

Студентите с четни факултетни номера решават 1та задача от домашното, а тези с нечетни 2та. Допълнително изискване **само** за студентите от бта група е разделянето на програмния код в header и source файлове.

### 1 Задача за кинозалони

Да се напише система за управление на кинозалони. Регистриран потребител в системата да може да прави следните операции:

- резервация на място/места
- показване на програма на киното
- показване на часовете в които се прожектира даден филм
- показване на прожектираните филми в даден час
- показване на всички резервации

За да реализирате системата имплементирайте следните класове:

**Seat**-клас представлява място в зала със следните член данни:

- seatNumber
- rowNumber
- isReserved

и предоставя следния метод:

- reserveSeat() – резервира мястото

**Hall** – клас представлява зала за прожекции, със следните член данни:

- seats – масив от места в залата

класа да предоставя следните методи:

- reserveSeats(int\* seatNumbers, int n) – резервира n места в залата с номерата от масива
- getFreeSeats() – връща броя на свободните места в залата

**Projection** – представя прожекцията в киното със следните член данни:

- hall – в коя зала се прожектира филма
- movie – кой филм се прожектира
- time – в колко часа е прожекцията

**Reservation** – клас представлява резервация за прожекцията със следните член данни:

- movieName – за кой филм е резервацията
- numSeats – за колко хора е резервацията
- time – от колко часа е резервацията
- username – име на потребителя който прави резервацията

Когато се конструира обект от тип Reservation да се проверяват следните неща:

- Има ли прожекция за този филм в този час
- Има ли искания брой места в залата

Резервациите и прожекциите да се пазят във файл. При стартиране на програмата да се зареждат отново от файл.

По-горе са описани минималните изисквания към класовете. Ако има нужда може да добавяте методи и член данни стига да спазвате принципите на ООП.

## 2 Система, съхраняваща информация за филми

Реализираната от вас система има два вида потребители:

1. Администратор
  - Той е единствен и има уникално име Admin
  - Единствено админът има право да добавя нови филми към колекцията. Филм може да бъде добавен, ако вече не е част от колекцията с филми.
2. Обикновен потребител
  - Има потребителско име и парола, като потребителското име на двама user-и не може да съвпада
  - Може да дава рейтинг на даден
  - Може да пише ревю на даден филм
  - Може да търси филми по рейтинг/актьори/ година/ режисьор/ жанр
  - Може да търси сортираната по рейтинг колекция от филми
  - Да търси сортираната последователност от филми от определена година
  - По зададени актьор и режисьор да търси всички филми, в които са участвали и двамата
  - Да търси всички римейкове на даден филм
  - Да търси главните актьори или режисьор на даден филм
  - Да търси всички ревюта за даден филм

Само регистрирани потребители могат да използват системата. За целта при стартиране на програмата се прави опит за логване чрез потребителско име и парола. Ако логването е успешно се показва меню със съответните опции, които има конкретният потребител(в зависимост от това дали е администратор или обикновен потребител). При 3 неуспешни поредни опита за логване, програмата приключва своята работа. За да реализирате системата, дефинирайте следните класове.

Клас **User** със следните член данни:

- Потребителско име
- Парола
- Каквито допълнителни член данни ви потрябват, за да реализирате поведнието на администратора и обикновения потребител

Клас **MovieStaff**, описващ участник във филми, със следните член данни:

- Име
- Дата на раждане
- Филми, в които е взел участие
- Роля(дали е актьор, режисьор, продуцент и т.н.). Приемаме, че даден човек може да бъде само актьор или режисьор, или продуцент.

Клас **Movie** със следните член данни:

- Име
- Режисьор
- Година
- Рейтинг
- Дали е римейк, ако е да пази указател към оригинала
- Главни актьори
- Жанр

Дефинирайте клас **Dispatcher**, който да реализира описаното по-горе поведение, като борави с 3те описани класа. Данните за потребителите, за филмите и за участниците във филмите да се пазят във файлове. Спазвайте принципите на ООП!!!