



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Ψηφιακών Συστημάτων

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφοριακά Συστήματα και Υπηρεσίες»

Ειδίκευση «Προηγμένα Πληροφοριακά Συστήματα»

Μάθημα: «Η Γλώσσα Προγραμματισμού Java»

Ακαδημαϊκό έτος 2023-2024

ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ

Προθεσμία Παράδοσης: 27/1/2025

1 Γενικά

Η εργασία αυτή θα σας βοηθήσει να εξοικειωθείτε με τις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού και με τον προγραμματισμό σε γλώσσα Java.

Στο πλαίσιο της εργασίας καλείστε να παραδώσετε μέσω της πλατφόρμας e-class «Λεύκιππος» α) τον **πηγαίο κώδικα** του συστήματος που θα αναπτύξετε και β) μια **αναφορά** που θα συνοψίζει την προσέγγιση που ακολουθήσατε.

Η εργασία μπορεί να εκπονηθεί είτε ατομικά ή σε ομάδες των δύο φοιτητών και θα πρέπει να παραδοθεί μέχρι τις **27/1/2025, ώρα 23:59**. Μετά την παρέλευση της ημερομηνίας αυτής δεν πρόκειται να γίνουν δεκτές εργασίες.

2 Περιγραφή Εργασίας

Η εργασία περιλαμβάνει την ανάπτυξη εφαρμογής – παιχνίδι γνώσεων – αξιοποιώντας δημόσια διαθέσιμες βάσεις γνώσεων μέσω των διεπαφών τους για τη δημιουργία των ερωτήσεων. Η εργασία θα υλοποιηθεί σε δύο μέρη την κατασκευή Α) της βιβλιοθήκης και Β) της εφαρμογής-παιχνίδι.

ΜΕΡΟΣ Α: ΚΑΤΑΣΚΕΥΗ ΒΙΒΛΙΟΘΗΚΗΣ “Trivia” API

Στο πλαίσιο του Μέρους Α της εργασίας θα πρέπει να υλοποιηθούν οι κλάσεις μέσω των οποίων θα γίνεται αναζήτηση των ερωτήσεων με βάση συγκεκριμένα κριτήρια. Συγκεκριμένα, οι κλάσεις που θα αναπτυχθούν θα υλοποιούν τις ακόλουθες λειτουργίες:

1. Ανάκτηση λίστας ερωτήσεων.
2. Παραμετροποίηση της ανάκτησης ερωτήσεων με την εφαρμογή φίλτρων:
 - i. Το πλήθος των ερωτήσεων που θα περιλαμβάνει η λίστα
 - ii. Τον τύπο των ερωτήσεων (Σωστό / Λάθος, Πολλαπλής επιλογής, ή όλες)
 - iii. Την κατηγορία των ερωτήσεων
 - iv. Το βαθμό δυσκολίας

Η λήψη όλων των παραπάνω δεδομένων θα γίνεται σε **πραγματικό χρόνο** μέσω ενός **REST API**, όπως για παράδειγμα το REST API “Open Trivia DB” (https://opentdb.com/api_config.php). Οι κλάσεις που θα αναπτυχθούν στο πλαίσιο του Μέρους Α της εργασίας θα διαχειρίζονται όλη την επικοινωνία με το REST

API, δηλαδή:

- **Κλήση API** για την **ανάκτηση ερωτήσεων** που ικανοποιούν συγκεκριμένα κριτήρια.
- **Λήψη των ερωτήσεων** σε μορφή JSON (**JSON response**).
- **Αποσειριοποίηση (deserialization)** του **response** στα σχετικά Java Objects. Για το σκοπό αυτό, θα πρέπει να υλοποιηθούν και οι αντίστοιχες POJO (Plain Old Java Object) κλάσεις, αντικείμενα των οποίων θα χρησιμοποιηθούν για την αποθήκευση των δεδομένων που θα εξάγονται από το JSON response.

Στη συνέχεια, από τις παραπάνω κλάσεις, θα **κατασκευαστεί μια βιβλιοθήκη (αρχείο .jar)**, η οποία θα **εγκατασταθεί στο τοπικό αποθετήριο του maven**, ώστε η λειτουργικότητα που παρέχει να μπορεί να ενσωματωθεί εύκολα και να αξιοποιηθεί από οποιαδήποτε εφαρμογή Java.

ΜΕΡΟΣ Β: ΚΑΤΑΣΚΕΥΗ ΕΦΑΡΜΟΓΗΣ “ΠΑΙΧΝΙΔΙ ΓΝΩΣΕΩΝ”

Στο πλαίσιο του Μέρους Β της εργασίας, θα **αναπτυχθεί μία εφαρμογή Java** (GUI ή web-based application) η οποία θα παρέχει στους χρήστες τη δυνατότητα να παίξουν ένα παιχνίδι γνώσεων και να διαγωνιστούν μέσα από την απάντηση σε ερωτήσεις πολλαπλής επιλογής και σωστό/λάθος.

1. Δημιουργία νέου παιχνιδιού με προεπιλεγμένες ρυθμίσεις ή επιλογή παραμέτρων του παιχνιδιού όπως κατηγορία ερωτήσεων, τύπος, πλήθος και βαθμός δυσκολίας.
2. Ανάκτηση των ερωτήσεων μέσω του API με βάση τις επιλογές του παιχνιδιού.
3. Εκκίνηση του παιχνιδιού με εμφάνιση κάθε ερώτησης, ανάλογα με τον τύπο της, και τη δυνατότητα στο χρήστη να την απαντήσει.
4. Μετά την απάντηση, ο χρήστης θα ενημερώνεται για την επιτυχία ή το λάθος του και θα γίνεται μετάβαση στην επόμενη ερώτηση.
5. Μετά την απάντηση της τελευταίας ερώτησης, ο χρήστης θα βλέπει το ποσοστό επιτυχίας και το συνολικό σκορ, με βάση ένα σύστημα βαθμολόγησης της επιλογής σας (π.χ. +10 στη σωστή απάντηση, -5 στη λάθος).
6. Το μέγιστο σκορ θα διατηρείται και ο χρήστης θα μπορεί να ξεκινήσει ένα νέο παιχνίδι.
7. Αν στο νέο παιχνίδι επιτευχθεί ένα καλύτερο μέγιστο σκορ, τότε θα εμφανίζεται στο χρήστη σχετικό μήνυμα.
8. Αν ο χρήστης επιλέξει διαφορετικές παραμέτρους για τις ερωτήσεις, τότε το μέγιστο σκορ θα διαγράφεται.

Για την υλοποίηση των ανωτέρω λειτουργιών της εφαρμογής θα πρέπει να χρησιμοποιηθούν οι κλάσεις της βιβλιοθήκης “Trivia” API που υλοποιήθηκε στο Μέρος Α της εργασίας.

3 Σημειώσεις

1. Η διαχείριση των projects που θα υλοποιηθούν στο πλαίσιο της εργασίας θα πρέπει να γίνει με χρήση του **maven** ή του **gradle** (π.χ. για την καλύτερη διαχείριση βιβλιοθηκών που θα χρησιμοποιηθούν, συμπεριλαμβανομένης και της βιβλιοθήκης που θα κατασκευαστεί στο Μέρος Α της εργασίας και θα αξιοποιηθεί στο Μέρος Β της εργασίας).
2. Σε όλες τις κλάσεις θα πρέπει να έχουν ενσωματωθεί κατάλληλοι **μηχανισμοί χειρισμού εξαιρέσεων (exception handling)**, ώστε το σύστημα να είναι σταθερό.
3. Ο πηγαίος κώδικας θα πρέπει να περιέχει **επαρκή σχολιασμό**.
4. Η **εφαρμογή** που θα υλοποιηθεί στο Μέρος Β της εργασίας θα πρέπει να είναι **φιλική προς το χρήστη** (π.χ. εμφάνιση μηνυμάτων που να ενημερώνουν το χρήστη για κάποια ενέργεια που έκανε ή πρόκειται να κάνει, εμφάνιση λίστας επιλογών όπου αυτό απαιτείται, όπως για

παράδειγμα οι κατηγορίες ή το επίπεδο δυσκολίας κ.λπ.).

5. Καθ' όλη τη διάρκεια ανάπτυξης της ζητούμενης εφαρμογής, θα πρέπει να πραγματοποιούνται **unit tests με τη χρήση του Junit** προκειμένου να ελέγχονται οι υπό ανάπτυξη λειτουργίες. Όσα tests πραγματοποιηθούν θα πρέπει να συμπεριληφθούν στην εργασία που θα παραδοθεί.
6. Η **διαχείριση των διάφορων εκδόσεων του κώδικα** θα πρέπει να γίνει με χρήση του **GIT**, ενώ η **τελική έκδοση του κώδικα θα πρέπει να αποθηκευτεί στο Github**.

4 Οδηγίες ανάπτυξης και εξαγωγής πηγαίου κώδικα εργασίας από το Eclipse

Για την ανάπτυξη του πηγαίου κώδικα της εργασίας και την εκτέλεση θα χρησιμοποιήσετε το ολοκληρωμένο περιβάλλον ανάπτυξης **Eclipse**. Για την εξαγωγή και αποθήκευση του πηγαίου κώδικα της εργασίας, πρέπει να ακολουθήσετε τα εξής βήματα μέσα από το Eclipse:

1. Από το μενού του Eclipse επιλέγετε **File ->Export**.
2. Στο παράθυρο που σας ανοίγει, επιλέγετε από την κατηγορία **"General"**, την επιλογή **"Archive File"** και πατάτε **"Next"**.
3. Επιλέγετε το **project** στο πάνω αριστερά μέρος (κάνετε τικ στο κουτάκι στα αριστερά του ονόματος) και πατάτε το κουμπί **"Browse..."** για να επιλέξετε τη διαδρομή που θα αποθηκευτεί το αρχείο.
4. Τέλος, πατάτε το κουμπί **"Finish"** για να αποθηκευτεί ο κώδικάς σας.

5 Παράδοση Εργασίας και Αναφορά

Η εργασία θα παραδοθεί μέσω την πλατφόρμας Lefkirros (ενότητα «Εργασίες») ως ένα zip αρχείο της μορφής AM1-AM2.zip, όπου AM1, AM2 οι αριθμοί μητρώου των μελών της ομάδας. Το αρχείο θα περιλαμβάνει δύο φακέλους, ένα φάκελο με την αναφορά σε μορφή AM1-AM2.docx και AM1-AM2.pdf και ένα φάκελο με τον κώδικα.

Η αναφορά θα περιλαμβάνει τα παρακάτω στοιχεία:

- Περιγραφή της χρήσης του REST API και της υλοποίησης των POJOs
- Ανάλυση της προσέγγισης που ακολουθήσατε για την υλοποίηση της εφαρμογής συνολικά αλλά και για κάθε μία από της λειτουργίες της
- Εικόνες από την εφαρμογή σε λειτουργία
- Ενδεικτικά αποτελέσματα των δοκιμών Junit
- Περιγραφή της χρήσης των εργαλείων που αξιοποιήσατε (π.χ. postman, eclipse, gradle, git)
- **Αναλυτική περιγραφή της συνεισφοράς κάθε μέλους της ομάδας στην εργασία**

Στην αναφορά **μην συμπεριλάβετε κώδικα**, εκτός εξαιρετικών περιπτώσεων όπου σε λίγες γραμμές θέλετε να αναδείξετε κάποια συγκεκριμένη πτυχή της υλοποίησής σας.

6 Κριτήρια Βαθμολόγησης

Για τη βαθμολογία της υλοποίησης του συστήματος θα χρησιμοποιηθούν τα ακόλουθα κριτήρια.

Κριτήριο	Ποσοστό Βαθμολογίας Υλοποίησης
Μέρος Α: Κατασκευή Βιβλιοθήκης "Trivia" API	
Λειτουργικότητα	25%
A1 - Επιτυχής ανάκτηση λίστας ερωτήσεων	10%
A2 - Χρήση παραμέτρων για τη λήψη της λίστας ερωτήσεων	10%
A3 - Σωστή εμφάνιση πληροφοριών για κάθε ερώτηση (ερώτηση, κατηγορία, τύπος, απαντήσεις κτλ.)	5%
Υλοποίηση	20%
A4 - Σωστή διαχείριση επικοινωνίας με το REST API (κλήση API, λήψη δεδομένων σε JSON/XML, αποσειριοποίηση)	10%
A5 - Σωστή υλοποίηση POJO κλάσεων για την αποθήκευση των δεδομένων	10%
Μέρος Β: Κατασκευή Εφαρμογής "Παιγνίδι Γνώσεων"	
Λειτουργικότητα	30%
B1 - Παραμετροποίηση του παιχνιδιού και παροχή επιλογών για ανάκτηση λίστας των ερωτήσεων	10%
B2 - Σωστή εμφάνιση των ερωτήσεων, απαντήσεων και αποτελεσμάτων βαθμολογίας	10%
B3 - Φιλικότητα προς το χρήστη (εμφάνιση μηνυμάτων, λίστας επιλογών)	5%
B4 - Σταθερότητα συστήματος (χειρισμός εξαιρέσεων)	5%
Υλοποίηση	25%
B5 - Χρήση maven ή gradle για τη διαχείριση του project	5%
B6 - Επαρκής σχολιασμός του πηγαίου κώδικα	5%
B7 - Unit tests με Junit για τον έλεγχο των λειτουργιών	10%
B8 - Διαχείριση εκδόσεων κώδικα με GIT και αποθήκευση στο Github	5%

ΣΗΜΕΙΩΣΗ 1: Για την τελική βαθμολογία της εργασίας θα ληφθεί επίσης υπόψη η αναφορά.

ΣΗΜΕΙΩΣΗ 2: Ο τελικός βαθμός του μαθήματος θα υπολογιστεί όπως ανακοινώθηκε στην 1η διάλεξη του μαθήματος και έχει αναρτηθεί στην ιστοσελίδα του μαθήματος στο e-class («Λεύκιππος»).

Καλή Επιτυχία!