



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Διαδίκτυο και Εφαρμογές Εαρινό Εξάμηνο 2021-2022

Επαναληπτική Εξεταστική

Θέμα 1ο (5 μονάδες)

Αξιοποιώντας τις δυνατότητες του Java AWT και ειδικά της Java Swing για την κατασκευή GUIs, καλείστε να αναπτύξετε ένα window-based application. Ο γενικός στόχος της εφαρμογής είναι η συνεχής σχεδίαση σε έναν καμβά νέων δίσκων με αυξανόμενη ακτίνα. Θα πρέπει να δίνεται η δυνατότητα να καθαρίζει ο καμβάς, να αποθηκεύεται η κατάσταση του σε ένα αρχείο, και να αλλάζει η ακτίνα όλων των δίσκων με σχετική είσοδο από τον χρήστη. Πιο αναλυτικά, η εφαρμογή πρέπει να έχει τις εξής ιδιότητες:

1. **Βασική μορφή παραθύρου εφαρμογής (1 μονάδα):**

- 1.1. Δημιουργήστε ένα frame διαστάσεων 750px x 750px με τίτλο "Inflating Disks".
- 1.2. Στο πάνω τμήμα του frame, προσθέστε ένα panel το οποίο θα έχει ένα Button που θα αναγράφει "Set Radius" και ένα Drop-down List με 4 τιμές (5, 10, 15, 20).
- 1.3. Στο μεσαίο τμήμα του frame, προσθέστε ένα canvas (με default διαστάσεις).
- 1.4. Στο κάτω τμήμα του frame, προσθέστε ένα Button που θα αναγράφει "Save Canvas" και ένα δεύτερο Button που θα αναγράφει "Clear Canvas".

2. **Παθητική λειτουργία παραθύρου εφαρμογής (2 μονάδες):**

- 2.1. Αρχικά, με το που τρέχετε το πρόγραμμα, θα εμφανίζεται το παράθυρο με τον καμβά να έχει σχεδιασμένο έναν δίσκο με ακτίνα 15px σε τυχαίο σημείο του.
- 2.2. Την ίδια στιγμή θα πρέπει κάθε 4 δευτερόλεπτα α) να εμφανίζεται ένας νέος δίσκος στον καμβά σε τυχαίο σημείο, β) να αυξάνεται η ακτίνα όλων των δίσκων κατά 10px (όλοι οι δίσκοι θα πρέπει να έχουν την ίδια ακτίνα), και γ) να εμφανίζεται στην κονσόλα το περιεχόμενο του αρχείου με όνομα "rocknroll" (βλ. σημείο 3.3) με μορφοποίηση φιλική προς τον χρήστη.

3. **Ενεργητική λειτουργία παραθύρου εφαρμογής (2 μονάδες):**

- 3.1. Το πλαίσιο/παράθυρο θα πρέπει να είναι resizable. Με το που κλείνετε το παράθυρο, η εφαρμογή θα πρέπει να σταματά.
- 3.2. Όταν πατάτε το κουμπί "Set Radius", το περιεχόμενο του Drop-down List δίπλα του θα πρέπει να χρησιμοποιείται για να αλλάξει η ακτίνα όλων των δίσκων στον καμβά (με το επόμενο repaint του). Στην συνέχεια, το πρόγραμμα θα συνεχίσει να λειτουργεί κανονικά, με την ακτίνα των δίσκων να ξεκινά να αυξάνεται και πάλι. Για την ενημέρωση του χρήστη, θα πρέπει να εμφανίζεται Warning με τίτλο "Radius Update" και περιεχόμενο "Successfully Updated Radius".
- 3.3. Όταν πατάτε το κουμπί "Save Canvas", θα πρέπει να αποθηκεύεται σε ένα αρχείο με όνομα "rocknroll" η κατάσταση του καμβά (λίστα objects, με κάθε object-δίσκο να έχει τις τιμές για την θέση του (x,y) στον καμβά, και την ακτίνα του).
- 3.4. Όταν πατάτε το κουμπί "Clear Canvas" θα πρέπει να καθαρίζει ο καμβάς (να ξαναρχικοποιείται και να εμφανίζει μόνο έναν δίσκο με ακτίνα 15px) και να συνεχίζει η ροή του προγράμματος κανονικά.

Θέμα 2ο (5 μονάδες)

Στα πλαίσια του δεύτερου θέματος θα θέλαμε να αναπτύξετε μια Web εφαρμογή για την καταγραφή των αποτελεσμάτων των αγώνων μπάσκετ. Η βάση για την αποθήκευση των δεδομένων σας δίνεται (αρχείο: [league_db.sql](#)).

Για τα ερωτήματα (1) και (2) θα πρέπει να συμπληρώσετε την υλοποίηση των δύο μεθόδων που υπάρχουν στο αρχείο «[ExamDbConnector.java](#)». Ακολουθώντας, μπορείτε να ελέγξετε την ορθότητά τους, χρησιμοποιώντας τον κώδικα που σας δίνεται (main μέθοδος της κλάσης αυτής).

1. Να υλοποιήσετε την μέθοδο «**examineUser**» που εξετάζει εάν τα στοιχεία του χρήστη (username και password-hash) είναι έγκυρα ή όχι με βάση τα δεδομένα που υπάρχουν στον αντίστοιχο πίνακα της βάσης.

```
public boolean examineUser(String username, String passwordHash) throws Exception;  
SQL Query: SELECT * FROM USERS WHERE USERNAME = ? and PASSWORDHASH = ?
```

2. Να υλοποιήσετε την μέθοδο «**recordData**» για την καταγραφή του τελικού αποτελέσματος ενός αγώνα στη βάση, συμπεριλαμβανομένου της ημερομηνίας που έγινε ο αγώνας (java.sql.Date), των δύο ομάδων (unique IDs) που συμμετείχαν σε αυτόν και τους πόντους που πέτυχαν κάθε μιας από αυτές.

```
public int recordData(Date date, int team1id, int team1score, int team2id, int team2score) throws Exception;  
SQL Query: INSERT INTO `game_data` VALUES (null, ?, ?, ?, ?, ?);
```

Για τα ερωτήματα (3) και (4) σας δίνεται η HTML σελίδα «[InsertData.html](#)» μέσω της οποίας μπορούμε να δώσουμε τα δεδομένα (ημερομηνία, ομάδες και πόντους που πέτυχαν) που θέλουμε να καταγράψουμε στην βάση. Ειδικά για το (3) ερώτημα μπορείτε να συμπληρώσετε το αρχείο αυτό.

3. Να υλοποιήσετε σε **JavaScript** κώδικα μια μέθοδο που ελέγχει όταν ο χρήστης «πατάει» το «**check**» button εάν τα στοιχεία που έχει δώσει ο χρήστης είναι σωστά ή όχι. Ειδικότερα θα θέλαμε να ελέγχει ότι οι δύο ομάδες που έχει ορίσει στα αντίστοιχα πεδία της φόρμας είναι διαφορετικές και να εμφανίζει κατάλληλο μήνυμα στην οθόνη.

4. Να φτιάξετε ένα Java **Servlet** για την εισαγωγή του αποτελέσματος ενός αγώνα στη βάση. Μετά την επιτυχή εισαγωγή των δεδομένων, θα πρέπει να εμφανίζονται στη σελίδα όλοι οι αγώνες που έχουν καταγραφεί στη βάση.

```
SQL Query: SELECT g.game_date, t1.name, g.team_1_score, t2.name, g.team_2_score FROM  
game_data g, teams t1, teams t2 WHERE g.team_1_id = t1.id and g.team_2_id = t2.id
```

Για τα ερωτήματα (5) σας δίνεται η Java κλάση «[Util.java](#)» που περιέχει την μέθοδο που δημιουργεί τον hash ενός string χρησιμοποιώντας τον SHA-256 αλγόριθμο καθώς και η «[Login.html](#)» σελίδα.

5. Να κάνετε τις απαραίτητες αλλαγές στον παραπάνω κώδικα έτσι ώστε η εισαγωγή των δεδομένων στην βάση να επιτρέπεται μόνο σε εγγεγραμμένους χρήστες.