

Proiect Programare Orientata pe Obiecte

Student: Duminica Alex Vasile

Anul: 3

Facultatea: Calculatoare

Grupa: 1

Aplicatia 1

Descriere a Funcționalității Aplicației 1

Această aplicație Java este un simplu sistem de gestionare a unei baze de date de cărți.

1. Clasa `Main`:

- Se inițializează o bază de date (`Database`) prin specificarea unui fișier de intrare (`books.txt`).
- Se afișează toate intrările existente în baza de date.
- Se oferă un meniu interactiv pentru utilizator, în care acesta poate alege dintre următoarele opțiuni:
 - **1. Cauta Carte:** Utilizatorul introduce autorul și titlul cărții, iar apoi se afișează stocul cărții din baza de date.
 - **2. Cauta Carte Binar:** Similar cu opțiunea 1, dar căutarea se face folosind o căutare binară într-o listă sortată.
 - **3. Stop:** Oprire a aplicației.

2. Clasa `Database`:

- Oferă o structură pentru gestionarea și manipularea bazei de date de cărți.
- În constructor, se specifică calea către fișierul care conține datele inițiale ale cărților.
- Metoda `loadDatabase` încarcă datele din fișier și le stochează într-o listă de obiecte `Entry`.
- Metodele `getEntry` și `getEntryBinary` realizează căutări în baza de date, fie liniar, fie prin căutare binară, și returnează informații despre o carte specificată de utilizator.
- Metoda `printEntries` afișează toate cărțile din baza de date.

3. Clasa `Entry`:

- Reprezintă o carte individuală cu informații precum autor, titlu și stoc.
- Conține metode de acces pentru obținerea informațiilor despre carte.
- Metoda `equals` compară două obiecte `Entry` pe baza autorului și titlului.
- Metoda `compareTo` compară două obiecte `Entry` pentru a facilita sortarea alfabetică.

În ansamblu, această aplicație permite utilizatorului să efectueze căutări de cărți într-o bază de date și să vizualizeze informații despre stocul acestora. Sortarea alfabetică și utilizarea căutării binare pot îmbunătăți eficiența operațiilor asupra bazei de date.

Main (Main.java)

```
import java.util.Scanner;
import Database.*;

public class Main {
    public static void main(String[] args) {
        Database database = new Database("books.txt");
        database.printEntries();
        Scanner scanner = new Scanner(System.in);
        String clientInput;
        Boolean RUN = true;

        while (RUN) {
            System.out.println("1. Cauta Carte");
            System.out.println("2. Cauta Carte Binar");
            System.out.println("3. Stop");

            clientInput = scanner.nextLine();

            try {
                switch (Integer.parseInt(clientInput)) {
                    case 1:
                        getEntry(scanner, database);
                        break;
                    case 2:
                        getEntryBinary(scanner, database);
                        break;
                    case 3:
                        RUN = false;
                        break;
                    default:
                        System.out.println("Invalid input!");
                        break;
                }
            } catch (Exception e) {
                System.out.println("Invalid input!");
            }
        }
    }
}
```

```

    }

}

scanner.close();

}

static void getEntry(Scanner scanner, Database database) {
    System.out.print("Author: ");
    String author = scanner.nextLine();
    System.out.print("Title: ");
    String title = scanner.nextLine();

    Entry entry = database.getEntry(author, title);
    System.out.println("Numar exemplare: " + entry.getStock());
}

static void getEntryBinary(Scanner scanner, Database database) {
    System.out.print("Author: ");
    String author = scanner.nextLine();
    System.out.print("Title: ");
    String title = scanner.nextLine();

    Entry entry = database.getEntryBinary(author, title);
    System.out.println("Numar exemplare: " + entry.getStock());
}
}

```

Database (Database.java)

```
package Database;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;

public class Database {
    String FilePath;
    ArrayList<Entry> Entries;

    public Database(String Path) {
        this.FilePath = Path;
        this.Entries = new ArrayList<Entry>();
        loadDatabase();
    }

    void loadDatabase() {
        try (BufferedReader reader = new BufferedReader(new
        FileReader(this.FilePath))) {
            String line;

            while ((line = reader.readLine()) != null) {
                String title = line.trim();
                String author = reader.readLine().trim();
                int stock = Integer.parseInt(reader.readLine().trim());

                Entry entry = new Entry(author, title, stock);
                Entries.add(entry);
            }
        } catch (IOException | NumberFormatException e) {
            e.printStackTrace();
        }
    }

    public Entry getEntry(String Author, String Title) {
        Entry search = new Entry(Author, Title, null);
        for(Entry entry : Entries) {
            if(entry.equals(search))
                return entry;
        }

        return search;
    }
}
```

```

public Entry getEntryBinary(String author, String title) {
    Entry search = new Entry(author, title, null);

    int left = 0;
    int right = Entries.size() - 1;

    while (left <= right) {
        int mid = left + (right - left) / 2;
        Entry midEntry = Entries.get(mid);

        int compareResult = midEntry.compareTo(search);

        if (compareResult == 0) {
            return midEntry;
        } else if (compareResult < 0) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }

    return search;
}

public void printEntries() {
    for (Entry entry : this.Entries) {
        System.out.println(entry.getAuthor() + ":" + entry.getTitle() +
            ":" + entry.getStock());
    }
}
}

```

Entry (Entry.java)

```
package Database;

public class Entry {
    String Author;
    String Title;
    Integer Stock;

    public Entry(String Author, String Title, Integer Stock) {
        this.Author = Author;
        this.Title = Title;
        this.Stock = Stock;
    }

    public String getAuthor() {
        return this.Author;
    }

    public String getTitle() {
        return this.Title;
    }

    public Integer getStock() {
        return this.Stock;
    }

    public boolean equals(Entry entry) {
        return entry.getAuthor().equals(this.Author) &&
entry.getTitle().equals(this.Title);
    }

    public int compareTo(Entry other) {
        int authorComparison = this.Author.compareTo(other.Author);

        if (authorComparison != 0) {
            return authorComparison;
        }

        return this.Title.compareTo(other.Title);
    }
}
```