



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

## ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

*Διαδίκτυο και Εφαρμογές Εαρινό Εξάμηνο 2021-2022*

### **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 (0.5 μονάδες):**

- 1.1. Δημιουργήστε ένα frame διαστάσεων 700px x 700px με τίτλο “Growing Disks”.
- 1.2. Στο πάνω τμήμα του frame, προσθέστε ένα panel το οποίο θα έχει ένα Button που θα αναγράφει “Set Radius” και ένα TextField με default τιμή “10” σε Fonts Times New Roman, Italic, 19.
- 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 10px σε τυχαίο σημείο του.
- 2.2. Την ίδια στιγμή θα πρέπει να εκκινεί και η εκτέλεση ενός thread (Thread 1) και να εμφανίζεται στην κονσόλα το μήνυμα “Starting hello thread...”. Μετά από 5 δευτερόλεπτα, θα εμφανίζεται και πάλι στην κονσόλα το μήνυμα “Hello”, οπότε και θα ολοκληρώνεται το thread.
- 2.3. **Αφού** ολοκληρωθεί το Thread 1, με την βοήθεια άλλων threads θα πρέπει κάθε 2 δευτερόλεπτα α) να εμφανίζεται ένας νέος δίσκος στον καμβά σε τυχαίο σημείο, β) να αυξάνεται η ακτίνα όλων των δίσκων κατά 2px (όλοι οι δίσκοι θα πρέπει να έχουν την ίδια ακτίνα), και γ) να εμφανίζεται στην κονσόλα το περιεχόμενο του αρχείου με όνομα “diskdata” (βλ. σημείο 3.3) με μορφοποίηση φιλική προς τον χρήστη.
- 2.4. Κάθε 5 δευτερόλεπτα, το background του button “Clear” θα πρέπει να αλλάζει χρώμα με την χρήση ενός Timer. Τα χρώματα που μπορεί να έχει το button είναι RED ή GREEN.

#### **3. Ενεργητική λειτουργία παραθύρου εφαρμογής (2.5 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 TextField δίπλα του θα πρέπει να χρησιμοποιείται για να αλλάζει 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 “diskdata” 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” ενώ είναι πράσινο, θα πρέπει να καθαρίζει ο καμβάς (να ξαναρχικοποιείται και να εμφανίζει μόνο έναν δίσκο με ακτίνα 10px) και να συνεχίζει η ροή του προγράμματος κανονικά. Όταν πατάτε το κουμπί “Clear Canvas” ενώ είναι κόκκινο, θα πρέπει να εμφανίζεται Warning με τίτλο “Warning” και περιεχόμενο “Clear button is red!”, χωρίς να αλλάζει τίποτα στον καμβά.

(\*) Αν δεν μπορείτε να ολοκληρώσετε κάποιο βήμα από το οποίο εξαρτάται κάποιο επόμενο (π.χ. βήμα 2.4 και 3.4), μπορείτε να «φυτέψετε» τις αντίστοιχες τιμές ή να προσπεράσετε το βήμα όπως θεωρείτε, ώστε να μπορέσετε να δείξετε τουλάχιστον την λειτουργικότητα του επόμενου βήματος.

## Θέμα 2ο (5 μονάδες)

Ο Κορονοϊός (COVID) εξακολουθεί να απασχολεί την κοινωνία μας. Συνεπώς η καταγραφή και παρακολούθηση των COVID ελέγχων που «βγήκαν» θετικοί είναι σημαντική. Στα πλαίσια της άσκησης αυτής θα πρέπει να αναπτύξετε μια διαδικτυακή εφαρμογή η οποία επιτρέπει σε εγγεγραμμένους χρήστες να έχουν πρόσβαση σε συγκεκριμένες υπηρεσίες/σελίδες, ανάλογα με τον ρόλο που έχουν στο σύστημα. Ειδικότερα:

### 1. Password Update (2 μονάδες):

Ένας εγγεγραμμένος χρήστης θα πρέπει να έχει την **δυνατότητα να αλλάξει τον κωδικό** του. Στην περίπτωση αυτή θα πρέπει να δώσει το username, τον υπάρχον κωδικό αλλά και τον νέο έτσι ώστε η εφαρμογή να ελέγξει εγκυρότητά του χρήστη και έπειτα να ανανεώσει τον κωδικό του (για την ακρίβεια, το hash του κωδικού). Καθώς ο χρήστης πληκτρολογεί τον νέο κωδικό (\*), η εφαρμογή θα πρέπει να του εμφανίζει στην οθόνη του (π.χ., σε κάποιο διπλανό πεδίο) εάν ο κωδικός είναι *εύκολος* (έχει μόνο γράμματα), *μέτριος* (έχει επιπρόσθετα και αριθμούς) ή *δύσκολος* (έχει επιπρόσθετα και κάποιο σημείο στίξης) (\*\*).

```
SELECT * FROM USERS WHERE USERNAME = ? and PASSWORD_HASH = ?  
UPDATE USERS SET PASSWORD_HASH = ? WHERE USERNAME = ?
```

(\*) Εναλλακτικά, ο έλεγχος θα μπορούσε να γίνεται «trigger» με κάποιο «button».

(\*\*) Για τον έλεγχο, μπορείτε να χρησιμοποιήσετε την JavaScript μέθοδο: `REG.test(STR)`, όπου REG π.χ. `/[a-zA-Z]/g`

### 2. Administrator Web Page (1 μονάδα):

Ο διαχειριστής του συστήματος (ROLE\_ID = 2) θα πρέπει να έχει πρόσβαση σε μία σελίδα στην οποία θα παρουσιάζονται (με την μορφή ενός πίνακα) ο **συνολικό αριθμό ελέγχων που έγιναν ανά κατηγορία**.

```
SELECT dm.NAME, count(*) FROM CASES c, DIAGNOSIS_METHODS dm  
WHERE c.DIAGNOSIS_METHOD_ID = dm.ID GROUP BY dm.NAME
```

### 3. User Web Pages (2 μονάδες):

Εάν ο χρήστης δεν είναι διαχειριστής του συστήματος (ROLE\_ID = 1) θα πρέπει να έχει πρόσβαση σε μία σελίδα μέσω της οποίας θα μπορεί να κάνει τα παρακάτω (\*\*):

(α) **Να δει τους ελέγχους COVID** που έχουν καταγραφεί για τον συγκεκριμένο χρήστη με την μορφή ενός πίνακα.

```
SELECT * FROM CASES c, DIAGNOSIS_METHODS dm  
WHERE c.DIAGNOSIS_METHOD_ID = dm.ID AND c.USER_ID = ?
```

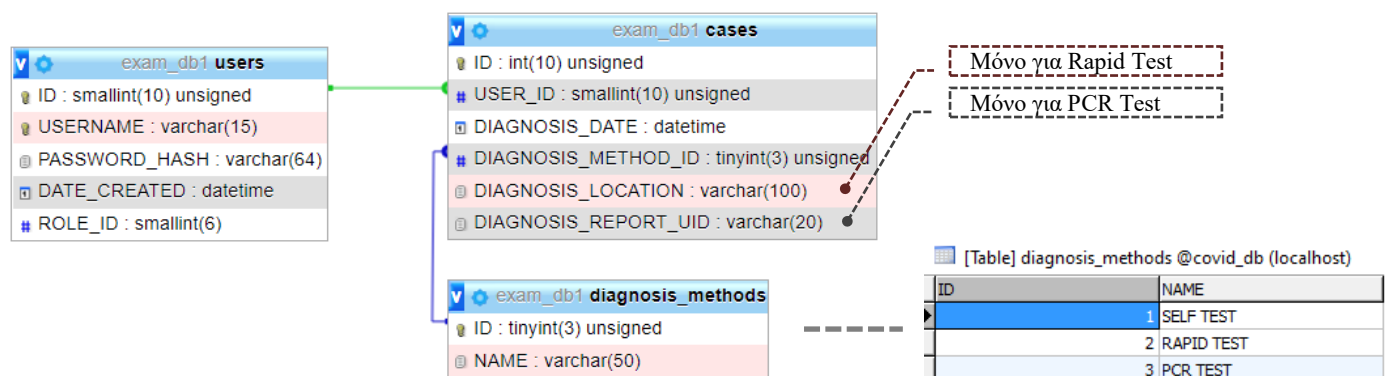
(β) **Να καταχωρήσει το αποτέλεσμα ενός νέου ελέγχου** παρέχοντας τα κατάλληλα δεδομένα, ανάλογα με τον τρόπο που έγινε η εξέταση (SELF/RAPID/PCR test). Ειδικότερα, στην περίπτωση ενός RAPID test θα πρέπει να αναφέρει την περιοχή που έγινε ο έλεγχος, ενώ στην περίπτωση ενός PCR test τον κωδικό του σχετικού εγγράφου.

```
INSERT INTO CASES VALUES (null, ?, ?, ?, ?, ?) -- check DB schema
```

(\*\*) Για καθένα από τα παραπάνω θα πρέπει το σύστημα να εξετάζει εάν ο χρήστης είχε κάνει σε προηγούμενο βήμα «successful login» με χρήση Session/Cookies και να εμφανίζει στο πάνω μέρος της οθόνης το USERNAME του χρήστη.

## Σχεσιακή Βάση και Κώδικας:

Για τον σκοπό αυτό έχουμε ήδη αναπτύξει μια σχεσιακή βάση (το SQL αρχείο βρίσκεται στον φάκελο DB) για την αποθήκευση των δεδομένων, στην οποία υπάρχουν ήδη ορισμένοι εγγεγραμμένοι χρήστες (username και password είναι **ίδια**) και έλεγχοι.



Στον φάκελο CODE υπάρχει η Java Class (Util.java) με την οποία μπορούμε να δημιουργήσουμε το Hash ενός String με τον αλγόριθμο SHA-256 καθώς και η Login σελίδα της εφαρμογής για το δεύτερο ερώτημα (Login.html)