



Universitatea Tehnică de Construcții București

Facultatea de Hidrotehnica

## **Programarea Calculatoarelor si Limbaje de Programare III**

**Profesor Coordonator:**

**Ing. Gabriela Olteanu**

**Student:**

**Popa Alexandru-Cristian**



Universitatea Tehnică de Construcții București

Facultatea de Hidrotehnica

## **Proiect JAVA**

### **Sistem de notare a filmelor**

**Profesor Coordonator:**

**Ing. Gabriela Olteanu**

**Student:**

**Popa Alexandru-Cristian**



Universitatea Tehnică de Construcții București

Facultatea de Hidrotehnica

## CUPRINS

Motivarea alegerii temei .....	4
Prezentare generala .....	4
Functionalitati .....	5
Tehnologii utilizate .....	7
Structura Aplicatiei .....	8
Persistenta datelor .....	10
Limitari .....	11
BIBLIOGRAFIE .....	11

## Motivarea alegerii temei

Am ales tema „Sistem de notare a filmelor” sub forma unei aplicații desktop dezvoltate în Java deoarece aceasta permite integrarea și aplicarea practică a unui număr mare de concepte fundamentale predate în cadrul cursului. Proiectul utilizează bazele limbajului Java, principiile programării orientate pe obiecte, lucrul cu fișiere pentru stocarea persistentă a datelor, precum și realizarea unei interfețe grafice folosind biblioteca Swing. Toate aceste elemente sunt combinate într-un mod coerent, oferind o imagine clară asupra modului în care componentele teoretice pot fi transformate într-o aplicație funcțională.

Tema aleasă reflectă, de asemenea, o problemă reală și actuală. Sistemele de evaluare a filmelor stau la baza unor platforme cunoscute precum IMDb (Internet Movie Database) sau MyAnimeList, unde utilizatorii pot nota și clasifica conținutul media. Acest aspect demonstrează utilitatea practică a aplicației și relevanța sa în viața de zi cu zi, chiar dacă proiectul este realizat la o scară redusă și cu scop educațional.

Un alt motiv important al alegerii acestei teme este flexibilitatea pe care o oferă din punct de vedere al dezvoltării. Proiectul permite extinderea ușoară cu funcționalități suplimentare, precum sortarea filmelor, gestionarea imaginilor, calculul statistic al scorurilor sau îmbunătățirea interfeței grafice. Astfel, tema nu se limitează strict la cerințele minime ale cursului, ci oferă posibilitatea de a experimenta și cu tehnologii sau concepte care nu au fost abordate în detaliu în cadrul acestuia.

## Prezentare generală

Aplicația „Sistem de Notare Filme” este o aplicație desktop realizată în Java folosind biblioteca Swing. Scopul aplicației este de a permite utilizatorului să gestioneze o listă de filme, să le acorde note și, opțional, să atașeze o imagine fiecărui film. Toate datele sunt salvate local într-un fișier JSON, aplicația funcționând complet offline.

Prin realizarea acestui proiect s-a urmărit aplicarea practică a conceptelor studiate în cadrul cursului, precum programarea orientată pe obiecte, lucrul cu fișiere pentru stocarea persistentă a datelor și dezvoltarea unei interfețe grafice de bază. Aplicația reprezintă un exemplu concret de îmbinare a acestor concepte într-un program funcțional, ușor de utilizat și adaptat pentru scopuri educaționale.

## Funcționalități

Aplicația „Sistem de Notare Filme” oferă un set complet de funcționalități care permit gestionarea eficientă a unei colecții de filme într-un mediu desktop. Funcționalitatea principală constă în administrarea unei liste de filme, fiecare film fiind reprezentat printr-un set de date bine definite: titlu, gen, an de apariție, imagine asociată (opțional), număr de evaluări și scor mediu.

Utilizatorul poate adăuga filme noi prin intermediul interfeței grafice, introducând informațiile obligatorii (titlu, gen și an). În timpul acestui proces, aplicația oferă posibilitatea selectării unei imagini de pe sistemul local, folosind un mecanism standard de alegere a fișierelor. Dacă utilizatorul nu selectează o imagine, aplicația utilizează automat o imagine implicită, asigurând astfel consistența vizuală a interfeței.

Aplicația implementează un sistem de evaluare numerică, în care fiecare film poate fi notat pe o scară de la 1 la 10. Evaluările sunt cumulative, iar scorul afișat pentru fiecare film este calculat automat ca medie aritmetică a tuturor notelor primite. În plus, aplicația păstrează evidența numărului total de evaluări, acest indicator fiind utilizat atât pentru afișare, cât și pentru sortarea filmelor în funcție de popularitate. Evaluările existente nu pot fi șterse, ci doar suplimentate, ceea ce asigură integritatea datelor statistice.

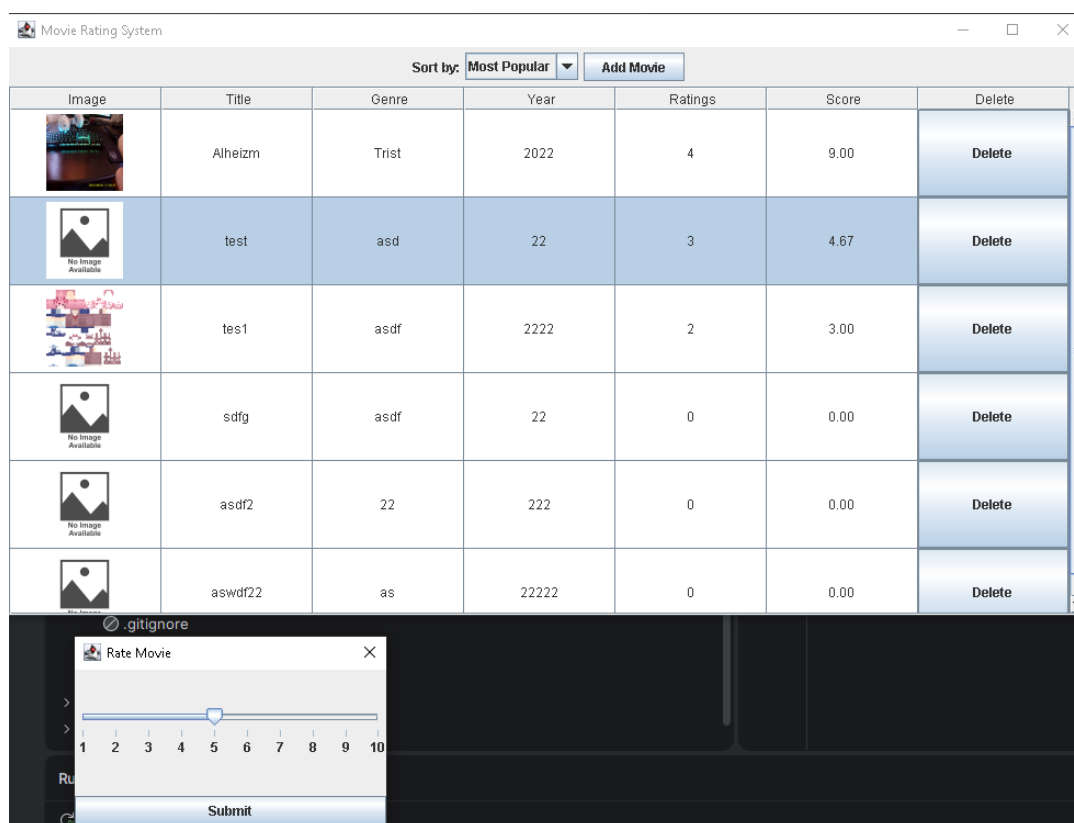
Lista de filme poate fi ordonată după mai multe criterii, în funcție de preferințele utilizatorului. Sortarea poate fi realizată după scorul mediu al filmelor sau după numărul de evaluări, permițând identificarea rapidă a filmelor cel mai bine cotate sau cel mai des evaluate. Sortarea este realizată dinamic, fără a afecta datele stocate, având impact doar asupra modului de afișare în interfața grafică.

Aplicația oferă și funcționalitatea de ștergere a filmelor din listă. În momentul ștergerii, toate datele asociate filmului respectiv sunt eliminate, iar modificarea este salvată imediat în fișierul de stocare. Această funcționalitate permite utilizatorului să mențină lista de filme actualizată și relevantă.

Un aspect esențial al aplicației este persistența datelor. Toate informațiile sunt salvate local într-un fișier JSON, folosind mecanisme de serializare și deserializare. Salvarea se realizează automat de fiecare dată când apar modificări în date, iar la pornirea aplicației fișierul este citit pentru a restaura starea anterioară. Acest lucru permite utilizarea aplicației pe termen lung, fără pierderea informațiilor introduse de utilizator.

Din punct de vedere al interfeței grafice, aplicația utilizează un tabel pentru afișarea filmelor, fiecare coloană fiind aliniată și dimensionată corespunzător pentru a asigura lizibilitatea datelor. Imaginile sunt redimensionate automat pentru a se potrivi în celulele tabelului, iar conținutul textual este centrat pentru o prezentare clară și uniformă.

Prin ansamblul funcționalităților implementate, aplicația oferă un exemplu complet de program desktop care îmbină logica de business, gestionarea datelor și interfața grafică într-o soluție coerentă și ușor de utilizat, fiind adecvată atât scopului educațional, cât și extinderii ulterioare cu funcționalități suplimentare.



## Tehnologii utilizate

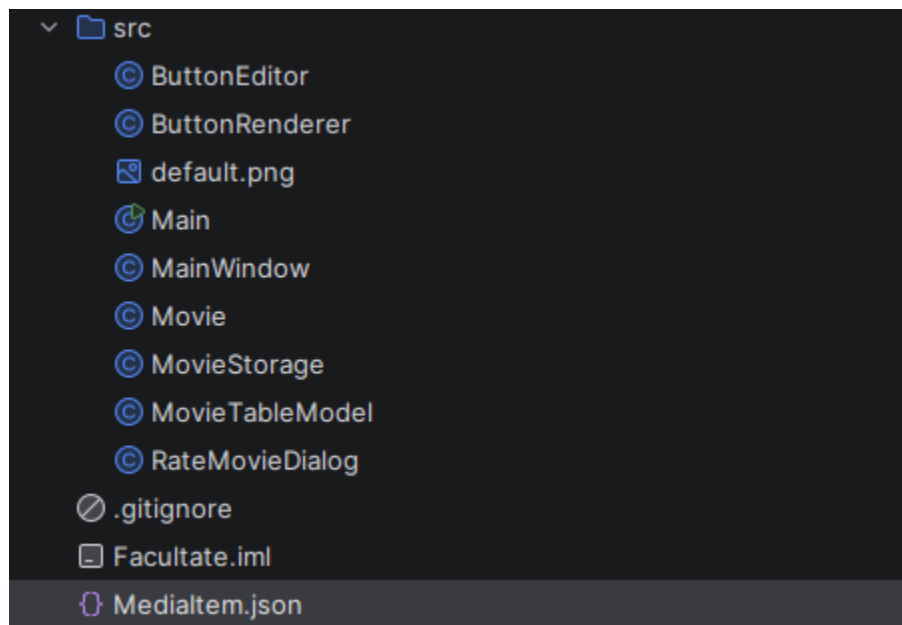
**JetBrains IntelliJ IDEA** este un mediu de dezvoltare integrat lansat în anul 2001 de compania JetBrains, conceput pentru a sprijini dezvoltarea eficientă a aplicațiilor software, în special în limbajul Java. Această tehnologie a fost creată ca răspuns la dificultățile întâmpinate în dezvoltarea aplicațiilor complexe, precum gestionarea codului de mari dimensiuni, identificarea erorilor și depanarea acestora. IntelliJ IDEA oferă un set avansat de instrumente pentru analiza statică a codului, completare automată inteligentă și depanare, contribuind la creșterea productivității și la reducerea erorilor de implementare.

**Java** a fost dezvoltat în anul 1995 de compania Sun Microsystems, având ca obiectiv principal crearea unui limbaj de programare portabil, sigur și independent de platformă. Java a fost conceput pentru a rezolva problema incompatibilității dintre sistemele de operare și arhitecturile hardware diferite, introducând conceptul de rulare a aplicațiilor prin intermediul mașinii virtuale Java (JVM). Această abordare permite ca același cod sursă să fie executat pe multiple platforme fără modificări, oferind totodată un nivel ridicat de securitate și stabilitate, aspecte esențiale în dezvoltarea aplicațiilor moderne.

**Swing** reprezintă o bibliotecă grafică introdusă la sfârșitul anilor 1990, ca parte a platformei Java, având rolul de a facilita dezvoltarea interfețelor grafice pentru aplicațiile desktop. Swing a fost creat pentru a depăși limitările bibliotecilor grafice anterioare, oferind componente vizuale avansate și un grad ridicat de personalizare. Prin utilizarea acestei tehnologii, dezvoltatorii pot crea interfețe grafice consistente și independente de sistemul de operare, contribuind la îmbunătățirea experienței utilizatorului.

**Gson** este o bibliotecă dezvoltată de Google, lansată în jurul anului 2008, care facilitează lucrul cu date în format JSON în aplicațiile Java. Aceasta a fost creată pentru a rezolva problema conversiei manuale dintre obiectele Java și reprezentările JSON, proces care era adesea complex și predispus la erori. Gson permite serializarea și deserializarea automată a datelor, asigurând un mod eficient, sigur și standardizat de stocare și transfer al informațiilor, fiind larg utilizată în aplicațiile care implică persistența sau schimbul de date.

## Structura Aplicației



Clasa Movie este containerul care conține toate informațiile necesare despre un film: titlul, genul, anul, calea către imagine, distribuția notelor, numărul total de evaluări și scorul mediu. Clasa are un constructor care nu cere nimic legat de rating dar are metoda “add Rating” care permite adăugarea unei note cand e chemata și recalculeaza automat a scorul. Recenziile sunt stocate sub forma unui vector de frecventa cu dimensiunea de 11. Elementele de la 1-10 sunt folosite pentru a stoca numarul de recenzii si numarul recenziei. Scorul mediu este calculat printr-un for loop care parcurge aceasta variabila.

Clasa MovieStorage este responsabilă exclusiv de salvarea și încărcarea datelor. Aceasta citește lista de filme din fișierul MediaItem.json la pornirea aplicației și scrie datele în același fișier ori de câte ori are loc o modificare.



Clasa MAIN are liniile de cod care declanșează lansarea programului swing. Programul în sine este scris în MainWindow. Acest lucru este necesar deoarece swing își creează propriul thread în care să ruleze.

Clasa MainWindow reprezintă fereastra principală a aplicației. Aceasta inițializează interfața, gestionează evenimentele utilizatorului (adăugare, ștergere, sortare, evaluare) și asigură sincronizarea dintre date și interfață. Ea apelează clasele Movie și MovieStorage pentru a inițializa datele iar după cheama clasa MovieTableModel pentru a afișa și umple tabelul cu informații.

Clasa MovieTableModel definește modul în care datele sunt afișate în JTable. Aici sunt definite coloanele, tipurile de date și logica prin care este afișată imaginea filmului. Dacă un film nu are o imagine asociată, este afișată automat imaginea implicită (default.png). MovieTableModel are o structură modulară, așa că aceasta apelează clasele care au legătură cu elementele din tabel (ButtonEditor și ButtonRenderer pentru funcționalitatea butoanelor și animația când este apăsă și RateMovieDialog pentru când un element este selectat din listă prin dublu-click)

Clasa Button Editor asigură funcționalitatea butonului.

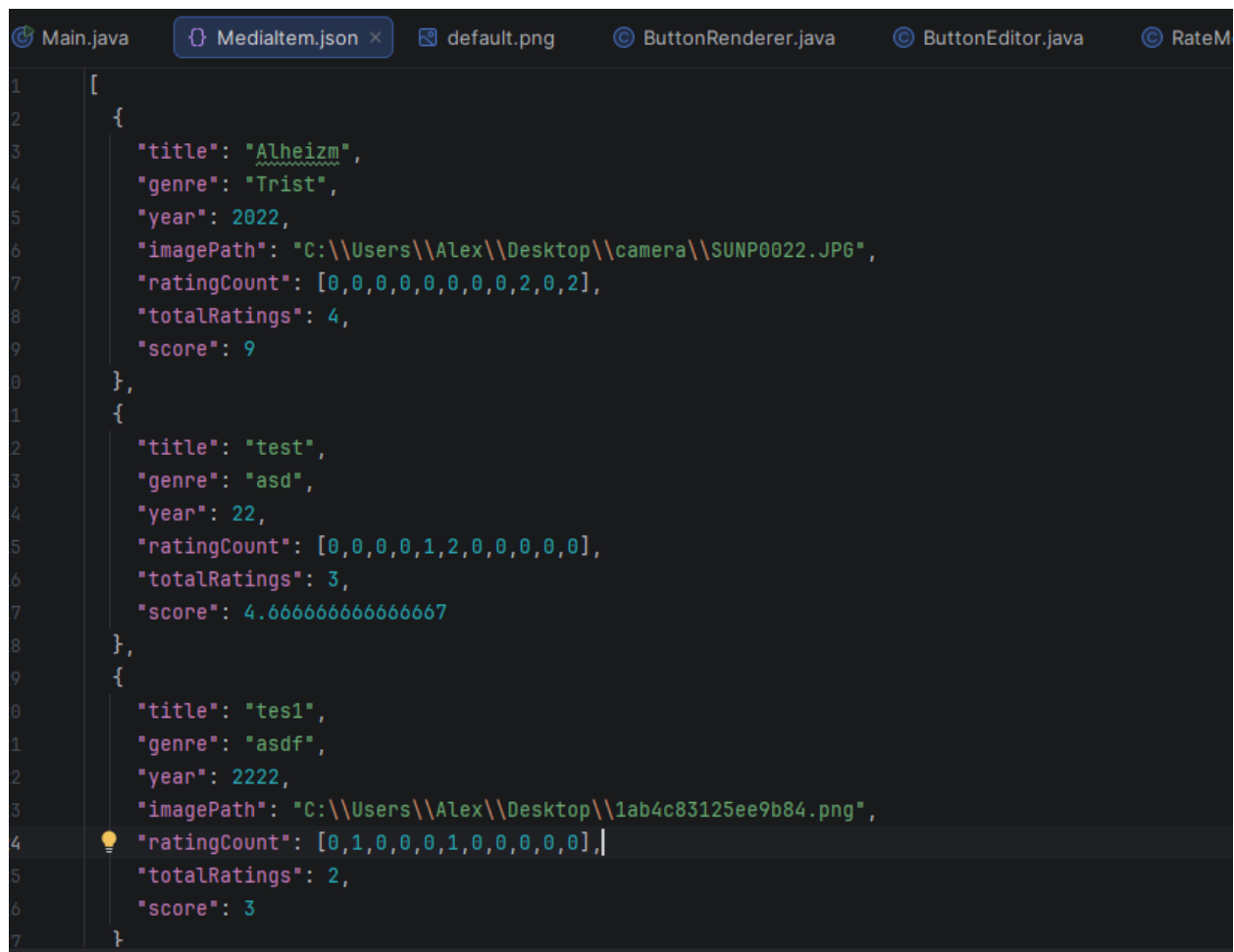
Clasa ButtonRender are rolul de a afișa animația când buton este apăsă.

Aceste clase sunt necesare deoarece la fiecare coloană creată trebuie să se inițializeze un nou buton.

Clasa RateMovieDialog are scopul de a crea o nouă fereastră care permite utilizatorului să acorde o notă unui film folosind un slider numeric de la 1 la 10.

## Persistenta datelor

Datele sunt salvate într-un fișier numit MediaItem.json, aflat în directorul principal al proiectului. Fișierul este creat și actualizat doar atunci când apar modificări în date, precum adăugarea, ștergerea sau evaluarea unui film. La pornirea aplicației, fișierul este citit automat, iar lista de filme este reconstruită în memorie.



```

1  [
2    {
3      "title": "Alheizm",
4      "genre": "Trist",
5      "year": 2022,
6      "imagePath": "C:\\Users\\Alex\\Desktop\\camera\\SUNP0022.JPG",
7      "ratingCount": [0,0,0,0,0,0,0,0,2,0,2],
8      "totalRatings": 4,
9      "score": 9
10   },
11   {
12     "title": "test",
13     "genre": "asd",
14     "year": 22,
15     "ratingCount": [0,0,0,0,1,2,0,0,0,0,0],
16     "totalRatings": 3,
17     "score": 4.666666666666667
18   },
19   {
20     "title": "tes1",
21     "genre": "asdf",
22     "year": 2222,
23     "imagePath": "C:\\Users\\Alex\\Desktop\\1ab4c83125ee9b84.png",
24     "ratingCount": [0,1,0,0,0,1,0,0,0,0,0],
25     "totalRatings": 2,
26     "score": 3
27   }
28 ]
  
```

## Limitari

Aplicația nu oferă posibilitatea de a edita un film existent sau de a modifica imaginea acestuia după creare. Imaginile sunt stocate prin cale absolută, ceea ce presupune ca fișierele să existe în continuare pe sistemul utilizatorului. De asemenea, aplicația nu folosește o bază de date, aceasta fiind o alegere intenționată pentru simplitate.

## BIBLIOGRAFIE

Documentația oficială Java, pusă la dispoziție de Oracle, a fost utilizată pentru studierea limbajului Java și a bibliotecii Swing. Aceasta a oferit informații detaliate despre clasele utilizate în dezvoltarea interfeței grafice, gestionarea evenimentelor și lucrul cu componente precum JFrame, JTable, JDialog, JSlider și JFileChooser.

Oracle, *Java Platform, Standard Edition Documentation*, <https://docs.oracle.com/javase/>

Documentația oficială a bibliotecii Gson a fost consultată pentru implementarea mecanismelor de serializare și deserializare a obiectelor Java în format JSON. Aceasta a fost esențială pentru stocarea persistentă a datelor aplicației într-un fișier local.

Google, *Gson User Guide*, <https://github.com/google/gson>

Platforma Stack Overflow a fost utilizată ca sursă suplimentară de informații pentru clarificarea unor probleme punctuale întâlnite în timpul dezvoltării, în special în ceea ce privește utilizarea JTable, renderizarea imaginilor și gestionarea evenimentelor în Swing.

Stack Overflow, <https://stackoverflow.com/>

Wikipedia a fost utilizată pentru obținerea unor informații generale privind tehnologiile folosite în cadrul proiectului.

Wikipedia, <https://www.wikipedia.org/>