

به نام خدا



درس: برنامه سازی پیشرفته

ترم: ۹۹۲

استاد: دکتر مرتضی یوسفی صنعتی

دانشجو: محمد مرادی

ش دانشجویی: ۹۸۱۲۳۵۸۰۳۴

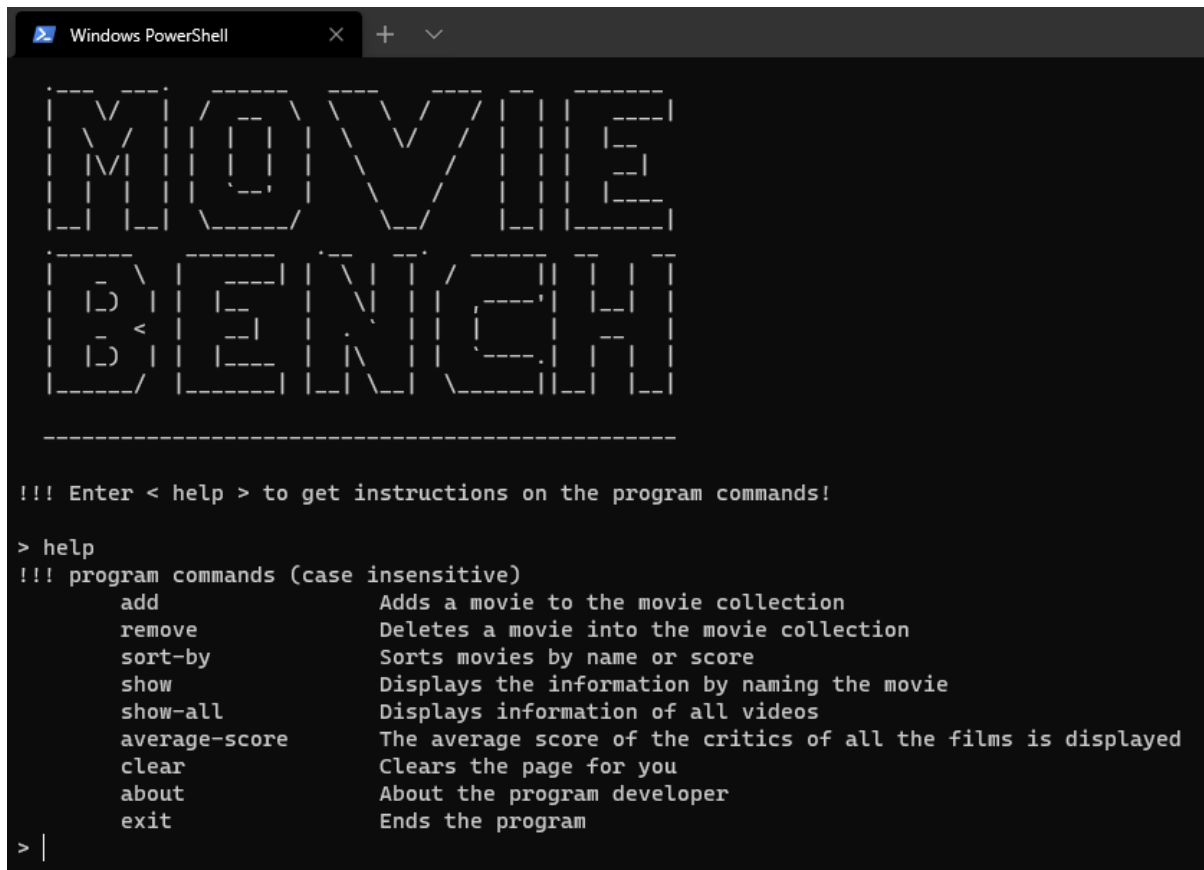
شرح تمرین

هدف : پیاده سازی کلاس
فیلم و روابط مربوط به آن

نام برنامه : Movie Bench
یک کنسول اپلیکیشن که با
دستورات مشخص اطلاعات
فیلم ها را اضافه، نمایش و
حذف می کند.

جزئیات ساخت برنامه

برنامه **Movie Bench** در محیط ادیتور **vscode** درون سیستم عامل ویندوز کد نویسی و با کامپایلر **GCC ۶.۳.۰** کامپایل شده است.



```
Windows PowerShell
MOVIE BENCH

!!! Enter < help > to get instructions on the program commands!

> help
!!! program commands (case insensitive)
    add           Adds a movie to the movie collection
    remove        Deletes a movie into the movie collection
    sort-by       Sorts movies by name or score
    show          Displays the information by naming the movie
    show-all     Displays information of all videos
    average-score The average score of the critics of all the films is displayed
    clear         Clears the page for you
    about         About the program developer
    exit          Ends the program

> |
```

دستورات مجاز برنامه با دستور **help** نمایش داده میشوند.

در این پروژه از سیستم ساخت (build system) ، **Cmake** استفاده شده است.

```
cmake_minimum_required(VERSION 3.0)
project("Movie Bench")
include_directories(include)
add_executable(app
    src/main.cpp
    src/app.cpp
    src/movie.cpp
    src/command.cpp
)
```

روند اجرای برنامه

```
int main()
{
    App app;
    return app.exec();
}
```

برنامه با فراخوانی تابع `exec()` از کلاس `App` اجرا میشود. که این کلاس توابع و داده های کلی برنامه را در خود نگهداری می کند.

```
class App
{
    friend class Command;
public:
    App();
    ~App();
    int exec();
    static void splashScreen();
private:
    Command command;
    Movie * movies;
    static bool status;
    static void clear();
};
```

در داده های خصوصی این کلاس دو شیء از کلاس های `Command` و `Movie` قرار دارد که کلاس `Command` مجموعه توابعی برای اجرای دستورات تعاملی برنامه با کاربر را در خود نگهداری می کند.

کلاس `Movie` هم اطلاعات هر فیلم را در خود نگه می دارد و توابعی برای اعمال تغییرات و پیمایش بر این ساختمان داده را در خود جای داده است. در کلاس `App` اشاره گری از نوع کلاس `Movie` برای تخصیص حافظه پویا در نظر گرفته شده است.

کلاس Command

تابع `exec0` از شیء `app` ساخته شده در تابع `main` پس از نمایش صفحه اول برنامه، تابع `getCommand` از شیء خصوصی `command` را فرا میخواند تا برنامه آماده دریافت دستورات از کاربر شود.

```
void Command::getCommand(Movie * movies)
{
    this->movies = movies;
}
```

اطلاعات کلی فیلم ها که دورن شیء `movies` (آرایه پویا فیلم ها) ذخیره شده است به تابع `getCommand` ارسال میشود

تا اشاره گری از نوع `Movie *` به آن اشاره کند و کلاس `Command` هم به اطلاعات کلی فیلم ها دسترسی داشته باشد.

```
class Command
{
public:
    void getCommand(Movie *);
    void add();
    void remove();
    void sortBy();
    void show() const;
    void showAll() const;
    void averageScore() const;
    void help() const;
    void clear() const;
    void about() const;
private:
    std::string inputText;
    std::string command; //The first word separated from inputText
    Movie * movies;
    std::string commandSeparator();
    void swapMovies(Movie &, Movie &);
    void operator<<(int); //shift back 1 unit movies that are after a particular movie
};
```

هر کدام از توابع عمومی این کلاس با توجه به دستورات ورودی کاربر در برنامه فراخوانده می شوند.

همچنین عملگر `<<` برای یک واحد شیفت دادن به عقب کل اشیا فیلم های درون آرایه پویا سربار گذاری شده است.

ارسال

به علت از دسترس خارج شدن سامانه **cw۱** دانشگاه، فایل های تمرین از طریق رایانامه برای دستیار آموزش ارسال شد.

machitgarha@chmail.ir

همچنین فایل ها درون مخزن گیت هاب به نشانی زیر موجود است.

<https://github.com/itismoradi/Movie-Bench>