Proiect POO (2022-2023)

Sistem smart de recomandări stream-uri

Responsabili: Marilena Panaite

Introducere

O aplicație de streaming cunoscută are nevoie de ajutorul vostru pentru a implementa un sistem smart de recomandări pentru utilizatorii existenți. Scopul acestui proiect este de a realiza un algoritm de recomandări pentru stream-uri (muzica, podcast-uri sau audiobooks) pe baza datelor existente despre utilizatorii aplicației (ascultători) și despre creatorii de stream-uri (muzicieni, moderatori de podcast, etc.) sau cele acumulate de-a lungul rularii aplicatiei. Detalii despre formatul datelor, cat și despre comenzile ce trebuiesc implementate sunt oferite în secțiunile următoare.

Funcționalitatea aplicației

Pentru a putea implementa sistemul de recomandări, veți primi datele despre fiecare entitate implicată în fișiere separate, după cum urmează:

Datele de intrare

Datele de intrare pentru proiect se vor afla în fișiere de tipul CSV (Comma Separated Values) [1] și puteți alege orice librărie de parsare din Java pentru a citi și parsa fisierele de intrare (ex: https://mvnrepository.com/artifact/com.opencsv/opencsv/5.7.1).

Pentru output, se vor scrie toate comenzile la consola.

Metoda *main* a proiectului se va afla in clasa *ProiectPOO* și va primi ca parametrii 3 fișiere csv (în ordinea *streamers.csv streams.csv și users.csv)* și un fișier text *comenzi.txt,* ce va contine comenzile date de-a lungul rularii aplicatiei.

[1] https://en.wikipedia.org/wiki/Comma-separated_values

Streamers

Datele despre autorii de stream-uri, fie muzicieni, gazde de podcast sau autorii audiobook-urilor se vor în fișierul *streamers.csv*, unde o linie va reprezenta datele unui streamer:

streamerType, id, name

Streamer	Tip de date	Descriere
streamerType	Integer	tipul de streamer cu următoarea codificare: 1 - muzician 2 - gazda unui podcast 3 - autor al audiobook-ului
id	Integer	identificatorul unic, citit din fișier
name	String	numele streamer-ului

Streams

Datele despre stream-urile postate în aplicație pana la acest moment se vor găsi în fișierul streams.csv, unde o linie va reprezenta datele unui stream:

streamType, id, streamGenre, noOfStreams,streamerId,length,dateAdded,name

Streams	Tip de date	Descriere
streamType	Integer	tipul de stream cu următoarea codificare: 1 - piesa muzicala 2 - podcast 3 - audiobook
id	Integer	identificatorul unic, citit din fișier
streamGenre	Integer	genul stream-ului ce va fi codificat în funcție de tip astfel: Pentru o piesa muzicala: 1 - pop 2 - latin 3 - house 4 - dance 5 - trap

		Pentru un podcast: 1 - documentary 2 - celebrities 3 - tech Pentru un audiobook: 1 - fiction 2 - personal development 3 - children
noOfStreams	Long	numărul de ascultari până în acest moment
streamerId	Integer	id-ul streamer-ului care a publicat stream-ul
length	Long	durata stream-ului în secunde
dateAdded	Long	data la care a fost adaugat stream-ul, folosind formatul Unix timestamp [2]
name	String	numele streamului

User

Datele despre un utilizator al platformei de streaming se vor afla în fișierul de intrare *users.csv*, unde o linie va reprezenta un utilizator și va avea formatul:

id,name,streams

User	Tip de date	Descriere
id	Integer	identificatorul unic, citit din fișier
name	String	numele utilizatorului
streams	List<Integer>	lista de stream id, reprezentand istoria stream-urilor ascultate, despărțite printr-un spațiu (ex: 1234 1231 1345)

[2] https://en.wikipedia.org/wiki/Unix_time

Pe baza informațiilor din fișierele de input aveți libertatea sa va modelati entitățile așa cum doriți, atat timp cat sunt respectate principiile POO invatate la curs și Design Pattern-urile alese de voi.

Comenzi

Pentru a implementa sistemul smart de recomandări, implementarea proiectul trebuie sa fie capabilă sa modifice datele existente atunci cand se rulează comenzi de către utilizatori sau stream-uri, pentru a putea face recomandări corecte (ex: dacă un utilizator asculta un nou stream, diferit de cele ascultate pana la momentul citirii din fișierul *users.txt*, atunci istoricul utilizatorului cat și numărul de ascultari al stream-ului respectiv se vor schimba).

Comenzile ce vor putea sa fie efectuate de streamers vor fi:

Nr	Nume și descriere	Format	Rezultat
1	Adaugă Stream	<streamerId:Integer> ADD <streamType: Integer> <id: Integer> <streamGenre: Integer> <length: Long> <name:String>	Nu se printeaza nimic la consola, dar se modifica datele aplicației
2	Listeaza streamurile unui streamer	<streamerId:Integer> LIST	Se va afișa în format json[3] o lista de streams după schema prezentată în anexa. *length* va fi afișat în formatul HH:MM:SS (dacă length este mai strict ca 60 minute, se va afișa doar MM:SS). Puteți folosi Duration din Java [4] *dateAdded* se va afișa în formatul DD-MM-YYYY. De exemplu: [{ "id": "12345", "name": "One Kiss", "streamerName": "Dua Lipa",

[3] https://en.wikipedia.org/wiki/JSON
[4] https://docs.oracle.com/javase/8/docs/api/java/time/Duration.html

Nr	Nume și descriere	Format	Rezultat
			"noOfStreams": "100", "length": "3:33", "dateAdded": "01-05-2018" }]
3	Șterge Stream	<streamerId:Integer> DELETE <streamId: Integer>	Nu se printeaza nimic la consola, dar se modifica datele aplicației

Comenzile ce vor putea sa fie efectuate de *utilizatori* vor fi:

Nr	Nume și descriere	Format	Rezultat
1	Listeaza istoria de ascultare a utilizatorului	<userId: Integer> LIST	Se va afișa în format json[5] o lista de streams după schema prezentată în anexa.
2	Asculta un stream	<userId: Integer> LISTEN <streamId:Integer>	Nu se printeaza nimic la consola, dar se modifica datele aplicației
3	Recomanda 5 stream-uri după preferințe	<userId: Integer> RECOMMEND [SONG | PODCAST | AUDIOBOOK]	Se va afișa în format json o lista de streams după schema prezentată în anexa. Algoritmul de recomandare stream-urilor va fi următorul: - Din lista de streamers ascultați de utilizator veți alege top 5 stream-uri (neascultate) cu cele mai multe ascultari - Recomandarea va fi făcută pentru tipul de stream pasat ca parametru (SONG, PODCAST sau AUDIOBOOK)
4	Recomanda 3 stream-uri surpriză	<userId: Integer> SURPRISE [SONG | PODCAST | AUDIOBOOK]	Se va afișa în format json o lista de streams după schema prezentată în anexa. Algoritmul de recomandare stream-urilor va fi următorul:

[5] https://en.wikipedia.org/wiki/JSON

			- Din lista de streamers din aplicație, ce nu au fost ascultați de utilizator veți alege 3 stream-uri ce au fost adaugate cel mai recent. Dacă au fost adaugate in acceasi zi, atunci veți alege stream-ul cu cele mai multe ascultari. - Recomandarea va fi făcută pentru tipul de stream pasat ca parametru (SONG, PODCAST sau AUDIOBOOK)

Punctaj

Punctajul va fi împărțit în felul următor:

Punctaj	Descriere
40 p	Testele automate pentru validarea functionalitatii implementate
50 p	Folosirea a minim 4 design patterns și argumentarea în parte a implementării fiecăruia. Atenție: Nu veți primi punctajul dacă nu veți argumenta corect folosirea desgin pattern-urilor alese în modelare. De asemenea, dacă veți implementa mai puțin de 4 DP, veți primi punctaj parțial.
10 p	Folosirea principiilor POO invatate (mostenire, encapsulare, colectii, etc.)

Observații:

- Pentru a primi punctajul pe acest proiect este important ca în implementare sa folosiți minim **4** design patterns, din cele invatate la curs și să fiți capabili să argumentați folosirea lor. Atenție, neimplementarea niciun design pattern sau implementarea fără respectarea principiilor POO invatate poate duce la pierderea punctajului 100%.
- Proiectul va fi prezentat în ziua examenului, așa ca va trebui să argumentați folosirea design pattern-urilor alese și să fiți pregătiți sa raspundeti și la întrebări teoretice referitoare la acestea.

Anexe


```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "array",
  "items": [
    {
      "type": "object",
      "properties": {
        "id": {
          "type": "string"
        },
        "name": {
          "type": "string"
        },
        "streamerName": {
          "type": "string"
        },
        "noOfStreams": {
          "type": "string"
        },
        "length": {
          "type": "string"
        },
        "dateAdded": {
          "type": "string"
        }
      },
      "required": [
        "id",
        "name",
        "streamerName",
        "noOfStreams",
        "length",
        "dateAdded"
      ]
    }
  ]
}
```