Documentație

Scrieți o aplicație pentru o firmă de închiriere de filme.

**Lista de functionalitati**

|  |  |
| --- | --- |
| F1 | Adaugare in lista de filme/clienti |
| F2 | Stergere in lista de filme/clienti |
| F3 | Modificare in lista de filme/clienti |
| F4 | Cautare in lista de filme/clienti |
| F5 | Inchiriere/returnare film |
| F6 | Clienți cu filme închiriate ordonat dupa: nume, după numărul de filme închiriate |
| F7 | Cele mai inchiriate filme. |
| F8 | Primi 30% clienti cu cele mai multe filme (nume client și numărul de filme închiriate) |

Plan iterații:

I1: Creare aplicatie pe arhitectura stratificata + F1, F2, F3

I2: F4, F5

I3: F6, F7, F8

**Scenariu de rulare pentru adaugarea unui client**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 1 din metoda alegere\_optiune() al obiectului UI |
| p = 1 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de intrare id, nume si prenume |
| Se apeleaza metoda add\_client al obiectului sv\_client |
| Se creaza un obiect de tipul Client() cu datele de intrare |
| Obiectul dat este validat prin metoda validare\_client al obiectului de tip ValidatorClient() |
| Se stocheaza clientul creat in repository prin metoda store\_client al obiectului de tip RepositoryClient() |

**Scenariu de rulare pentru adaugarea unui film**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 1 din metoda alegere\_optiune() al obiectului UI |
| p = 2 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de intrare id, titlu si gen |
| Se apeleaza metoda add\_film al obiectului sv\_film |
| Se creaza un obiect de tipul Film() cu datele de intrare |
| Obiectul dat este validat prin metoda validare\_film al obiectului de tip ValidatorFilm() |
| Se stocheaza filmul creat in repository prin metoda store\_film al obiectului de tip RepositoryFilm() |

**Scenariu de rulare pentru stergerea unui client**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 2 din metoda alegere\_optiune() al obiectului UI |
| p = 1 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de intrare id |
| Se apeleaza metoda stergere\_client al obiectului sv\_client |
| Se apeleaza metoda delete\_client(id) al obiectului de tip RepositoryClient() |
| Se valideaza id-ul dat prin metoda validare\_id al obiectului de tip RepositoryClient(); Daca id-ul nu este validat, se va arunca o eroare de tip ValueError |
| Se va sterge clientul corespunzator cu id-ul dat, validat |

**Scenariu de rulare pentru stergerea unui film**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 2 din metoda alegere\_optiune() al obiectului UI |
| p = 2 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de intrare id |
| Se apeleaza metoda stergere\_film al obiectului sv\_film |
| Se apeleaza metoda delete\_film(id) al obiectului de tip RepositoryFilm() |
| Se valideaza id-ul dat prin metoda validare\_id al obiectului de tip RepositoryFilm(); Daca id-ul nu este validat, se va arunca o eroare de tip ValueError |
| Se va sterge filmul corespunzator cu id-ul dat, validat |

**Scenariu de rulare pentru modificarea numelui unui client**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 3 din metoda alegere\_optiune() al obiectului UI |
| p = 1 din metoda alegere\_optiune() al obiectului UI |
| P = 1 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de intrare id, nume |
| Se apeleaza metoda modificare\_client al obiectului sv\_client |
| Se valideaza numele dat prin metoda validare\_modificator al obiectului ValidatorClient(); daca numele nu este validat, se va arunca o eroare de tipul ValueError |
| Se apeleaza metoda modifica\_nume\_client() al obiectului de tip RepositoryClient() |
| Se valideaza id-ul dat prin metoda validare\_id al obiectului de tip RepositoryClient(); Daca id-ul nu este validat, se va arunca o eroare de tip ValueError |
| La clientul corespunzator cu id-ul dat se va schimba numele cu numele dat ca data de intrare prin metoda set\_nume |

**Scenariu de rulare pentru modificarea titlului unui film**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 3 din metoda alegere\_optiune() al obiectului UI |
| p = din metoda alegere\_optiune() al obiectului UI |
| P = 1 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de intrare id, titlu |
| Se apeleaza metoda modifica\_film al obiectului sv\_film |
| Se valideaza titlul dat prin metoda validare\_modificator al obiectului ValidatorFilm(); daca titlul nu este validat, se va arunca o eroare de tipul ValueError |
| Se apeleaza metoda modifica\_titlu\_film() al obiectului de tip RepositoryFilm() |
| Se valideaza id-ul dat prin metoda validare\_id al obiectului de tip RepositoryFilm(); Daca id-ul nu este validat, se va arunca o eroare de tip ValueError |
| La filmul corespunzator cu id-ul dat se va schimba titlul cu titlul dat ca data de intrare prin metoda set\_titlu |

**Scenariu de rulare pentru cautarea tuturor clientilor dupa un nume dat**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 6 din metoda alegere\_optiune() al obiectului UI |
| p = 1 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de intrare un nume |
| Se apeleaza metoda cautare\_clienti\_dupa\_nume al obiectului sv\_client |
| Se vor adauga intr-o lista auxiliara toti clientii gasiti in lista de clienti cu numele dat, ce va fi returnata |
| Se va crea o lista ce va primi lista auxiliara returnata din urma cautarii |
| Daca lista returnata este goala, se va afisa un mesaj special |
| Daca lista nu este goala se va apela metoda afisare\_rezultat\_cautare\_clienti |
| Se va afisa fiecare client din lista creata |

**Scenariu de rulare pentru inchirierea unui film**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 8 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de intrare id\_client, id |
| Se apeleaza metoda inchiriere\_film al obiectului sv\_film |
| Se valideaza id\_client si id; Daca unul din ele nu este validat, se arunca o eroare de tipul IndexError |
| Se verifica daca filmul corespunzator cu id-ul dat este inchiriat sau nu; Daca filmul este inchiriat, se va arunca o eroare de tipul ValueError |
| Se seteaza statusul filmului corespunzator cu id-ul dat la True(este inchiriat), i creste numarul de inchirieri, iar la clientul corespunzator cu id\_client i creste numarul de filme inchiriate |

**Scenariu de rulare pentru returnarea unui film**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 9 din metoda alegere\_optiune() al obiectului UI |
| Se vor da ca date de id |
| Se apeleaza metoda returnare\_film al obiectului sv\_film |
| Se valideaza id; Daca id-ul nu este validat, se arunca o eroare de tipul IndexError |
| Se verifica daca filmul corespunzator cu id-ul dat este inchiriat sau nu; Daca filmul nu este inchiriat, se va arunca o eroare de tipul ValueError |
| Se seteaza statusul filmului corespunzator cu id-ul dat la False(nu este inchiriat) |

**Scenariu de rulare pentru afisarea clientilor ordonati dupa nume**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 10 din metoda alegere\_optiune() al obiectului UI |
| p = 1 din metoda alegere\_optiune() al obiectului UI |
| Se apeleaza metoda ordonare\_clienti\_dupa\_nume al obiectului sv\_client |
| Se vor adauga intr-o lista auxiliara toti clientii din lista de clienti ordonati crescator dupa nume, ce va fi returnata |
| Se va crea o lista ce va primi lista auxiliara returnata din urma cautarii |
| Daca lista returnata este goala, se va afisa un mesaj special |
| Daca lista nu este goala se va apela metoda afisare\_rezultat\_rapoarte\_clienti |
| Se va afisa fiecare client din lista creata |

**Scenariu de rulare pentru afisarea celor mai inchiriate filme**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 10 din metoda alegere\_optiune() al obiectului UI |
| p =31 din metoda alegere\_optiune() al obiectului UI |
| Se apeleaza metoda cele\_mai\_inchiriate\_filme al obiectului sv\_film |
| Se parcurge o data lista de filme pentru a determina numarul maxim de inchirieri |
| Se va creea o lista ce va contine toate filmele ce au numarul de inchirieri cu maximul determina anterior, lista ce va fi returnata |
| Daca lista returnata este goala, se va afisa un mesaj special |
| Daca lista nu este goala se va apela metoda afisare\_ cele\_mai\_inchiriate\_filme |
| Se va afisa fiecare film din lista creata |

**Scenariu de rulare pentru afisarea primilor 30% clienti cu cele mai inchiriate filme**

|  |
| --- |
| Se incarca datele din fisiere in repository\_client si repository\_film |
| Se porneste UI |
| p = 10 din metoda alegere\_optiune() al obiectului UI |
| p = 4 din metoda alegere\_optiune() al obiectului UI |
| Se apeleaza metoda primii\_30lasuta\_clienti\_cu\_filme\_inchiriate al obiectului sv\_client |
| Se apeleaza metoda ordonare\_clienti\_dupa\_nr\_filme\_inchiriate |
| Se va crea o lista ce va primi lista auxiliara returnata din urma ordonarii |
| Se va calcula cat reprezinta 30% din numarul total de clienti |
| In caz ca rezultatul va fi un numar intre 0 si 1, numarul de clienti ce vor fi afisati va fi 1 |
| Se vor pune intr-o lista primii 30% clienti din lista ordonata, folosind calculul facut |
| Se va afisa ultima lista creata, ce contine primii 30% clienti |