



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

DVD Rental

z przedmiotu

Języki programowania obiektowego

Elektronika i Telekomunikacja, III rok

Lidia Wiśniewska

poniedziałek 14:40

prowadzący: Rafał Frączek

09.02.2021r.

1. Opis projektu

Projekt realizuje system ułatwiający obsługę w wypożyczalni DVD. Obsługa programu jest możliwa od strony klienta jak i od strony administratora.

2. Project description

Project is a program that helps in operating a DVD rental. It can be operated by the client and the administrator.

3. Instrukcja użytkownika

Po uruchomieniu programu widoczny jest panel logowania. Użytkownik może wprowadzić swoje dane i zalogować się. W przypadku kiedy logowanie się powiedzie, pojawia się menu główne, które różni się wyświetlanymi opcjami w zależności od typu zalogowanego użytkownika (klient, administrator). Użytkownik może łatwo poruszać się w programie dzięki instrukcjom, które zawiera każde menu.

4. Kompilacja

Wystarczy standardowa kompilacja.

5. Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

- *Account.h*, *Account.cpp* – deklaracja oraz implementacja klasy *Account*,
- *Admin.h*, *Admin.cpp* – deklaracja oraz implementacja klasy *Admin*,
- *Client.h*, *Client.cpp* – deklaracja oraz implementacja klasy *Client*
- *Movie.h*, *Movie.cpp* – deklaracja oraz implementacja klasy *Movie*
- *DVDRent.h*, *DVDRent.cpp* – deklaracja oraz implementacja klasy *DVDRent*
- *functions.h*, *functions.cpp* – deklaracja i instancje zmiennych globalnych w projekcie oraz implementacja różnych przydatnych funkcji.

6. Zależności

brak

7. Opis klas

W projekcie utworzono następujące klasy:

- *Account* – reprezentuje konto w wypożyczalni filmów.
 - `void print()` – wypisuje typ konta,
 - `void showStatus()` – wypisuje dane logowania,
 - settery oraz gettery
- *Admin* – reprezentuje konto administratora w wypożyczalni filmów.
 - Dziedziczy metody z klasy *Account*.
- *Client* – reprezentuje konto klienta w wypożyczalni filmów.

- Dziedziczy metody z klasy `Account`.
- `Movie` – reprezentuje film.
 - `void addComment(string comment)` – dodaje komentarz `comment` do filmu.
 - `void newRate(int value)` – zwiększa ilość ocen o jeden oraz zwiększa sumę ocen o wartość `value`
 - `void print()` – wypisuje dane filmu,
 - `setter`y oraz `getter`y
- `DVD Rental` – reprezentuje zbiór użytkowników oraz filmów i zawiera metody operujące na nich.
 - `int searchUser(string login)` – zwraca miejsce w wektorze, na którym znajduje się użytkownik o nazwie `login`.
 - `void getUsers()` – wczytuje użytkowników z plików do wektora użytkowników
 - `void getMovies()` – wczytuje użytkowników z plików do wektora użytkowników
 - `void printUsers()` – wypisuje użytkowników
 - `string getPasswordByIndex(int idx)` – zwraca hasło użytkownika o indeksie `idx`
 - `bool adminCheck(int idx)` – zwraca `true` kiedy użytkownik o indeksie `idx` jest administratorem, w przeciwnym razie zwraca `false`
 - `void showMovieList(bool admin)` – pokazuje listę filmów odpowiednio zmodyfikowaną dla klienta i administratora
 - `int searchMovie(int id)` – zwraca indeks filmu o `id` `id`
 - `void printDetails(int idx)` – wyświetla detale filmu o indeksie `idx`
 - `void showClientStatus(string login)` – pokazuje wypożyczone filmy użytkownika o loginie `login`
 - `void showClientList()` – wyświetla listę klientów
 - `bool showUserDetail(string login)` – pokazuje szczegóły konta o nazwie `login`
 - `void updateAdmins()` – uaktualnia plik `admins.txt`
 - `void updateClients()` – uaktualnia plik `users.txt`
 - `void updateMovies()` – uaktualnia plik `admins.txt`
 - `void addUser(string login, string passwd)` – dodaje klienta o loginie `login` i hasle `passwd`
 - `void addAdmin(string login, string passwd)` – dodaje administratora o loginie `login` i hasle `passwd`
 - `void addMovie(string title, string director, string genre)` – dodaje film o tytule `title` i gatunku `genre` w reżyserii `director`
 - `void rentMovie(int idx, string user, string date)` – wypożycza film o indeksie
 - `void returnMovie(int idx)` – zwraca wypożyczony film o indeksie `idx`
 - `void rateMovie(int idx, int value)` – dodaje ocenę `value` do filmu o indeksie `idx`

```
void commentMovie(int idx, string comment) – dodaje komentarz comment do filmu o indeksie  
idx
```

8. Zasoby

W projekcie wykorzystywane są następujące pliki zasobów:

- `movies.txt` – plik zawierający dane filmów. Struktura pliku:
 - struktura jednej linii:
ID_filmu|tytuł|reżyser|gatunek|ilość_ocen|suma_ocen|wypożyczający_użytkownik|data_zwrotu|komentarz;komentarz
 - każda kolejna linia reprezentuje dane następnego filmu
- `users.txt` – plik zawierające dane klientów. Struktura pliku:
 - struktura jednej linii: **login|hasło**
 - każda kolejna linia zawiera dane następnego klienta
- `admins.txt` – plik zawierające dane administratorów. Struktura pliku:
 - struktura jednej linii: **login|hasło**
 - każda kolejna linia zawiera dane następnego administratora

9. Dalszy rozwój i ulepszenia

Możliwość rozbudowania projektu o kolejne opcje takie jak usuwanie filmów czy użytkowników oraz sortowanie filmów według zadanej kategorii np. gatunek. Dodanie dodatkowych funkcjonalności takich jak np. tworzenie konta bez udziału administratora.

10. Inne

brak