

**Marin Stefania, 324CC**

Tema a avut un grad de dificultate mediu si rezolvarea ei mi-a luat cam 3 saptamani.

### **IMDB App**

Prima clasa este AccountType care este o enumerare. In ea am tipurile de useri: admini, contributori si regulari.

A doua clasa este Actor. In aceasta clasa sunt implementati actorii. Implementeaza Comparable deoarece vor trebui comparati actorii dupa nume. Folosim JsonCreator pentru a putea crea acest obiect si apoi JsonProperty pentru a stii fiecare variabila carui camp din json apartine. Am scris getterii si setterii pe care ii vom folosi in alte clase si am facut si clasa ProductionRole. Aceasta ne ajuta pentru performances care contine 2 campuri: title si type. Mai exista metoda toString care afiseaza actorii si apoi am diverse functii care ne ajuta sa afisam actorii dupa mai multe criterii.

Clasa AddSystemStrategy implementeaza interfata ExperienceStrategy si returneaza experienta user-ului care adauga un actor/productie in sistem.

Clasa Admin extinde Comparable si Staff. Contine listele pentru actorii si productiile de care este responsabil, listele de la favorite si mai are getteri si setteri.

Clasa ConsoleUserInterface are 2 metode: afisare si getInput care ma ajuta sa citesc de la tastatura si sa afisez pe ecran un mesaj.

Clasa Contributor contine aceleasi liste si are acelasi constructor ca adminul doar ca in plus mai are niste metode care sunt implementate in clasa Flow. Aceasta mai implementeaza si RequestsManager care contine functiile: createRequest si removeRequest. In aceasta clasa mai gasim si getteri si setteri.

Clasa Credentials contine e-mailul si parola pentru user. Aici am implementata si metoda toString pentru a afisa aceste informatii pe ecran. Pe langa astea mai am si getteri si setteri.

Clasa Episode contine detalii despre episoade de care avem nevoie in clasa Series. Avem getteri, setteri si metoda toString pentru a afisa numele si durata fiecarui episod.

Interfata ExperienceStrategy contine metoda calculateExperience care ajuta la calculul experientei unui user si este un design pattern.

Clasa Flow contine 2 metode care sunt exceptii pentru comenzile invalide si pentru informatiile care sunt incomplete. Contine si cele 3 meniuri ale userului si diferite metode pentru a putea folosi meniurile. Metoda loginAgain reconecteaza userul.

- Metodele pe care le contine Flow sunt:

- removeFavoriteProduction care sterge din lista de favorite o anumita productie citind de la tastatura indexul corespunzator.

- removeFavoriteActor care sterge din lista de favorite actorul corespunzator indexului citit de la tastatura.

- addFavoriteProduction care adauga o productie in lista de favorite. Daca productia nu exista se va afisa un mesaj si se va iesi din functie.

- addFavoriteActor care adauga un actor in lista de favorite. Daca actorul nu exista se va afisa un mesaj si se va iesi din functie.

- createReview adauga un review in lista de ratings a productiei. Se vor citi de la tastatura datele pentru variabilele care se afla in Ratings.

- createRequest adauga un Request in lista de Requests si se vor citi de la tastatura datele. In functie de tipul userului care este responsabil pentru rezolvarea cererilor se vor imparti in 2 categorii: rezolvate de admini sau de toti userii.

- functia findUserByUsername care va cauta un user dupa username si va returna userul sau null daca nu il gaseste.

- removeRequest sterge un anumit request in functie de indexul care va fi citit de la tastatura.

- deleteReview sterge un review. Acesta este ales dintr-o lista cu ajutorul unui index.

- viewNotifications afiseaza un mesaj daca nu exista notificari sau afiseaza notificari in functie de tipul userului.

- resolveRequest rezolva requestul de care este responsabil userul si apoi sterge din lista requestul respectiv.

- createActorFromSystem creaza un actor citind de la tastatura datele.

- createNewProductionMovie si createNewProductionSeries sunt functii care creaza un film sau un serial citind de la tastatura datele.

- removeProductionFromSystem sterge din sistem o productie alegand un index citit de la tastatura.

- removeUser sterge din sistem un user alegand un index citit de la tastatura.

- removeActorFromSystem sterge un actor citind indexul respectiv de la tastatura.> -
- createUser creaza un user citind datele de la tastatura si afisand un mesaj daca informatiile sunt incomplete.
- updateProduction actualizeaza datele lasand userul sa aleaga ce vrea sa modifice.
- updateActor actualizeaza datele lasand userul sa aleaga ce vrea sa modifice.
- functiile findProductionByName si findActorByName cauta o productie sau un actor si ii returneaza daca ii gaseste sau null in caz contrar.
- movieSeriesActorMenu este un meniu de unde poate userul sa aleaga ce vrea sa caute(film/serial/actor) si se vor afisa pe ecran informatiile.
- actorsMenu si productionMenu suunt meniuri care contin mai multe metode de a afisa actorii sau productiile si lasa userul sa aleaga metoda de afisare.

Clasa FlowApp este o clasa asemanatoare cu Flow doar ca ea este folosita pentru a lucra in interfata grafica. Contine in plus obiectul Component parent care ajuta la mostenirea proprietatilor unei interfete grafice. Folosirea parent asigura ca dialogurile sunt centrate corespunzator si se comporta cum trebuie in raport cu restul aplicatiei.

Clasa Genre contine genuri de filme sau seriale.

Clasa GUI contine 2 metode: displayOutput si getInput care afiseaza un mesaj si citeste de la tastatura informatii pentru metoda Login din interfata grafica.

Clasa IMDB este pagina principala a aplicatiei. Aceasta contine mai multe metode:

- contine Constructorul
- se creaza singletonul care este un design pattern.
- metoda run care contine alte metode pentru a putea citi datele din json si pentru a alege meniul userului.
- loadDataJSON incarca datele din fisierul Json
- startFlow este metoda de unde se alege meniul in functie de tipul userului.
- metoda login este metoda prin care un user se logheaza in terminal.
- metoda loginGUI este metoda prin care un user se logheaza in interfata grafica.
- presentInterfaceOptions permite utilizatorului sa aleaga cum vrea sa se conecteze(terminal sau interfata)
- functia main cu care porneste aplicatia

- clasa RequestsHolder care contine metodele addRequest, removeRequest si getRequests

Clasa IssueStrategy returneaza experienta pe care un user o primeste cand rezolva un request.

Clasa MainPage creaza pagina principala a aplicatiei.

Clasa Movie implementeaza unul din tipurile productiei. In plus are campurile: releaseYear si duration. Contine getteri si setteri dar si metoda de afisare toString.

Clasa MovieApp implementeaza interfata grafica si are mai multe metode:

- constructorul MovieApp care seteaza dimensiunea paginii, contine butonul de inchidere, parsarea si pagina principala

- initializeMainPage care contine butonul de selectare pentru a afisa actorii sau productiile.

- updateMainContent care actualizeaza pagina daca userul vrea cealalta optiune de afisare si seteaza locul in care vor aparea butonul de selectare si textul.

- createDetailPanelActor care contine detaliile despre actor si butonul de back dupa ce userul a ales un anumit actor

si vrea sa se intoarca la pagina principala.

- createActorsPanel care afiseaza pe ecran toate numele de actor si cu ajutorul unui MouseListener cand apasa dublu click ii va afisa detalii despre actorul respectiv.

- createDetailPanelProduction si createProductionPanel se comporta la fel ca cele 2 de mai sus.

- initializeComponent creaza pagina de login

- initializeMenu creaza bara de meniu care are File sau Exit.

- createLoginPanel pagina de login unde userul isi adauga datele pentru a se conecta.

Clasa Observer este o interfata care ajuta ca userii sa primeasca notificari si este un design pattern.

Clasa Parser parseaza datele. Obiectul objectMapper ajuta la transformarea datelor din json in liste si am metode pentru cele 4 fisiere din input.

Clasa PasswordGenerator genereaza o parola random pentru metoda de creare a unui user iar lungimea este citita de la tastatura

Clasa Production contine JsonSerializer pentru a serializa elementele, JsonTypeInfo pentru a stii tipul de productie si JsonSubType care este o subclasa a lui Type si ajuta la identificarea obiectelor Movie si Series.

- Mai contine un constructor, getteri si setteri
- metoda pentru a calcula media ratingurilor.
- metoda averageRating care ajuta la calcularea mediei.
- metoda toString

Pe langa astea am mai multe functii pentru a ajuta la afisarea productiilor in mai multe moduri

Clasa ProductionToStringSerializer face convertirea in string a obiectelor

Clasa ProductionType este o enumerare pentru tipul productiei(Movie/Series)

Clasa Rating care contine 3 campuri: username, coment si rating. Avem constructorul, getteri si setteri dar si metoda toString pentru a afisa datele.

Clasa Regular care extinde User. Contine listele ca si adminul si contirbutorul, getteri si setteri si cele 2 metode pentru a crea si sterge un request, metode scrise in Flow.

Clasa Request contine constructor, getter, setter si metode de rezolvare a unei cereri, adaugare de review-uri si metoda toString pentru a afisa datele.

Interfata RequestManager care contine cele 2 metode: createRequest si removeRequest.

Clasa RequestTypes care este o enumerare a tipurilor de cerere.

Clasa ReviewStrategy care returneaza experienta pe care o primeste userul atunci cand scrie un review.

Clasa Series care este celalalt tip al productiei. In plus contine campurile: plot, releaseYear, numberOfSeasons si seasons. Mai avem getteri si setteri, constructor si metoda toString pentru a afisa datele

Clasa Staff care implenteaza interfata StaffInterface contine metodele addProductionSystem, addActorSystem, removeProductionSystem, removeActorSystem, updateProduction, updateActor, resolveRequests care sunt implementate in Flow. Mai avem getteri, setteri si construtor.

Interfata StaffInterface contine metodele de mai sus.

Clasa TerminalUI implementeaza metodele displayOutput si getInput pentru a putea citi de la tastatura emailul si parola

Clasa User are TypeInfo si SubType ca la Production, subtype-ul fiind dat de Regular, Admin si Contributor. Mai avem si clasa Information care contine informatiile despre user, constructor pentru Information si User, getteri si setteri si metoda de a creste experienta in functie de tipul actiunii pe care o face userul, metoda de a adauga si vizualiza o notificare si metoda toString pentru a afisa datele. Pe langa acestea mai exista si metoda logout pentru a deconecta userul si a goli listele si notificarile.

Clasa UserFactory care ajuta la crearea unui user este un desing pattern si contine un constructor si un switch pentru a alege tipul de user atunci cand adminul il creaza.

Interfata UserInterface contine metodele displayOutput si getInput pentru a afisa si citi de la tastatura date despre user.