

ΠΡΟΗΓΜΕΝΑ ΘΕΜΑΤΑ ΑΝΤΙΚΕΙΜΕΝΟΣΤΕΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

ΠΜΣ «Προηγμένα Συστήματα Πληροφορικής - Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

Μέγιστη Βαθμολογία: 6 μονάδες

Ομαδική/ατομική εργασία: Για την εργασία αυτή μπορείτε αν δουλέψετε είτε ατομικά, είτε σε ομάδες 2 ατόμων. Η δημιουργία των ομάδων γίνεται από εσάς

Αριθμός θεμάτων: 2. Είστε ελεύθεροι να επιλέξετε 1 από τα 2 διαθέσιμα θέματα.

Θέμα 1: «Unipi World Cup 2022 Stats»

Περιγραφή: Στην παρούσα εργασία θα πρέπει να υλοποιήσετε μια εφαρμογή Java η οποία έχει ως στόχο την υλοποίηση της εφαρμογής WeatherApp, η οποία επεξεργάζεται δεδομένα καιρικών συνθηκών που λαμβάνει από την υπηρεσία ιστού (web service) της ιστοσελίδας wttr.in: <https://wttr.in/>

Λογική: Η περιγραφή της υπηρεσίας βρίσκεται στο url: <https://github.com/chubin/wttr.in?#readme>

Η wttr.in παρέχει τις εν λόγω διαδικτυακές υπηρεσίες μέσω web services τύπου REST (Representational State Transfer), προκειμένου να χρησιμοποιηθούν από οποιοδήποτε ενδιαφερόμενο μέρος. Τα σχετικά δεδομένα επιστρέφονται σε διάφορες μορφές, οι οποίες συμπεριλαμβάνουν σύντομο κείμενο και εικόνες, ωστόσο εμάς μας ενδιαφέρει η μορφή JSON (JavaScript Object Notation), η οποία και υποστηρίζεται.

Η εν λόγω υπηρεσία μας παρέχει αρκετά μεγάλη πληροφόρηση σχετικά με δεδομένα καιρού (ηλιοφάνεια, βροχή, θερμοκρασία, υγρασία, ταχύτητα ανέμου, κ.λπ.), καθώς επίσης εκτός από την πληροφόρηση για τα τρέχοντα δεδομένα καιρού, μας δίνει και δεδομένα πρόβλεψης για έως και τις 2 επόμενες ημέρες. Επιπλέον, εκτός από πόλεις, η εν λόγω υπηρεσία μπορεί να μας επιστρέψει δεδομένα βάσει τοποθεσίας, κωδικού περιοχής, γεωγραφικών συντεταγμένων, κωδικού αεροδρομίου, κ.α.

Στην παρούσα εργασία μας ενδιαφέρουν μόνο η αναζήτηση δεδομένων καιρικών συνθηκών ανά όνομα πόλης. Στην αναζήτηση καιρικών δεδομένων πόλης, εφόσον η πόλη υπάρχει, μας ενδιαφέρει η εφαρμογή να αναγράψει τις εξής πληροφορίες/πεδία (και όχι όλες):

- Θερμοκρασία (temp_C)
- Υγρασία (humidity)
- Ταχύτητα ανέμου (windspeedKmph)
- Δείκτης UV (uvIndex)
- Λεκτική πρόγνωση καιρού (weatherDesc)

Κάθε αναζήτηση αποτελεσμάτων από το χρήστη θα αποθηκεύεται σε ΒΔ της επιλογής σας, με όποιο τρόπο αποθήκευσης θέλετε εσείς (οι πίνακες και οι μεταξύ τους σχέσεις είναι δικό σας θέμα). Για τη διευκόλυνσή σας, προτείνω τη χρήση μιας τοπικής ΒΔ SQLite/Derby/H2 κ.λπ.

Σε κάθε εγγραφή στη ΒΔ θα καταγράφετε εκτός από τα δεδομένα της αναζήτησης του χρήστη, και ένα timestamp, του πότε έγινε η εν λόγω αναζήτηση.

Τονίζεται ότι δεν μας ενδιαφέρει να «κατεβάσουμε» όλα τα δεδομένα που έχει το εν λόγω web service, παρά μόνο αυτά που αναζητά ο εκάστοτε χρήστης της εφαρμογής. Αντίστοιχα, θα παράγονται στατιστικά από όσα δεδομένα έχουν αποθηκευτεί στη ΒΔ.

Θέμα 2: «Automatic Code Injection for Database Apps using Annotations and Reflection»

Περιγραφή: Στην παρούσα εργασία θα πρέπει να υλοποιήσετε μια εφαρμογή Java η οποία θα λειτουργεί με βάση τη λογική ενός απλού ORM Framework. Συγκεκριμένα, η εφαρμογή σας θα δέχεται ως input μια κλάση εμπλουτισμένη με annotations με στόχο την αυτοματοποίηση της χρήσης κώδικα για τη σύνδεση της εφαρμογής με μια ΒΔ και ειδικότερα, την ταύτιση ενός πίνακα της ΒΔ με την εν λόγω δοθείσα κλάση.

Οι πληροφορίες που θα δίδονται ως annotations μέσα στην κλάση στόχο θα είναι οι εξής:

- Ο τύπος της ΒΔ (διαθέσιμοι τύποι Derby, SQLite, H2)
- Το όνομα της ΒΔ (πχ. MyDB)
- Το όνομα του πίνακα ο οποίος θα αντιστοιχηθεί στην κλάση στόχο
- Τα πεδία της κλάσης μαζί με τους τύπους τους, τα οποία θα αποτελέσουν αντίστοιχα πεδία του πίνακα μέσα στην ΒΔ
- Το πεδίο το οποίο θα είναι πρωτεύον κλειδί
- Προαιρετικά πληροφορίες πεδίων όπως not null, unique

Επιπλέον μέσα στην κλάση θα υπάρχουν και μέθοδοι ανάκτησης των υπαρχόντων εγγραφών από τον δοθέντα πίνακα (Select *), καθώς και διαγραφή συγκεκριμένης εγγραφής από τον πίνακα βάση ενός κλειδιού (Delete by id). Τόσο η μέθοδος ανάκτησης δεδομένων, όσο και η μέθοδος διαγραφής, θα δηλώνονται από αντίστοιχο annotation.

Ακολουθεί ένα ενδεικτικό input κλάσης (δεν χρειάζεται να είναι έτσι η δική σας υλοποίηση)

```
import java.util.List;
@Database(name="UnipiDB", dbtype="Derby")
@Table(name="Student")
public class Student {
    @PrimaryKey
    @Field(name="AM", type="Text")
    String AM;
    @Field(name="Email", type="Text")
    String email;
    @Field(name="Year", type="Integer")
    int yearOfStudies;
    @Field(name="FullName", type="Text")
    String fullName;
    @Field(name="PostGraduate", type="Boolean")
    boolean postGraduate;
    //Για τη μέθοδο αυτή μπορείτε να δοκιμάσετε να επιστρέψετε
    List<Student>
    @DBMethod(type="SelectAll")
    public List<String> getAllStudents() {
        return null;
    }
    //Ο επιστρεφόμενος ακέραιος υποδηλώνει τον αριθμό των εγγραφών
    που διαγράφηκαν
    @DBMethod(type="DeleteOne")
    public int deleteStudents(String AM) {
        return 0;
    }
}
```

Προσδοκώμενο αποτέλεσμα: Ένα αρχείο .java (σε άλλον φάκελο) με την κλάση, όπου θα έχει συμπληρωθεί από το πρόγραμμά σας όλος ο απαραίτητος κώδικας ώστε να είναι έτοιμη η κλάση προς εκτέλεση και να υποστηρίζει πλήρως τις δοθείσες λειτουργίες με τη ΒΔ από τα δηλωθέντα annotations.

Σημαντικά:

- Στην εφαρμογή σας καλό θα ήταν να γίνει χρήση κάποιου design pattern κατά την υλοποίηση των όποιων κλάσεων ή/και διεπαφών, στην αρχιτεκτονική σας
- Στην εφαρμογή σας καλό θα ήταν να γίνει χρήση του Stream API για την παραγωγή κάποιων αποτελεσμάτων, εφόσον αυτό είναι εφικτό
- Θέλουμε να αποφύγετε τη χρήση βιβλιοθηκών τρίτων και να βασιστείτε στα εργαλεία που σας παρέχει η Java. Εξαίρεση αποτελεί η ανάγνωση/μεταχείριση δεδομένων JSON, εφόσον τα χρειάζεστε, όπου μπορείτε να χρησιμοποιήσετε μια από τις γνωστές λύσεις (π.χ. Gson, Jackson, Json-simple).

Παραδοτέο :

Στον ειδικό χώρο αποστολής εργασιών σας στο e-class, «Εργασίες».

Θα πρέπει να αποστείλετε ένα .zip αρχείο (ένας φοιτητής ανά ομάδα) που θα περιέχει:

(1.) Το Project σας έτοιμο να τρέξει,

(2.) Αναλυτικά σχόλια (comments) στον κώδικά σας,

(3.) Σύντομο video επίδειξης της εφαρμογής σας ενώ τρέχει, με χρήση από εσάς (μπορείτε να χρησιμοποιήσετε ένα πρόγραμμα screen-recording της επιλογής σας).

Τελικές οδηγίες:

Ακόμα και αν δεν μπορείτε να υλοποιήσετε ό,τι αναφέρεται στην εργασία, μπορείτε να υλοποιήσετε μέρος της εργασίας. Μην απογοητευτείτε αν «κολλήσετε» σε κάποιο τμήμα. Ακόμα και μερικώς υλοποιημένη εφαρμογή μπορεί να πάρει καλό βαθμό.

Μπορείτε να κάνετε όσες επιπλέον παραδοχές θέλετε και να εμπλουτίσετε (σε λογικά πλαίσια) την εφαρμογή σας, εφόσον όμως καλύψετε πρώτα όλες τις προδιαγραφές που αναφέρονται στην παρούσα εκφώνηση.

Καλή επιτυχία!

Ε. Αλέπης