

README POO

Nume : Stanciu Vlad-Mihai

Grupa : 321CC

Numele meu este Stanciu Vlad-Mihai si sunt student la grupa 321CC.

Consider ca nivelul de dificultate al temei a fost unul mediu, deoarece a inglobat toata materia invata de-a lungul semestrului. Eu am alocat un timp de 14 zile rezolvarii acestei teme, in care am lucrat in medie 4 ore pe zi.

Pentru implementarea temei voi vorbi despre fiecare implementare in parte :

1) La general am implementat o functie numita loadData() pentru a citi din fisiere informatiile necesare utilizand Jackson si o functie numita unloadData() pentru a reactualiza informatiile. Am implementat 3 functii, respectiv : flowRegular(), flowContributor() si flowAdmin() pentru fiecare dintre tipurile de utilizatori.

2) In primul rand, aplicatia doreste sa stie daca va functiona in CLI sau in GUI . Am implementat o functie de login() care cere pe rand credentialele, dupa care cauta in lista de utilizatori un cont care sa aiba aceleasi informatii. Daca se gaseste, atunci va porni una dintre functiile de flow, daca nu vei putea alege intre a inchide aplicatia si a incerca din nou sa te loghezi.

3) Odata logat, in functie de tipul contului, alaturi de un mesaj de intampinare, iti vor fi prezentate toate optiunile disponibile pentru tipul contului.

4) Pentru apasarea tastei „1” se vor putea afisa productiile. Aici am implementat 3 moduri. Primul mod este o afisare simpla a productiilor in ordinea in care au fost citite. Al doilea mod, vei introduce de la tastatura un gen si vor fi afisate toate productiile ce contin acel gen, sau in functie de numarul de rating-uri. Aceste functii fac doar afisari, nesortand cu adevarat lista.

5) Pentru apasarea tastei „2” se vor putea afisa actorii. Acestia pot fi vizualizati in ordinea adaugarii sau in ordine alfabetica. Din nou, afisarea alfabetica este realizata prin sortarea unei liste copii ale listei de actori, pentru a pastra lista originala nesortata.

6) Pentru apasarea tastei „3” se vor afisa toate notificarile user-ului curent printr-o simpla afisare de lista.

7) Pentru apasarea tastei „4” se vor putea cauta un actor/serial/film cu ajutorul unei functii statice auxiliare implementate „find” care parcurge toate cele 3 liste (Users, Actors, Productions) in functie de ceea ce iti doresti sa cauti. In cazul in care find intoarce null, se afiseaza un mesaj in conform in care se precizeaza ca „Actorul/Filmul/Serialul nu a fost gasit”.

8) Pentru apasarea tastei „5” user-ul va trebui sa introduca numele actorului/filmului/serialului pe care vrea fie sa il adauge, fie sa il stearga. Apoi este rugat sa apese 1 pentru a adauga sau 2 pentru a sterge. Daca functia find nu gaseste productia gasita, nu se va adauga nimic si nici nu se va sterge nimic din lista de favorite, programul continuandu-si flow-ul.

Acum voi parcurge fiecare functionalitate in parte, deoarece vor diferi tastele in functie de tipul utilizatorului.

9) Functiile de adaugare si de stergere a cererilor sunt diferite intre tipurile regular si contributor, deoarece pentru contributor trebuie pusa o conditie in plus, atat la „movie issue” cat si „actor issue”, deoarece trebuie verificat daca sunt contributile contributorului care doreste sa creeze cererea, caz in care primeste un mesaj in conformitate cu cauza erorii, dupa care programul continua sa ruleze. In principiu, creez un request cu toate attributele nule. Si apoi utilizatorul introduce pe rand datele necesare. Numele celui care trebuie sa se ocupe de cerere este facut automat printr-o functie statica creata intr-o clasa numita Functions unde am implementat si alte functii auxiliare. Pentru stergere, vor fi afisate toate cererile create de utilizator indexate, iar acesta introduce numarul indexului cererii pe care vrea sa o stearga.

10) Pentru adaugarea unei productii/actor din sistem, utilizatorul introduce pe rand fiecare parametru. Pentru lista de regizori si actori, terminarea introducerii elementelor in lista se termina atunci cand utilizatorul introduce „end” lucru specificat in mesaj. Pentru introducerea genurilor se afiseaza toate genurile posibile indexate, iar utilizatorul introduce pe rand indexii genurilor pe care vor sa le introduca. Totul este facut cu un while infinit pana la introducerea unei valori egale cu numarul tuturor genurilor plus unu care reprezinta indexul genului End care semnifica finalizarea introducerii genurilor. Pentru stergerea din sistem, utilizatorul vede toate

filmele/serialele/actorii asupra carora are drepturi de modificare/stergere asa cum a fost prezentat in cerinta.

11) Pentru vizualizarea si rezolvarea cererilor, contributorii si adminii au functii diferite. Ceea ce difera este ca adminii sunt pusi sa aleaga daca doresc sa vizualizeze/rezolve cererile adresate lor in mod special sau cele dedicate intregii echipe de admini, pe cand contributorii au posibilitatea doar de a alege dintre vizualizare/stergere. Pentru vizualizare sunt afisate cererile in mod indexat, iar pentru stergere, prima oara sunt afisate cererile in mod indexat dupa care se asteapta ca utilizatorul sa introduca indexul cererii pe care vrea sa o rezolve.

12) Pentru actualizarea informatiilor atat despre productii, cati si despre actori, in principiu a fost reciclat codul de la functionalitatea de adaugare in sistem, insa structurat pe etape pentru a putea modifica cate un atribut al productiei/actorului in parte. Utilizatorul poate alege ce isi doreste sa modifice prin tastarea unui index dintr-o lista afisata anterior.

13) Pentru adaugarea unei recenzii, initial am verificat lista productiei respective de rating-uri pentru a vedea daca utilizatorul a mai adaugat inainte o recenzie. Daca da, acesta este rugat sa o stearga prima oara, iar in caz contrar, initializez un rating nou dupa care utilizatorul introduce datele necesare pentru creerea rating-ului si sunt setate secvential. Daca utilizatorul alege sa stearga o recenzie, pur si simplu se parcurge lista de rating-uri a productiei respective, se retine index-ul unicei lui recenzii, iar dupa parcurgerea listei este eliminata.

14) Pentru adaugarea unui utilizator, utilizatorul introduce manual informatiile necesare. Algoritmul de generare a unei parole puternice, insereaza initial 8 litere mari si mici alternativ random, apoi 5 cifre total random, iar apoi 3 semne speciale random. Username-ul este generat prin transformare numelui noului utilizator. Folosim `toLowerCase()` pentru a face toate literele mici, unde intalnim spatiu utilizam „_”, iar apoi introducem un numar de 4 cifre total random. Se verifica unicitatea lui cu un while infinit prin care se formeaza acest username pana cand nu este gast in lista actuala de useri.

15) Au fost implementate toate pattern-urile, mai putin „Observer pattern”.