

1. Opis projektu

DVD Rental to aplikacja konsolowa przeznaczona do ułatwienia kontroli nad systematyzowaniem wypożyczeń filmów DVD. Zawiera w sobie najważniejsze funkcje służące do tego celu. Do takowych należą m.in. aktualizacje asortymentu oraz raporty z transakcji wypożyczeń.

2. Project description

DVD Rental is a console application designed to facilitate control over systematizing rentals. It contains the most important functions for this purpose. There are some of them: assortment's updates, raports of transactions or searching products in bases.

3. Instrukcja użytkownika

Poniżej przedstawiono menu główne aplikacji. W zależności od wprowadzonej liczby, wywołana zostanie odpowiednia funkcja. Wprowadzenie jakichkolwiek innych wartości (również liter i znaków) skutkować będzie zamknięciem aplikacji.

```
Welcome to program for DVD Rental!

What would you like to do?

    1 - Show complete list of owned movies
    2 - Show current state of magazine
    3 - Find movie (complete list)
    4 - Find movie (current state)
    5 - Add new movie
    6 - Delete movie (permanently)

    7 - Show complete list of clients
    8 - Find client in base
    9 - Add new client
   10 - Lend film
   11 - Return movie
   12 - Show Lends
   13 - Show raport
   14 - EXIT

-----
Your choice:
```

Opcja nr 1 i 2 wyświetla odpowiednio stan ogólny asortymentu oraz jego stan bieżący.

Opcja nr 3 i 4 wyszukuje odpowiednio na stanie ogólnym oraz bieżącym konkretny – wpisany przez użytkownika (bez polskich znaków) tytuł filmu i wyświetla wszystkie informacje na jego temat

Opcja nr 5 spowoduje włączenie kreatora nowego filmu DVD. Użytkownik będzie kolejno pytany o tytuł filmu, typ (film animowany, komedia itp.), rok produkcji, imię i nazwisko reżysera (odseparowane spacją), imiona i nazwiska głównych aktorów (odseparowanych między sobą przecinkiem), ilość sztuk.

Opcja nr 6 spowoduje usunięcie filmu z bazy. Poprosi ona najpierw o podanie tytułu, a następnie zatwierdzenie lub anulowanie opcji.

Opcja nr 7 – Wyświetla kompletną listę klientów dostępnych w bazie.

Opcja nr 8 – Wyszukuje w bazie klientów o zadanym imieniu i nazwisku – wprowadzanych kolejno przez użytkownika.

Opcja nr 9 – spowoduje włączenie kreatora nowego klienta – jak w przypadku opcji nr 5 użytkownik zostanie po kolei przez nią przeprowadzony.

Opcja nr 10 – Sprawdza czy dany film jest dostępny w bazie – jeśli tak – użytkownik zostanie poproszony o kolejno: imię klienta, nazwisko klienta oraz numer telefonu. Wyświetlą się wtedy dane o obecnym stanie produktu (zaktualizowane). Do pliku z obecnymi wypożyczeniami zostaną wpisane dane klienta oraz dane filmu, który został wypożyczony.

Opcja nr 10 – Użytkownik zostaje poproszony o wpisanie tytułu filmu oraz danych klienta. Wtedy dane o wypożyczeniu zostaną zaktualizowane – dane klienta znikną z pliku z aktualnymi wypożyczeniami i pojawią się w zakończonych transakcjach oraz ilość produktów w bazie zostanie zaktualizowana.

Opcja nr 11 – umożliwia podgląd wypożyczonych (i jeszcze nie zwróconych) produktów wraz z klientami, które tego dokonały

Opcja nr 12 – umożliwia podgląd raportu o zakończonych transakcjach

Opcja nr 13 – Wyjście z programu.

4. Kompilacja

Projekt skompilowano za pomocą środowiska Visual Studio 2019

5. Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

- *Movies.h*, *Movies.cpp* – deklaracja oraz implementacja klasy *Movies*, deklaracja oraz implementacja różnych przydatnych funkcji,
- *Raports.h*, *Raports.cpp* – deklaracja oraz implementacja klasy *Raports*, deklaracja oraz implementacja różnych przydatnych funkcji,
- *Clients.h*, *Clients.cpp* – deklaracja oraz implementacja klasy *Clients*, deklaracja oraz implementacja różnych przydatnych funkcji. Te dwa pliki stanowią potencjalne rozszerzenie aplikacji konsolowej.

6. Zależności

W projekcie wykorzystano następujące biblioteki:

- *iostream*
- *string*
- *cstring*
- *filesystem*
- *fstream*
- *sstream*
- *cstdlib*

- ctime
- time.h
- conio.h
- cstdio
- chrono

7. Opis klas

W projekcie utworzono następujące klasy:

- **Movies** – klasa reprezentuje film w systemie oraz zawiera publiczne metody i zmienne.
 - `int ID_Generator()` - generuje numer ID. Funkcja przeznaczona do rozszerzenia aplikacji
 - `void read_movies()` - wczytuje obiekty z pliku zawierającego cały asortyment
 - `void write_movies()` - przygotowuje obiekty do wypisania na ekranie
 - `void show_all_movies()` - wypisuje cały dostępny asortyment
 - `void show_movies_in_magazine()` - wypisuje obecny stan magazynu
 - `void find_movie()` - znajduje i wyświetla film o zadanej nazwie lub powiadamia o braku dostępności obiektu w całym asortymencie
 - `void find_movie_in_magazine()` - znajduje i wyświetla film o zadanej nazwie lub powiadamia o braku dostępności obiektu w danym momencie
 - `void new_movie()` - dodaje nowy film o zadanych przez użytkownika danych
 - `void read_state_in_magazine()` - wczytuje obiekty z pliku zawierającego obecny stan magazynu
 - `void update_after_return(string kryterium)` - aktualizuje bazę filmów po zwróceniu
 - `void delete_movie()` - permanentnie usuwa film z asortymentu
- **Raports** – klasa reprezentuje najważniejsze dane potrzebne do przechowywania raportów o wypożyczeniach, zawierają publiczne statyczne metody i zmienne.
 - `void read_raport()` - wczytuje plik zawierający dane o przebiegu transakcji
 - `void read_lending()` - wczytuje dane zawierające dane o bieżących wypożyczeniach
 - `void write_raports()` - przygotowuje obiekty (dotyczące przebiegu transakcji) do wypisania na ekranie
 - `void write_lendings()` - przygotowuje obiekty zawierające bieżące wypożyczenia do wypisania na ekranie
 - `void show_lending_movies()` - wypisuje stan obecny wypożyczeń (dane klienta, dane dotyczące filmu)

- `void show_reports()` - wypisuje zakończone transakcje,
- `void lend_movie()` - wyszukuje film do pożyczenia, oraz aktualizuje listę asortymentu oraz stanu bieżących wypożyczeń
- `void return_movie()` - wyszukuje w obrębie aktywnych wypożyczeń dane na temat konkretnego wypożyczenia oraz aktualizuje listę asortymentu , bieżących wypożyczeń oraz generuje raport.
- `Clients` – klasa reprezentuje najważniejsze dane potrzebne do przechowywania informacji o klientach, zawierają publiczne statyczne metody i zmienne. Klasa przeznaczona do rozszerzenia aplikacji.
 - `void read_client()` - wczytuje plik zawierający dane o klientach
 - `void write_clients()` - przygotowuje obiekty (dotyczące informacji o klientach) do wypisania na ekranie
 - `void show_all_clients()` - wypisuje kompletną listę obecnych klientów
 - `void find_client()` - znajduje i wyświetla informacje na temat klienta o zadanym imieniu i nazwisku lub powiadamia o braku takiej osoby w spisie klientów
 - `void new_movie()` - dodaje nowego klienta o wpisanych przez użytkownika danych do bazy
 - `string cli_ID_Generator()` - generuje numer ID

8. Zasoby

W projekcie wykorzystywane są następujące pliki zasobów:

- `Asortyment.txt` – plik zawierający dane na temat całego asortymentu wypożyczalni. Struktura pliku:
 - pierwsza linia: tytuł filmu;
 - druga linia: typ (animowany, akcji, dokumentalny, itp.),
 - trzecia linia: rok produkcji,
 - czwarta linia: czas trwania filmu,
 - piąta linia: Imię i Nazwisko reżysera lub nazwa wytwórni,
 - szósta linia: Imiona i Nazwiska głównej obsady (separacja spacją lub przecinkiem),
 - siódma linia: numer ID filmów,
 - ósma linia: ilość sztuk dostępnych w obiegu,
 - dziewiąta linia: pusta – pełni rolę separatora pomiędzy kolejnymi rekordami.
 - w następnych liniach dane kolejnych rekordów wypisane w powyższej kolejności.
- `Current_state_of_magazine.txt` – plik zawierający dane na temat dostępnego w danej chwili asortymentu wypożyczalni. Struktura pliku:
 - pierwsza linia: tytuł filmu;
 - druga linia: typ (animowany, akcji, dokumentalny, itp.),

- trzecia linia: rok produkcji,
 - czwarta linia: czas trwania filmu,
 - piąta linia: Imię i Nazwisko reżysera lub nazwa wytwórni,
 - szósta linia: Imiona i Nazwiska głównej obsady (separacja spacją lub przecinkiem),
 - siódma linia: numer ID filmów,
 - ósma linia: ilość sztuk dostępnych w danej chwili,
 - dziewiąta linia: pusta – pełni rolę separatora pomiędzy kolejnymi rekordami.
 - w następnych liniach dane kolejnych rekordów wypisane w powyższej kolejności.
- `Lended_movies.txt` – plik zawierający dane na temat wypożyczonych w tym momencie egzemplarzy. Struktura pliku:
 - pierwsza linia: Imię klienta,
 - druga linia: nazwisko klienta,
 - trzecia linia: numer telefonu klienta,
 - czwarta linia: tytuł wypożyczonego filmu,
 - piąta linia: numer ID filmu,
 - szósta linia: data wypożyczenia filmu,
 - siódma linia: pusta – pełni rolę separatora pomiędzy kolejnymi rekordami.
 - w następnych liniach dane kolejnych rekordów wypisane w powyższej kolejności.
- `Raport.txt` – plik zawierający dane na temat całego asortymentu wypożyczalni. Struktura pliku:
 - pierwsza linia: Imię klienta,
 - druga linia: nazwisko klienta,
 - trzecia linia: numer telefonu klienta,
 - czwarta linia: tytuł wypożyczonego filmu,
 - piąta linia: numer ID filmu,
 - szósta linia: data wypożyczenia filmu,
 - siódma linia: data oddania filmu,
 - ósma linia: pusta – pełni rolę separatora pomiędzy kolejnymi rekordami.
 - w następnych liniach dane kolejnych rekordów wypisane w powyższej kolejności.
- `Clients.txt` – plik zawierający dane na temat całego asortymentu wypożyczalni. Struktura pliku:
 - pierwsza linia: imię klienta;
 - druga linia: nazwisko klienta,
 - trzecia linia: numer PESEL klienta,
 - czwarta linia: numer telefonu klienta,

- piąta linia: data dołączenia,
- szósta linia: pusta – pełni rolę separatora pomiędzy kolejnymi rekordami.
- w następnych liniach dane kolejnych rekordów wypisane w powyższej kolejności.

9. Dalszy rozwój i ulepszenia

Projekt wypożyczalni można w przyszłości poszerzyć o:

- dodanie opcji zapisywania stałych klientów i zbiorczych reportów o danym kliencie
- przedłużanie wypożyczeń
- generację ID zarówno klientów jak i dodawanych filmów
- generację dat

10. Inne

Brak.