

## Tema 1 POO (2022-2023)

### Generator de chestionare

*Responsabili: Laur Neagu, Andreea Duțulescu*

#### Introducere

Scopul acestei teme este de a vă familiariza cu programarea orientată pe obiecte și cu limbajul Java. În același timp, puteți aprofunda cunoștințele cu privire la structuri de date și aveți șansa de a implementa un prim proiect de dezvoltare software similar industriei.

Scheletul de cod furnizat include și un set de teste publice prin care puteți să vedeți, în timp real, procentul de acoperire al soluției voastre raportat la cerințele temei. Vă recomandăm să studiați testele publice furnizate pentru a vedea formatul așteptat al procesării din vectorul args[].

Totodată, nu modificați numele fișierului sursă unde este metoda main(), folosiți numele furnizat în schelet – Tema1.java!

Vă puteți instala și configura în paralel Gradle<sup>1</sup>, și rula aceste teste pe mediile voastre local (gradle test). Totodată, la încărcarea codului pe GitHub Classroom (git push), aceste teste vor rula automat și veți vedea punctajul. Important de menționat că aceste teste publice sunt doar un set parțial de teste redactate pentru această temă. Tema include și un set de teste private ce se vor rula ulterior.

#### Descrierea problemei

Tema își propune implementarea unui Generator simplu de chestionare. Utilizatorii acestui program se vor autentifica în aplicație la orice apel din sistem (pentru simplificare), în afară de acela de creare de utilizator. Utilizatorii autentificați în sistem vor putea: **crea întrebări** (cu un singur răspuns corect, sau cu mai multe răspunsuri corecte), **crea chestionare** în baza întrebărilor adăugate anterior și vor putea **răspunde chestionarelor celorlalți**, doar o singură dată.

Orice întrebare este caracterizată de un identificator, un text și o listă de răspunsuri posibile. Un răspuns este caracterizat de un identificator, un text, și valoarea de adevăr a acestuia (False/True).

Exemplu întrebare cu un singur răspuns corect:

*Cerul este albastru ( [1] Adevărat / [2] Fals)*

În contextul chestionarelor, întrebările sunt punctate în stil francez. Dacă răspunsul ales este cel corect, întrebarea este punctată cu 1, dacă nici un răspuns nu este oferit este punctată cu 0 Dacă în schimb răspunsul greșit este ales, acea întrebare este punctată negativ, cu valoarea -1.

Exemplu întrebare cu mai multe răspunsuri corecte:

*Temperatura se poate măsura în ( [1] Celsius / [2] Litri / [3] Fahrenheit / [4] Grame / [5] Kelvin)*

---

<sup>1</sup> <https://gradle.org/>

Fiecare răspuns corect selectat adaugă o pondere pozitivă la această întrebare, pe când orice răspuns greșit ales scade punctajul întrebării.

Așa cum ați învățat în școală, răspunsurile corecte sunt Celsius, Kelvin și Fahrenheit. Din cele 5 răspunsuri posibile, 3 dintre acestea sunt corecte. Așadar, fiecare răspuns corect are pondere de  $\frac{1}{3}$  (ceea ce reprezintă 0.333). 2 răspunsuri (în acest caz, Litri și Grame) sunt greșite, așadar ponderea fiecăruia este de  $\frac{1}{2}$  (ceea ce reprezintă 0.5).

Să luăm câteva exemple de răspunsuri pentru această întrebare:

- Dacă utilizatorul selectează Celsius + Kelvin, întrebarea va fi punctată cu  $\frac{2}{3} = 0.66$  puncte.
- Dacă utilizatorul selectează Celsius + Fahrenheit + Kelvin, întrebarea va fi punctată cu 1 punct.
- Dacă utilizatorul alege Litri + Fahrenheit + Kelvin, întrebarea va fi punctată cu  $0.333 + 0.333 - 0.5 = 0.166$  puncte (echivalentul 16% dintr-un punct)
- Dacă utilizatorul alege toate variantele Celsius + Litri + Fahrenheit + Grame + Kelvin =  $0.333 - 0.5 + 0.333 - 0.5 + 0.333 = 0$  puncte
- Dacă sunt selectate opțiunile Litri + Grame =  $-0.5 - 0.5 = -1$  punct.

Pentru a crea un chestionar, este necesar ca întrebările să fie adăugate în prealabil. Orice utilizator poate crea întrebări, și mai departe poate crea chestionare.

#### **Reguli generale ale generatorului de chestionare:**

- Nu pot exista două întrebări cu același text.
- Nu pot exista două chestionare cu același nume.
- Utilizatorii pot folosi în crearea chestionarelor și întrebări create de alți utilizatori.
- Un chestionar poate avea între 1 și 10 întrebări.
- În cadrul unui chestionar, punctajul maxim care se poate obține este 100 puncte, și fiecare întrebare are o pondere egală, rotunjită la două zecimale (exemplu: 10 întrebări – fiecare întrebare este punctată cu 10% din punctajul chestionarului, sau 7 întrebări – fiecare întrebare este punctată cu 14.28%).
- Pentru răspunsuri greșite la întrebări (scor negativ per total), se va scădea în același mod, proporțional (exemplu, dacă la chestionarul cu 7 întrebări, una dintre ele primește punctaj -0.5, atunci ponderea întrebării în chestionar va fi de -7.14%).
- Mai departe, procentul total rezultat al chestionarului se va rotunji la număr întreg (exemplu: dacă procentul obținut este 28.42%, se va întoarce 28, iar dacă este spre exemplu 35.67% se va întoarce 36).
- În cazul în care suma punctajelor întrebărilor dintr-un chestionar rezultă într-o valoare negativă, chestionarul va fi punctat la valoarea 0.
- O întrebare poate avea între 1 și 5 răspunsuri posibile.
- Nu ar trebui permis la crearea unei întrebări cu un singur răspuns corect, să se trimită mai multe răspunsuri corecte.
- Utilizatorii ar trebui să își poată vedea scorul pe fiecare chestionar rezolvat.

- Chestionarele pot fi șterse doar de către cei care le-au creat.
- Identificatorii pentru fiecare obiect (proprietățile ID) se vor seta incremental la fiecare instanțiere, folosind câmpuri statice, precum ați învățat la laborator.
- Pentru persistența datelor (utilizatori, întrebări, chestionare, punctaje per chestionar completat) se vor folosi operații cu fișiere. Lucrul cu fișierele este descris la finalul acestui document.

Tema va fi implementată sub forma unei aplicații Java de tip consolă, și va accepta o serie de argumente în linie de comandă ce vor trebui interpretate.

Următoarele comenzi vor trebui procesate în tema voastră. Documentația completă a acestor comenzi, cu răspunsurile așteptate în funcție de apel, este disponibilă în documentul Tema1-DocumentațieComenzi.pdf.

1. Creare utilizator
  2. Creare întrebare
  3. Întoarce identificator întrebare după nume
  4. Întoarce toate întrebările din system
  5. Creare chestionar
  6. Întoarce identificator chestionar după nume
  7. Întoarce toate chestionarele din sistem
  8. Întoarce detaliile unui chestionar în funcție de identificator (fără valoarea de adevăr a răspunsurilor)
  9. Încarcă răspunsuri chestionar
  10. Șterge chestionar
  11. Întoarce soluțiile mele
  12. Curăță date din aplicație
- \* Acest apel va fi folosit pentru a curăța datele istorice (utilizatori, întrebări, chestionare, soluții) existente în sistem. În practică, va trebui să curățați fișierele unde veți salva aceste date.*

Pentru citirea din fișier, regăsiți mai jos un snippet de cod prin care puteți face asta:

```
try (BufferedReader br = new BufferedReader(new FileReader("myfile.txt"))) {
    String line;
    while ((line = br.readLine()) != null) {
        // process the line
    }
}
```

Pentru scriere (mod append) în fișier, regăsiți mai jos un snippet de cod prin care puteți face asta:

```
try (FileWriter fw = new FileWriter("myfile.txt", true);
    BufferedWriter bw = new BufferedWriter(fw);
    PrintWriter out = new PrintWriter(bw)) {

    out.println("the text");
    //more code
    out.println("more text");
}
```

```
//more code
} catch (IOException e) {
    //exception handling left as an exercise for the reader
}
```

Vă recomandăm să stocați persistent informația pentru creare utilizatori, creare întrebări, creare chestionare, încărcare soluții, în fișiere de tip text separate (se poate folosi formatul CSV – Comma Separated Value<sup>2</sup>).

Spre exemplu, pentru utilizatori, puteți crea un fișier users.csv, unde să aveți pe fiecare linie informația în următorul format:

*Username,Password*

*Test1,test1234!*

*Test2,test2222*

În mod similar, puteți stoca informația și pentru celelalte date necesare.

#### **PUNCTAJ:**

DOMENIU	TASK	PUNCTAJ	TOTAL
Test	<i>Suită de teste rulată automat</i>	80 p	80 PUNCTE
Cod	<i>Folosirea conceptelor de POO studiate</i> <i>Folosirea structurilor de date studiate</i> <i>Lizibilitate cod, comentarii (acolo unde e cazul)</i>	5 p 5 p 5 p	15 PUNCTE
README	<i>Redactarea unui fișier Readme în care să se detalieze implementarea temei (200-500 cuvinte)</i>	5 p	5 PUNCTE

#### **BONUS:**

*Ce alte cazuri limită ați mai trata în această aplicație (minim 3 cazuri, a se descrie în README)?*

*Cum ați refactoriza comenzile și răspunsurile din aplicație (minim 3 propuneri, a se descrie în README)?*

– 5 PUNCTE

**Notițele autorilor:** Chiar dacă răspunsurile primite din fiecare comandă sunt formate precum JSONs<sup>3</sup> în tabelele anterioare, ele vor fi trimise de către voi, de fapt, ca string-uri, formate în stil JSON (exemplu: '{ "status" : "error", "message" : "You need to be authenticated"}').

Totodată, pentru a vă maximiza punctajul în cadrul acestei teme, încercați să acoperiți cât mai multe dintre cazurile întoarse de fiecare comandă, și din lanțuri de comenzi succesive (exemplu create-user -> create-user -> create-question -> create-question -> get-question-id-by-text -> get-question-id-by-text -> create-quizz -> submit-quizz -> get-my-solutions -> delete-quizz).

<sup>2</sup> [https://en.wikipedia.org/wiki/Comma-separated\\_values](https://en.wikipedia.org/wiki/Comma-separated_values)

<sup>3</sup> <https://www.json.org/json-en.html>