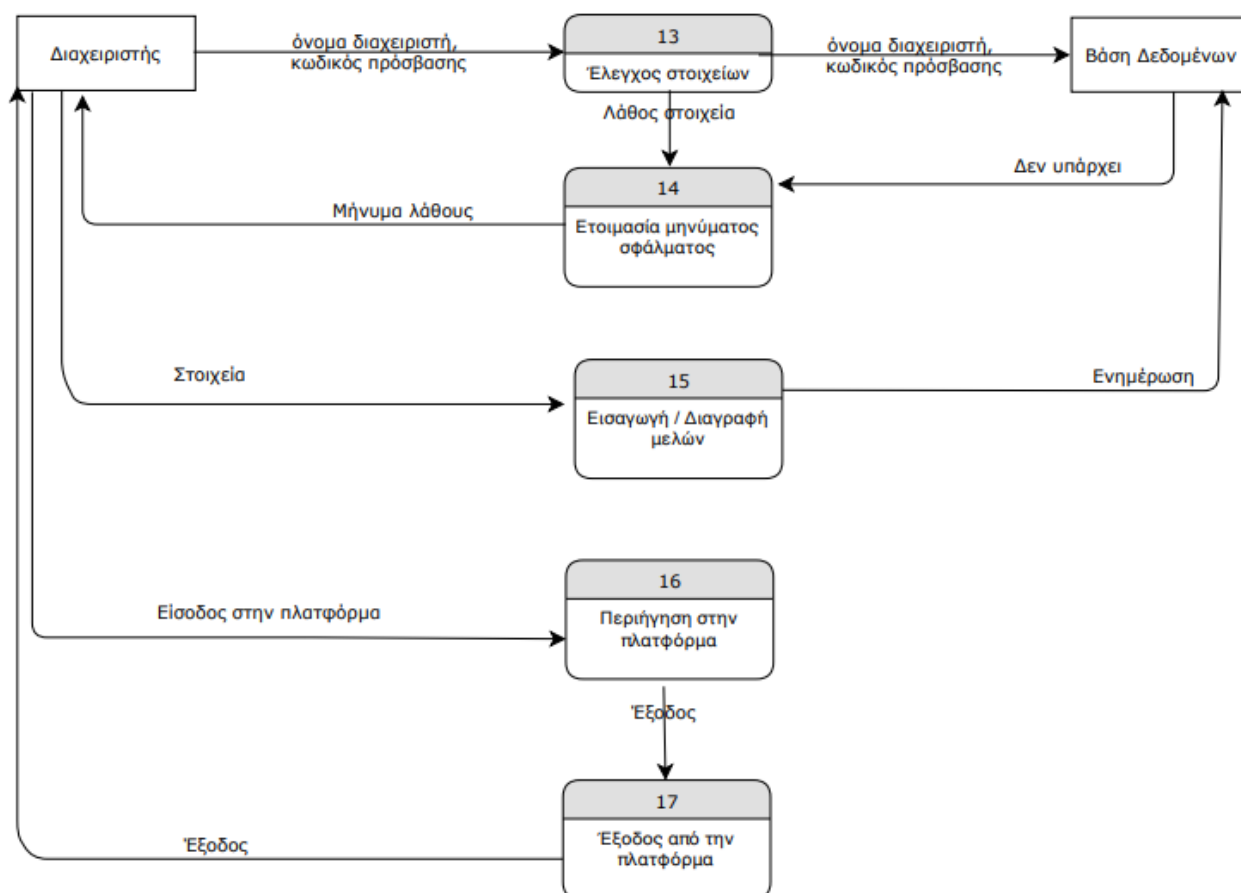
Στο πιο πάνω σχήμα (σχήμα 4.2.2) παρουσιάζονται οι εξωτερικοί πράκτορες του συστήματος. Οι εξωτερικοί πράκτορες είναι ο διαχειριστής (διδάσκοντας) και οι χρήστες (φοιτητές). Οι χρήστες θα κάνουν είσοδο (login) στο σύστημα καταχωρώντας το username και τον κωδικό τους, τα οποία θα δίνονται από το διδάσκοντα τους μαθήματος. Αφού κάποιος φοιτητής έχει λογαριασμό, θα μπορεί να εισέρχεται στο σύστημα μέσω του username του και του κωδικού του. Οι διαχειριστές του συστήματος θα εισέρχονται στο σύστημα και θα μπορούν να εισάγουν / εξάγουν φοιτητές στο σύστημα.

4.2.3 Διάγραμμα Ροής Δεδομένων – Επίπεδο 1



Σχήμα 4.2.3.1: Διάγραμμα Ροής Δεδομένων – Επίπεδο 1

Στο πιο πάνω σχήμα (σχήμα 4.2.3.1) γίνεται η αναπαράσταση των συνδέσεων των λειτουργιών μεταξύ των χρηστών (φοιτητών) και της βάσης δεδομένων. Ο χρήστης μπορεί να κάνει είσοδο (log-in) στο σύστημα (διαδικασία 3) δίνοντας το username του και τον κωδικό του (διαδικασία 1), και ακολούθως ελέγχεται αν υπάρχουν αυτά τα στοιχεία στη βάση δεδομένων. Αν τα στοιχεία δεν υπάρχουν το σύστημα επιστρέφει μήνυμα λάθους (διαδικασία 2) και ζητά από το χρήστη να ξαναδώσει τα στοιχεία του (διαδικασία 1). Ακολούθως, ο χρήστης αφού κάνει είσοδο στην πλατφόρμα, μπορεί να περιηγηθεί στο σύστημα (διαδικασία 3). Εντός του συστήματος μπορεί με βάση το κεφάλαιο το οποίο επιθυμεί να εκπαιδευτεί (διαδικασία 4), να επιλέξει τον τρόπο διδασκαλίας που προτιμά μεταξύ διαλέξεων, ερωτήσεων πολλαπλής επιλογής, και συμπλήρωσης κώδικα στην γλώσσα προγραμματισμού Java (διαδικασίες 6, 7, 8, 9). Μετά την ολοκλήρωση κάθε δραστηριότητας, εμφανίζεται ένα μενού (διαδικασία 11), μέσω του οποίου ο χρήστης μπορεί να μεταφερθεί στις υπόλοιπες δραστηριότητες του συγκεκριμένου κεφαλαίου ή στην αρχική σελίδα μέσω τις οποίας μπορεί να μεταφερθεί και σε υπόλοιπα κεφάλαια. Στο τέλος, ο φοιτητής μπορεί να βγει από το σύστημα (logout) (διαδικασία 12) . [16]



Σχήμα 4.2.3.2: Διάγραμμα Ροής Δεδομένων – Επίπεδο 1

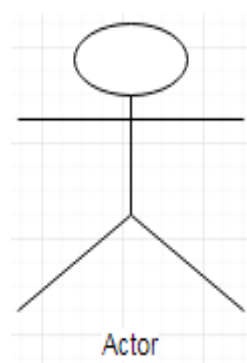
Στο πιο πάνω σχήμα (σχήμα 4.2.3.2) γίνεται η αναπαράσταση των συνδέσεων των λειτουργιών μεταξύ των διαχειριστών (καθηγητών) και της βάσης δεδομένων. Ο διαχειριστής μπορεί να κάνει είσοδο (log-in) στο σύστημα δίνοντας το username του και τον κωδικό του (διαδικασία 13), και ακολούθως ελέγχεται αν υπάρχουν αυτά τα στοιχεία στη βάση δεδομένων. Αν τα στοιχεία δεν υπάρχουν, το σύστημα επιστρέφει μήνυμα λάθους (διαδικασία 14) και ζητά από το διαχειριστή να ξαναδώσει τα στοιχεία του (διαδικασία 13). Επιπρόσθετα, ο διαχειριστής, αφού κάνει είσοδο, εκτός από το ότι μπορεί να περιηγηθεί στο σύστημα (διαδικασία 16), μπορεί και να προσθέτει ή να αφαιρεί χρήστες από το σύστημα (διαδικασία 15). Στο τέλος ο φοιτητής μπορεί να βγει από το σύστημα (logout) (διαδικασία 17).

4.3 Αντικειμενοστραφής Ανάλυση

4.3.1 Use Case Diagrams (Διαγράμματα Σεναρίου Χρήσης)

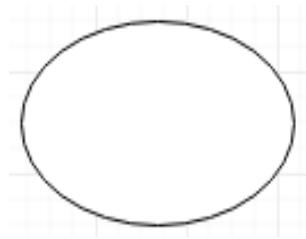
Ένα διάγραμμα σεναρίου χρήσης (use case diagram) χρησιμοποιείται για να αναγνωρίσει τα πρωτεύοντα στοιχεία και διαδικασίες που σχετίζονται με το σύστημα. Τα πρωτεύοντα στοιχεία αναφέρονται ως actors και οι διαδικασίες αναφέρονται ως use cases. [16]

Χρήστης: είναι πρόσωπα, μονάδες του οργανισμού, κάποιο άλλο σύστημα ή άλλος οργανισμός. Αλληλεπιδρούν με το σύστημα ανταλλάσσοντας δεδομένα.



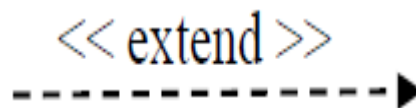
Σχήμα 4.3.1.1: Συμβολισμός για Χρήστη

Σενάριο: είναι λειτουργίες του συστήματος όπου υπάρχουν στο σύστημα και παράγουν μια συμπεριφορά.



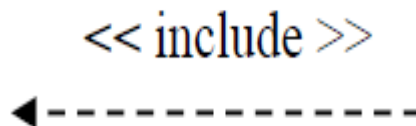
Σχήμα 4.3.1.2: Συμβολισμός για Σενάριο

Επέκταση: τοποθετείται στις περιπτώσεις που ένα σενάριο ή ένας δράστης χρησιμοποιούν δύο ή περισσότερα σενάρια.



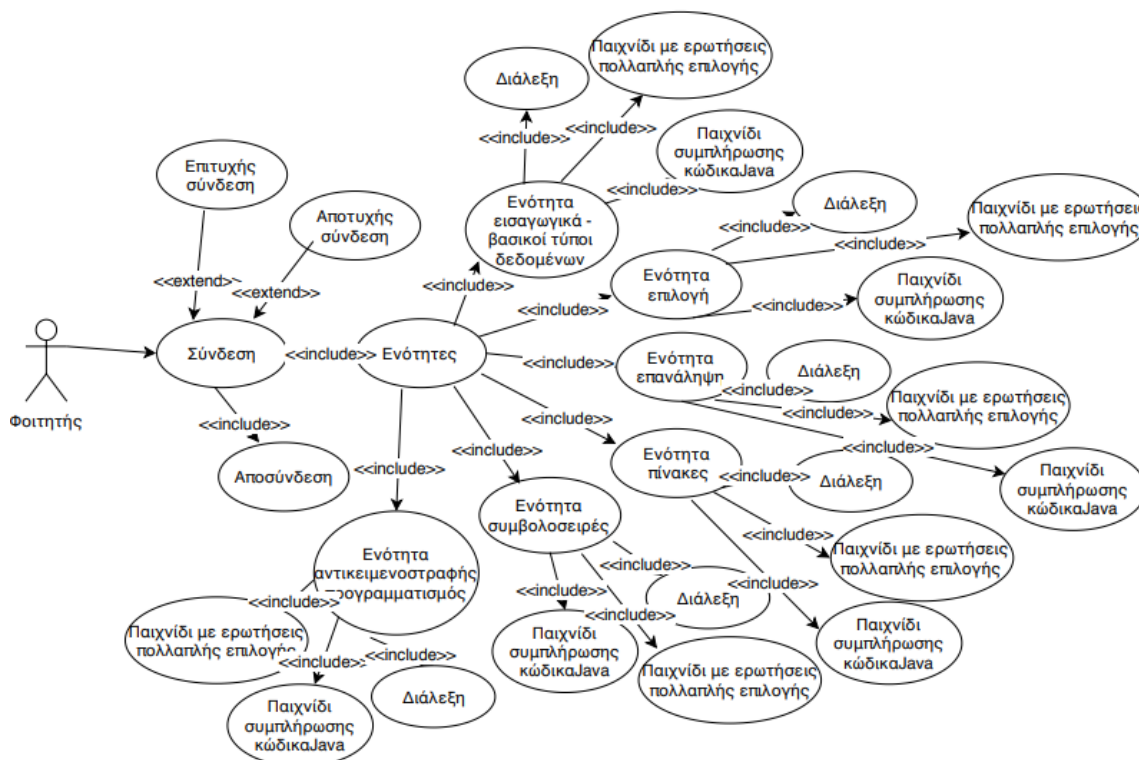
Σχήμα 4.3.1.3: Συμβολισμός για Επέκταση

Ενσωμάτωση: τοποθετείται στις περιπτώσεις που δύο ή περισσότεροι δράστες έχουν την ίδια συμπεριφορά.



Σχήμα 4.3.1.4: Συμβολισμός για Ενσωμάτωση

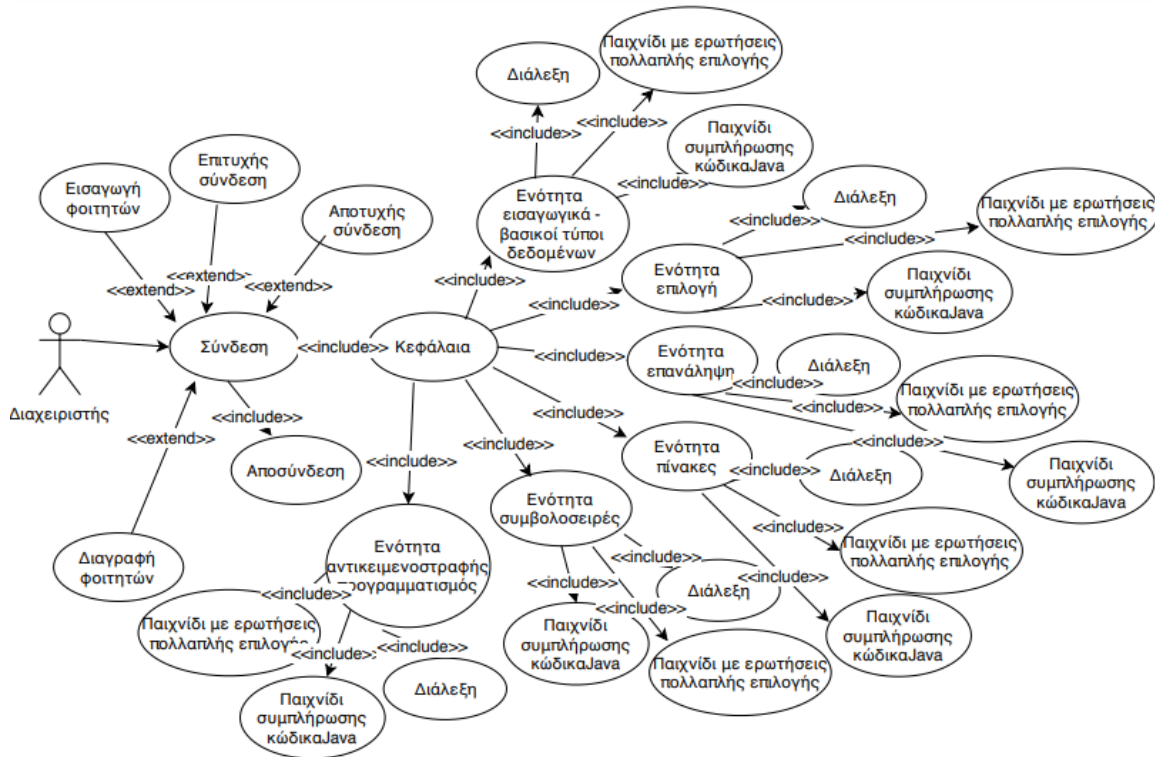
4.3.1.1 Διάγραμμα Σεναρίου Φοιτητή



Σχήμα 4.3.1.5: Διάγραμμα Σεναρίου Φοιτητή

Σύμφωνα με το πιο πάνω διάγραμμα (σχήμα 4.3.1.5) ο χρήστης μπορεί να κάνει είσοδο (login) στο σύστημα δίνοντας το username του και τον κωδικό του, και ακολούθως ελέγχεται αν υπάρχουν αυτά τα στοιχεία στη βάση δεδομένων. Αν τα στοιχεία δεν υπάρχουν το σύστημα, επιστρέφει μήνυμα λάθους και ζητά από το χρήστη να ξαναδώσει τα στοιχεία του. Ακολούθως, ο χρήστης αφού κάνει είσοδο στην πλατφόρμα, μπορεί να περιηγηθεί στο σύστημα. Εντός του συστήματος μπορεί με βάσει την ενότητα στην οποία επιθυμεί να εκπαιδευτεί, να επιλέξει τον τρόπο διδασκαλίας που προτιμά μεταξύ διαλέξεων, ερωτήσεων πολλαπλής επιλογής, και συμπλήρωσης κώδικα στη γλώσσα προγραμματισμού Java. Μετά την ολοκλήρωση κάθε δραστηριότητας εμφανίζεται ένα μενού, μέσω του οποίου ο χρήστης μπορεί να μεταφερθεί στις υπόλοιπες δραστηριότητες της συγκεκριμένης ενότητας ή στην αρχική σελίδα μέσω τις οποίας μπορεί να μεταφερθεί και σε άλλες ενότητες. Στο τέλος, ο φοιτητής μπορεί να βγει από το σύστημα (logout). [16]

4.3.1.2 Διάγραμμα Σεναρίου Διαχειριστή



Σχήμα 4.3.1.6: Διάγραμμα Σεναρίου Διαχειριστή

Σύμφωνα με το πιο πάνω διάγραμμα (σχήμα 4.3.1.6), ο διαχειριστής μπορεί να κάνει είσοδο (log-in) στο σύστημα δίνοντας το username του και τον κωδικό του, και ακολούθως ελέγχεται αν υπάρχουν αυτά τα στοιχεία στη βάση δεδομένων. Αν τα στοιχεία δεν υπάρχουν, το σύστημα επιστρέφει μήνυμα λάθους και ζητά από το διαχειριστή να ξαναδώσει τα στοιχεία του. Επιπρόσθετα, ο διαχειριστής, αφού κάνει είσοδο, εκτός από το ότι μπορεί να περιηγηθεί στο σύστημα, μπορεί και να προσθέτει ή να αφαιρεί χρήστες από το σύστημα. Στο τέλος ο φοιτητής μπορεί να αποχωρήσει από το σύστημα (logout).

4.3.2 Entity Relationship Diagram

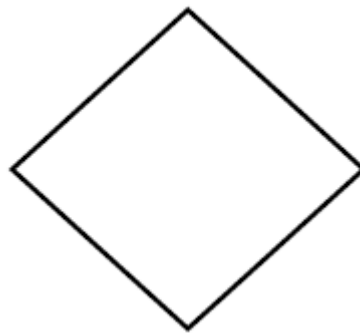
Μέσω των διαγραμμάτων οντοτήτων – συσχετίσεων μπορούμε να αναπαραστήσουμε με αφηρημένο τρόπο τα δεδομένα μας και πώς συσχετίζονται στη βάση δεδομένων.

Οντότητα: συγκεκριμένα αντικείμενα που υπάρχουν και μπορούν να αναπαρασταθούν στην βάση δεδομένων



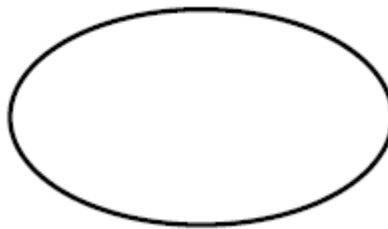
Σχήμα 4.3.1.3: Συμβολισμός για Οντότητα

Σχέση: (ειδικά) αντικείμενα που αντιστοιχούν δύο ή περισσότερες ξεχωριστές οντότητες με ένα συγκεκριμένο νόημα

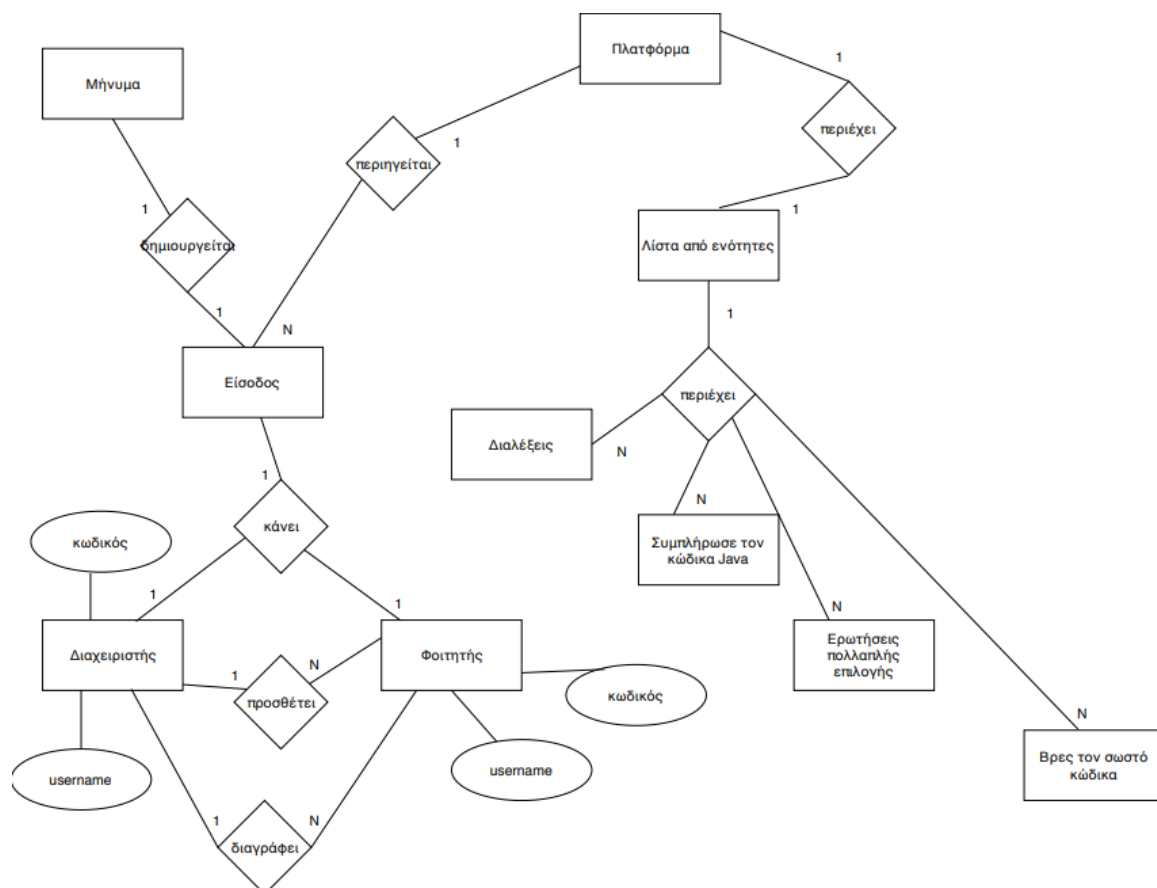


Σχήμα 4.3.1.3: Συμβολισμός για Σχέση

Γνώρισμα: χαρακτηριστικό μια σχέσης ή οντότητας



Σχήμα 4.3.1.3: Συμβολισμός για Γνώρισμα



Σχήμα 4.3.2.1: Διάγραμμα οντοτήτων – συσχετίσεων

Σύμφωνα με το πιο πάνω σχήμα (4.3.2.1), ο χρήστης μπορεί να κάνει είσοδο (log-in) στο σύστημα δίνοντας το username του και τον κωδικό του, και ακολούθως ελέγχεται αν υπάρχουν αυτά τα στοιχεία στη βάση δεδομένων. Αν τα στοιχεία δεν υπάρχουν το σύστημα επιστρέφει μήνυμα λάθους και ζητά από το χρήστη να ξαναδώσει τα στοιχεία του. Ακολούθως, ο χρήστης αφού κάνει είσοδο στην πλατφόρμα, μπορεί να περιηγηθεί στο σύστημα. Εντός του συστήματος, μπορεί με βάση την ενότητα την οποία επιθυμεί να εκπαιδευτεί, να επιλέξει τον τρόπο διδασκαλίας που προτιμά μεταξύ διαλέξεων, ερωτήσεων πολλαπλής επιλογής, και συμπλήρωσης κώδικα στην γλώσσα προγραμματισμού Java. Μετά την ολοκλήρωση κάθε δραστηριότητας εμφανίζεται ένα μενού, μέσω του οποίου ο χρήστης μπορεί να μεταφερθεί στις υπόλοιπες δραστηριότητες της συγκεκριμένης ενότητας ή στην αρχική σελίδα μέσω της οποίας μπορεί να μεταφερθεί και σε άλλες ενότητες. Στο τέλος, ο φοιτητής μπορεί να βγει από το σύστημα (logout). Επιπρόσθετα, ο διαχειριστής έχει όλες τις ιδιότητες του φοιτητή, με την επιπλέον ιδιότητα να προσθέτει και να αφαιρεί χρήστες από την πλατφόρμα. [16]

Κεφάλαιο 5

Σχεδίαση Συστήματος

5.1 Εισαγωγή	40
5.1.1 Σκοπός	40
5.1.2 Αποδέκτες	41
5.2 Αντικειμενοστραφής σχεδίαση	41
5.2.1 Διαγράμματα Αλληλεπίδρασης (Interaction Diagrams)	41
5.2.1.1 Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για επιτυχή είσοδο φοιτητή στο σύστημα	44
5.2.1.2 Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για αποτυχή είσοδο φοιτητή στο σύστημα	45
5.2.1.3 Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για επιτυχή είσοδο διαχειριστή στο σύστημα	46
5.2.1.4 Διάγραμμα Αλληλεπίδρασης (Interaction Diagrams) για αποτυχή είσοδο διαχειριστή στο σύστημα	47

5.1 Εισαγωγή

5.1.1 Σκοπός

Σκοπός του κεφαλαίου αυτού είναι η λεπτομερής καταγραφή των όσων καταγράφηκαν στη φάση των απαιτήσεων και των προδιαγραφών του συστήματος. Μέσα από αυτό το έγγραφο θα σχεδιαστεί το σύστημα το οποίο πρόκειται να υλοποιηθεί. Έχοντας ένα πλήρες και τεκμηριωμένο έγγραφο για τη φάση της σχεδίασης, διευκολύνεται η φάση της υλοποίησης, αφού με την καθοδήγηση των διαγραμμάτων θα έχουμε μια πιο ξεκάθαρη και σαφή αντίληψη ως προς το ποιες είναι οι λειτουργίες και πώς θα αλληλεπιδρούν μεταξύ τους για τη σωστή λειτουργία του συστήματος.

5.1.2 Αποδέκτες

Το συγκεκριμένο έγγραφο περιέχει τις απαραίτητες πληροφορίες για τα άτομα τα οποία ενδιαφέρονται να καταλάβουν πως λειτουργεί το σύστημα, για τους μελλοντικούς συντηρητές του συστήματος και για τα άτομα που θα βελτιώνουν το σύστημα, οι οποίοι πρόκειται να κληθούν να αντιληφθούν πλήρως τον τρόπο με τον οποίο έχει σχεδιαστεί και υλοποιηθεί το σύστημα. Οι πληροφορίες αυτές θα πρέπει να είναι σαφείς και κατανοητές από τους μελλοντικούς χρήστες του συστήματος, καθώς και πλήρως τεκμηριωμένες με σκοπό να διευκολύνουν τα άτομα που θα αναλάβουν τη συντήρηση και αναβάθμιση του συστήματος αυτού στο μέλλον.

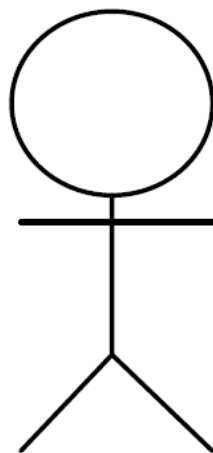
5.2 Αντικειμενοστραφής Σχεδίαση

5.2.1 Διαγράμματα Αλληλεπίδρασης (Interaction Diagrams)

Τα διαγράμματα αλληλεπίδρασης παρουσιάζουν την αλληλεπίδραση μεταξύ αντικειμένων με ανταλλαγή μηνυμάτων. Υπάρχουν δύο ειδών διαγράμματα αλληλεπίδρασης τα οποία παρουσιάζουν το ίδιο πράγμα αλλά με διαφορετικό τρόπο:

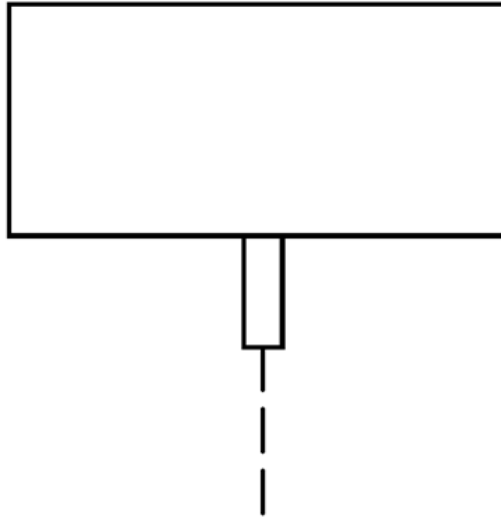
1. τα ακολουθιακά διαγράμματα: δίνουν έμφαση στην ακριβή χρονολογική ακολουθία των μηνυμάτων που ανταλλάσσονται μεταξύ των αντικειμένων και είναι χρήσιμα για την παρουσίαση της σειράς εμφάνισης των γεγονότων και εξελίσσονται από πάνω προς τα κάτω

Αντικείμενο (actor):



Σχήμα 5.2.1.1: Συμβολισμός για Αντικείμενο (actor)

Αντικείμενο:



Σχήμα 5.2.1.2: Συμβολισμός για Αντικείμενο

Μήνυμα μεταξύ αντικειμένων:



Σχήμα 5.2.1.3: Συμβολισμός για Μήνυμα μεταξύ αντικειμένων

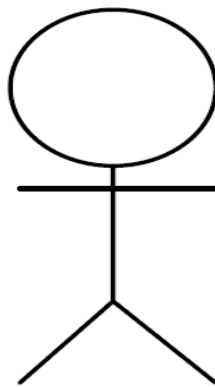
Αντανεκλαστικό μήνυμα:



Σχήμα 5.2.1.4: Συμβολισμός για Αντανακλαστικό μήνυμα

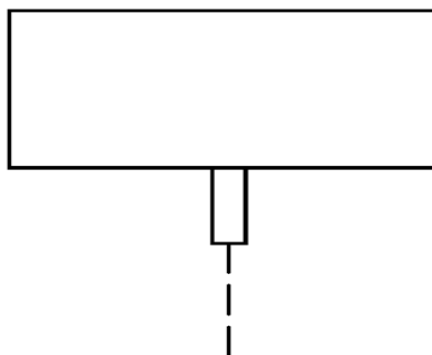
2. τα διαγράμματα συνεργασίας: παρουσιάζουν τη ροή διαμέσου ενός συγκεκριμένου σεναρίου και εστιάζονται περισσότερο στις σχέσεις μεταξύ των αντικειμένων

Αντικείμενο (actor) : ένας τύπος ρόλου που αναπαριστά μια οντότητα που αλληλεπιδρά με το σύστημα



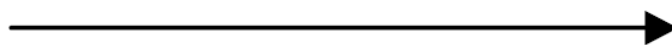
Σχήμα 5.2.1.5: Συμβολισμός για Αντικείμενο (actor)

Αντικείμενο: αντιπροσωπεύει έναν μεμονωμένο συμμετέχοντα στην Αλληλεπίδραση



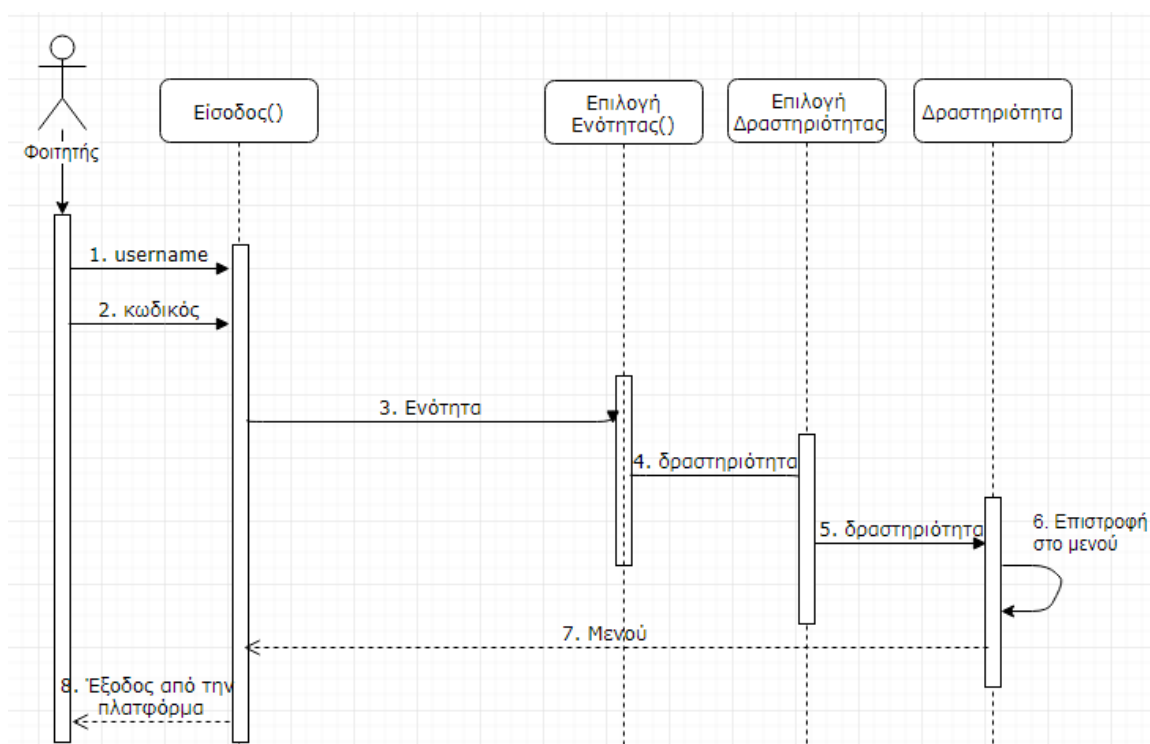
Σχήμα 5.2.1.6: Συμβολισμός για Αντικείμενο

Μήνυμα μεταξύ αντικειμένων: ορίζει μια συγκεκριμένη επικοινωνία μεταξύ των αντικειμένων μιας Αλληλεπίδρασης



Σχήμα 5.2.1.7: Συμβολισμός για Μήνυμα μεταξύ αντικειμένων

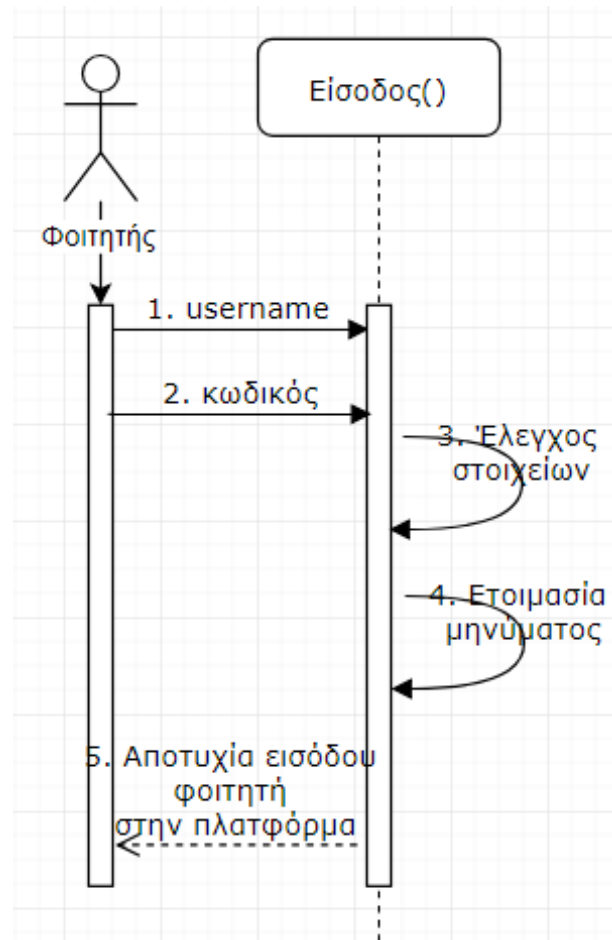
5.2.1.1 Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για επιτυχή είσοδο φοιτητή στο σύστημα



Σχήμα 5.2.1.8: Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για επιτυχή είσοδο φοιτητή στο σύστημα

Σύμφωνα με το πιο πάνω σχήμα (σχήμα 5.2.1.8), μέσω της λειτουργίας αυτής ο φοιτητής μπορεί να κάνει είσοδο (login) στο σύστημα χρησιμοποιώντας το username του και τον κωδικό του. Ελέγχονται τα στοιχεία αυτά και αφού είναι σωστά, ο φοιτητής μπορεί ανάλογα με την ενότητα που επιθυμεί, να επιλέξει δραστηριότητα και να εκπαιδευθεί. [16]

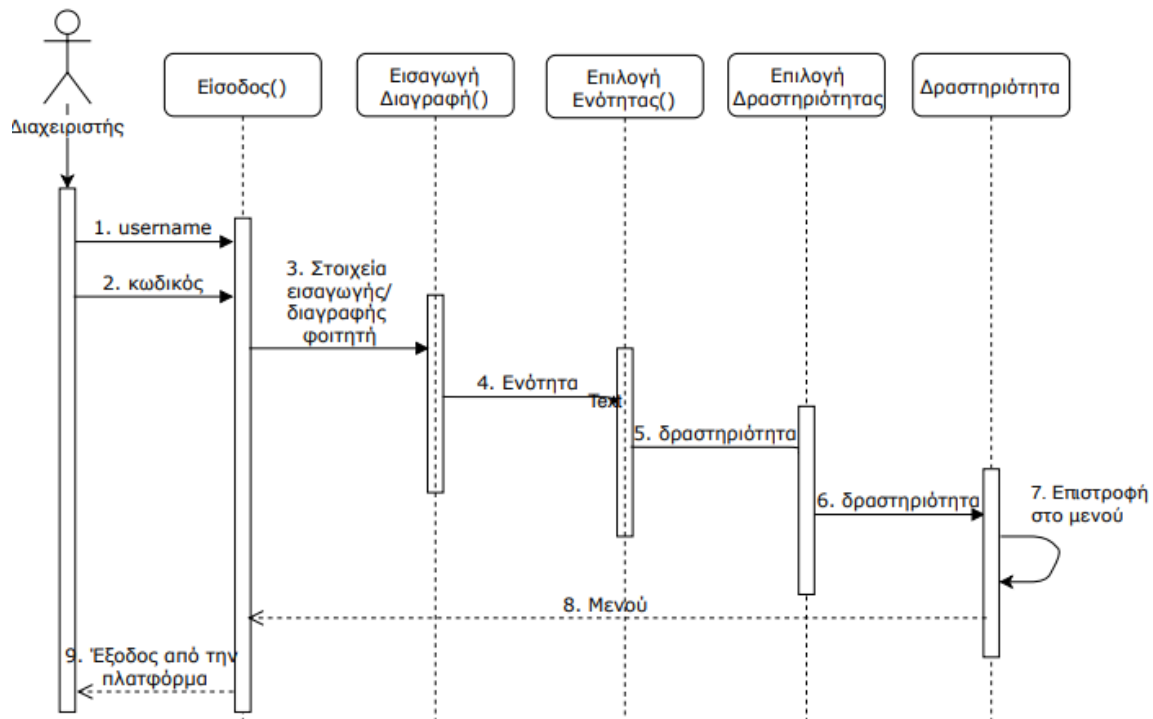
5.2.1.2 Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για αποτυχή είσοδο φοιτητή στο σύστημα



Σχήμα 5.2.1.9: Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για αποτυχή είσοδο φοιτητή στο σύστημα

Σύμφωνα με το πιο πάνω σχήμα (σχήμα 5.2.1.9), μέσω της λειτουργίας αυτής, ο φοιτητής μπορεί να κάνει είσοδο (login) στο σύστημα χρησιμοποιώντας το username του και τον κωδικό του. Ελέγχονται τα στοιχεία αυτά, και σε περίπτωση που είναι λάθος επιστρέφεται στον χρήστη μήνυμα λάθους.

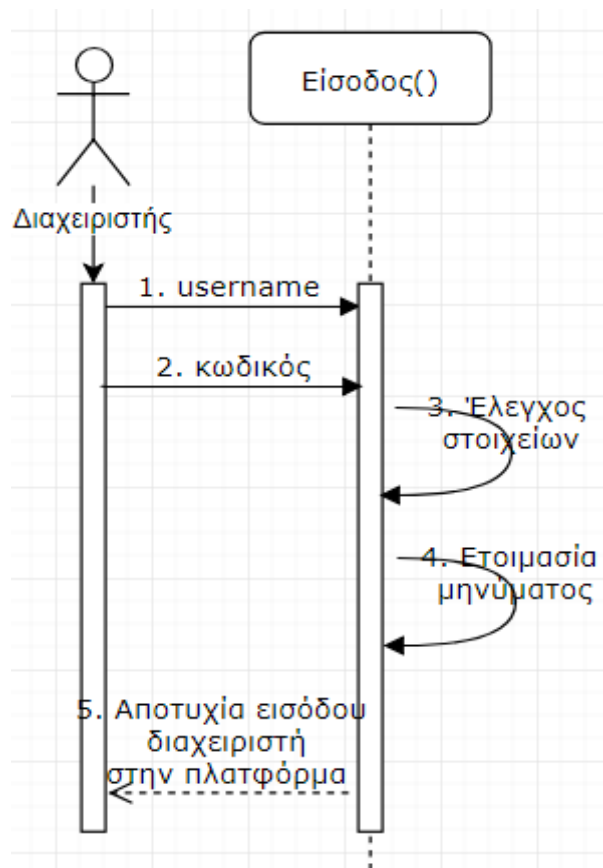
5.2.1.3 Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για επιτυχή είσοδο διαχειριστή στο σύστημα



Σχήμα 5.2.1.10: Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για επιτυχή είσοδο διαχειριστή στο σύστημα

Σύμφωνα με το πιο πάνω σχήμα (σχήμα 5.2.1.10), μέσω της λειτουργίας αυτής, ο διαχειριστής – διδάσκοντας μπορεί να κάνει είσοδο (login) στο σύστημα χρησιμοποιώντας το username του και τον κωδικό του. Ελέγχονται τα στοιχεία αυτά, και αφού είναι σωστά, ο διαχειριστής μπορεί να προσθέσει ή να αφαιρέσει χρήστες – φοιτητές, και επίσης ανάλογα με το κεφάλαιο που επιθυμεί, μπορεί να επιλέξει δραστηριότητα να ασχοληθεί.

5.2.1.4 Διάγραμμα Αλληλεπίδρασης (Interaction Diagrams) για αποτυχή είσοδο διαχειριστή στο σύστημα



Σχήμα 5.2.1.11: Διάγραμμα Αλληλεπίδρασης (Interaction Diagram) για αποτυχή είσοδο διαχειριστή στο σύστημα

Σύμφωνα με το πιο πάνω σχήμα (σχήμα 5.2.1.11), μέσω της λειτουργίας αυτής, ο διαχειριστής μπορεί να κάνει είσοδο (login) στο σύστημα χρησιμοποιώντας το username του και τον κωδικό του. Ελέγχονται τα στοιχεία αυτά, και σε περίπτωση που είναι λάθος, επιστρέφεται μήνυμα λάθους.

Κεφάλαιο 6

Υλοποίηση Συστήματος

6.1 Εισαγωγή	48
6.2 Παρουσίαση εργαλείων, γλωσσών και βοηθητικών προγραμμάτων που χρησιμοποιήθηκαν	49
6.2.1 Java	49
6.2.2 Hyper Text Markup Language – HTML	50
6.2.3 Cascading Styling Sheet - CSS	50
6.2.4 JavaScript	51
6.2.5 JSP	51
6.2.6 Bootstrap	51
6.2.7 MySQL	52
6.2.8 Draw.io	52
6.2.9 PHP Hypertext Preprocessor	52
6.2.10 Eclipse IDE for Java Developers	53
6.2.11 Microsoft Word	53
6.2.12 Φόρμες Google	54
6.2.13 Windows Live Movie Maker	54
6.3 Υλοποίηση Διαδικτυακής Σελίδας	54
6.3.1 Διαδικασία υλοποίησης	54
6.3.2 Προβλήματα που αντιμετωπίστηκαν	55
6.4 Παρουσίαση Συστήματος	57

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο, θα αναφερθούμε αρχικά στα εργαλεία, στις γλώσσες προγραμματισμού, στην πλατφόρμα Eclipse JEE, στους διακομιστές (server) καθώς και στην βάση δεδομένων, που χρησιμοποιήθηκαν για την επιτυχή κατασκευή του συστήματος υποστήριξης φοιτητών για τις Αρχές Προγραμματισμού Ι. Στην συνέχεια, θα περιγράψουμε

λεπτομερώς όλη την διαδικασία που ακολουθήσαμε για να καταφέρουμε να υλοποιήσουμε το σύστημα, καθώς και τις δυσκολίες που αντιμετωπίσαμε κατά την διάρκεια της υλοποίησης. Θα παρουσιαστούν εικόνες, οι οποίες παρουσιάζουν την λειτουργία του συστήματος μαζί με μια μικρή περιγραφή για να μπορεί να γίνει πιο κατανοητή η λειτουργικότητα του συστήματος. Σκοπός του κεφαλαίου αυτού, είναι η παρουσίαση της φάσης της υλοποίησης της ανάπτυξης του παρόντος έργου ανάπτυξης λογισμικού, με απώτερο στόχο τη διευκόλυνση και την εξομάλυνση της φάσης συντήρησης του συστήματος καθώς και για την καλύτερη κατανόηση των επιμέρους διαδικασιών που ακολουθήθηκαν μέχρι να ολοκληρωθεί η πλατφόρμα.

6.2 Παρουσίαση εργαλείων, γλωσσών και βοηθητικών προγραμμάτων που χρησιμοποιήθηκαν

6.2.1 Java

Η Java είναι μια αντικειμενοστραφής γλώσσα γενικού σκοπού που βασίζεται στην δομημένη σε κλάσεις λογική προγραμματισμού. Σκοπός της Java είναι να επιτρέπει στους προγραμματιστές να γράφουν κώδικα ο οποίος εκτελείται οπουδήποτε (write once run anywhere), που σημαίνει ότι ο μεταγλωτισμένος κώδικας Java μπορεί να τρέξει σε όλες τις πλατφόρμες που υποστηρίζουν την Java, χωρίς την ανάγκη να γίνει ξανά μεταγλώττιση. Χρησιμοποιώντας την Java, ένας προγραμματιστής μπορεί να γράψει προγράμματα που μπορούν να εκτελεστούν μέσα από μια ιστοσελίδα για να μπορέσει να αλληλεπιδράσει με τον υπολογιστή του χρήστη.

Μέσω της πλατφόρμας Eclipse JEE και του TomCat server, τρέχουμε Java Servlets (κώδικας Java στην μεριά του server) για να υλοποιήσουμε το παιχνίδι συμπλήρωσης κώδικα στη Java. Στο παιχνίδι αυτό, ο χρήστης (φοιτητής), συμπληρώνει τα κομμάτια κώδικα που λείπουν από το κείμενο, και ακολούθως ο συμπληρωμένος Java κώδικας από τον φοιτητή, τρέχει στον server και γίνεται έλεγχος ορθότητας και επιστροφής των κατάλληλων μηνυμάτων.

6.2.2 Hyper Text Markup Language – HTML

Η HTML είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML, τα οποία αποτελούνται από ετικέτες (tags). Ο σκοπός ενός web browser, είναι να διαβάζει τα έγγραφα HTML και να συνθέτει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας. Τα στοιχεία της HTML, χρησιμοποιούνται για να κτίσουν όλους τους ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. [5]

Ένα μεγάλο ποσοστό των αρχείων έχει δημιουργηθεί σε HTML κώδικα, εφ' όσον είναι η κύρια γλώσσα για να γράψουμε και να παρουσιάσουμε το περιεχόμενο μιας ιστοσελίδας.

6.2.3 Cascading Styling Sheet - CSS

Η CSS είναι μια γλώσσα υπολογιστή, που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους, που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Δηλαδή, χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στην γλώσσα HTML (ιστοσελίδας). Ο σκοπός αυτής, είναι να διαμορφώνει τα χαρακτηριστικά, τα χρώματα, τη στοίχιση μιας ιστοσελίδας. Με λίγα λόγια ο σκοπός της είναι να ομορφαίνει και να εμπλουτίζει την παρουσίαση ενός εγγράφου HTML. [6]

Έχει χρησιμοποιηθεί για την υλοποίηση της γραφικής διαπροσωπείας σε όλα τα HTML και JSP αρχεία που περιέχονται στην πλατφόρμα.

6.2.4 JavaScript

Η JavaScript είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για τη δημιουργία διαδραστικών ιστοσελίδων και δυναμικών συναρτήσεων από την μεριά του χρήστη. Παρ' όλο που οι σελίδες HTML είναι καλές για την εμφάνιση στατικού περιεχομένου, εν τούτοις, σπάνια συναντάς στις μέρες μας στατικές σελίδες. Η JavaScript βοηθά τους προγραμματιστές για την αλληλεπίδραση και τον μετασχηματισμό της στατικής σελίδας HTML, σε δυναμική.

Έχει χρησιμοποιηθεί για την δυναμική εκτέλεση κάποιων λειτουργιών από την μεριά του client. Έχει χρησιμοποιηθεί εκτενώς για τα παιχνίδια πολλαπλών επιλογών, καθώς για τα προγράμματα συμπλήρωσης κώδικα, τόσο για την εμφάνιση δυναμικών κουμπιών, για την εμφάνιση επιλεκτικών εικόνων και δυναμικής αλλαγής του περιεχομένου.

6.2.5 JSP

Τα Servlets είναι προγράμματα Java με ενσωματωμένο κώδικα Hyper Text Markup Language (HTML). Υπό αυτή την έννοια, οι JSP είναι ιστοσελίδες HTML με ενσωματωμένο κώδικα Java. Ο κώδικας των Servlets, είναι δομημένος σε κλάσεις, που γράφονται, μεταγλωττίζονται και εκτελούνται όπως κάθε κλάση Java, ενώ οι JSP αποτελούνται από στατικό κώδικα HTML και δυναμικά παραγόμενο περιεχόμενο HTML, το οποίο είναι διαχωρισμένο με ειδικές ετικέτες (tags). Τα Servlets και οι JSP θεωρούνται από πολλούς ταυτόσημα, επειδή οι JSP μεταφράζονται σε Servlets, δηλαδή βασίζονται στην τεχνολογία των Java Servlets. Τα Servlets είναι κατάλληλα για εργασίες προσανατολισμένες στην λογική και στην επεξεργασία, ενώ οι JSP είναι πιο κατάλληλες για εργασίες προσανατολισμένες στην παρουσίαση. [7]

Έχει χρησιμοποιηθεί για την υλοποίηση του παιχνιδιού Java.

6.2.6 Bootstrap

Το Bootstrap είναι ένα εργαλείο ανοικτού κώδικα που βοηθά τους προγραμματιστές να δημιουργήσουν ένα φιλικό γραφικό περιβάλλον ως προς την χρήση. Η χρήση του είναι πολύ

εύκολη και έχει κοινό στυλ με τα περισσότερα στοιχεία της HTML. Υπάρχουν πολλά components που μπορούν να χρησιμοποιηθούν και κάποια έχουν συνδεδεμένα πρόσθετα της JavaScript. [9]

Το Bootstrap χρησιμοποιήθηκε μόνο για την εμφάνιση του modal στα παιχνίδια που περιέχουν επίπεδο βοήθειας, καθώς επίσης και σε όλα τα παιχνίδια που περιέχουν απάντηση – λύση του προβλήματος

6.2.7 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που χρησιμοποιείται κυρίως σε διαδικτυακές εφαρμογές. Τρέχει στη μεριά του εξυπηρετητή και θεωρείται πολύ γρήγορη, αξιόπιστη και εύκολη στη χρήση. Τα δεδομένα στη βάση δεδομένων αποθηκεύονται μέσα σε πίνακες. Ένας πίνακας είναι μια συλλογή από δεδομένα που αποτελείται από γραμμές και στήλες. Έχει χρησιμοποιηθεί για την αποθήκευση των προσωπικών στοιχείων των φοιτητών και των διαχειριστών

Έχει χρησιμοποιηθεί για την υλοποίηση του login, καθώς επίσης και για την διαγραφή και προσθήκη νέων μελών (φοιτητών) από τον διαχειριστή.

6.2.8 Draw.io

Το Draw.io είναι ένα δωρεάν διαδικτυακό λογισμικό κατασκευής περίπλοκων διαγραμμάτων διαφόρων τύπων. Έχει χρησιμοποιηθεί για τη δημιουργία Διαγραμμάτων Ροής Δεδομένων (Data Flow Diagram – DFD), Διαγραμμάτων Σεναρίου Χρήσης (Use Case Diagrams), Διαγραμμάτων Οντοτήτων Συσχετίσεων (Entity Relationship Diagram) και Διαγραμμάτων Αλληλεπίδρασης (Interaction Diagrams).

6.2.9 PHP Hypertext Preprocessor

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό

περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργαστεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script. Για την ορθή εκτέλεση της γλώσσας PHP σε ένα HTML αρχείο ή αντίστροφα., πρέπει να χρησιμοποιηθεί ένας διακομιστής ο οποίος χρησιμοποιείτε σε διάφορα λειτουργικά συστήματα. Είναι ευρέως διαδεδομένη λόγω του ότι είναι εύκολο να δημιουργηθεί κώδικας με αυτή. Επιπλέον, μπορεί να βρεθεί σχεδόν σε όλους τους διακομιστές, και είναι ανοικτού κώδικα, οπότε είναι άμεσα διαθέσιμη και δωρεάν. Ο χρόνος ανταπόκρισης των προγραμμάτων της PHP είναι πολύ μικρός. [8]

Έχει χρησιμοποιηθεί για την υλοποίηση των λειτουργιών εισόδους του χρήστη και διαχειριστή στο σύστημα (login). Επίσης έχει χρησιμοποιηθεί για την υλοποίηση της διαγραφής και προσθήκης χρήστη στο σύστημα από τον διαχειριστή.

6.2.10 Eclipse IDE for Java EE Developers

Εργαλεία για προγραμματιστές Java που δημιουργούν εφαρμογές Java EE και Web. Το IDE για Java EE υποστηρίζει την ανάπτυξη επιχειρηματικών εφαρμογών χρησιμοποιώντας το J2EE - Servlets / JSP / EJB κλπ. Υπάρχει η δυνατότητα προσθήκης ενός διακομιστή σε runtime J2EE με server επιλογής (TomCat, Glassfish, Weblogic) και δυνατότητα αποσφαλμάτωσης εφαρμογών.

Η πλατφόρμα αυτή, χρησιμοποιήθηκε για την ανάπτυξη και σύνδεση των Java servlets μαζί τον TomCat server.

6.2.11 Microsoft Word

Η Microsoft Word είναι η πιο συχνά χρησιμοποιούμενη εφαρμογή για επεξεργασία κειμένου. Το τρέχον έγγραφο της ατομικής διπλωματικής εργασίας έχει γραφτεί χρησιμοποιώντας αυτή την εφαρμογή.

6.2.12 Φόρμες Google

Οι φόρμες Google έχουν χρησιμοποιηθεί για τη δημιουργία του ερωτηματολογίου κατά τη φάση συλλογής απαιτήσεων. Είναι μια απλή, εύκολη, και φιλική προς τον χρήστη σελίδα, όπου μπορεί κάποιος να δημιουργήσει και να αποστείλει φόρμες μέσω διαφόρων εφαρμογών. Για παράδειγμα, στην περίπτωση μας χρησιμοποιήθηκε το ηλεκτρονικό ταχυδρομείο. Επιπλέον, ανάλογα με τις απαντήσεις, τα αποτελέσματα καταγράφονται και αναδεικνύονται μέσω γραφικών παραστάσεων.

6.2.13 Windows Live Movie Maker

Το πρόγραμμα Windows Live Movie Maker είναι ένα πολύ χρήσιμο και εύκολο στη χρήση πρόγραμμα για τη δημιουργία και επεξεργασία βίντεο. Το πρόγραμμα αυτό χρησιμοποιήθηκε για την επεξεργασία των βίντεο που βρίσκονται στις διαλέξεις.

6.3 Υλοποίηση Διαδικτυακής Σελίδας

6.3.1 Διαδικασία υλοποίησης

Αρχικά, έγινε η υλοποίηση του μενού χρησιμοποιώντας HTML και CSS κώδικα καθώς και το framework Bootstrap. Μετά τη δημιουργία του μενού, προχωρήσαμε προς την υλοποίηση των διαλέξεων, καταγράφοντας επιλεκτικά κομμάτια της ύλης. Επίσης, για κάποια από τα δυσνόητα κομμάτια της θεωρίας, τοποθετήθηκαν παραδείγματα και βίντεο. Για μερικά παραδείγματα χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java. Όσον αφορά τα βίντεο που εμφανίζονται στην πλατφόρμα, επιλέχθηκαν τα πιο επεξηγηματικά, σαφή και κατανοητά με βάση τις γνώσεις ενός αρχάριου, ενώ τα βίντεο αυτά, τα επεξεργαστήκαμε μέσω του προγράμματος Movie Maker.

Στην συνέχεια υλοποιήθηκε το παιχνίδι με τις ερωτήσεις πολλαπλής επιλογής. Για την παρουσία του περιεχομένου, χρησιμοποιήθηκαν οι γλώσσες HTML, CSS, framework Bootstrap, ενώ για τη λειτουργικότητα του παιχνιδιού χρησιμοποιήθηκε η γλώσσα προγραμματισμού JavaScript. Κάποιες λειτουργίες των ερωτήσεων πολλαπλής επιλογής,

είναι η εμφάνιση πολλαπλών βοηθειών (ανάλογα με το ερώτημα, δημιουργούσαμε και διαφορετικό αριθμό βοηθειών), χρωματισμός επιλογής απάντησης (πράσινο χρώμα σωστή επιλογή, ενώ με κόκκινο χρώμα η λάθος επιλογή), εμφάνιση εικόνων μέσα στο παιχνίδι (είτε σαν ερώτηση, είτε σαν επιλογή, είτε σαν σωστή απάντηση, είτε σαν βοήθεια) και εμφάνιση απάντησης με χρήση Modal.

Τέλος, έγινε η υλοποίηση του παιχνιδιού συμπλήρωσης κώδικα Java. Για τη λειτουργία αυτή, έγινε χρήση του εργαλείου Eclipse Jee και του διακομιστή (server) TomCat v8.5. Για να ενωθεί ο server μαζί με το Eclipse Jee, έπρεπε να γίνουν οι κατάλληλες ρυθμίσεις τόσο πάνω στο ίδιο το Eclipse, όσο και στον ίδιο τον υπολογιστή. Σαν πρώτο στάδιο έγινε η συγγραφή κώδικα σε Java servlet, όπου στέλλοντας μέσω μια φόρμας HTML τον κώδικα της Java, έπρεπε να γίνει πρώτα το HTTP request για να ζητήσει την πληροφορία αυτή. Στο Java Servlet, παίρναμε την πληροφορία αυτή βάσει του id element που περιείχε τον κώδικα Java ώστε να την επεξεργαστούμε. Παίρνοντας τον κώδικα Java αυτό, τον γράψαμε σε ένα άλλο αρχείο και τον αποθηκεύσαμε. Στην συνέχεια μέσω της Java, δημιουργήσαμε μια νέα διεργασία (process) ώστε να εκτελέσουμε ένα batch αρχείο πάνω στον υπολογιστή μας. Αυτό το batch αρχείο, καλεί μέσω της κονσόλας, ένα jar αρχείο. Το jar αρχείο αυτό, περιέχει κώδικα java, ο οποίος διαβάζει ένα αρχείο (στην περίπτωση μας αυτό που γράφτηκε από το Servlet), το κάνει compile και ανάλογα αν τρέχει ή όχι βγάζει το αποτέλεσμα σε ένα άλλο αρχείο out.txt. Το αποτέλεσμα του αρχείου out.txt το διαβάζουμε αργότερα από το Java servlet. Το αποτέλεσμα αυτό αποστέλλεται πίσω στα HTML/JSP αρχεία ανοίγοντας ένα session (request.session) καθώς και με τη χρήση της συνάρτησης setAttribute. Ανάλογα με το id του element που θέλαμε να καταγραφεί η απάντηση, το χρησιμοποιήσαμε στη συνάρτηση setAttribute. Όλα τα πιο πάνω που προαναφέρθηκαν, έπρεπε επίσης να συγχρονιστούν, ώστε η ανάγνωση από τα αρχεία, η μεταγλώττιση και η εκτέλεση του κώδικα καθώς και η επιστροφή της απάντησης να γίνεται στο σωστό χρόνο.

6.3.2 Προβλήματα που αντιμετωπίστηκαν

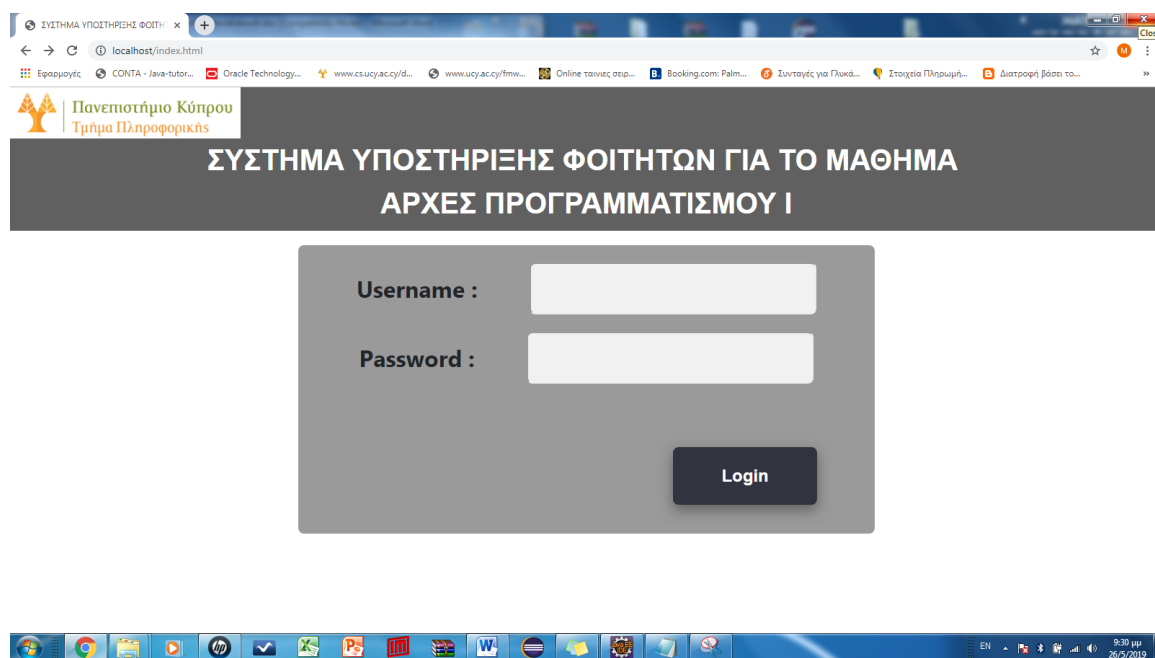
Τα προβλήματα που αντιμετωπίστηκαν κατά την ανάπτυξη του συστήματος είναι αρκετά και θα μιλήσουμε για κάποια από αυτά. Καταρχάς, ένα πρόβλημα που αντιμετωπίσαμε, ήταν σε ποιιά/ές γλώσσα/ες προγραμματισμού θα αναπτυσσόταν το σύστημα υποστήριξης των φοιτητών του μαθήματος Αρχές Προγραμματισμού Ι. Παρά το γεγονός ότι καταλήξαμε πως το σύστημα θα αναπτυσσόταν πάνω σε τεχνολογίες διαδικτύου (ώστε να υπάρχει το

λογισμικό στο διαδίκτυο), εν τούτοις, δεν είχαμε καμία γνώση για τις τεχνολογίες αυτές. Έπρεπε να μάθουμε ποιές γλώσσες ήταν οι πιο ευκολόχρηστες και οι πιο πλούσιες σε υλικό, ενώ ταυτόχρονα έπρεπε να μάθουμε και πάνω σε πιο editor έπρεπε να γραφτούν οι τεχνολογίες αυτές. Επιπλέον, έγινε και η χρήση ερωτηματολογίου ώστε να μας βοηθήσει να καταλήξουμε στην μορφή που θα είχε το τελικό μας σύστημα, καθώς και πώς θα βοηθούσαμε τους φοιτητές αυτούς. Ένα πρόβλημα στο στάδιο αυτό, ήταν να βρούμε πρόθυμους φοιτητές να απαντήσουν στο ερωτηματολόγιο, όπως επίσης και στην αμέλεια κάποιων φοιτητών να απαντούν εντελώς τυχαία τα ερωτήματα. Μετά από κάποια μελέτη, βρήκαμε τους φοιτητές με τις πιο αξιόπιστες απαντήσεις. Στην συνέχεια, ένα από τα μεγαλύτερα προβλήματα που είχαμε να αντιμετωπίσουμε, ήταν να μάθουμε όλες αυτές τις γλώσσες προγραμματισμού (HTML, CSS, PHP, JavaScript) χωρίς να έχουμε κανένα υπόβαθρο στις τεχνολογίες αυτές. Κατά την ανάπτυξη της πλατφόρμας, χρειαστήκαμε ένα μεγάλο μερίδιο του χρόνου μας, διορθώνοντας λάθη, λόγω της έλλειψης γνώσης πάνω στις τεχνολογίες αυτές. Το ίδιο ίσχυε και για την πλατφόρμα Eclipse JEE, σε συνδυασμό με την χρήση του TomCat server, τα οποία ήταν εντελώς άγνωστα. Λόγω συμβατικότητας του server με το Eclipse και του λειτουργικού συστήματος του υπολογιστή, χρειαστήκαμε να κάνουμε κάποιες διαδικασίες μέχρι να ενωθεί ο server και χρειάστηκε αρκετό χρονικό διάστημα. Ακόμη ένα μεγάλο πρόβλημα που είχαμε να αντιμετωπίσουμε, ήταν πώς να εκτελέσουμε κώδικα Java πάνω στην μεριά του Server, και να επιστρέψει η απάντηση πίσω στον χρήστη. Έπρεπε να μάθουμε την τεχνολογία των Java servlets, όπου έπρεπε να στέλλουμε τον κώδικα από την μεριά του client και έπρεπε κάπως ο server, να εκτελεί και να επεξεργάζεται τον κώδικα αυτό, καθώς και να επιστρέφει πίσω το αποτέλεσμα. Μεγάλο πρόβλημα με την τεχνολογία των Java servlets αφορούσε την ανανέωση της σελίδας μετά από κάποια αλλαγή. Παρ' όλο που αλλάζαμε το περιεχόμενο κάποιου αρχείου, η σελίδα δεν άλλαζε. Χρειάστηκε καιρό μέχρι να ανακαλύψουμε πως το περιεχόμενο της σελίδας, φυλαγόταν τοπικά στην cache του browser. Ο μόνος τρόπος για να φανεί η αλλαγή ήταν να πατήσουμε clt+F5 [15], καθώς να γίνει και refresh η σελίδα του server. Τώρα, όσον αφορά τη εκτέλεση και επιστροφή του κώδικα, έπρεπε να βρούμε τρόπο να εκτελούμε το κώδικα από την μεριά του διακομιστή και να επιστρέφουμε την απάντηση πίσω. Δημιουργήσαμε ένα jar file, το οποίο δημιουργεί νέες διεργασίες, μεταγλωττίζει ένα java αρχείο, καθώς και το τρέχει. Το αποτέλεσμα αυτού του εκτελέσιμου java αρχείου, το αποθηκεύουμε σε ένα txt αρχείο, ενώ μέσω exceptions, φυλάγουμε το αποτέλεσμα του αρχείου αυτού, ακόμη και όταν δεν μπορεί να μεταγλωττιστεί. Εκτός αυτού, μέσω του κώδικα Java από την μεριά του server, δεν μπορούσαμε να τρέξουμε το jar file, έτσι έπρεπε να δημιουργήσουμε windows batch file, για να τρέξουμε το jar file αυτό. Ακόμη ένα πρόβλημα που αντιμετωπίσαμε, λίγες μέρες πριν την παρουσίαση της διπλωματικής εργασίας, ήταν ότι δεν μπορούσε ο server να

βρει την Java μέσα στο path, και μας έδινε μηνύματα λάθους στο request. Έπρεπε να δημιουργήσουμε νέο project στο Eclipse JEE, ωστόσο να δουλέψει σωστά ο Server.

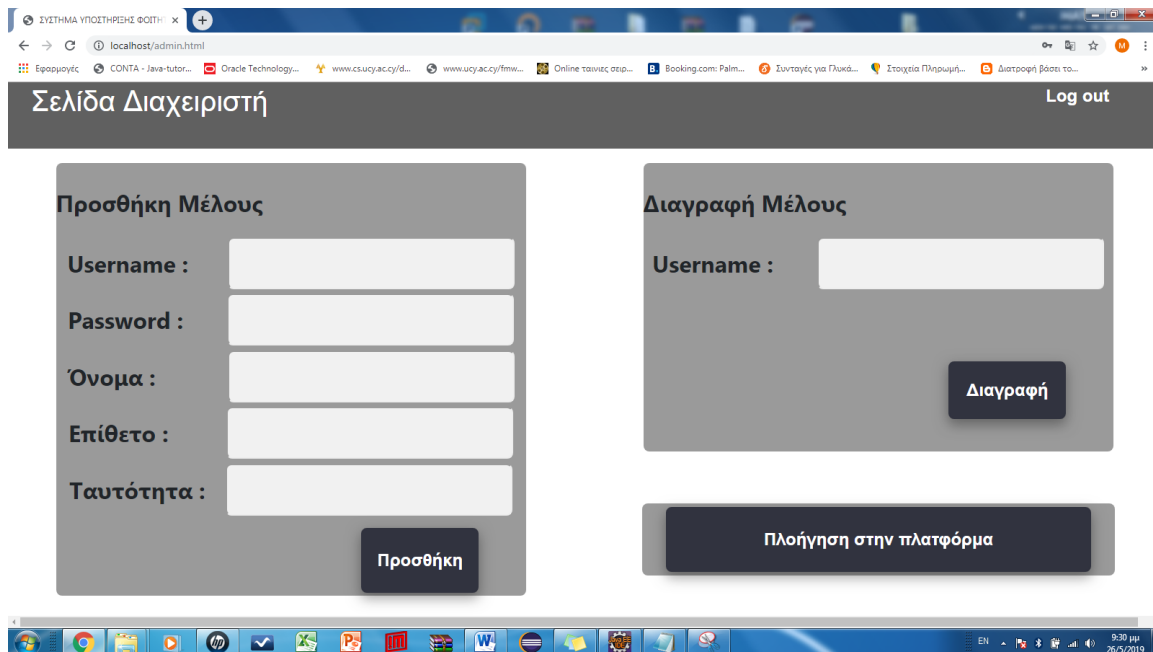
Συνοψίζοντας, κατά την διάρκεια ανάπτυξης της πλατφόρμας, αντιμετωπίσαμε πάρα πολλά προβλήματα που χρειάστηκαν διάβασμα και χρόνο για να λυθούν, ενώ στο πάνω μέρος αναφερθήκαμε μόνο για ένα υποσύνολο των προβλημάτων αυτών που αντιμετωπίσαμε.

6.4 Παρουσίαση Συστήματος



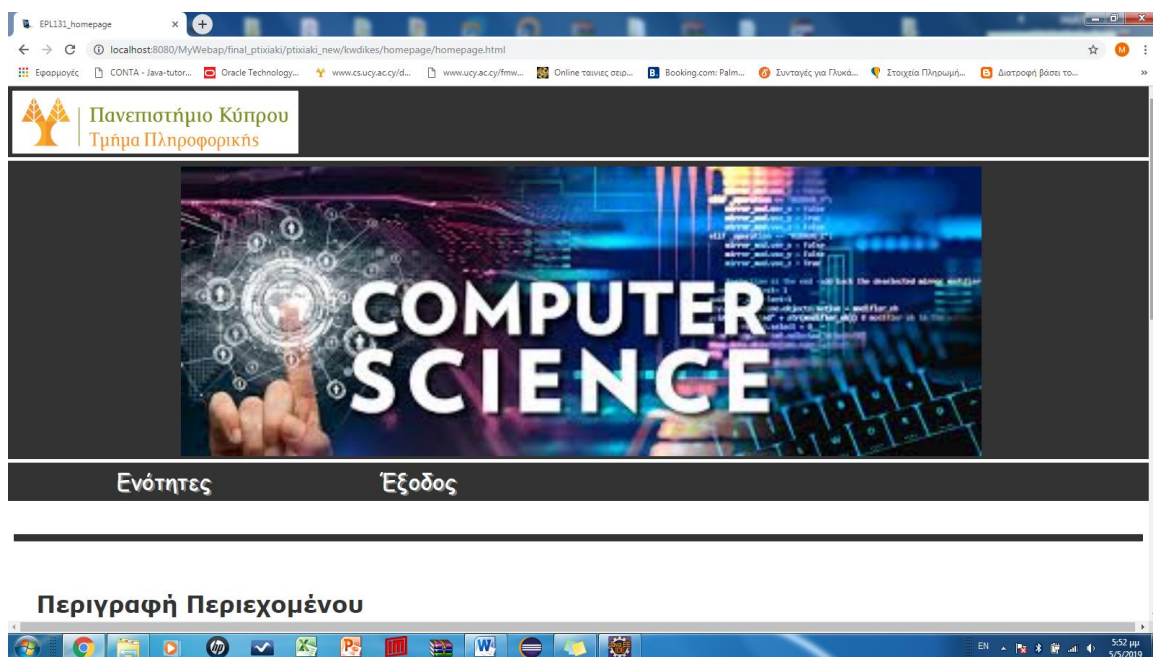
Εικόνα 6.4.1

Στην εικόνα 6.4.1 βλέπουμε την οθόνη για το login του χρήστη και του διαχειριστή. Ανάλογα με το username που δίνεται, αν δηλαδή είναι του διαχειριστή ή του χρήστη, μεταφερόμαστε και στην ανάλογη σελίδα.

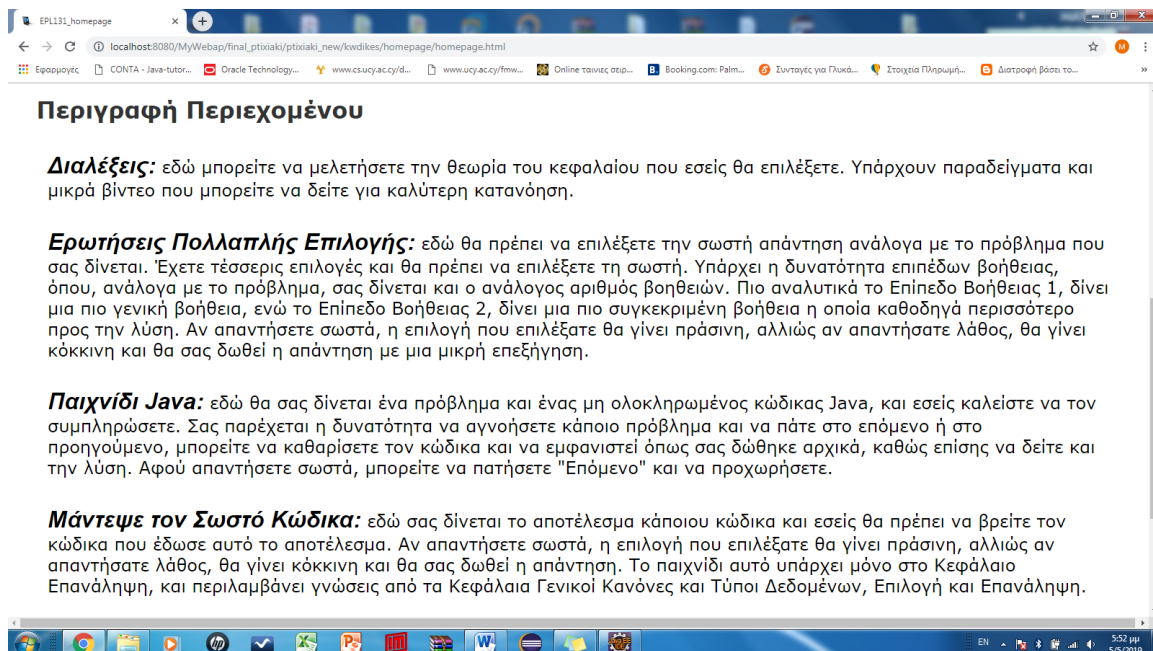


Εικόνα 6.4.2

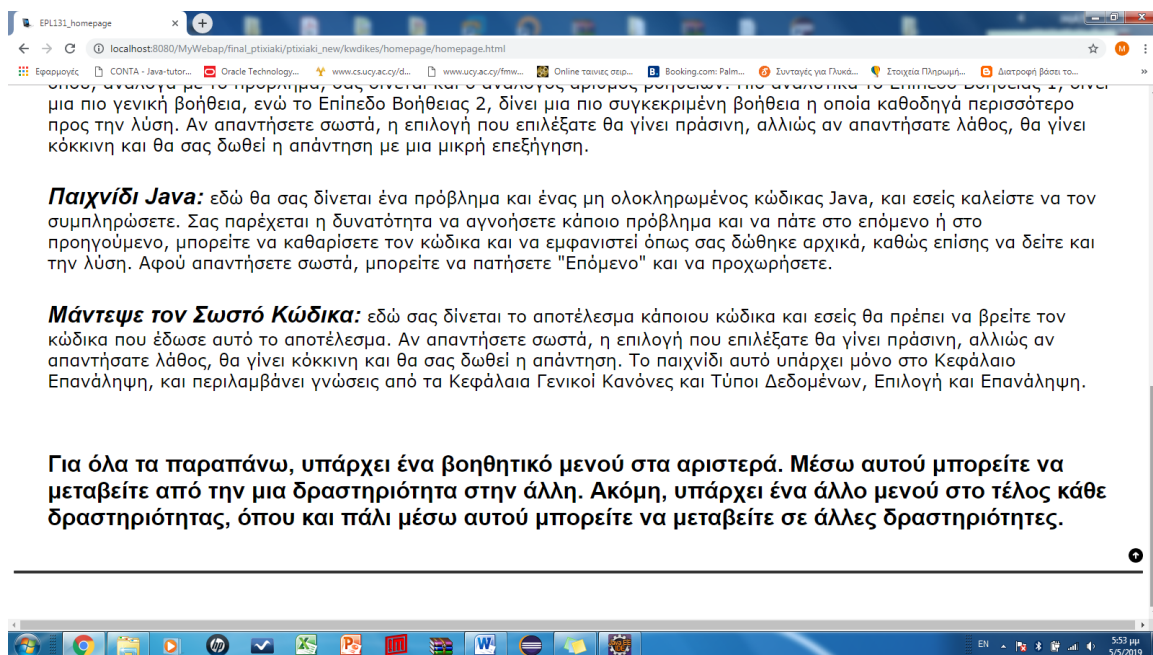
Στην εικόνα 6.4.2 βλέπουμε την οθόνη αφού κάνει login ο διαχειριστής στο σύστημα. Ο διαχειριστής εκτός από τη δυνατότητα πλοήγησης στο σύστημα, μπορεί να προσθέσει ή να αφαιρέσει μέλη από το σύστημα.



Εικόνα 6.4.3

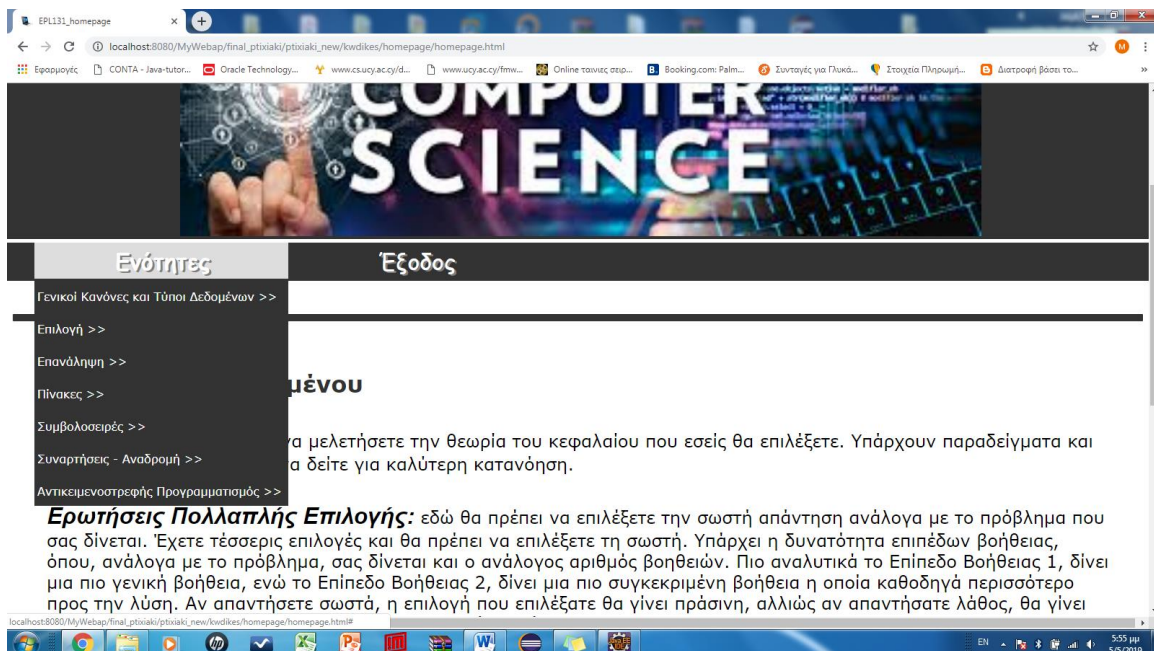


Εικόνα 6.4.4

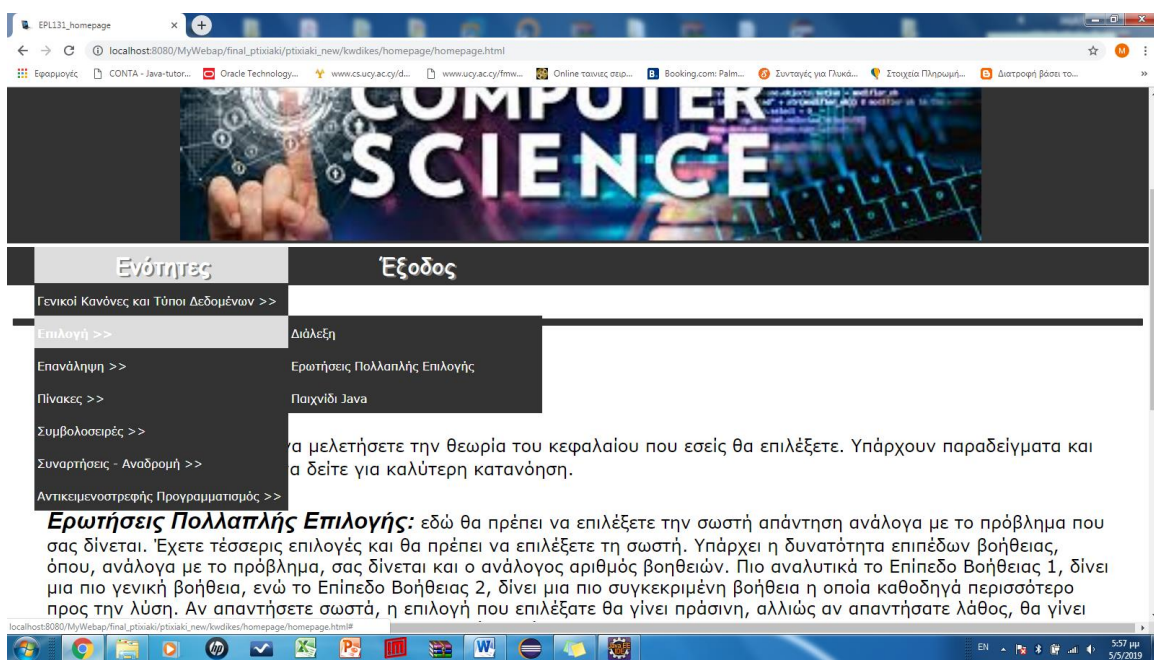


Εικόνα 6.4.5

Στις εικόνες 6.4.3, 6.4.4, και 6.4.5, φαίνεται η αρχική σελίδα. Η αρχική σελίδα εμφανίζεται αφού κάνει είσοδο (login) ο χρήστης στο σύστημα. Η αρχική σελίδα περιέχει το μενού και μια σύντομη επεξήγηση του τι περιέχει η πλατφόρμα. Το μενού περιέχει δύο επιλογές, Ενότητες και Έξοδο.



Εικόνα 6.4.6

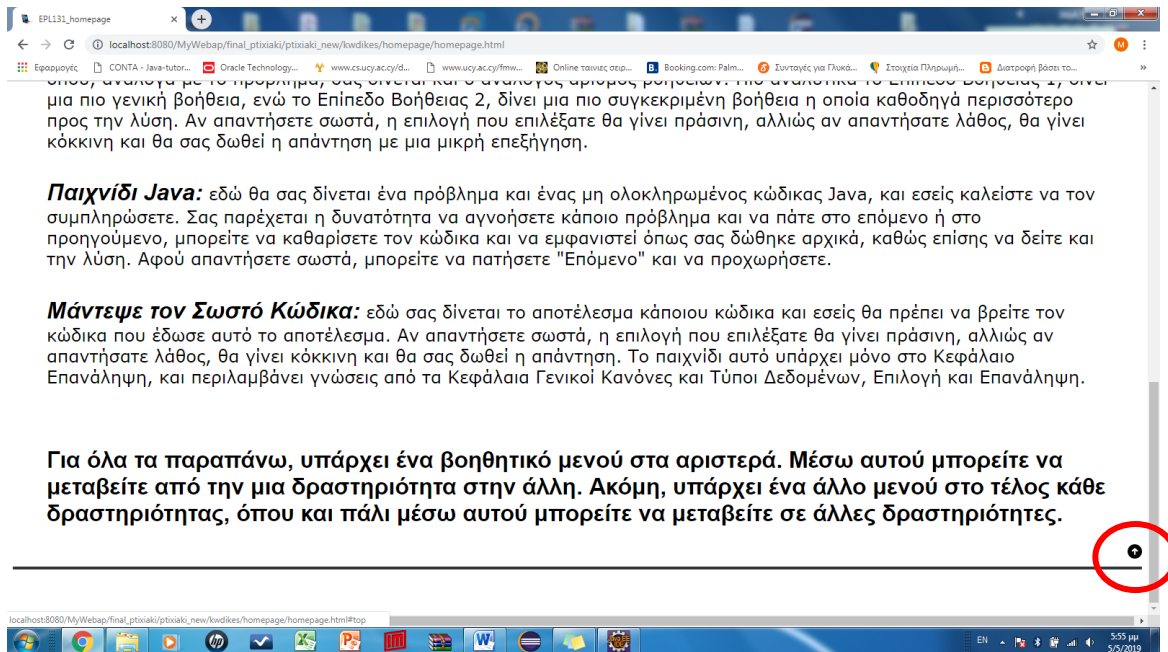


Εικόνα 6.4.7

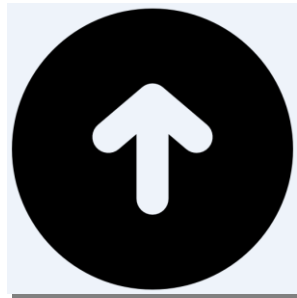


Εικόνα 6.4.8

Πατώντας ο χρήστης το κουμπί «Ενότητες», όπως φαίνεται στην Εικόνα 6.4.6, ξεδιπλώνεται μια λίστα με τις ενότητες με τις οποίες είναι δομημένη η πλατφόρμα. Όπως φαίνεται στις εικόνες 6.4.6, 6.4.7 και 6.4.8 οι ενότητες που περιέχει είναι οι εξής: Γενικοί Κανόνες και Τύποι Δεδομένων, Επιλογή, Επανάληψη, Πίνακες, Συμβολοσειρές, Συναρτήσεις – Αναδρομή, Αντικειμενοστρεφής Προγραμματισμός. Επίσης, για τις ενότητες Γενικοί Κανόνες και Τύποι Δεδομένων, Επιλογή, Πίνακες, Συμβολοσειρές, Συναρτήσεις – Αναδρομή και Αντικειμενοστρεφής Προγραμματισμός ο χρήστης έχει τη δυνατότητα να επιλέξει να ασχοληθεί μεταξύ διάλεξης, ερωτήσεων πολλαπλής επιλογής και παιχνιδιού Java. Για την ενότητα Επανάληψη, έχει τη δυνατότητα να επιλέξει μεταξύ διάλεξης, ερωτήσεων πολλαπλής επιλογής, παιχνιδιού «Μάντεψε τον σωστό κώδικα» και παιχνιδιού Java.



Εικόνα 6.4.9α



Εικόνα 6.4.9β

Στο τέλος της αρχικής σελίδας όπως φαίνεται από την εικόνα 6.4.9α, βρίσκεται ένα μικρό εικονίδιο (εικόνα 6.4.9β). Πατώντας το σε μεταφέρει στην αρχή της σελίδας.

ΕΡΛ131

localhost:8080/MyWebapp/final_ptixiaki/ptixiaki_new/kwdikes/dialekseis_html/dialeksi_epilogi/dialeksi_epilogi.html

Εφαρμογές

CONTA - Java-tutor... Oracle Technology... www.cs.ucy.ac.cy/d... www.ucy.ac.cy/fmk... Online ταινίες στο... Booking.com: Palm... Συνταγές για Γλυκά... Στοιχεία Πληρωμ... Διατροφή βάσει το...

Αλγοριθμικές Δομές Επιλογής

The diagram shows two types of control flow. On the left, a vertical sequence of four boxes labeled 'εντολή 1', 'εντολή 2', 'εντολή 3', and 'εντολή 4' is connected by downward arrows, representing sequential flow. On the right, a flowchart starts with a decision diamond 'συνθήκη 1'. If 'true', it goes to 'εντολή 1', then to another decision diamond 'συνθήκη 2'. If 'συνθήκη 2' is 'true', it goes to 'εντολή 2', and if 'false', it goes to 'εντολή 3'. If 'συνθήκη 1' is 'false', it goes directly to 'εντολή 3'. Below the sequential flow is the text 'ακολουθιακή ροή ελέγχου'. Below the conditional flow is the text 'ροή ελέγχου με συνθήκες και βρόγχους'.

Υπάρχουν δύο ειδών ροές ελέγχου. Η ακολουθιακή ροή ελέγχου και η ροή ελέγχου με συνθήκες και βρόγχους. Στην ακολουθιακή ροή, το πρόγραμμα εκτελεί όλες τις

EN 5:58 μμ 5/5/2019

Εικόνα 6.4.10

ΕΡΛ131

localhost:8080/MyWebapp/final_ptixiaki/ptixiaki_new/kwdikes/dialekseis_html/dialeksi_epilogi/dialeksi_epilogi.html

Εφαρμογές

CONTA - Java-tutor... Oracle Technology... www.cs.ucy.ac.cy/d... www.ucy.ac.cy/fmk... Online ταινίες στο... Booking.com: Palm... Συνταγές για Γλυκά... Στοιχεία Πληρωμ... Διατροφή βάσει το...

συνθήκη με τιμή true. Για λόγους αποδοτικότητας θα ήταν καλύτερα να γράψουμε στα αριστερά την συνθήκη με το λιγότερο κόστος έτσι ώστε σε περίπτωση που η έκφραση μας γίνει ψευδής, να μην προχωρήσει στις επόμενες εκφράσεις.

The screenshot shows an IDE with a code editor. The code is in Java and shows a conditional statement. The code is as follows:

```

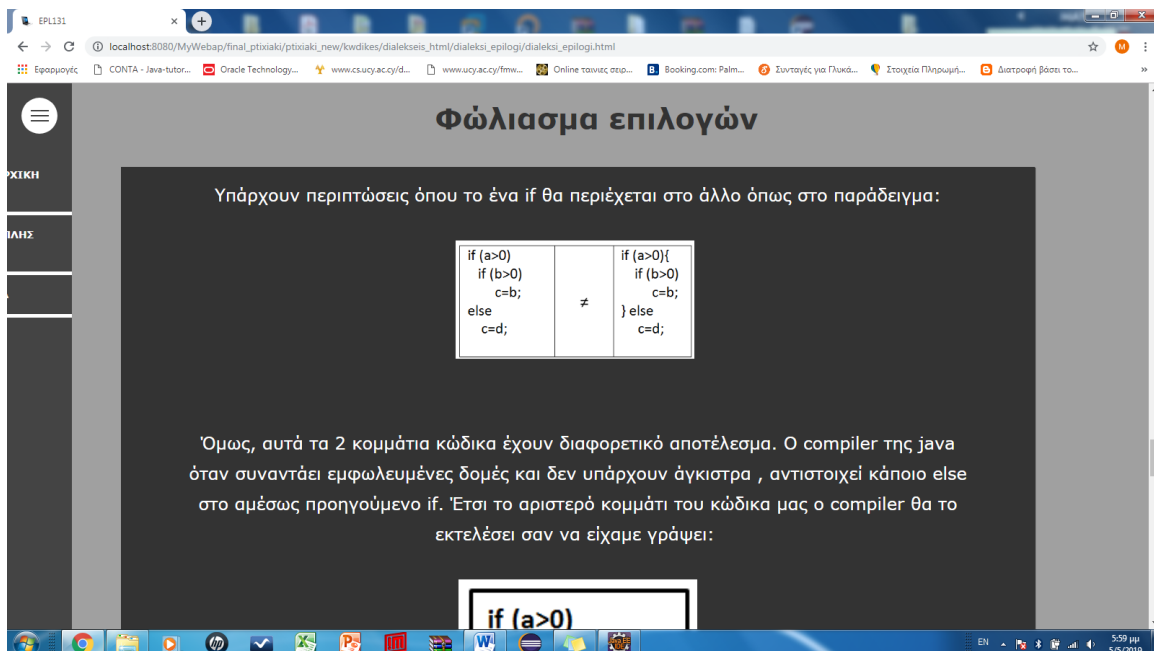
1 public boolean isEven(int n) {
2     return n % 2 == 0;
3 }
4
5 public static void main(String[] args) {
6     int n = 4;
7     if (isEven(n)) {
8         System.out.println("Ο αριθμός " + n + " είναι άρτιος.");
9     }
10 }

```

The video player interface at the bottom shows the video is at 0:00 / 23:48.

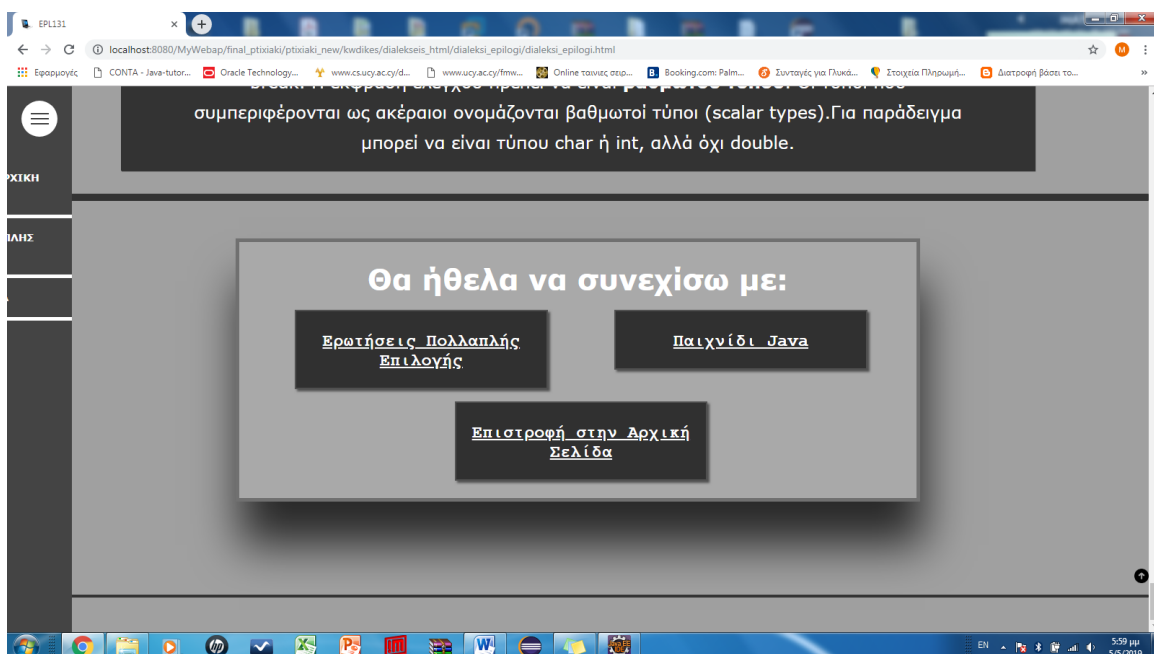
EN 5:58 μμ 5/5/2019

Εικόνα 6.4.11

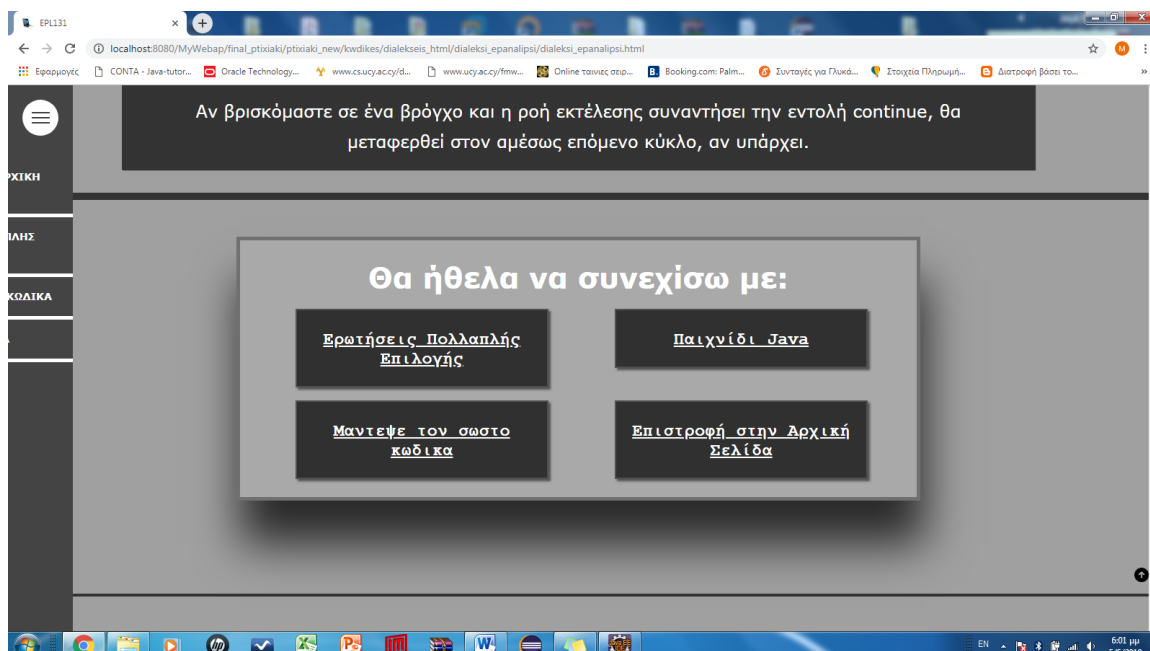


Εικόνα 6.4.12

Στην Εικόνα 6.4.10 παρουσιάζεται ένα στιγμιότυπο οθόνης από τις διαλέξεις. Οι διαλέξεις είναι εμπλουτισμένες με βίντεο [10] (εικόνα 6.4.11) και παραδείγματα (εικόνα 6.4.12) για καλύτερη κατανόηση της θεωρίας.

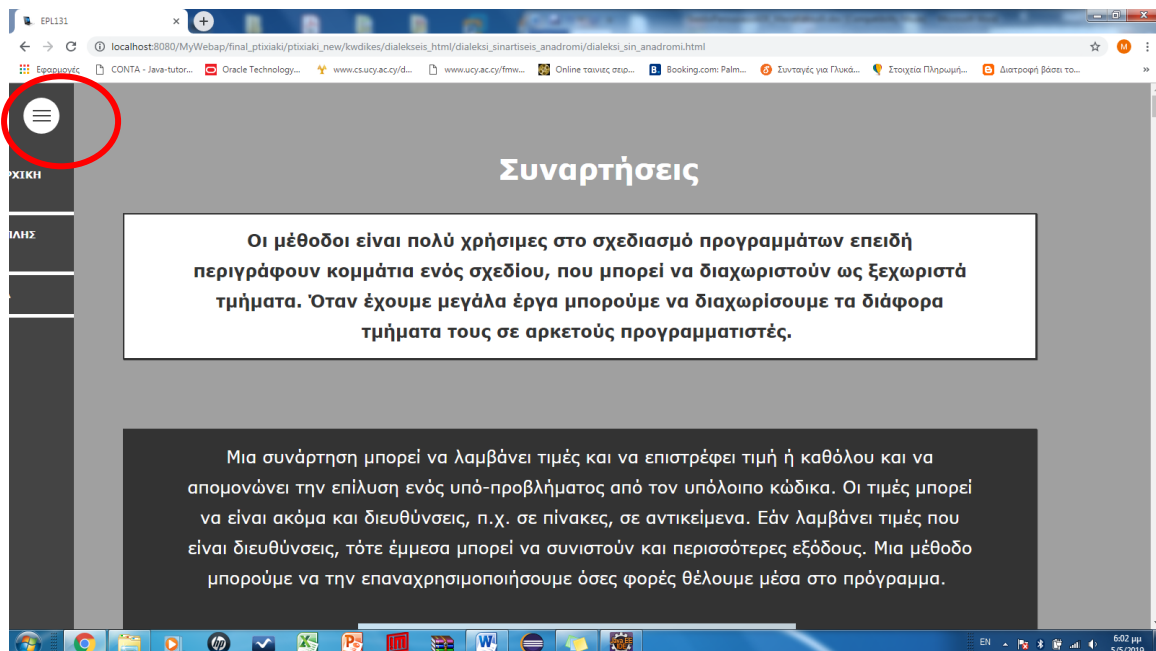


Εικόνα 6.4.13

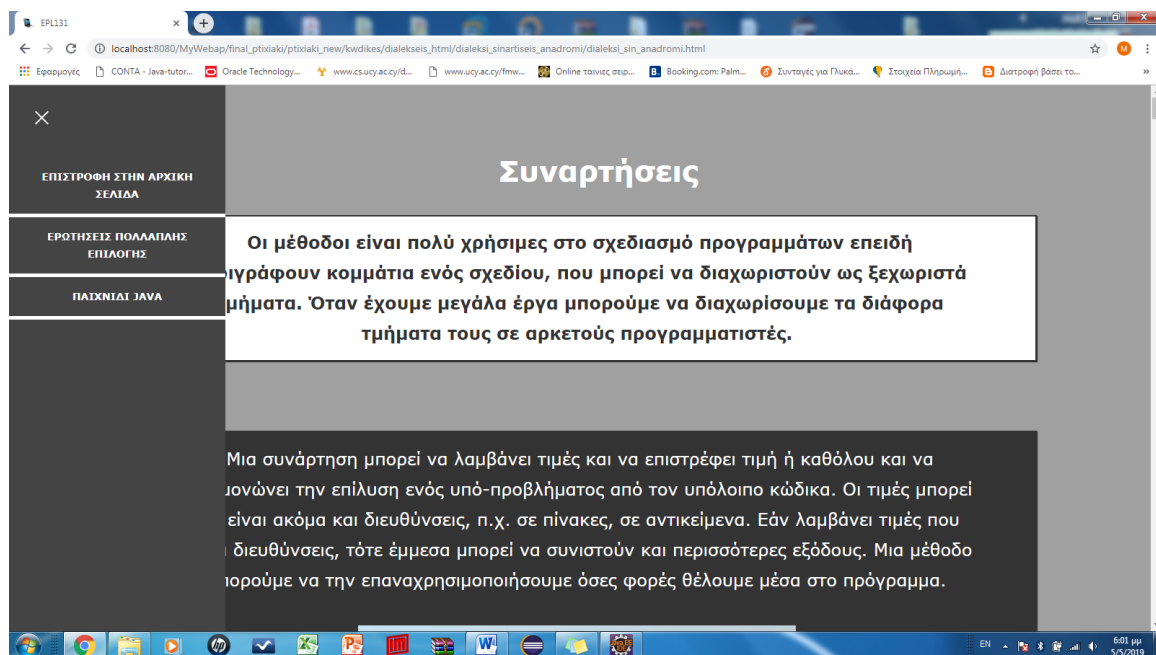


Εικόνα 6.4.14

Στο τέλος των διαλέξεων υπάρχει ένα μενού, που παρέχει επιλογές στο χρήστη για να συνεχίσει (εικόνα 6.4.13). Το μενού που παρουσιάζεται στην εικόνα 6.4.13 αφορά τις ενότητες Γενικοί Κανόνες και Τύποι Δεδομένων, Επιλογή, Πίνακες, Συμβολοσειρές, Συναρτήσεις – Αναδρομή και Αντικειμενοστρεφής Προγραμματισμός οι οποίες περιέχουν τις διαλέξεις, τις ερωτήσεις πολλαπλής επιλογής και το παιχνίδι Java. Από την άλλη, στην εικόνα 6.4.14 παρουσιάζεται το μενού που βρίσκεται στην ενότητα Επανάληψη, καθώς η συγκεκριμένη ενότητα περιέχει διάλεξη, ερωτήσεις πολλαπλής επιλογής, το παιχνίδι «Μάντεψε τον σωστό κώδικα» και το παιχνίδι Java. Αυτό το μενού υπάρχει στο τέλος όλων των δραστηριοτήτων.

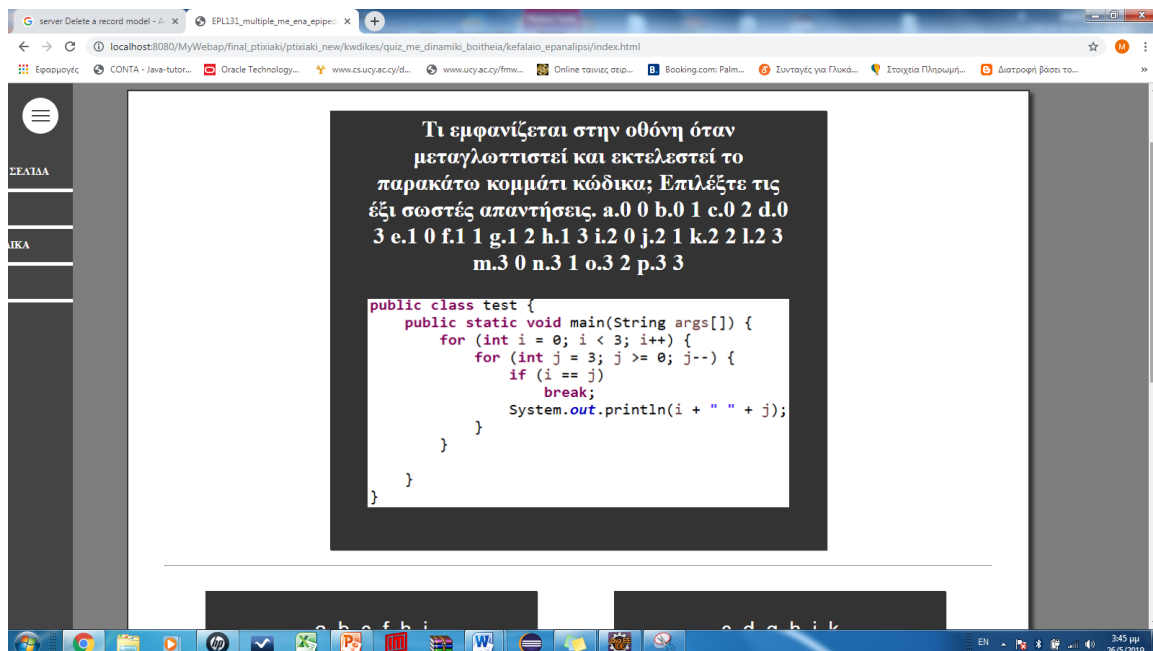


Εικόνα 6.4.15

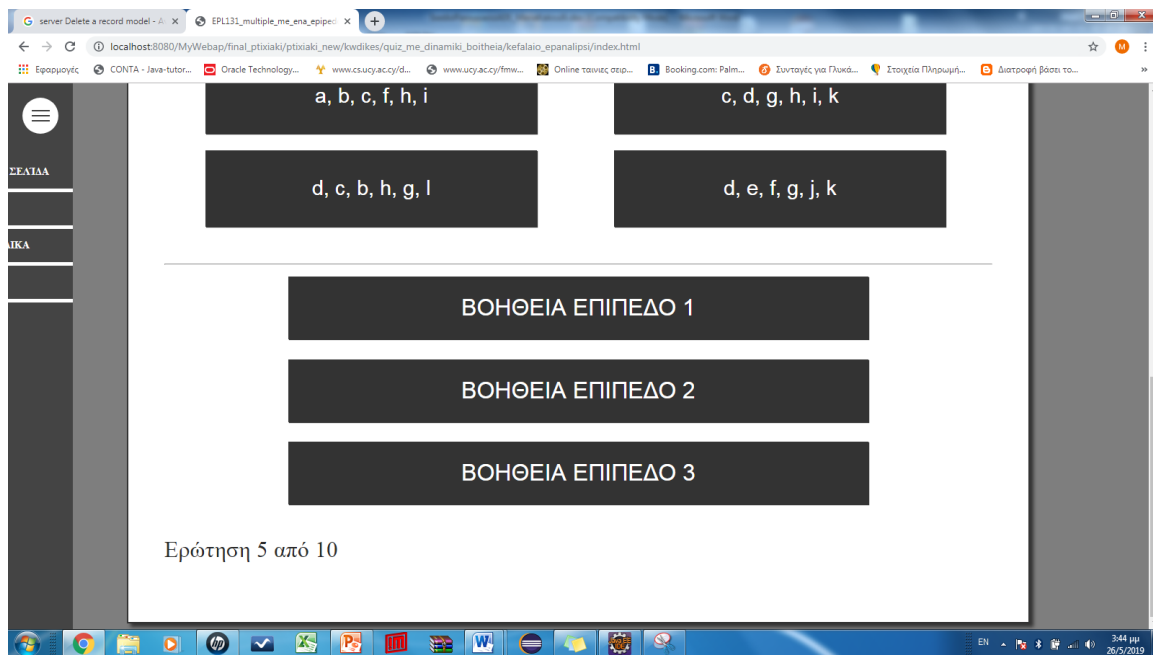


Εικόνα 6.4.16

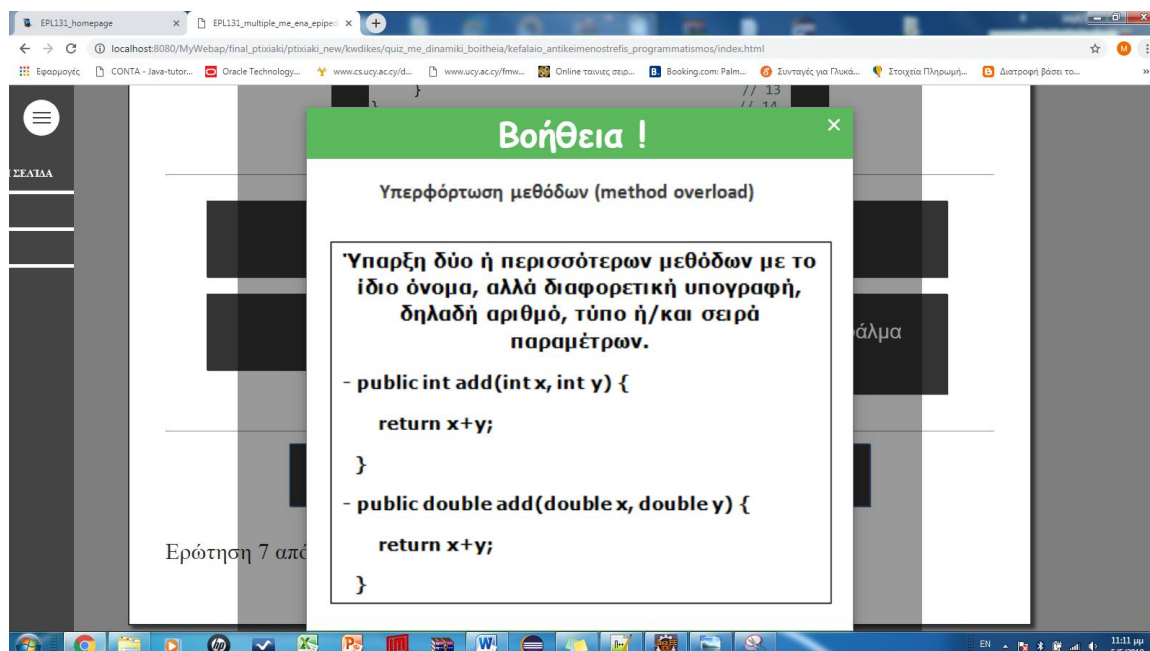
Όπως φαίνεται στην εικόνα 6.4.15, στην αριστερή πλευρά της σελίδας βρίσκεται ένα εικονίδιο. Πατώντας το εμφανίζεται το μενού που παρουσιάζεται στην εικόνα 6.4.16. Μέσω αυτού ο χρήστης μπορεί οποιαδήποτε στιγμή να διακόψει τη δραστηριότητα που κάνει και να συνεχίσει με κάτι άλλο. Αυτό το μενού υπάρχει σε όλες τις δραστηριότητες.



Εικόνα 6.4.17

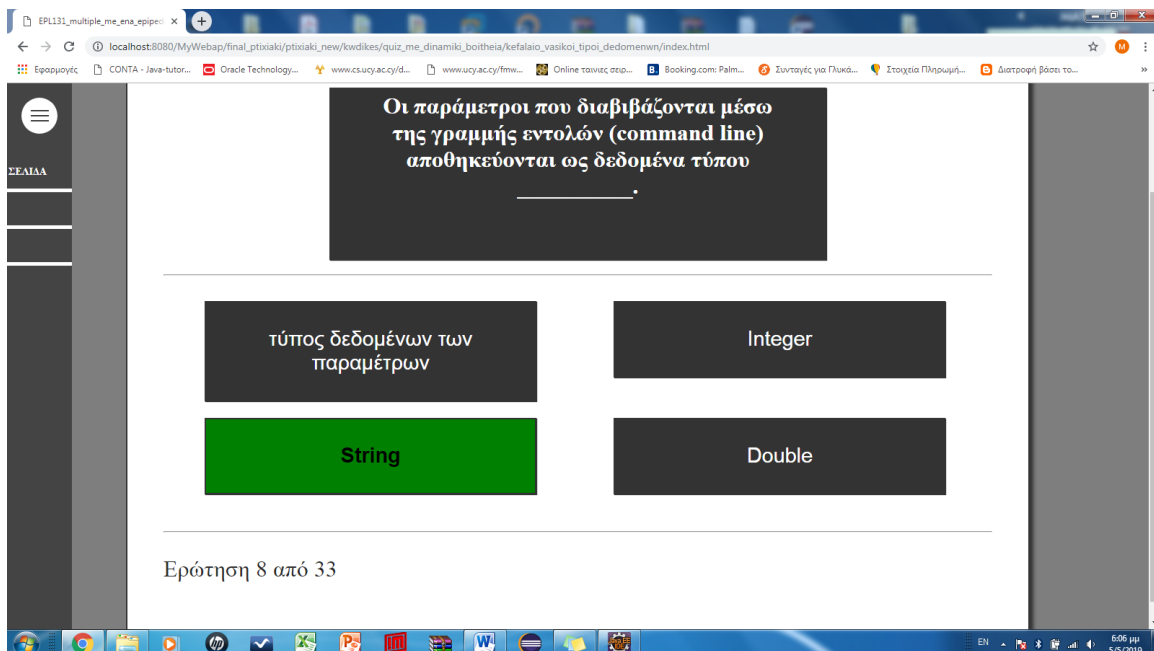


Εικόνα 6.4.18



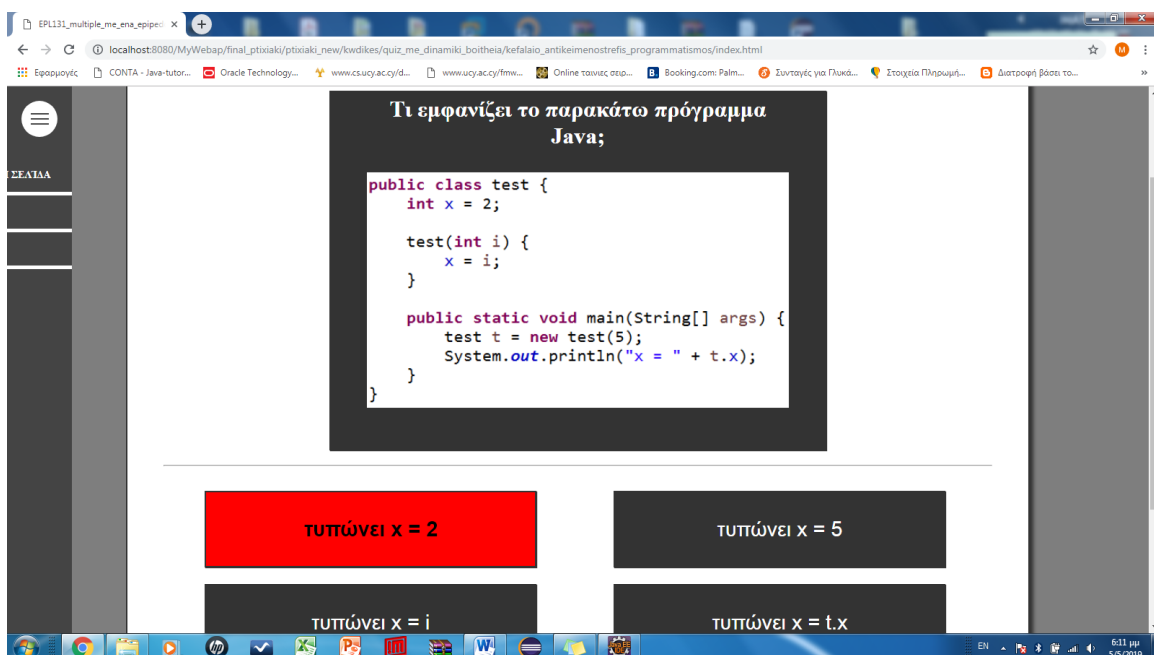
Εικόνα 6.4.19

Στις εικόνες 6.4.17 και 6.4.18, παρουσιάζεται ένα στιγμιότυπο οθόνης από τις ερωτήσεις πολλαπλής επιλογής. Για κάθε ερώτηση υπάρχουν τέσσερις πιθανές απαντήσεις. Ο χρήστης καλείται να βρει τη σωστή απάντηση. Επιπρόσθετα, υπάρχει η δυνατότητα χρήσης βοήθειας. Ανάλογα με το πρόβλημα, παρουσιάζεται και ο αντίστοιχος αριθμός βοηθειών. Για καλύτερη επεξήγηση, το «Επίπεδο Βοήθειας 2» περιέχει μια πιο συγκεκριμένη βοήθεια απ' ότι το «Επίπεδο Βοήθειας 1», το οποίο καθοδηγεί περισσότερο προς την λύση, δηλαδή δίνει μια πιο συγκεκριμένη πληροφορία. Στην εικόνα 6.4.19 παρουσιάζεται δείγμα βοήθειας. Η οθόνη «παγώνει» μέχρι ο χρήστης να πατήσει το σύμβολο «X» που βρίσκεται πάνω δεξιά, και αφού κλείσει το παράθυρο ο χρήστης μπορεί να χρησιμοποιήσει τις υπόλοιπες βοήθειες ή να δώσει απάντηση.

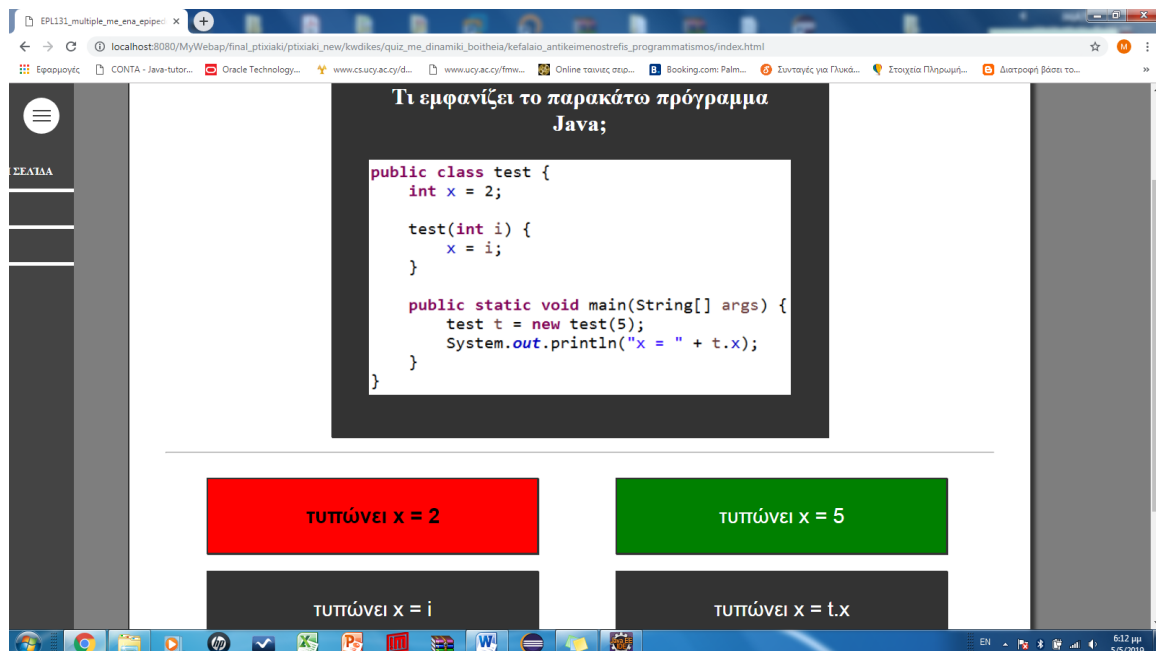


Εικόνα 6.4.20

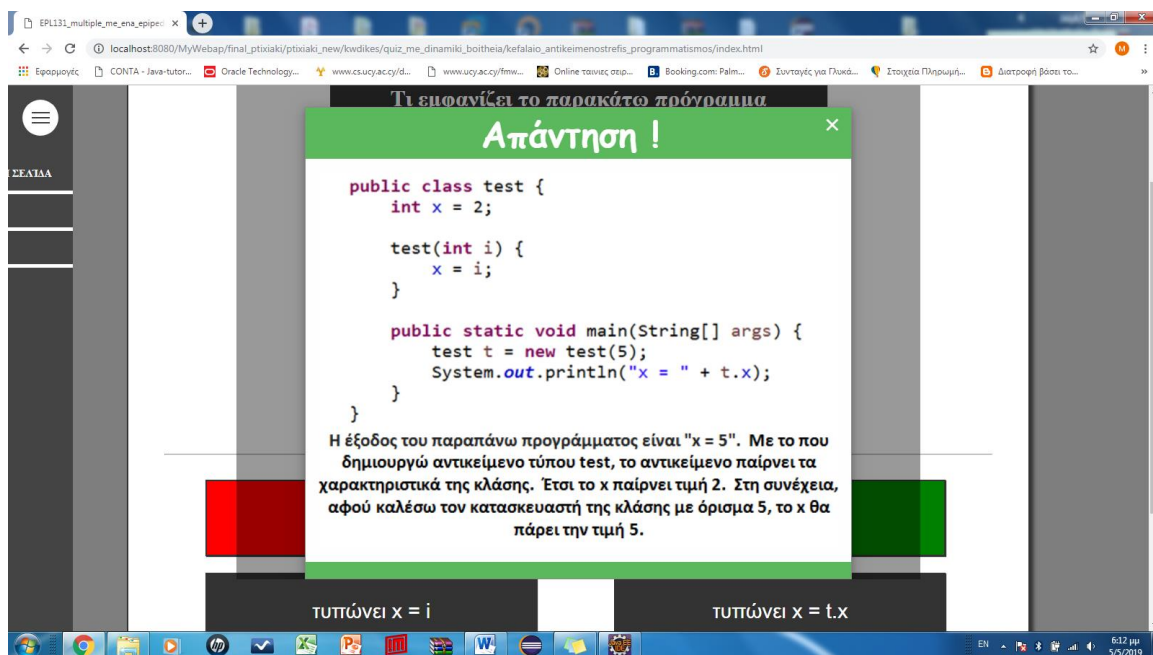
Συνεχίζοντας με τις ερωτήσεις πολλαπλής επιλογής, αν ο χρήστης απαντήσει ορθά, η επιλογή που επέλεξε παίρνει πράσινο χρώμα (εικόνα 6.4.20).



Εικόνα 6.4.21

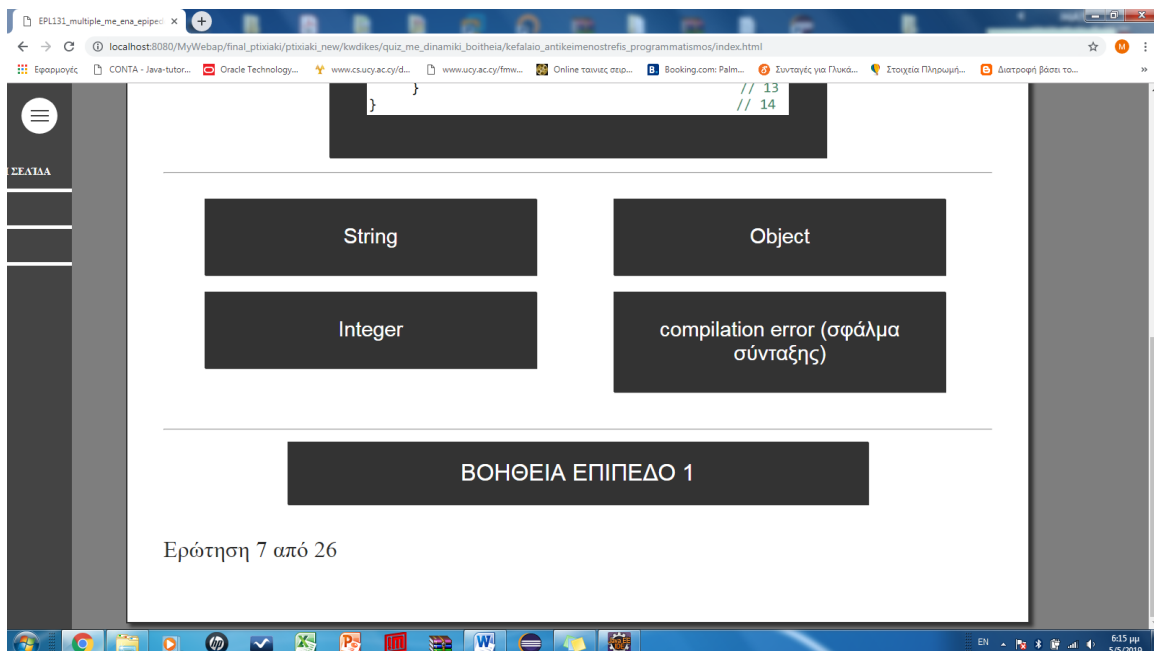


Εικόνα 6.4.22



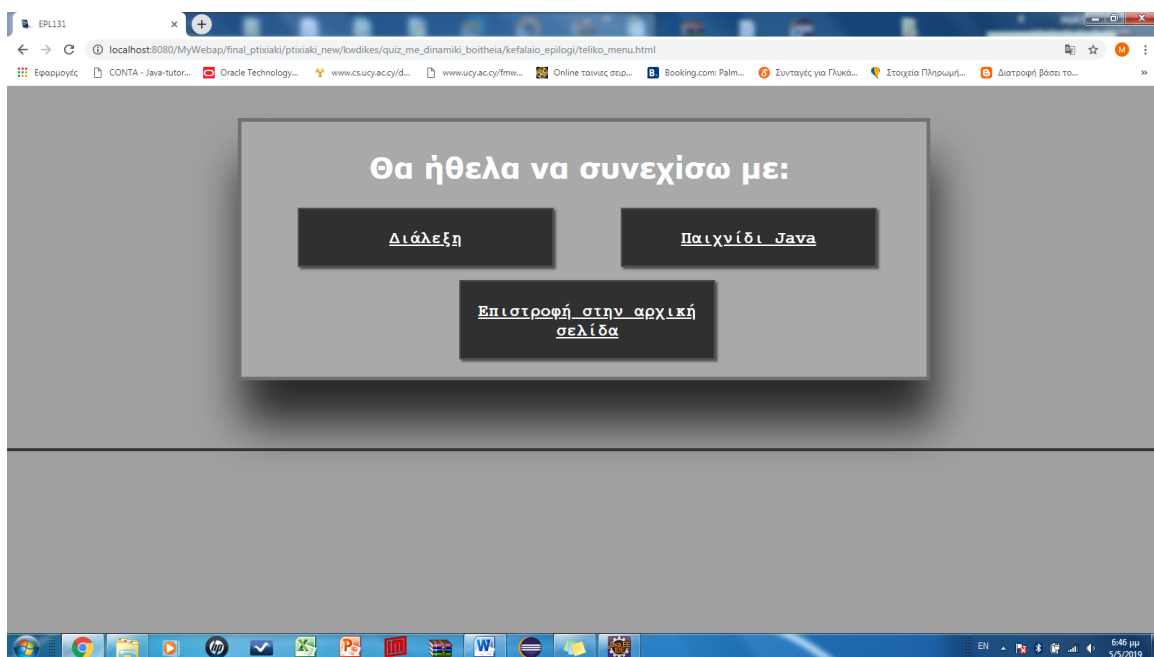
Εικόνα 6.4.23

Σε περίπτωση που ο χρήστης απαντήσει λανθασμένα, η επιλογή που επέλεξε παίρνει κόκκινο χρώμα (εικόνα 6.4.21) και σε λίγα δευτερόλεπτα η σωστή απάντηση θα πάρει πράσινο χρώμα (εικόνα 6.4.22). Αμέσως μετά εμφανίζεται η σωστή απάντηση μαζί με μια μικρή επεξήγηση (εικόνα 6.4.23).

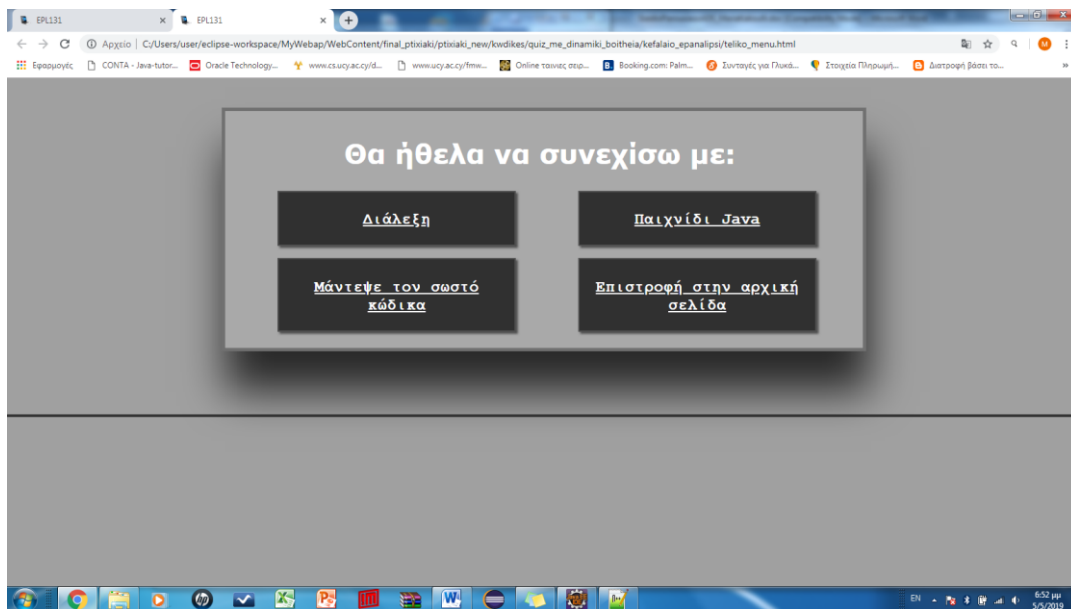


Εικόνα 6.4.24

Κάτω αριστερά στην οθόνη βρίσκεται ο αριθμός της ερώτησης του συγκεκριμένου παιχνιδιού (εικόνα 6.4.24).



Εικόνα 6.4.25



Εικόνα 6.4.26

Μετά την ολοκλήρωση του παιχνιδιού, εμφανίζεται μια νέα οθόνη που περιέχει ένα μενού και παρέχει επιλογές στο χρήστη για να συνεχίσει (εικόνες 6.4.25 και 6.4.26). Το μενού που παρουσιάζεται στην εικόνα 6.4.25 αφορά τις ενότητες Γενικοί Κανόνες και Τύποι Δεδομένων, Επιλογή, Πίνακες, Συμβολοσειρές, Συναρτήσεις – Αναδρομή και Αντικειμενοστρεφής Προγραμματισμός, οι οποίες περιέχουν τις διαλέξεις, τις ερωτήσεις πολλαπλής επιλογής και το παιχνίδι Java. Από την άλλη, στην εικόνα 6.4.26 παρουσιάζεται το μενού που βρίσκεται στην ενότητα Επανάληψη, καθώς η συγκεκριμένη ενότητα περιέχει διάλεξη, ερωτήσεις πολλαπλής επιλογής, το παιχνίδι «Μάντεψε τον σωστό κώδικα» και το παιχνίδι Java.

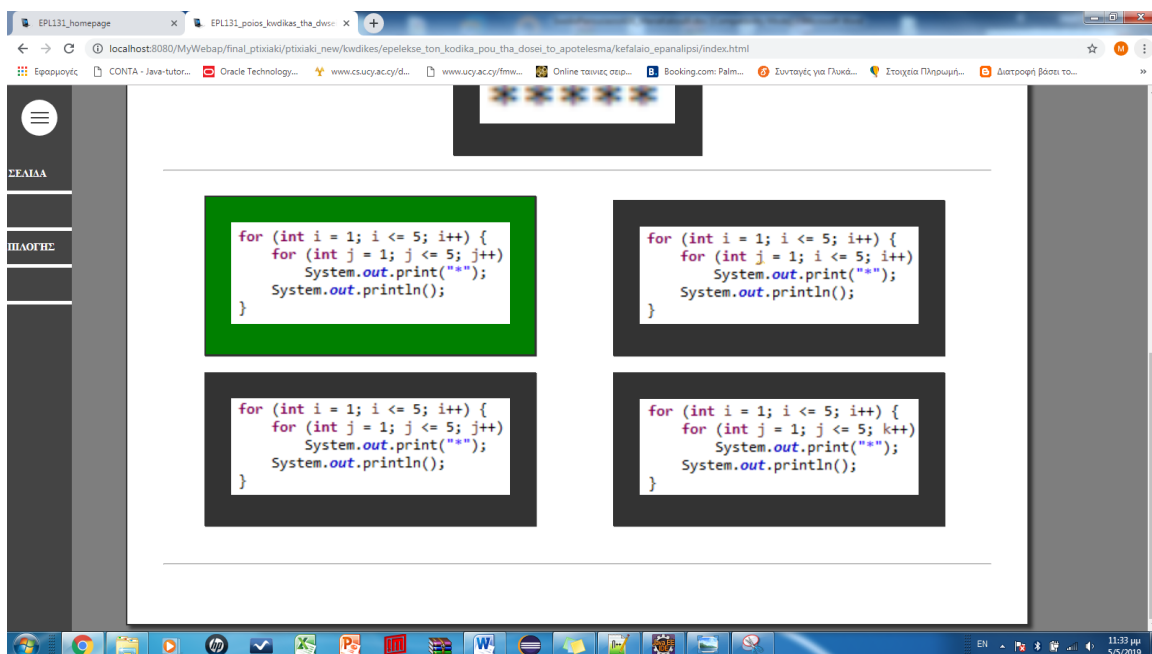


Εικόνα 6.4.27



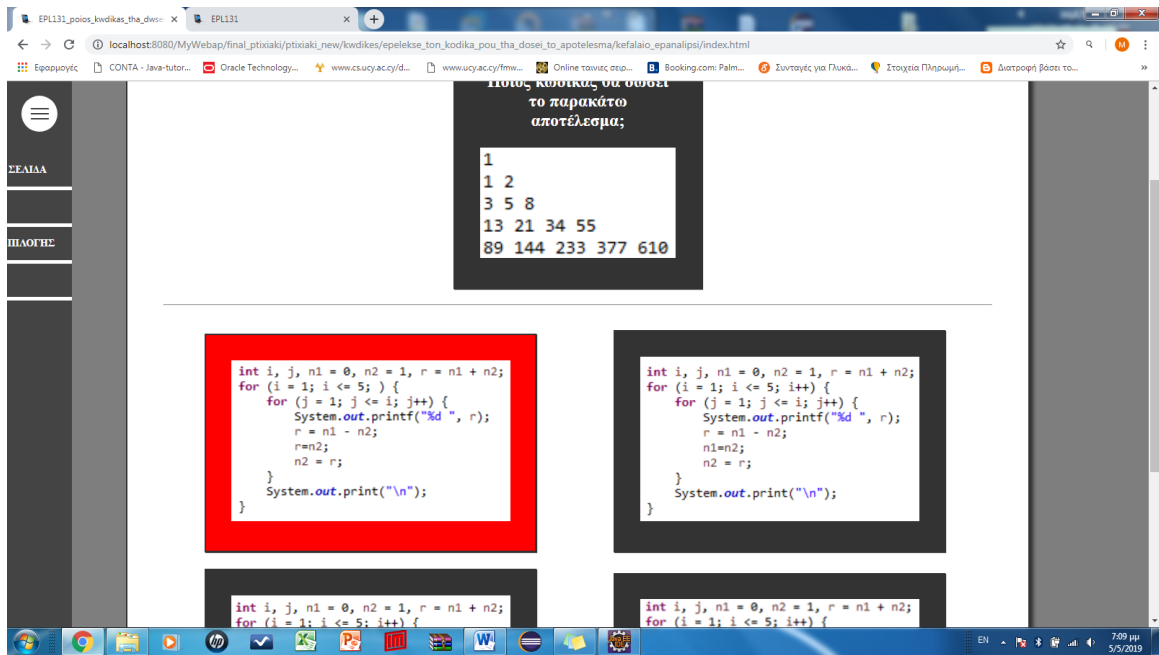
Εικόνα 6.4.28

Ακολουθώντας τα βήματα των εικόνων 6.4.27 και 6.4.28, ο χρήστης μπορεί να μεταβεί στο παιχνίδι «Μάντεψε τον σωστό κώδικα». Σε αυτό το παιχνίδι δίνεται ένα αποτέλεσμα κώδικα, και τέσσερις πιθανοί κώδικες που να έδωσαν αυτό το αποτέλεσμα. Ο χρήστης καλείται να βρει ποιος κώδικας έδωσε το συγκεκριμένο αποτέλεσμα.

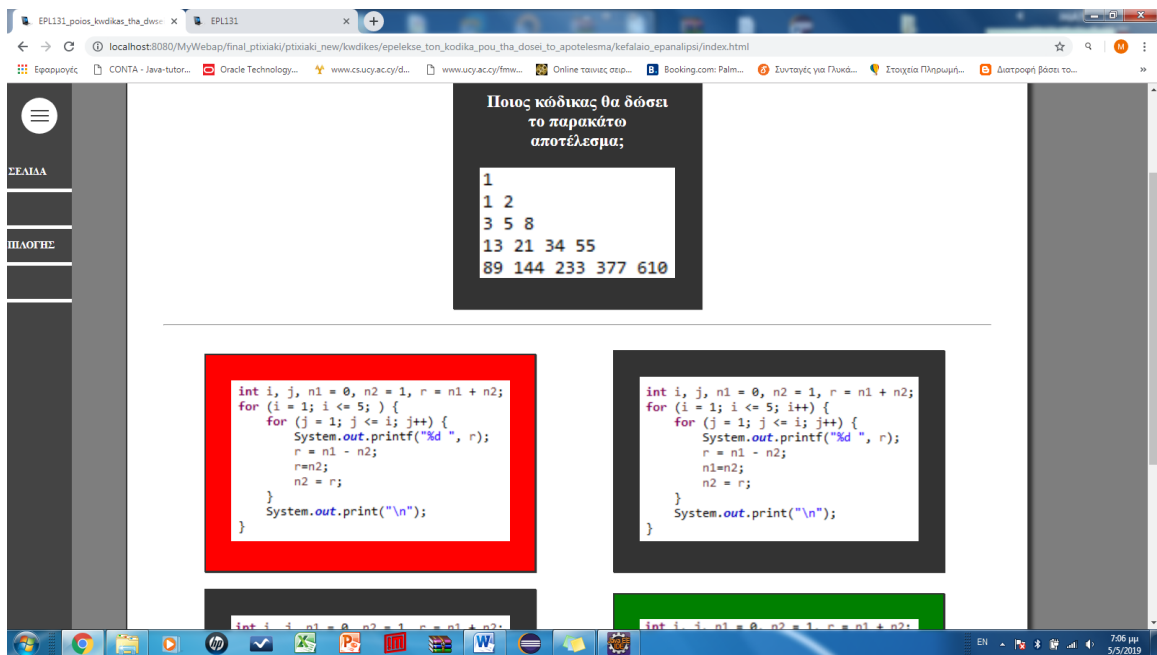


Εικόνα 6.4.29

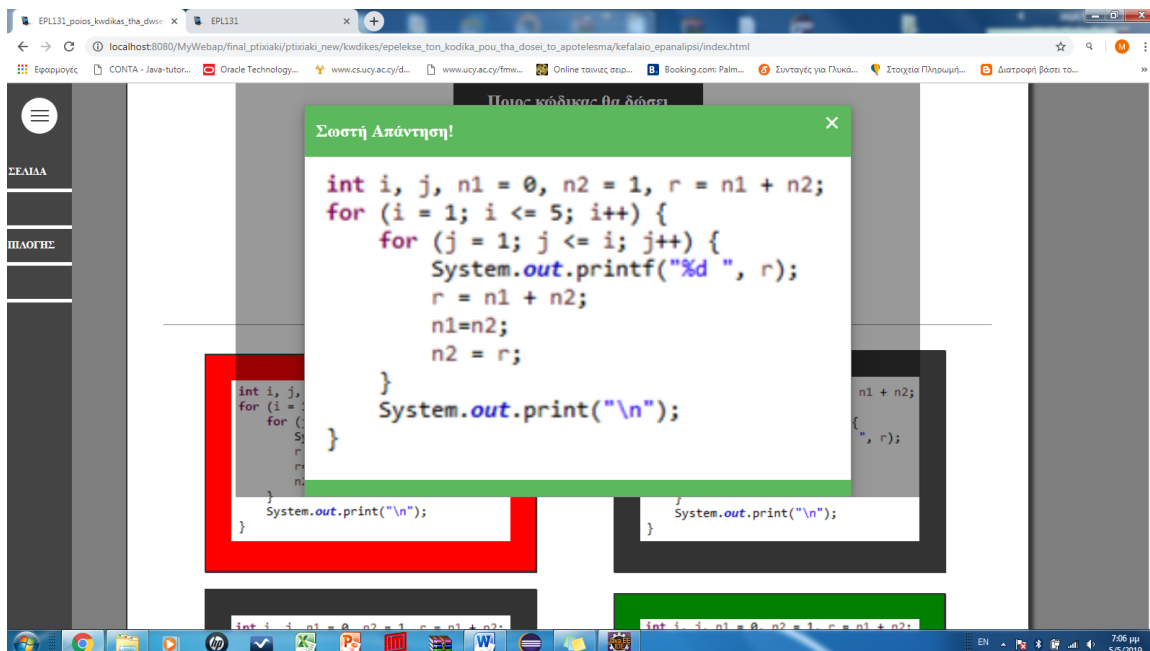
Αν ο χρήστης βρει την σωστή απάντηση, τότε η απάντηση που έδωσε παίρνει πράσινο χρώμα (εικόνα 6.4.29) για μερικά δευτερόλεπτα, και έπειτα το παιχνίδι προχωρά με το επόμενο ερώτημα.



Εικόνα 6.4.30

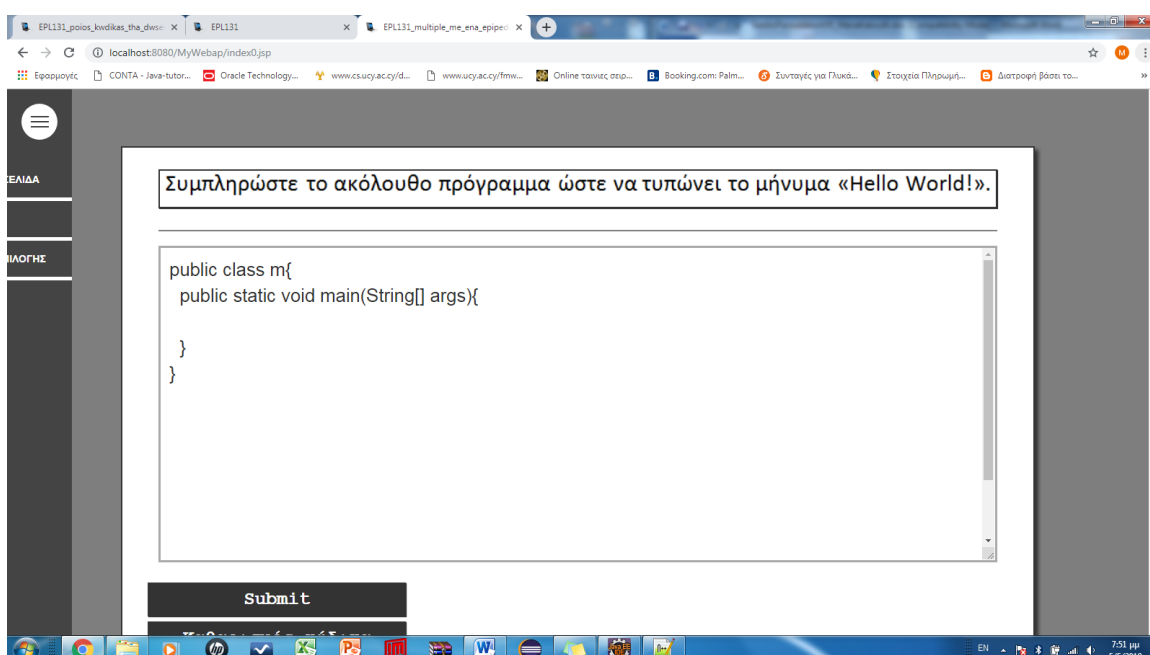


Εικόνα 6.4.31

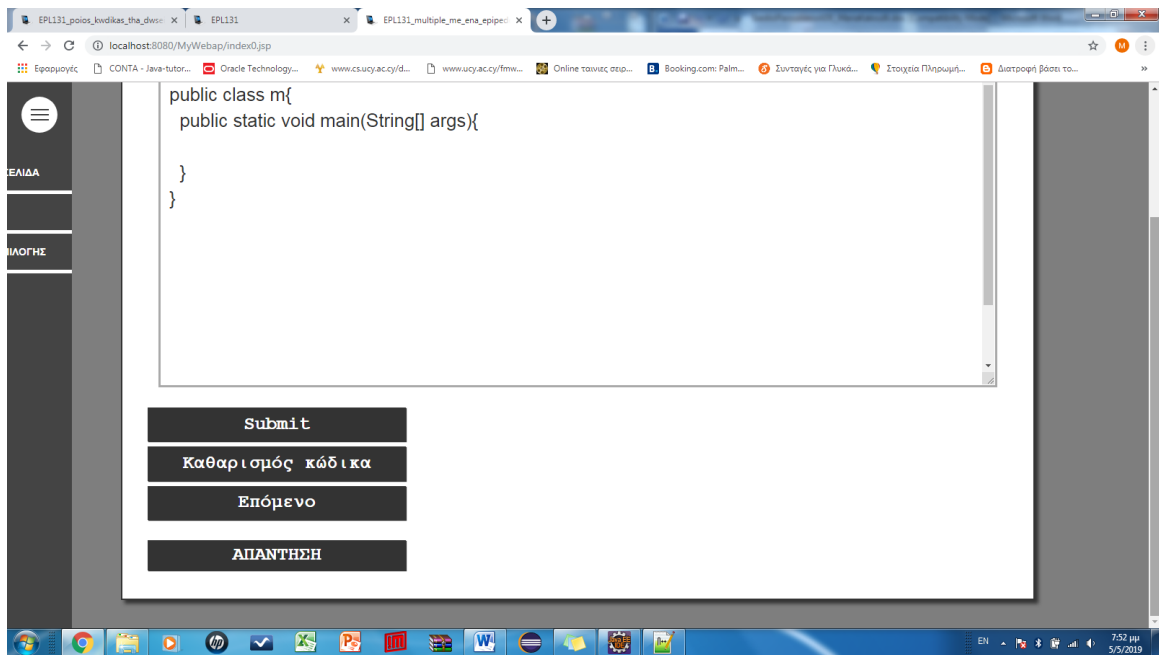


Εικόνα 6.4.32

Αν ο χρήστης δώσει λάθος αποτέλεσμα, τότε η επιλογή που έδωσε παίρνει κόκκινο χρώμα (εικόνα 6.4.30). Σε μερικά δευτερόλεπτα η σωστή επιλογή (εικόνα 6.4.31) παίρνει πράσινο χρώμα και εμφανίζεται ο σωστός κώδικας (εικόνα 6.4.32). Όταν ο χρήστης πατήσει το σύμβολο «X» πάνω δεξιά, τότε το παράθυρο της σωστής απάντησης κλείνει και το παιχνίδι προχωρά. Ενώσω είναι ανοιχτό το παράθυρο της σωστής απάντησης, η οθόνη «παγώνει» μέχρι ο χρήστης να το κλείσει.

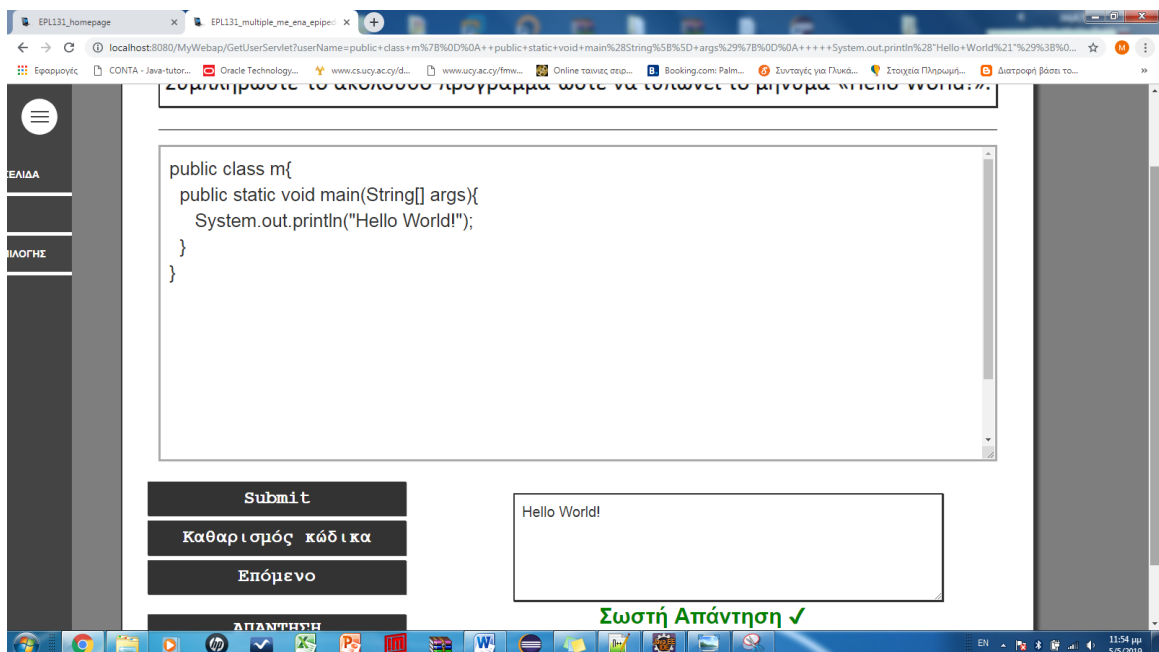


Εικόνα 6.4.33

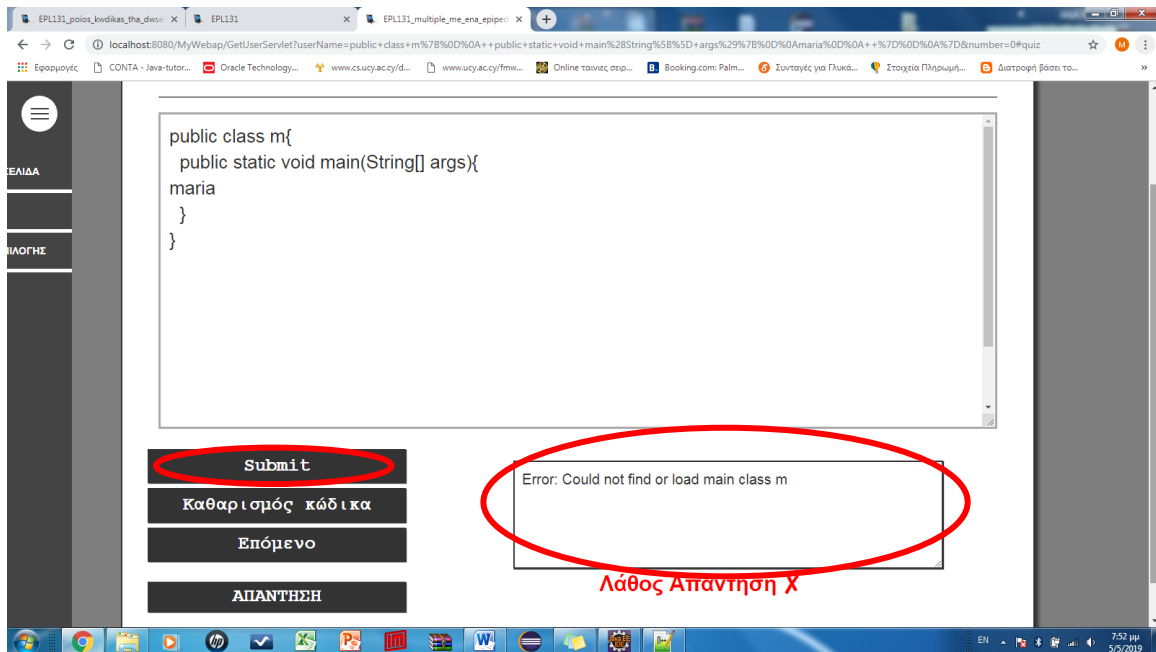


Εικόνα 6.4.34

Στις εικόνες 6.4.33 και 6.4.34 εμφανίζεται το παιχνίδι συμπλήρωσης κώδικα Java. Βάση κάποιου προβλήματος, ο χρήστης θα πρέπει να συμπληρώσει τον υπολειπόμενο κώδικα ώστε να εκτελεί αυτό που ζητείτε στην οδηγία.

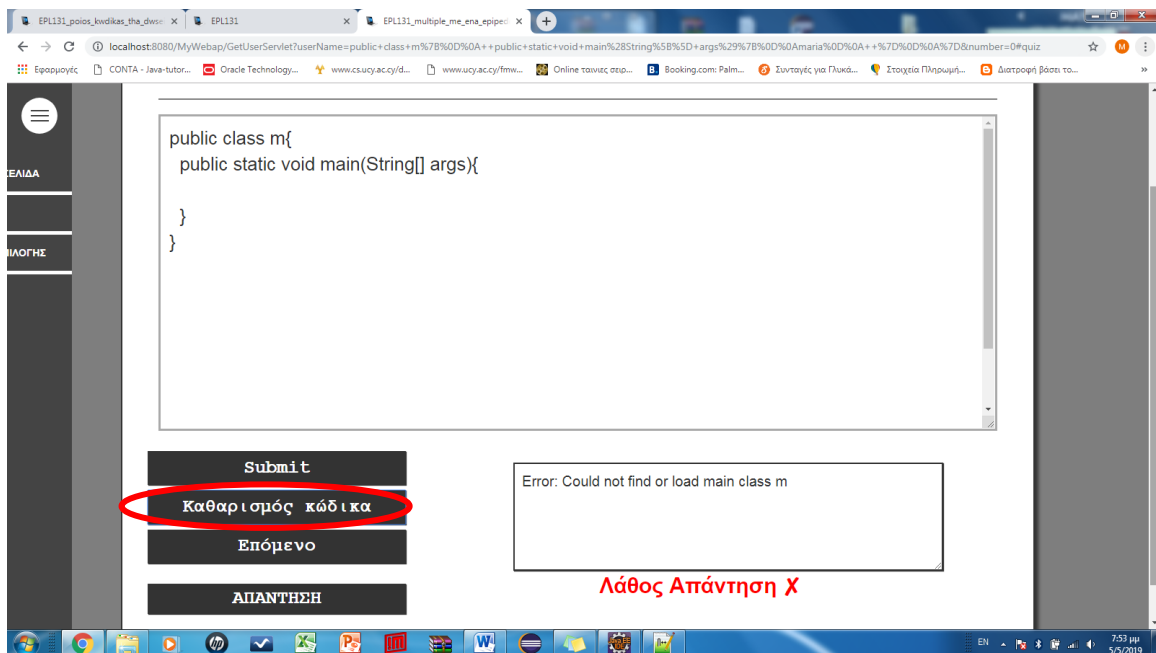


Εικόνα 6.4.35



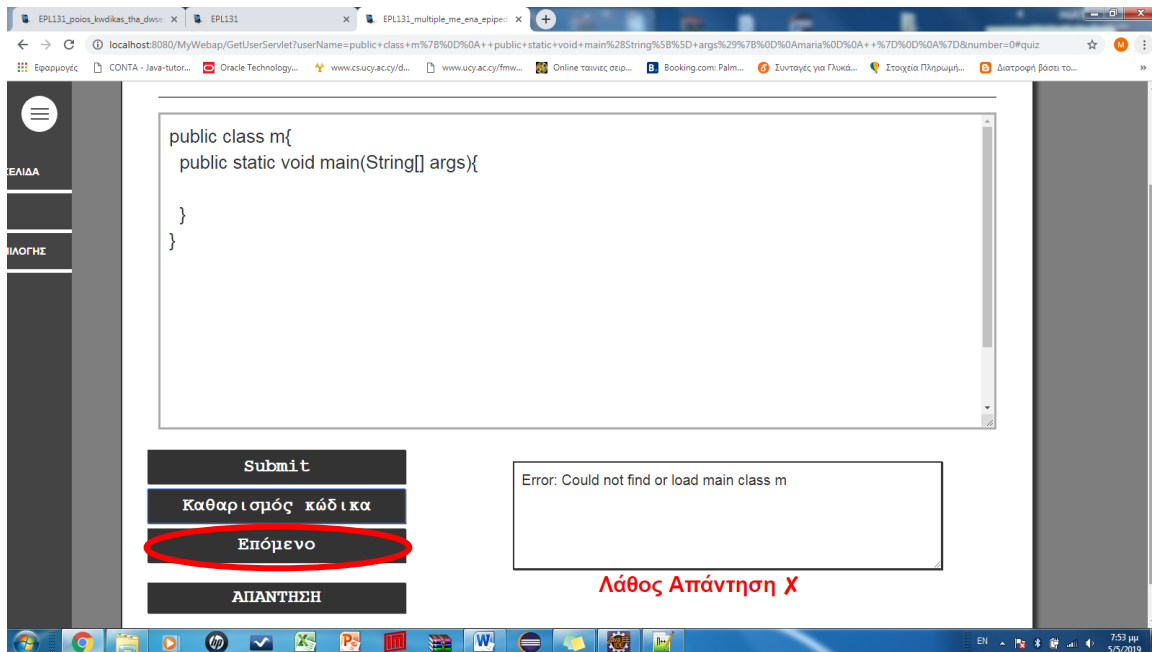
Εικόνα 6.4.36

Ο χρήστης θα πρέπει να συμπληρώσει τον κώδικα και ακολούθως να πατήσει το κουμπί «Submit» (εικόνα 6.4.36) για να σταλθεί ο κώδικας στην Java διαμέσου του server. Μόλις έρθει η απάντηση από τον server, εμφανίζεται ένα πλαίσιο που περιέχει το αποτέλεσμα που επιστρέφει η κονσόλα (εικόνα 6.4.36). Το αποτέλεσμα ελέγχεται μέσω της βάσης, και αν είναι σωστό εμφανίζεται το μήνυμα «Σωστή Απάντηση» (εικόνα 6.4.35), αλλιώς το μήνυμα «Λάθος Απάντηση» (εικόνα 6.4.36).



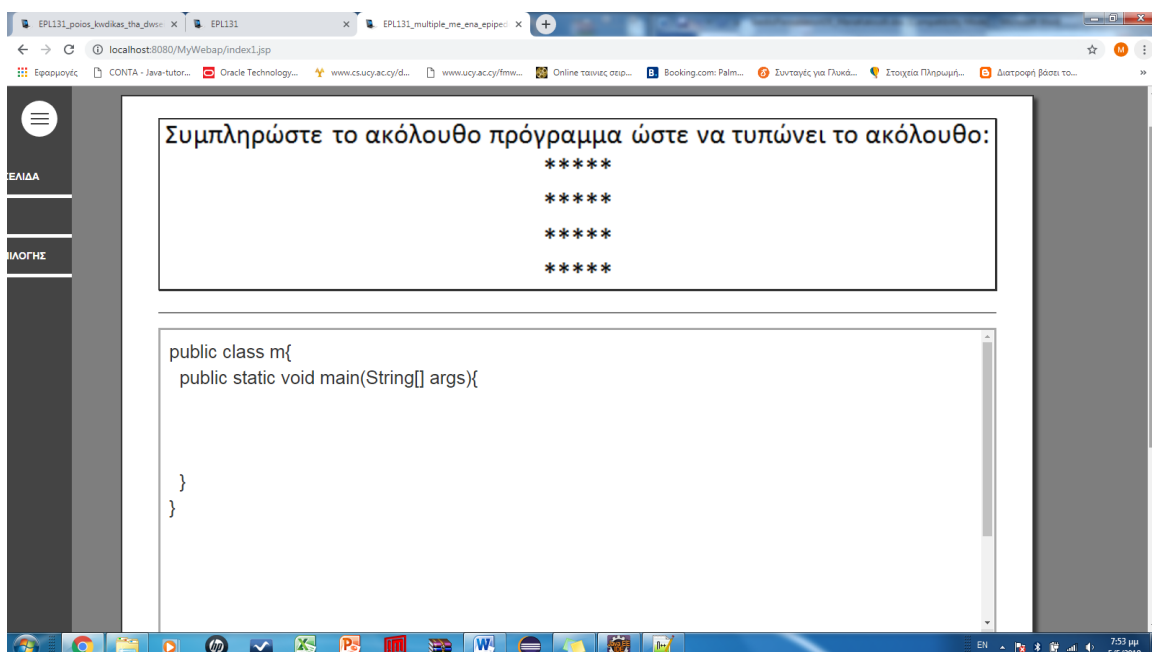
Εικόνα 6.4.37

Όταν ο χρήστης δώσει κάποιον κώδικα, και επιθυμεί να διαγραφούν τα όσα έγραψε τότε πατά το κουμπί «Καθαρισμός κώδικα», και επανέρχεται ο κώδικας όπως του δόθηκε στην αρχή (εικόνα 6.4.37).



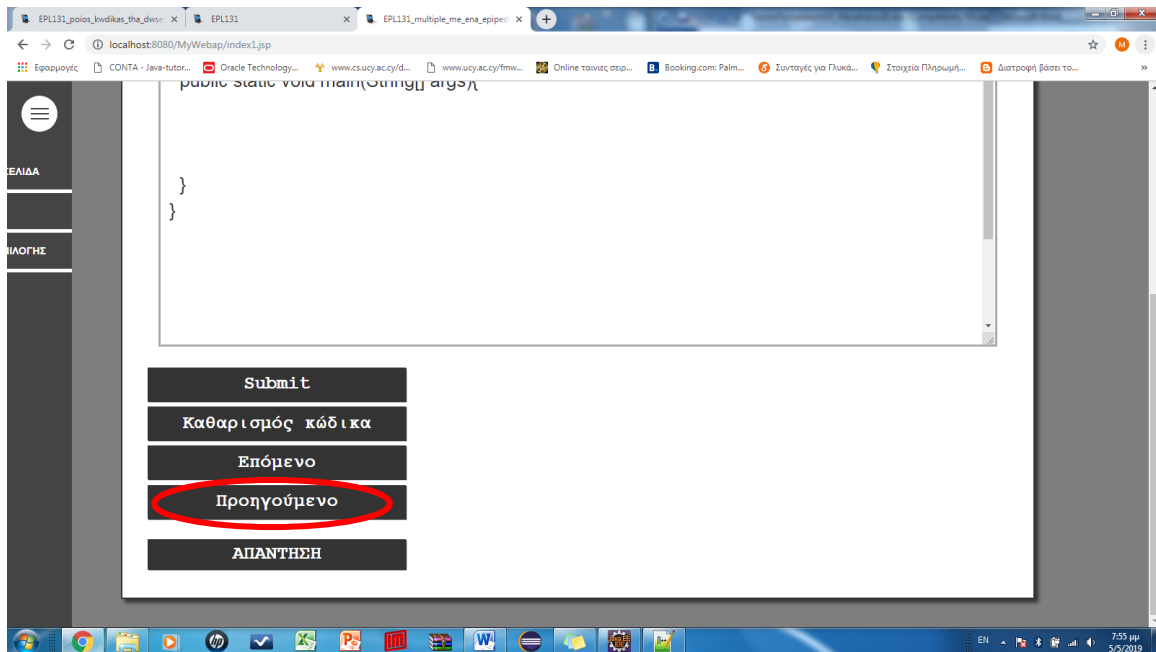
Εικόνα 6.4.38

Πατώντας ο χρήστης το κουμπί «Επόμενο» (εικόνα 6.4.38), μπορεί να μεταβεί στο επόμενο ερώτημα, χωρίς να είναι αναγκαστικό να ολοκληρώσει το συγκεκριμένο ερώτημα. Οι εικόνες 6.4.33 έως 6.4.38, αφορούν το πρώτο ερώτημα του παιχνιδιού.



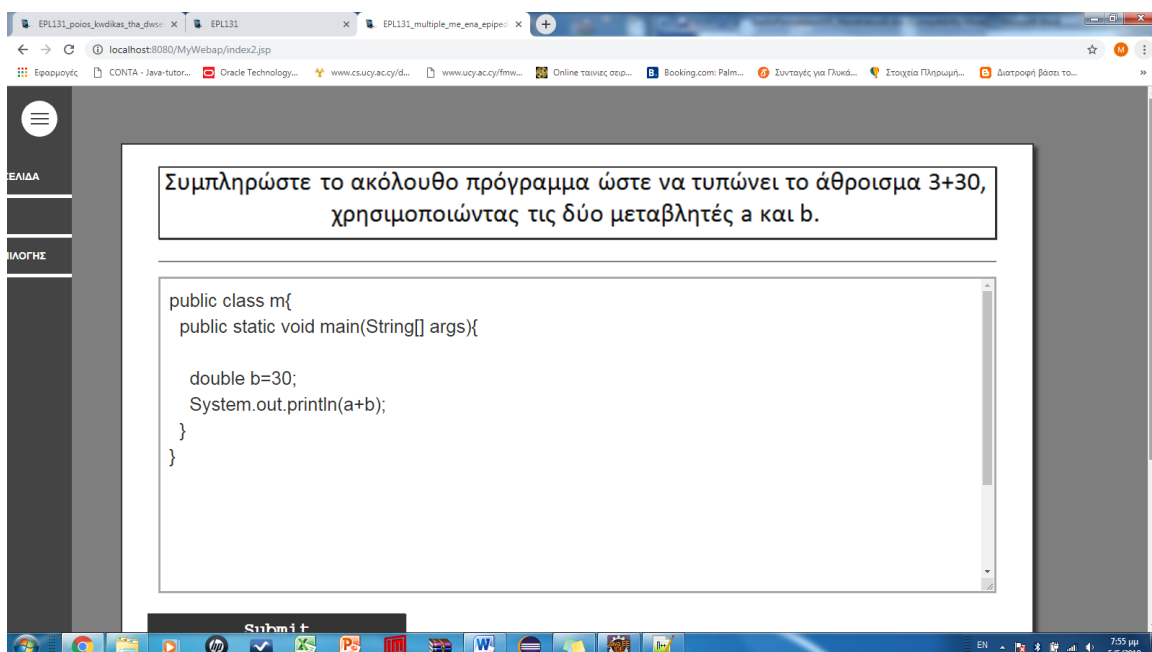
Εικόνα 6.4.39

Πατώντας ο χρήστης το κουμπί «Επόμενο» (εικόνα 6.4.38), μπορεί να μεταβεί στο επόμενο ερώτημα (εικόνα 6.4.39), χωρίς να είναι αναγκαστικό να ολοκληρώσει το συγκεκριμένο ερώτημα.

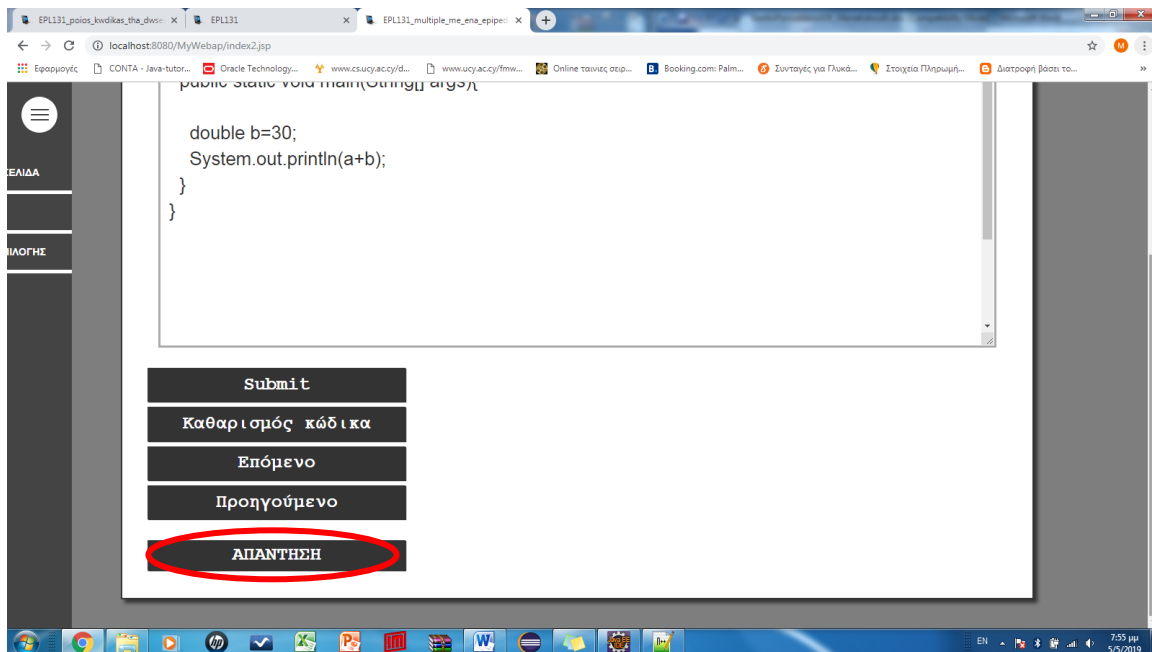


Εικόνα 6.4.40

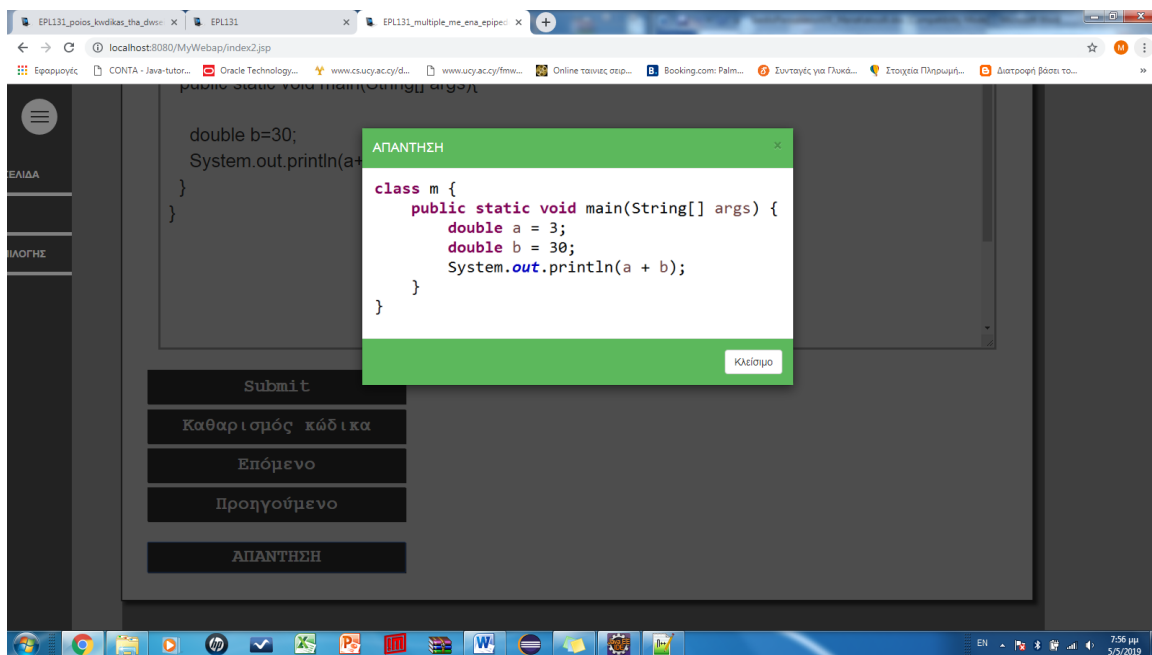
Εκτός από το πρώτο ερώτημα (οι εικόνες 6.4.33 έως 6.4.38, αφορούν το πρώτο ερώτημα του παιχνιδιού), τα υπόλοιπα ερωτήματα περιλαμβάνουν και τη δυνατότητα για επιστροφή στο προηγούμενο ερώτημα (εικόνα 6.4.40).



Εικόνα 6.4.41

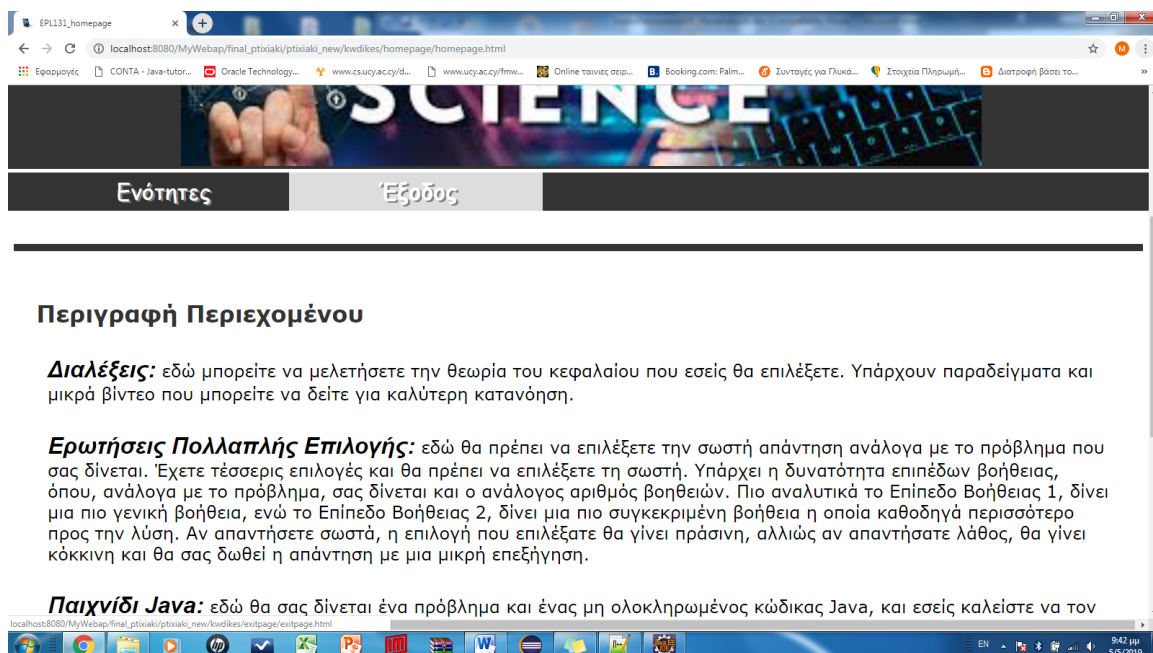


Εικόνα 6.4.42

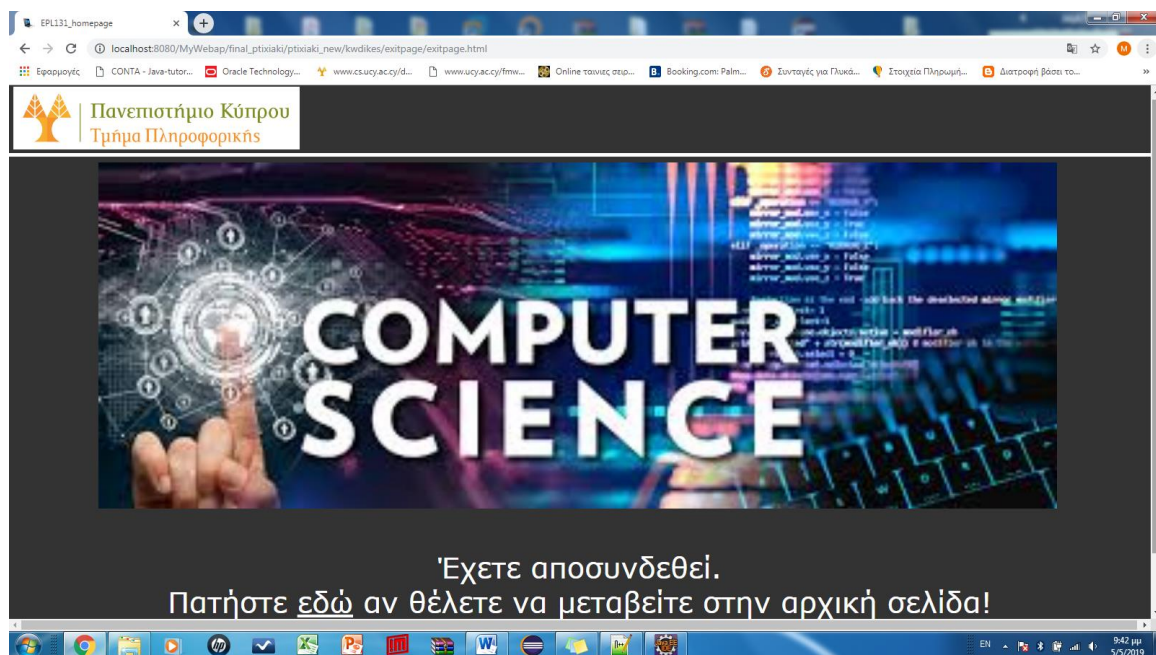


Εικόνα 6.4.43

Τις εικόνες 6.4.41 και 6.4.42, φαίνεται μια άλλη οθόνη για το παιχνίδι συμπλήρωσης κώδικα Java. Πατώντας ο χρήστης την επιλογή «ΑΠΑΝΤΗΣΗ» (εικόνα 6.4.42), εμφανίζεται η λύση για το τρέχον πρόβλημα (εικόνα 6.4.43).



Εικόνα 6.4.44



Εικόνα 6.4.45

Αφού ο χρήστης επιθυμεί να αποχωρήσει από την πλατφόρμα, μπορεί να πατήσει την επιλογή «Εξοδος» από την αρχική σελίδα (εικόνα 6.4.44). Τότε θα οδηγηθεί σε μια νέα οθόνη που τον ενημερώνει ότι έχει αποσυνδεθεί και του δίνεται η ευκαιρία αν θέλει να ξαναμεταβεί στην αρχική σελίδα.

Κεφάλαιο 7

Συμπεράσματα

7.1 Εισαγωγή	82
7.2 Συμπεράσματα	82
7.3 Μελλοντική Επέκταση	83

7.1 Εισαγωγή

Κατά τη διάρκεια εκπόνησης της εν λόγω διπλωματικής εργασίας αυτής και μέσα από τη μελέτη των αναγκών των φοιτητών του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου, προέκυψαν μερικά συμπεράσματα. Σε αυτό το κεφάλαιο θα παρουσιαστούν τα συμπεράσματα αυτά, καθώς επίσης και μελλοντικές εργασίες που μπορεί να γίνουν για βελτίωση της πλατφόρμας που έχει υλοποιηθεί.

7.2 Συμπεράσματα

Η ατομική διπλωματική εργασία έχει υλοποιηθεί με επιτυχία σε μεγάλο βαθμό, εφόσον το σύστημα έχει αναπτυχθεί με σκοπό την επιπλέον υποστήριξη των φοιτητών του τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου στο μάθημα ΕΠΛ131 Αρχές Προγραμματισμού. Το σύστημα έχει υλοποιηθεί λαμβάνοντας υπόψη τους χρήστες που θα το χρησιμοποιούν και δημιουργήθηκαν οι ανάλογες λειτουργίες με τρόπο ώστε να είναι φιλικό προς τους χρήστες. Επιπλέον, το σύστημα είναι πλήρως λειτουργικό καθώς ικανοποιεί τις ανάγκες και τις απαιτήσεις που αναφέρθηκαν, άρα και τον σκοπό που αρχικά τέθηκε.

7.3 Μελλοντική Επέκταση

Η έρευνα που διεξήχθη στο πλαίσιο αυτής της διπλωματικής εργασίας κατά την ανάπτυξη, έχει οδηγήσει στην επισήμανση κάποιων θεμάτων τα οποία αναμένεται να υλοποιηθούν για καλύτερη λειτουργία της εφαρμογής. Αρχικά, θα μπορούσε οι διαλέξεις να εμπλουτιστούν με περισσότερα παραδείγματα και βίντεο. Μέσω αυτών ο φοιτητής θα μπορεί να κατανοήσει καλύτερα το ό,τι αναγράφεται στις διαλέξεις. Επίσης στο παιχνίδι με τις ερωτήσεις πολλαπλής επιλογής θα μπορούσε να υπάρξει βαθμολογία, ώστε ο φοιτητής να έχει περισσότερο κίνητρο να ασχοληθεί με το παιχνίδι. Μέσω αυτού, θα αυξηθεί ο ανταγωνισμός και ο φοιτητής θα προσπαθεί να επιτύχει υψηλότερη βαθμολογία από τους συμμαθητές του. Επιπρόσθετα, το παιχνίδι συμπλήρωσης κώδικα, θα μπορούσε να γίνει ομαδικό, ενισχύοντας την συνεργασία, και έχοντας πάλι βαθμολογία ομάδων. Ακόμη θα μπορούσε να υπάρξει ακόμα ένα παιχνίδι, στο οποίο να μπορούσαν να συνεργαστούν δύο φοιτητές ως αντίπαλοι και σ' αυτό το παιχνίδι θα καλούνταν να διορθώσουν ένα μικρό τμήμα κώδικα Java. Αυτός που θα το διόρθωνε πρώτος θα ήταν ο νικητής και στο τέλος του παιχνιδιού ο κάθε παίκτης θα έπαιρνε μια βαθμολογία. Οι βαθμολογίες για όλα αυτά τα παιχνίδια θα μπορούσαν να αποθηκευτούν σε μια βάση, και ο κάθε φοιτητής να μπορεί να βλέπει τις βαθμολογίες των συμμαθητών του για όλες τις δραστηριότητες. Εκτός από αυτά, θα μπορούσε να υπάρχει μια λειτουργία στην οποία ο κάθε φοιτητής να έχει το δικαίωμα να υποβάλλει ερωτήσεις, και όποιος γνωρίζει την απάντηση να απαντά. Με τον τρόπο αυτό, θα επιλύονται οι απορίες των φοιτητών. Άλλωστε η απορία ενός φοιτητή, μπορεί να είναι και απορία άλλων φοιτητών.

Εκτός από αυτά, λόγω του ότι η πλατφόρμα βρίσκεται σε τοπική βάση, θα μπορούσε να μεταφερθεί στον server του Πανεπιστημίου Κύπρου, ώστε οι φοιτητές να έχουν πρόσβαση στην πλατφόρμα. Λόγω του ότι το σύστημα τρέχει σε tomcat server, ενώ το Πανεπιστήμιο χρησιμοποιεί τον server Electra, ίσως χρειαστεί να γίνουν αλλαγές στους κώδικες. Μετά από αυτό, θα μπορούσε να γίνουν έλεγχοι από τους φοιτητές για το πόσο εύχρηστο, χρήσιμο, λειτουργικό και βοηθητικό είναι γι' αυτούς και μέσα από την περιήγηση τους στην πλατφόρμα να εισηγηθούν βελτιώσεις ή ακόμα και λειτουργίες που πιστεύουν πως θα τους βοηθούσε. Ακόμη, προτού μεταφερθεί το σύστημα στον server του Πανεπιστημίου Κύπρου, θα πρέπει να γίνουν επιπλέον έλεγχοι για να είναι πλήρως ολοκληρωμένο το σύστημα σε θέματα ασφάλειας.

Βιβλιογραφία

- [1] Κ. Κεραυνού-Παπαηλιού Ελπίδα, Σημειώσεις μαθήματος «ΕΠΑ131 - Αρχές Προγραμματισμού Ι»
- [2] Κ. Δημήτρης Αποστόλου, «Ευφυή Συστήματα Υποστήριξης Αποφάσεων», <https://gunet2.cs.unipi.gr/modules/document/file.php/TMF126/KDSS%20Apostolou.pdf>
- [3] Κ. Βάσος Βασιλείου, Σημειώσεις μαθήματος «ΕΠΑ344 : Τεχνολογίες Διαδικτύου»
- [4] Jeff Kramer, «Abstraction – the key to Computing?», <http://www.sofismo.ch/links/kramer-abstraction.pdf>
- [5] HTML, <https://el.wikipedia.org/wiki/HTML>
- [6] CSS, <https://el.wikipedia.org/wiki/CSS>
- [7] JSP, https://en.wikipedia.org/wiki/JavaServer_Pages
- [8] PHP, <https://el.wikipedia.org/wiki/PHP>
- [9] BootStrap, <https://getbootstrap.com/>
- [10] Βίντεο, https://www.youtube.com/watch?v=B5KFv-Kzo0M&list=PLn_PwitlR7q1Lns-cckzSMTgtkSGorl6v&index=1
- [11] <https://www.w3.org/>
- [12] TomCat, Eclipse, <https://www.youtube.com/watch?v=h-yKgQtpbco>
- [13] JSP, <https://stackoverflow.com/questions/29983007/if-i-need-to-return-value-from-servlet-to-jsp>

- [14] client-server, https://en.wikipedia.org/wiki/Client%E2%80%93server_model
- [15] web cache, <https://refreshyourcache.com/en/cache/>
- [16] Κ. Γεωργία Καπιτσάκη, «ΕΠΛ 361 – Τεχνολογία Λογισμικού Ι»

Παράρτημα Α

Εισάγεται το θέμα που προτίθεστε να διατυπώσετε στο παράρτημα αυτό. Εισάγεται το θέμα που προτίθεστε να διατυπώσετε στο παράρτημα αυτό. Εισάγεται το θέμα που προτίθεστε να διατυπώσετε στο παράρτημα αυτό.

Ερωτηματολόγιο για Ατομική Διπλωματική Εργασία

Αγαπητοί συμμετέχοντες,

Το ερωτηματολόγιο αυτό απευθύνεται σε φοιτητές Πληροφορικής και στοχεύει στην μελέτη αξιολόγησης της ιδέας σχετικά με την ανάπτυξη συστήματος ενισχυτικής υποστήριξης φοιτητών για το μάθημα Αρχές Προγραμματισμού Ι (ΕΠΛ 131) .

Η έρευνα αυτή διεξάγεται στα πλαίσια εκπόνησης της διπλωματικής μου εργασίας για την απόκτηση προπτυχιακού τίτλου σπουδών στον κλάδο της «Πληροφορικής» του Πανεπιστημίου Κύπρου.

Δεν υπάρχουν σωστές ή λάθος απαντήσεις, θα ήθελα μόνο τη γνώμη σας. Η έρευνα είναι ανώνυμη και οι απαντήσεις σας θα χρησιμοποιηθούν μόνο στα πλαίσια ανάλυσης της έρευνας.

* Απαιτείται

Έτος φοίτησης *

- ☐ 1ο
- ☐ 2ο
- ☐ 3ο
- ☐ 4ο
- ☐ Άλλο

Πριν αρχίσεις τις σπουδές σου στο Πανεπιστήμιο Κύπρου, είχες παρακολουθήσει κάποιο μάθημα Πληροφορικής; *

☐ Ναι

☐ Όχι

Αντιμετώπισες δυσκολίες στο μάθημα ΕΠΛ 131; *

☐ Ναι

☐ Όχι

Αν Ναι, τι σε δυσκόλεψε περισσότερο;

Η απάντησή σας

Αν είχες τη δυνατότητα να χρησιμοποιούσες ένα σύστημα για καλύτερη εμπέδωση της ύλης του μαθήματος το οποίο να περιείχε διαλέξεις και διάφορα παιχνίδια, θα το χρησιμοποιούσες; *

1

2

3

4

Καθόλου

☐☐☐☐

Πάρα πολύ

Αν το συγκεκριμένο σύστημα παρείχε σημειώσεις για την θεωρία του κάθε κεφαλαίου με παραδείγματα και μικρά επεξηγηματικά βίντεο, πόσο χρήσιμο θα σου φαινόταν; *

	1	2	3	4	
Καθόλου	☐	☐	☐	☐	Πάρα πολύ

Αν το συγκεκριμένο σύστημα παρείχε ερωτήσεις πολλαπλής επιλογής για την ύλη κάθε κεφαλαίου, πόσο χρήσιμο θα σου φαινόταν; *

	1	2	3	4	
Καθόλου	☐	☐	☐	☐	Πάρα πολύ

Αν το συγκεκριμένο σύστημα παρείχε ασκήσεις συμπλήρωσης κώδικα βαθμωτής δυσκολίας (δηλαδή αρχίζοντας από τις πολύ απλές ασκήσεις και στη συνέχεια στις πιο δύσκολες) βάση κάποιου προβλήματος, πόσο χρήσιμο θα σου φαινόταν; *

	1	2	3	4	
Καθόλου	☐	☐	☐	☐	Πάρα πολύ

Αν το συγκεκριμένο σύστημα παρείχε ασκήσεις πολλαπλής επιλογής του τύπου δοθέντος ενός αποτελέσματος, να επιλέξεις τον σωστό κώδικα που θα έδινε το συγκεκριμένο αποτέλεσμα, πόσο χρήσιμο θα σου φαινόταν; *

	1	2	3	4	
Καθόλου	☐	☐	☐	☐	Πάρα πολύ

Στα παιχνίδια που προαναφέρθηκαν πόσο χρήσιμο πιστεύεις θα ήταν αν υπήρχε η δυνατότητα παροχής επιπέδων βοήθειας (δηλαδή το επίπεδο βοήθειας 1 να παρέχει μια πιο γενική βοήθεια και όσο αυξάνεται το επίπεδο βοήθειας να σε κατευθύνει πιο συγκεκριμένα προς την λύση); *

	1	2	3	4	
Καθόλου	☐	☐	☐	☐	Πάρα πολύ

Στα παιχνίδια που προαναφέρθηκαν πόσο χρήσιμο πιστεύεις θα ήταν αν υπήρχε η δυνατότητα παροχής λύσης μαζί με μια σύντομη επεξήγηση της λύσης; *

	1	2	3	4	
Καθόλου	☐	☐	☐	☐	Πάρα πολύ

Στα παιχνίδια που προαναφέρθηκαν πόσο χρήσιμο πιστεύεις
θα ήταν αν υπήρχε η δυνατότητα παροχής λύσης μαζί με μια
σύντομη επεξήγηση της λύσης; *

	1	2	3	4	
Καθόλου	☐	☐	☐	☐	Πάρα πολύ

Θα μπορούσες να εισηγηθείς άλλα παιχνίδια που πιστεύεις θα
βοηθούσαν για καλύτερη εμπέδωση της ύλης του μαθήματος;

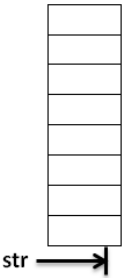
Η απάντησή σας

ΥΠΟΒΟΛΗ

Μην υποβάλετε ποτέ κωδικούς πρόσβασης μέσω των Φορμών Google.

Παράρτημα Β

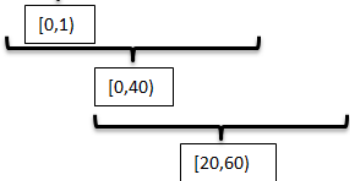
Ερώτηση	Τι είδους δεδομένα μπορεί να αποθηκεύσει η μεταβλητή char ch;																																							
Επιλογή 1	ένα μόνο χαρακτήρα																																							
Επιλογή 2	Αριθμούς																																							
Επιλογή 3	true/false τιμή																																							
Επιλογή 4	Κείμενο																																							
Απάντηση	<table><thead><tr><th>Τύποι Δεδομένων</th><th>Προεπιλεγμένη Τιμή</th><th>Προεπιλεγμένο Μέγεθος</th><th>Περιγραφή</th></tr></thead><tbody><tr><td>boolean</td><td>false</td><td>1 bit</td><td>Αυτός ο τύπος μπορεί να πάρει μόνο την τιμή True ή False</td></tr><tr><td>char</td><td>'\u0000'</td><td>1 byte</td><td>Αυτός ο τύπος μπορεί να κρατήσει έναν μόνο χαρακτήρα</td></tr><tr><td>byte</td><td>0</td><td>1 byte</td><td>Τύπος ακέραιου μικρού μεγέθους που πιάνει 1 byte μνήμη μπορεί να πάρει τιμές από -128 έως 127</td></tr><tr><td>short</td><td>0</td><td>2 byte</td><td>Τύπος ακέραιου που πιάνει 2 byte μνήμη μπορεί να πάρει τιμές από -32 768 έως 32 767</td></tr><tr><td>int</td><td>0</td><td>4 byte</td><td>Τύπος ακέραιου που πιάνει 4 byte μνήμη μπορεί να πάρει τιμές από -2 147 483 648 έως 2 147 483 647</td></tr><tr><td>long</td><td>0</td><td>8 byte</td><td>Τύπος ακέραιου μικρού μεγέθους που πιάνει 8 byte μνήμη μπορεί να πάρει τιμές από -9 223 372 036 854 775 808 έως 9 223 372 036 854 775 807</td></tr><tr><td>float</td><td>0.0</td><td>4 byte</td><td>Αυτός ο τύπος είναι για να ορίσουμε έναν αριθμό με δεκαδικά ψηφία μεγέθους 4byte (απλή ακρίβεια)</td></tr><tr><td>double</td><td>0.0</td><td>8byte</td><td>Είναι τύπος που μας επιτρέπει την τύπωση δεκαδικών αριθμών με πιο μεγάλο πλήθος δεκαδικών ψηφίων (διπλή ακρίβεια)</td></tr></tbody></table>				Τύποι Δεδομένων	Προεπιλεγμένη Τιμή	Προεπιλεγμένο Μέγεθος	Περιγραφή	boolean	false	1 bit	Αυτός ο τύπος μπορεί να πάρει μόνο την τιμή True ή False	char	'\u0000'	1 byte	Αυτός ο τύπος μπορεί να κρατήσει έναν μόνο χαρακτήρα	byte	0	1 byte	Τύπος ακέραιου μικρού μεγέθους που πιάνει 1 byte μνήμη μπορεί να πάρει τιμές από -128 έως 127	short	0	2 byte	Τύπος ακέραιου που πιάνει 2 byte μνήμη μπορεί να πάρει τιμές από -32 768 έως 32 767	int	0	4 byte	Τύπος ακέραιου που πιάνει 4 byte μνήμη μπορεί να πάρει τιμές από -2 147 483 648 έως 2 147 483 647	long	0	8 byte	Τύπος ακέραιου μικρού μεγέθους που πιάνει 8 byte μνήμη μπορεί να πάρει τιμές από -9 223 372 036 854 775 808 έως 9 223 372 036 854 775 807	float	0.0	4 byte	Αυτός ο τύπος είναι για να ορίσουμε έναν αριθμό με δεκαδικά ψηφία μεγέθους 4byte (απλή ακρίβεια)	double	0.0	8byte	Είναι τύπος που μας επιτρέπει την τύπωση δεκαδικών αριθμών με πιο μεγάλο πλήθος δεκαδικών ψηφίων (διπλή ακρίβεια)
Τύποι Δεδομένων	Προεπιλεγμένη Τιμή	Προεπιλεγμένο Μέγεθος	Περιγραφή																																					
boolean	false	1 bit	Αυτός ο τύπος μπορεί να πάρει μόνο την τιμή True ή False																																					
char	'\u0000'	1 byte	Αυτός ο τύπος μπορεί να κρατήσει έναν μόνο χαρακτήρα																																					
byte	0	1 byte	Τύπος ακέραιου μικρού μεγέθους που πιάνει 1 byte μνήμη μπορεί να πάρει τιμές από -128 έως 127																																					
short	0	2 byte	Τύπος ακέραιου που πιάνει 2 byte μνήμη μπορεί να πάρει τιμές από -32 768 έως 32 767																																					
int	0	4 byte	Τύπος ακέραιου που πιάνει 4 byte μνήμη μπορεί να πάρει τιμές από -2 147 483 648 έως 2 147 483 647																																					
long	0	8 byte	Τύπος ακέραιου μικρού μεγέθους που πιάνει 8 byte μνήμη μπορεί να πάρει τιμές από -9 223 372 036 854 775 808 έως 9 223 372 036 854 775 807																																					
float	0.0	4 byte	Αυτός ο τύπος είναι για να ορίσουμε έναν αριθμό με δεκαδικά ψηφία μεγέθους 4byte (απλή ακρίβεια)																																					
double	0.0	8byte	Είναι τύπος που μας επιτρέπει την τύπωση δεκαδικών αριθμών με πιο μεγάλο πλήθος δεκαδικών ψηφίων (διπλή ακρίβεια)																																					

Ερώτηση	Ποια είναι η προεπιλεγμένη τιμή (default value) της μεταβλητής String;
Επιλογή 1	""
Επιλογή 2	0
Επιλογή 3	Null
Επιλογή 4	δεν ορίζεται (not defined)
Απάντηση	Η μεταβλητή String δεν είναι αρχέγονος τύπος δεδομένων. Οπότε δεν έχει συγκεκριμένο μέγεθος, ούτε αρχικοποιείται με συγκεκριμένη τιμή. Απλώς αν την δηλώσω, χωρίς να την αρχικοποιήσω, θα δείχνει στο null (κενό). <pre>String str;</pre> Μνήμη 

Ερώτηση	Με ποια μέθοδο ξεκινά ένα πρόγραμμα java;
Επιλογή 1	system class
Επιλογή 2	main method
Επιλογή 3	static keyword
Επιλογή 4	κανένα από αυτά
Απάντηση	Ένα πρόγραμμα Java ξεκινά με την μέθοδο main. Αν δεν βάλουμε την main μέθοδο στο πρόγραμμα μας, κατά τη μεταγλώττιση του προγράμματος δεν θα πάρουμε σφάλμα (αφού δεν είναι κάτι συντακτικό), θα πάρουμε όμως σφάλμα κατά το χρόνο εκτέλεσης αφού το πρόγραμμα δεν θα βρίσκει από πού να ξεκινήσει. <pre> 1 class test { 2 3 public static void main(String args[]) { 4 5 6 } 7 } 8 </pre>

Ερώτηση	Το x=x+1 είναι ισοδύναμο με:
Επιλογή 1	+ + x
Επιλογή 2	- - x
Επιλογή 3	x + +
Επιλογή 4	x + 1
Απάντηση	<pre> class test { public static void main(String args[]) { int x=1; System.out.printf("1. x=%d\n", x++); System.out.printf("2. x=%d\n", x); System.out.printf("3. x=%d\n", x+1); System.out.printf("4. x=%d\n", --x); int y=5; System.out.printf("5. y=%d\n", ++y); System.out.printf("6. y=%d\n", y); System.out.printf("7. y=%d\n", y+1); System.out.printf("8. y=%d\n", --y); } } </pre> Για το παραπάνω πρόγραμμα τα αποτελέσματα είναι τα εξής: 1. x=1 2. x=2 3. x=3 4. x=1 5. x=6 6. x=6 7. x=7 8. x=5 Άρα το x++ είναι ισοδύναμο με το x=x+1.

Ερώτηση	Ο ακόλουθος κώδικας <code>int a = 4,6;</code> θα έχει ως αποτέλεσμα:
Επιλογή 1	σφάλμα σύνταξης (compilation error)
Επιλογή 2	σφάλμα χρόνου εκτέλεσης (runtime error)",
Επιλογή 3	να γίνει 4.6
Επιλογή 4	να γίνει 4
Απάντηση	Ο τύπος δεδομένων <code>int</code> παίρνει μόνο ακέραιες τιμές. Για να μπορέσω να φυλάξω την τιμή 4,6 σε τύπο <code>int</code> θα πρέπει να την κάνω casting σε τύπο <code>int</code>. <code>int a = (int) 4.6;</code>

Ερώτηση	Ο κώδικας <code>double r=(int) (Math.random() *40+20);</code> θα δημιουργήσει έναν τυχαίο αριθμό μεταξύ του εύρους;
Επιλογή 1	20 – 40
Επιλογή 2	0 – 40
Επιλογή 3	20 – 60
Επιλογή 4	0 – 60
Απάντηση	Η Java παρέχει την κλάση <code>Math</code> στο πακέτο <code>java.util</code> για τη δημιουργία τυχαίων αριθμών. Η κλάση <code>Math</code> περιέχει τη στατική μέθοδο <code>Math.random()</code> για τη δημιουργία τυχαίων αριθμών τύπου <code>double</code>. Η μέθοδος <code>random()</code> επιστρέφει μια <code>double</code> τιμή, μεγαλύτερη από ή ίση με 0,0 και μικρότερη από 1,0. Όταν καλείτε το <code>Math.random()</code>, δημιουργείται και χρησιμοποιείται ένα αντικείμενο γεννήτριας ψευδοτυχαίων αριθμών. Μπορείτε να χρησιμοποιήσετε τη μέθοδο <code>Math.random()</code>, με ή χωρίς παραμέτρους. Εάν δώσετε παραμέτρους, η μέθοδος παράγει τυχαίους αριθμούς μέσα στις συγκεκριμένες παραμέτρους. <code>double r = (int) (Math.random() * 40 + 20);</code> 
Βοήθεια 1	Η Java παρέχει την κλάση <code>Math</code> στο πακέτο <code>java.util</code> για τη δημιουργία τυχαίων αριθμών. Η κλάση <code>Math</code> περιέχει τη στατική μέθοδο <code>Math.random()</code> για τη δημιουργία τυχαίων αριθμών τύπου <code>double</code>. Η μέθοδος <code>random()</code> επιστρέφει μια <code>double</code> τιμή, μεγαλύτερη από ή ίση με 0,0 και μικρότερη από 1,0. Όταν καλείτε το <code>Math.random()</code>, δημιουργείται και χρησιμοποιείται ένα αντικείμενο γεννήτριας ψευδοτυχαίων αριθμών. Μπορείτε να χρησιμοποιήσετε τη μέθοδο <code>Math.random()</code>, με ή χωρίς παραμέτρους. Εάν δώσετε παραμέτρους, η μέθοδος παράγει τυχαίους αριθμούς μέσα στις συγκεκριμένες παραμέτρους.

Ερώτηση	Ποια είναι η έξοδος του ακόλουθου προγράμματος; <pre> public class test { public static void main(String args[]) { int value = 3, sum = 6 + --value; int data = --value + ++value / sum++ * value++ + ++sum % value--; System.out.println(data); } } </pre>																																
Επιλογή 1	1																																
Επιλογή 2	2																																
Επιλογή 3	0																																
Επιλογή 4	3																																
Απάντηση	<pre> int value = 3, sum = 6 + -- value; </pre> $\begin{array}{c} \text{---} \\ \boxed{2} \\ \text{---} \\ \boxed{8} \end{array}$ <pre> int data = --value + ++value / sum++ * value++ + ++sum % value--; </pre> $\begin{array}{cccccc} \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \boxed{1} & \boxed{2} & \boxed{8} & \boxed{3} & \boxed{9} & \boxed{2} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ & \boxed{\frac{1}{4} \rightarrow 0} & & & & \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ & \boxed{0} & & & \boxed{1} & \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ & & & & \boxed{2} & \end{array}$																																
Βοήθεια 1	<table border="1"> <thead> <tr> <th colspan="2">Προτεραιότητα τελεστών</th> </tr> </thead> <tbody> <tr><td colspan="2">. [] ()</td></tr> <tr><td colspan="2">++ -- ! ~ instanceof</td></tr> <tr><td colspan="2">* / %</td></tr> <tr><td colspan="2">+ -</td></tr> <tr><td colspan="2"><< >> >>></td></tr> <tr><td colspan="2">< > <= >=</td></tr> <tr><td colspan="2">== !=</td></tr> <tr><td colspan="2">&</td></tr> <tr><td colspan="2">^</td></tr> <tr><td colspan="2"> </td></tr> <tr><td colspan="2">&&</td></tr> <tr><td colspan="2"> </td></tr> <tr><td colspan="2">?:</td></tr> <tr><td colspan="2">= += -= *= /= %= &= = ^= <<= >>= >>>=</td></tr> <tr><td colspan="2">,</td></tr> </tbody> </table>	Προτεραιότητα τελεστών		. [] ()		++ -- ! ~ instanceof		* / %		+ -		<< >> >>>		< > <= >=		== !=		&		^				&&				?:		= += -= *= /= %= &= = ^= <<= >>= >>>=		,	
Προτεραιότητα τελεστών																																	
. [] ()																																	
++ -- ! ~ instanceof																																	
* / %																																	
+ -																																	
<< >> >>>																																	
< > <= >=																																	
== !=																																	
&																																	
^																																	
&&																																	
?:																																	
= += -= *= /= %= &= = ^= <<= >>= >>>=																																	
,																																	

Ερώτηση	Ποια είναι η τιμή του 'n' μετά την εκτέλεση του παρακάτω κώδικα; <pre> public class test { public static void main(String args[]) { int n = 20; switch (n) { case 10: n = n + 1; case 15: n = n + 2; case 20: n = n + 3; case 25: n = n + 4; case 30: n = n + 5; case 35: n = n + 6; } System.out.printf("n= %d", n); } } </pre>				
Επιλογή 1	25				
Επιλογή 2	38				
Επιλογή 3	23				
Επιλογή 4	αυτός ο κώδικας δεν μεταγλωττίζεται],				
Απάντηση	<pre> public class test { public static void main(String args[]) { int n = 20; switch (n) { case 10: n = n + 1; case 15: n = n + 2; case 20: n = n + 3; case 25: n = n + 4; case 30: n = n + 5; case 35: n = n + 6; } System.out.printf("n= %d", n); } } </pre> n=20 => n=23 Δεν υπάρχει break, άρα θα συνεχίσει στην επόμενη γραμμή. n=23 => n=27 Δεν υπάρχει break, άρα θα συνεχίσει στην επόμενη γραμμή. n=27 => n=32 Δεν υπάρχει break, άρα θα συνεχίσει στην επόμενη γραμμή. n=32 => n=38				
Βοήθεια 1	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 5px;">Ψευδοκώδικας switch</th> <th style="text-align: left; padding: 5px;">Σύνταξη switch</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px; vertical-align: top;"> Στην περίπτωση που (η μεταβλητή αυτή){ Έχει την τιμή (αυτή1): κάνε αυτό; Έχει την τιμή (αυτή2): κάνε αυτό; Έχει την τιμή (αυτή3): κάνε αυτό; Αλλιώς: Κάνε αυτό; } </td> <td style="padding: 5px; vertical-align: top;"> <pre> switch (έκφραση ελέγχου){ case (τιμή1): Εντολή 1; ... Εντολή n; break; case (τιμή2): Εντολή 1; ... Εντολή m; break; case (τιμή3): Εντολή 1; ... Εντολή z; break; default : Εντολή 1; ... Εντολή y; } </pre> </td> </tr> </tbody> </table>	Ψευδοκώδικας switch	Σύνταξη switch	Στην περίπτωση που (η μεταβλητή αυτή){ Έχει την τιμή (αυτή1): κάνε αυτό; Έχει την τιμή (αυτή2): κάνε αυτό; Έχει την τιμή (αυτή3): κάνε αυτό; Αλλιώς: Κάνε αυτό; } 	<pre> switch (έκφραση ελέγχου){ case (τιμή1): Εντολή 1; ... Εντολή n; break; case (τιμή2): Εντολή 1; ... Εντολή m; break; case (τιμή3): Εντολή 1; ... Εντολή z; break; default : Εντολή 1; ... Εντολή y; } </pre>
Ψευδοκώδικας switch	Σύνταξη switch				
Στην περίπτωση που (η μεταβλητή αυτή){ Έχει την τιμή (αυτή1): κάνε αυτό; Έχει την τιμή (αυτή2): κάνε αυτό; Έχει την τιμή (αυτή3): κάνε αυτό; Αλλιώς: Κάνε αυτό; } 	<pre> switch (έκφραση ελέγχου){ case (τιμή1): Εντολή 1; ... Εντολή n; break; case (τιμή2): Εντολή 1; ... Εντολή m; break; case (τιμή3): Εντολή 1; ... Εντολή z; break; default : Εντολή 1; ... Εντολή y; } </pre>				
Βοήθεια 2	Τα break χρησιμοποιείται για να βγω από το case στο οποίο βρίσκομαι αλλιώς το πρόγραμμα προχωρά στην επόμενη γραμμή.				

Ερώτηση	Τι εμφανίζεται στην οθόνη όταν μεταγλωττιστεί και εκτελεστεί το παρακάτω κομμάτι κώδικα; Επιλέξτε τις έξι σωστές απαντήσεις. a.0 0 b.0 1 c.0 2 d.0 3 e.1 0 f.1 1 g.1 2 h.1 3 i.2 0 j.2 1 k.2 2 l.2 3 m.3 0 n.3 1 o.3 2 p.3 3 <pre>public class test { public static void main(String args[]) { for (int i = 0; i < 3; i++) { for (int j = 3; j >= 0; j--) { if (i == j) break; System.out.println(i + " " + j); } } } }</pre>																																																																																																			
Επιλογή 1	a, b, c, f, h, i																																																																																																			
Επιλογή 2	c, d, g, h, i, k																																																																																																			
Επιλογή 3	d, c, b, h, g, l																																																																																																			
Επιλογή 4	d, e, f, g, j, k																																																																																																			
Απάντηση	<table><tr><th>i</th><th>i < 3</th><th>true / false</th><th>j</th><th>j >= 0</th><th>true / false</th><th>i == j</th><th>true / false</th><th>i + " " + j</th></tr><tr><td>0</td><td>0 < 3</td><td>true</td><td>3</td><td>3 >= 0</td><td>true</td><td>0==3</td><td>false</td><td>0 3</td></tr><tr><td>0</td><td>0 < 3</td><td>true</td><td>2</td><td>2 >= 0</td><td>true</td><td>0==2</td><td>false</td><td>0 2</td></tr><tr><td>0</td><td>0 < 3</td><td>true</td><td>1</td><td>1 >= 0</td><td>true</td><td>0==1</td><td>false</td><td>0 1</td></tr><tr><td>0</td><td>0 < 3</td><td>true</td><td>0</td><td>0 >= 0</td><td>true</td><td>0==0</td><td>true -> break; -> i = 1</td><td></td></tr><tr><td>1</td><td>1 < 3</td><td>true</td><td>3</td><td>3 >= 0</td><td>true</td><td>1==3</td><td>false</td><td>1 3</td></tr><tr><td>1</td><td>1 < 3</td><td>true</td><td>2</td><td>2 >= 0</td><td>true</td><td>1==2</td><td>false</td><td>1 2</td></tr><tr><td>1</td><td>1 < 3</td><td>true</td><td>1</td><td>1 >= 0</td><td>true</td><td>1==1</td><td>true -> break; -> i = 2</td><td></td></tr><tr><td>2</td><td>2 < 3</td><td>true</td><td>3</td><td>3 >= 0</td><td>true</td><td>2==3</td><td>false</td><td>2 3</td></tr><tr><td>2</td><td>2 < 3</td><td>true</td><td>2</td><td>2 >= 0</td><td>true</td><td>2==2</td><td>true -> break; -> i = 3</td><td></td></tr><tr><td>3</td><td>3 < 3</td><td>false</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	i	i < 3	true / false	j	j >= 0	true / false	i == j	true / false	i + " " + j	0	0 < 3	true	3	3 >= 0	true	0==3	false	0 3	0	0 < 3	true	2	2 >= 0	true	0==2	false	0 2	0	0 < 3	true	1	1 >= 0	true	0==1	false	0 1	0	0 < 3	true	0	0 >= 0	true	0==0	true -> break; -> i = 1		1	1 < 3	true	3	3 >= 0	true	1==3	false	1 3	1	1 < 3	true	2	2 >= 0	true	1==2	false	1 2	1	1 < 3	true	1	1 >= 0	true	1==1	true -> break; -> i = 2		2	2 < 3	true	3	3 >= 0	true	2==3	false	2 3	2	2 < 3	true	2	2 >= 0	true	2==2	true -> break; -> i = 3		3	3 < 3	false						
i	i < 3	true / false	j	j >= 0	true / false	i == j	true / false	i + " " + j																																																																																												
0	0 < 3	true	3	3 >= 0	true	0==3	false	0 3																																																																																												
0	0 < 3	true	2	2 >= 0	true	0==2	false	0 2																																																																																												
0	0 < 3	true	1	1 >= 0	true	0==1	false	0 1																																																																																												
0	0 < 3	true	0	0 >= 0	true	0==0	true -> break; -> i = 1																																																																																													
1	1 < 3	true	3	3 >= 0	true	1==3	false	1 3																																																																																												
1	1 < 3	true	2	2 >= 0	true	1==2	false	1 2																																																																																												
1	1 < 3	true	1	1 >= 0	true	1==1	true -> break; -> i = 2																																																																																													
2	2 < 3	true	3	3 >= 0	true	2==3	false	2 3																																																																																												
2	2 < 3	true	2	2 >= 0	true	2==2	true -> break; -> i = 3																																																																																													
3	3 < 3	false																																																																																																		
Βοήθεια 1	<pre>if (<συνθήκη>) { <αν ισχύει η συνθήκη κάνει αυτό> } else { <αν δεν ισχύει η συνθήκη κάνει αυτό> }</pre>																																																																																																			
Βοήθεια 2	<pre>for (έκφραση αρχικοποίησης; συνθήκη; έκφραση ενημέρωσης){ εντολή 1; ... εντολή m; }</pre>																																																																																																			
Βοήθεια 3	break Αν βρισκόμαστε σε ένα βρόγχο και η ροή εκτέλεσης συναντήσει την εντολή break, θα μεταφερθεί αμέσως εκτός του βρόγχου και θα συνεχίσει εκτελώντας την αμέσως επόμενη εντολή.																																																																																																			

Ερώτηση	Ποια έκφραση μπορεί να χρησιμοποιηθεί για την πρόσβαση στο τελευταίο στοιχείο ενός πίνακα;										
Επιλογή 1	array[array.length]										
Επιλογή 2	array[array.length()]										
Επιλογή 3	array[array.length-1]										
Επιλογή 4	array[array.length()-1]										
Απάντηση	Με την δήλωση array.length, μπορώ να πάρω το μέγεθος του πίνακα array. Όχι σε μνήμη, αλλά το πόσες θέσεις έχει. Για παράδειγμα, έχουμε ένα πίνακα 5 θέσεων και θέλω να πάρω το τελευταίο του στοιχείο. Η αρίθμηση των κελιών του πίνακα θα ξεκινά από το 0 έως το 4. <table><tr><td>array[0]</td><td>array[1]</td><td>array[2]</td><td>array[3]</td><td>array[4]</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 5 θέσεις => array.length = 5 Για να έχω πρόσβαση στο τελευταίο στοιχείο του πίνακα θα έγραφα array[array.length-1] 5 - 1 = 4	array[0]	array[1]	array[2]	array[3]	array[4]	0	0	0	0	0
array[0]	array[1]	array[2]	array[3]	array[4]							
0	0	0	0	0							
Βοήθεια 1	Με την δήλωση array.length, μπορώ να πάρω το μέγεθος του πίνακα array. Όχι σε μνήμη, αλλά το πόσες θέσεις έχει.										

Ερώτηση	Τι εμφανίζει το παρακάτω πρόγραμμα Java; <pre> public class test { int x = 2; test(int i) { x = i; } public static void main(String[] args) { test t = new test(5); System.out.println("x = " + t.x); } } </pre>
Επιλογή 1	τυπώνει x = 2
Επιλογή 2	τυπώνει x = 5
Επιλογή 3	τυπώνει x = i
Επιλογή 4	τυπώνει x = t.x

Απάντηση	<pre> public class test { int x = 2; test(int i) { x = i; } public static void main(String[] args) { test t = new test(5); System.out.println("x = " + t.x); } } </pre> Η έξοδος του παραπάνω προγράμματος είναι "x = 5". Με το που δημιουργώ αντικείμενο τύπου test, το αντικείμενο παίρνει τα χαρακτηριστικά της κλάσης. Έτσι το x παίρνει τιμή 2. Στη συνέχεια, αφού καλέσω τον κατασκευαστή της κλάσης με όρισμα 5, το x θα πάρει την τιμή 5.
Βοήθεια 1	Για τη δημιουργία αντικειμένων χρησιμοποιούμε την ακόλουθη εντολή: <όνομα κλάσης> <όνομα αντικειμένου> = new <όνομα κλάσης>(<παράμετροι κατασκευαστή>); Χρησιμοποιώντας τον τελεστή new, καλείται ο κατασκευαστής της κλάσης και δημιουργείται το νέο αντικείμενο. Μέσα στην παρένθεση δίνουμε τα ορίσματα που δέχεται ως παράμετρους ο κατασκευαστής της κλάσης. Οι κατασκευαστές εκτελούνται κατά τη δημιουργία αντικειμένων και φροντίζουν για την αρχικοποίηση τους. Έχουν το ίδιο όνομα με την κλάση στην οποία ορίζονται. Ποτέ δεν επιστρέφουν τίποτα. Στην Java υπάρχει ο προκαθορισμένος κατασκευαστής όπου αν δεν τον καθορίσουμε, τότε ο μεταγλωττιστής παρέχει αυτόματα έναν κενό κατασκευαστή χωρίς παραμέτρους.

Παράρτημα Γ

Επέλεξε τον κώδικα που θα δώσει το παρακάτω αποτέλεσμα	<pre> * ** *** **** ***** </pre>
Επιλογή 1	<pre> for (i = 1; i <= 5; i++) { for (j = 1; j <= i; j++) System.out.print("*"); System.out.println(); } </pre>
Επιλογή 2	<pre> for (i = 1; i <= 5; i++) { for (j = 1; i <= i; j++) System.out.print("*"); System.out.println(); } </pre>
Επιλογή 3	<pre> for (i = 1; i <= 5; i++) { for (j = 1; j <= i; j++) System.out.println("*"); System.out.println(); } </pre>
Επιλογή 4	<pre> for (i = 1; i <= i; i++) { for (j = 1; j <= 5; j++) System.out.print("*"); System.out.println(); } </pre>

Επέλεξε τον κώδικα που θα δώσει το παρακάτω αποτέλεσμα	<pre> * ** *** **** *** ** * </pre>
Επιλογή 1	<pre> int z = 3, i = 10, j = 20; for (i = z; i >= -z; i--) { for (j = z; j >= Math.abs(i); j--) { System.out.print("*"); } System.out.println(); } </pre>

Επιλογή 2	<pre> int z = 3, i = 10, j = 20; for (i = z; i >= z; i--) { for (j = z; j >= Math.abs(i); j--) { System.out.print("*"); } System.out.println(); } </pre>
Επιλογή 3	<pre> int z = 3, i = 10, j = 20; for (i = z; i >= z; i++) { for (j = z; j >= Math.abs(i); j--) { System.out.print("*"); } System.out.println(); } </pre>
Επιλογή 4	<pre> int z = 3, i = 10, j = 20; for (i = z; i >= z; i++) { for (j = z; j >= Math.abs(i); j--) { System.out.print("*"); } System.out.print("\t"); } </pre>

Επέλεξε τον κώδικα που θα δώσει το παρακάτω αποτέλεσμα	<pre> 3 23 123 0123 123 23 3 </pre>
Επιλογή 1	<pre> int z = 3, i, j, k; for (i = z; i >= -z; i--) { for (j = 1; j <= Math.abs(i); j++) { System.out.print(" "); } for (k = Math.abs(i); k <= z; k++) { System.out.print(k); } System.out.print("\n"); } </pre>

Επιλογή 2	<pre> int z = 3, i, j, k; for (i = z; i >= -z; i--) { for (j = 1; j <= Math.abs(i); j++) { System.out.print(" "); } for (k = Math.abs(j); k <= z; k++) { System.out.print(k); } System.out.print("\n"); } </pre>
Επιλογή 3	<pre> int z = 3, i, j, k; for (i = z; i >= -z; i--) { for (j = 1; j <= Math.abs(z); j++) { System.out.print(" "); } for (k = Math.abs(j); k <= z; k++) { System.out.print(k); } System.out.print("\n"); } </pre>
Επιλογή 4	<pre> int z = 3, i, j, k; for (i = z; i >= -z; i--) { for (j = 1; j <= Math.abs(i); j++) { System.out.print(" "); } for (k = Math.abs(j); k <= z; k--) { System.out.print(k); } System.out.print("\n"); } </pre>

Επέλεξε τον κώδικα που θα δώσει το παρακάτω αποτέλεσμα	<pre> * ** * * * * * * * * * * ***** </pre>
--	--

Επιλογή 1	<pre> int i, j, n = 5; System.out.print("\n"); for (i = 1; i <= n; i++) { System.out.printf("*"); for (j = 1; j < i; j++) { System.out.print(" "); } System.out.print("\n"); } j = 1; while (j <= n + 2) { System.out.print("*"); j++; } </pre>
Επιλογή 2	<pre> int i, j, n = 5; System.out.print("\n"); for (i = 1; i <= n; i++) { System.out.printf("*"); for (j = 1; j < i; j++) { System.out.print(" "); } System.out.print("\n"); } j = 1; while (j <= n - 2) { System.out.print("*"); j++; } </pre>
Επιλογή 3	<pre> int i, j, n = 5; System.out.print("\n"); for (i = 1; i <= n; i++) { System.out.printf("*"); for (j = 1; j > i; j++) { System.out.print(" "); } System.out.print("\n"); } j = 1; while (j <= n - 2) { System.out.print("*"); j++; } </pre>
Επιλογή 4	<pre> int i, j, n = 5; System.out.print("\n"); for (i = 1; i <= n; i++) { System.out.printf("*"); for (j = 1; j > i; j++) { System.out.print(" "); } System.out.print("\n"); } j = 1; while (j <= n - 2) { System.out.print("*"); j++; } </pre>

Παράρτημα Δ

Ερώτηση	Συμπληρώστε το ακόλουθο πρόγραμμα το οποίο θα τυπώνει «hot» αν το temperature είναι πάνω από 25, αλλιώς θα τυπώνει «cold». Χρησιμοποιείτε τριαδικό τελεστή.
Απάντηση	<pre>class test { public static void main(String[] args) { int temperature = 20; String msg = (temperature > 25) ? "hot" : "cold"; System.out.println(msg); } }</pre>
Ασυμπλήρωτος κώδικας	<pre>class test { public static void main(String[] args) { int temperature = 20; String msg = System.out.println(msg); } }</pre>
	cold

Ερώτηση	Συμπληρώστε το ακόλουθο πρόγραμμα με do/while ώστε να τυπώνει την μεταβλητή x, όταν έχει τιμές μικρότερες του 6.
Απάντηση	<pre>class test { public static void main(String[] args) { int x = 3; do { System.out.print(x++ + " "); } while (x < 6); } }</pre>
Ασυμπλήρωτος κώδικας	<pre>class test { public static void main(String[] args) { int x = 3; { System.out.print(x++ + " "); } } }</pre>

	<pre> } } }</pre>
	3 4 5

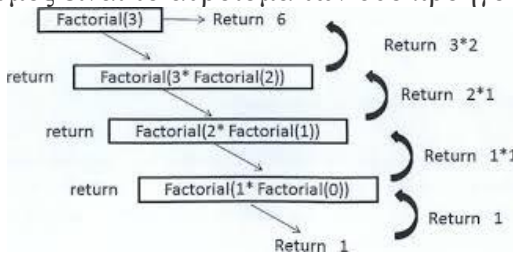
Ερώτηση	Συμπληρώστε το ακόλουθο πρόγραμμα το οποίο θα ελέγχει αν ένας αριθμός είναι παλίνδρομος. Παράδειγμα παλίνδρομου αριθμού είναι ο αριθμός 146787641.
Απάντηση	<pre> public class test { public static void main(String[] args) { int num = 121, reversedInteger = 0, remainder; int originalInteger = num; while (num != 0) { remainder = num % 10; reversedInteger = reversedInteger * 10 + remainder; num /= 10; } if (originalInteger == reversedInteger) System.out.println(originalInteger + " is a palindrome number"); else System.out.println(originalInteger + " is not a palindrome number"); } } }</pre>
Ασυμπλήρωτος κώδικας	<pre> public class test { public static void main(String[] args) { int num = 121, reversedInteger = 0, remainder; int originalInteger = num; if (originalInteger == reversedInteger) System.out.println(originalInteger + " is a palindrome number"); else System.out.println(originalInteger + " is not a palindrome number"); } } }</pre>
	121 is a palindrome number

Ερώτηση	Συμπληρώστε τον παρακάτω κώδικα ώστε να αντικαταστήσετε το «huawei» με «lg».
Απάντηση	<pre> class test { public static void main(String[] args) { String[] phones = { "iphone", "samsung", "huawei" }; phones[2] = "lg"; for (String i : phones) System.out.printf(i + " "); } } </pre>
Ασυμπλήρωτος κώδικας	<pre> class test { public static void main(String[] args) { String[] phones = { "iphone", "samsung", "huawei" }; for (String i : phones) System.out.printf(i + " "); } } </pre>
	iphone samsung lg

Ερώτηση	Σας δίνεται ένας πίνακας a και ένας πίνακας b. Συμπληρώστε τον παρακάτω κώδικα ώστε να βρείτε το άθροισμα των γινομένων αντίστοιχων στοιχείων από τους πίνακες a και b. Παράδειγμα: a <table><tr><td>1</td><td>6</td><td>3</td><td>7</td><td>5</td></tr></table> b <table><tr><td>3</td><td>5</td><td>7</td><td>2</td><td>5</td></tr></table> sum <- 1*3+6*5+3*7+7*2+5*5=93	1	6	3	7	5	3	5	7	2	5
1	6	3	7	5							
3	5	7	2	5							
Απάντηση	<pre>class test { public static void main(String[] args) { int i = 0, sum = 0; int[] a = { 1, 6, 3, 7, 5 }; int[] b = { 3, 5, 7, 2, 5 }; for (i = 0; i < b.length; i++) { sum += a[i] * b[i]; } System.out.printf("sum = %d", sum); } }</pre>										

	}
Ασυμπλήρωτος κώδικας	<pre> class test { public static void main(String[] args) { int i = 0, sum = 0; int[] a = { 1, 6, 3, 7, 5 }; int[] b = { 3, 5, 7, 2, 5 }; for (i = 0; i < b.length; i++) { } System.out.printf("sum = %d", sum); } } </pre>
	sum = 93

Ερώτηση	Συμπληρώστε το ακόλουθο πρόγραμμα υλοποιώντας την μέθοδο smallest, η οποία θα δέχεται τρεις αριθμούς και θα επιστρέφει τον μικρότερο.
Απάντηση	<pre> class test { public static void main(String[] args) { double x = 48; double y = 30; double z = 60; System.out.print("The smallest value is " + smallest(x, y, z) + "\n"); } public static double smallest(double x, double y, double z) { return Math.min(Math.min(x, y), z); } } </pre>
Ασυμπλήρωτος κώδικας	<pre> class test { public static void main(String[] args) { double x = 48; double y = 30; double z = 60; System.out.print("The smallest value is " + smallest(x, y, z) + "\n"); } public static double smallest() { } } </pre>
	The smallest value is 30

Ερώτηση	Συμπληρώστε το ακόλουθο πρόγραμμα υλοποιώντας την μέθοδο <i>fibonacci</i>, η οποία θα επιστρέφει τον n-οστό αριθμό της ακολουθίας <i>fibonacci</i>. Χρησιμοποιήστε αναδρομή. Υπενθύμιση: ακολουθία <i>fibonacci</i>: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,... $fibonacci(0)=0$ $fibonacci(1)=1$ Ο επόμενος αριθμός είναι το άθροισμα των δύο προηγούμενων αριθμών. 
Απάντηση	<pre> class test { public static void main(String[] args) { int num = 8; System.out.printf("Ο %δος όρος της ακολουθίας είναι το %d.", num + 1, fibonacci(num)); } public static int fibonacci(int n) { if (n == 1) { return 1; } else if (n == 0) { return 0; } else { return fibonacci(n - 1) + fibonacci(n - 2); } } } </pre>
Ασυμπλήρωτος κώδικας	<pre> class test { public static void main(String[] args) { int num = 8; System.out.printf("Ο %δος όρος της ακολουθίας είναι το %d.", num + 1, fibonacci(num)); } public static int fibonacci(int n) { } } </pre>
	Ο 9ος όρος της ακολουθίας είναι το 21.

Ερώτηση	Συμπληρώστε το ακόλουθο πρόγραμμα υλοποιώντας την μέθοδο <code>print_ch</code>, η οποία θα τυπώνει τους χαρακτήρες μεταξύ δύο χαρακτήρων. Η συνάρτηση θα δέχεται τους δύο χαρακτήρες καθώς και τον αριθμό των χαρακτήρων που θα τυπώνει σε κάθε γραμμή. <pre>print_ch('(', 'z', 20);</pre> το πρόγραμμα θα τυπώνει: <pre>() * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\] ^ _ ` a b c d e f g h i j k l m n o p q r s t u v w x y z</pre>
Απάντηση	<pre>class test { public static void main(String[] args) { print_ch('+', 'r', 20); } public static void print_ch(char char1, char char2, int n) { for (int ctr = 1; char1 <= char2; ctr++, char1++) { System.out.print(char1 + " "); if (ctr % n == 0) System.out.println(""); } System.out.print("\n"); } }</pre>
Ασυμπλήρωτος κώδικας	<pre>class test { public static void main(String[] args) { print_ch('+', 'r', 20); } public static void print_ch(char char1, char char2, int n) { } }</pre>
	<pre>+ , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\] ^ _ ` a b c d e f g h i j k l m n o p q r</pre>

Ερώτηση	Συμπληρώστε το ακόλουθο πρόγραμμα ώστε να καλέσετε τις συναρτήσεις <code>myStaticMethod</code> και <code>myPublicMethod</code>.
Απάντηση	<pre> class m { static void myStaticMethod() { System.out.println("Static methods can be called without creating objects"); } public void myPublicMethod() { System.out.println("Public methods must be called by creating objects"); } public static void main(String[] args) { myStaticMethod(); m myObj = new m(); myObj.myPublicMethod(); } } </pre>
Ασυμπλήρωτος κώδικας	<pre> class m { static void myStaticMethod() { System.out.println("Static methods can be called without creating objects"); } public void myPublicMethod() { System.out.println("Public methods must be called by creating objects"); } public static void main(String[] args) { m myObj = new m(); } } </pre>
	Static methods can be called without creating objects Public methods must be called by creating objects

Ερώτηση	
---------	--

	Συμπληρώστε το ακόλουθο πρόγραμμα ώστε να υπολογίζεται το παραγοντικό του αριθμού 5. Συμπληρώστε τις συναρτήσεις function και paragontiko χωρίς να αλλάξετε την main.
Απάντηση	<pre> class m { public static int function(int num) { return paragontiko(num); } public static int paragontiko(int num) { int p = 1; for (int i = 1; i <= num; i++) p = p * i; return p; } public static void main(String[] args) { int b = function(5); System.out.printf("Το παραγοντικό του αριθμού 5 είναι: %d", b); } } </pre>
Ασυμπλήρωτος κώδικας	<pre> class m { public static int function(int num) { return ; } public static int paragontiko(int num) { int p = ; for (int i = 1;) return p; } public static void main(String[] args) { int b = function(5); System.out.printf("Το παραγοντικό του αριθμού 5 είναι: %d", b); } } </pre>
	Το παραγοντικό του αριθμού 5 είναι: 120