

Aplicație pentru gestiunea eficientă a colecțiilor de cărți
ale unei biblioteci

Dulgheru Mihai-Nicolae
Grupa 1122

Cuprins

Introducere	3
Descriere clase	4
Clase abstracte	4
Clase	4
Enum-uri și constante	5
Interfețe	5
Descriere meniu	6
Concluzii	7
Bibliografie	7

Introducere

Aplicația reprezintă un mediu de lucru ce permite gestiunea eficientă a colecțiilor de cărți ale unei librării. Aceasta pune la dispoziția utilizatorului un meniu clar și intuitiv, ce integrează opțiuni precum „Adaugă carte în colecție”, „Șterge carte din colecție”, „Generează raport privind cărțile unui anumit autor” etc. Există două tipuri de conturi: **bibliotecar** și **cititor**. Un cititor nu poate executa operații care să altereze baza de date a bibliotecii (de exemplu: nu poate șterge o carte dintr-o colecție). Acesta se poate înregistra, poate împrumuta și returna o carte. Un bibliotecar poate adăuga și șterge cărți și are posibilitatea de a genera rapoarte privind cărțile aparținând unui anumit autor, lista cărților dintr-o anumită colecție și lista cărților împrumutate la o anumită data.

Persistența datelor este asigurată folosind fișiere text. La închiderea aplicației, datele vor fi salvate automat, respectând toate modificările aduse pe parcursul fluxului logic al aplicației, urmând ca acestea să fie restaurate la repornire. Locația unde vor fi stocate datele este „~/data/db/”. După prima execuție, se vor genera cinci fișiere text, și anume:

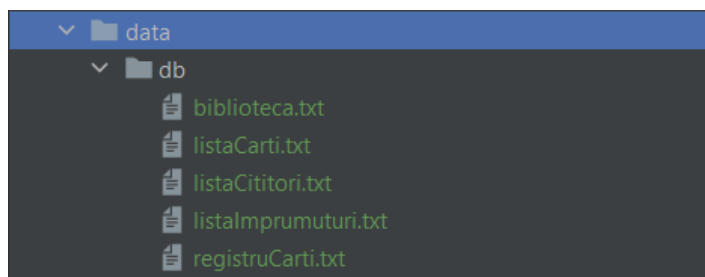


Figura 1. Structura fișierelor generate

Rapoartele vor fi generate în directorul „~/data/reports” și vor avea formă tabelară. Un exemplu de raport ce poate fi generat în aplicație este cel privind cărțile din colecția FICȚIUNE:

ISBN	Titlu	Autor	Editură	An apariție	Număr pagini
978-973-6479-24-3	Theodoros	Mircea Cărtărescu	Humanitas	2022	632
978-973-7879-92-5	Imagini ascunse	Jason Rekulak	Nemira	2022	440
978-973-9194-30-3	Ultimul gambit	Jennifer Lynn Barnes	Storia	2022	432
978-973-5465-28-3	Nopti albe	Feodor Mihailovici Dostoievski	Art	2022	120
978-973-1029-93-1	Nuvele	Ion Luca Caragiale	Cartex	2022	328

Figura 2. Raport colecție FICTIUNE.txt

În implementare sunt utilizate atât masive unidimensionale de tipuri fundamentale (int[], String[], byte[]), cât și colecții de obiecte din clasele Carte, Cititor, Categorie, ColectieCarti și Imprumut.

Datele citite de la consolă și cele preluate din fișierele de intrare sunt validate folosind mecanismul de gestiune a excepțiilor try – catch – finally prin folosirea claselor standard de excepții (Exception, ParseException, InputMismatchException, IllegalArgumentException, IOException etc.) și prin definirea unor excepții specifice (ExceptieCititorExistent, ExceptieCititorInexistent, ExceptieCNPInvalid, ExceptieImprumutInexistent, ExceptieIncarcareDateDinFisier, ExceptieValidareData, ExceptieValidareISBN, ExceptieValidareNumeAutor și ExceptieValidareNumeColectie).

Descriere clase

Clase abstracte

Clasa Cititor conține câmpurile cnp, dataNasterii, nume, prenume, adresa, nrTelefon, email și este extinsă de clasele Student și Elev. Aceasta este corespondentul contului de cititor. Clasa Meniu conține metoda abstractă *public abstract void afiseazaMeniu()* și este extinsă de clasele MeniuBibliotecar, MeniuCititor și MeniuCont.

Clase

Clasa singleton Biblioteca este responsabilă cu gestiunea centralizată a tuturor colecțiilor de cărți. Pe lângă map-ul *Map<Categorie, ColectieCarti> colectii*, clasa mai conține și câmpurile *nume, adresa, nrTelefon, email* și *orar*. O colecție de cărți este formată dintr-o listă de string-uri care reprezintă ISBN-urile cărților și un câmp *Integer nrCarti*.

Clasa Carte conține câmpurile *isbn, titlu, autor, editura, anAparitie, nrPagini* și *categorie*. Un împrumut își are corespondentul din lumea reală în clasa Imprumut, ce conține CNP-ul cititorului, lista de cărți împrumutate, data împrumutului și un map cu datele de retur pentru toate cărțile împrumutate. Aceasta implementează interfața *Imprumut*.

Clasele Carte, Cititor și Imprumut au câte o clasă echivalentă menită să stocheze câte o colecție de fiecare tip: ListaCarti, ListaCititori, respectiv ListaImprumuturi.

Pe lângă aceste clase, aplicația mai utilizează și clasele Raport și RegistruCarti. Clasa Raport conține metodele răspunzătoare cu generarea raporturilor privind cărțile aparținând unui anumit autor, lista cărților dintr-o anumită colecție și lista cărților împrumutate la o anumită dată. Clasa RegistruCarti stochează inventarul cărților din bibliotecă (toate cărțile disponibile la un moment dat).

Enum-uri și constante

Aplicația folosește două enum-uri și o clasă de variabile constante, și anume căile fișierelor unde sunt stocate datele (CaiFișiere). Clasele enum sunt Categorie (se referă la categoria unei cărți) și TipCititor (ELEV, respectiv STUDENT).

Interfețe

Visual Paradigm Online Free Edition

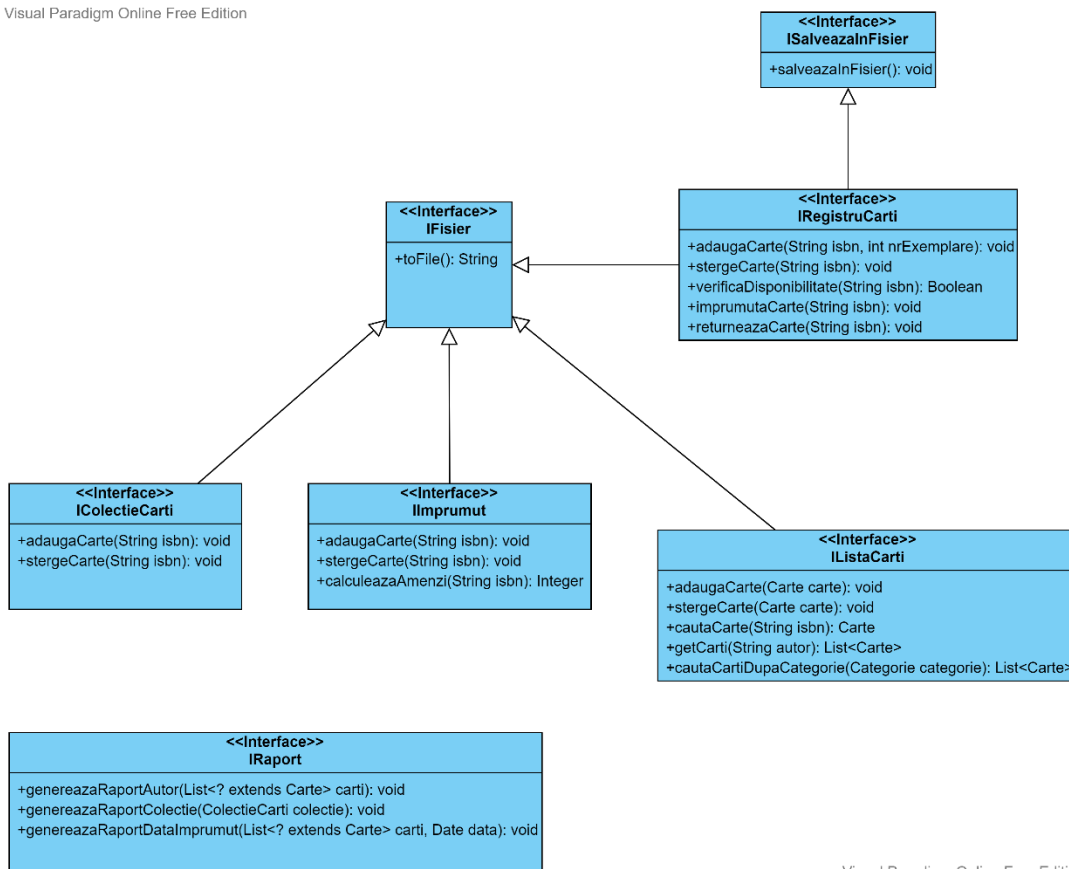


Figura 3. Diagrama de clase – Interfețe

Descriere meniu

Meniul aplicației este împărțit în 3 secțiuni:

1. Meniu alegere cont. Opțiuni: 1. Bibliotecar; 2. Cititor; 0. Ieșire

```
Vă rugăm să alegeți un cont:  
1. Bibliotecar  
2. Cititor  
0. Ieșire  
> |
```

Figura 4. Meniu alegere cont

2. Meniu bibliotecar. Opțiuni: 1. Adaugă carte în colecție; 2. Șterge carte din colecție; 3. Generează raport privind cărțile unui anumit autor; 4. Generează raport privind cărțile dintr-o anumită colecție; 5. Generează raport privind cărțile împrumutate la o anumită dată; 0. Ieșire

```
Meniu Bibliotecar  
1. Adaugă carte în colecție  
2. Șterge carte din colecție  
3. Generează raport privind cărțile unui anumit autor  
4. Generează raport privind cărțile dintr-o anumită colecție  
5. Generează raport privind cărțile împrumutate la o anumită dată  
0. Ieșire  
> |
```

Figura 5. Meniu bibliotecar

3. Meniu cititor. Opțiuni: 1. Înregistrează-te; 2. Împrumută o carte; 3. Returnează o carte; 0. Ieșire

```
Meniu Cititor  
1. Înregistrează-te  
2. Împrumută o carte  
3. Returnează o carte  
0. Ieșire  
>
```

Figura 6. Meniu cititor

Meniu contului Bibliotecar, respectiv al contului Cititor rulează până când utilizatorul introduce valoarea 0 sau orice altă valoare care nu este întreagă. În ultimul caz aplicația va afișa mesajul de eroare „EROARE: Opțiunea trebuie să fie un număr întreg!”.

Concluzii

Soluția informatică prezentată mai sus integrează conceptele fundamentale ale Programării orientate obiect și de design pattern pentru a gestiona eficient colecțiile de cărți ale unei biblioteci. Meniul este limpede, perceptibil și acoperă principalele funcționalități ale aplicației, iar raporturile generate sunt ușor de implementat.

Bibliografie

- [1] <https://docs.oracle.com/javase/tutorial/essential/io/file.html>
- [2] <https://carturesti.ro/raft/carte-109>
- [3] <https://www.javatpoint.com/java-date-to-string>
- [4] <https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>
- [5] <https://en.wikipedia.org/wiki/ISBN>
- [6] <https://www.isbn-international.org/content/what-isbn>