

**Общее описание:**

Мы продолжаем совершенствовать модель кинотеатра. Во втором домашнем задании вам необходимо будет имплементировать основы объектно-ориентированного подхода и создать модель, учитывающую большее количество деталей.

Итак, в нашем кинотеатре показывают теперь сразу несколько фильмов, их показывают в нескольких залах, а также в разные даты и время. Естественно, для всех этих сеансов должна быть доступна онлайн-покупка билетов. Именно это и предстоит вам реализовать в рамках улучшенного консольного приложения для онлайн-покупки билетов.

**Входные данные:**

В начале программы, при запуске, в консоль вводятся следующие исходные данные (позднее, все они могут быть обновлены и дополнены администратором):

- Количество фильмов, показываемых в ближайшее время в нашем кинотеатре. Про каждый из этих фильмов мы, как минимум, хотим знать его название и возрастной рейтинг (напомним, в России это «0+», «6+», «12+», «16+» и «18+»). Подумайте, какая еще информация про фильм могла бы быть полезной в этом контексте.
- Для каждого фильма должен присутствовать набор залов, в котором данный фильм демонстрируется. Вы можете просто пронумеровать залы, а можете воспользоваться более хитрыми обозначениями, как, например, сделано в достаточно популярном в Москве кинотеатре – ОККО Москва-Сити (обратите внимание на первый столбец в таблице ниже).





















Название	Мест	Тип	Кресла	Звук	Изображение
№1 Лос Анджелес	177	Обычный	Стандарт		
№2 Нью Йорк	230	Обычный	Стандарт		
№3 Сидней	208	Обычный	Стандарт		
№4 Шанхай	231	Обычный	Стандарт		
№5 Гонконг Premium	73	VIP	VIP		
№6 Лондон	232	Обычный	Стандарт		
№7 Дубай	232	Обычный	Стандарт		
№8 Сеул	232	Обычный	Стандарт		
№9 Токио	177	Обычный	Стандарт		
№10 IMAX Sapphire	78	Обычный	Стандарт		 IMAX

Рисунок 1. Пример обозначения залов

При этом вы также можете добавить информацию, например, о типе зала. Это может быть полезно и реалистично с точки зрения влияния типа зала (в качестве примера: «Обычный», «Премиум», «Luxe», «VIP», «IMAX», другие) на значения количества мест (размер зала), стоимости билетов и т.д., которые вы также должны хранить для каждого зала. Вы можете добавить и любую другую валидную информацию о каждом зале.

- Для упрощения нашей модели, мы пока избегаем типы связей «Многие ко многим» и предполагаем, что каждый зал используется для показов только какого-то одного фильма. В будущем мы улучшим модель!
- Каждый фильм в каждом зале показывается не один раз, а множество раз в разные даты и время. Вам необходимо предусмотреть возможность многочисленных показов одного и того же фильма в разные моменты. Разумеется, для каждого сеанса должен быть свой уникальный шаблон бронирования (если место 1-1 выкуплено на фильм «Матрица: Воскрешение» на 21:00 29 января, это место должно все равно быть доступно на тот же фильм в тот же зал на 21:00 30 января, так как это другой сеанс в другое время – хоть и того же фильма в том же зале).

Ниже представлен начальный шаблон примера, как входные данные могут быть внесены в консоль и в каком формате могут быть считаны. Вы можете придумать любой другой формат, в котором вам было бы удобнее это реализовать.

```
Введите количество фильмов:
5

В следующих 5 строках введите названия этих фильмов:
King's Man: Начало
Человек-Паук: Нет пути домой
Крик
Зверопой 2
Дом Gucci

В следующих 5 строках введите возрастные рейтинги для этих фильмов:
18+
12+
18+
6+
18+

Теперь укажем информацию о фильме King's Man: Начало
Введите количество залов для этого фильма:
4

В следующих 4 строках введите названия залов для фильма King's Man: Начало:
. . . .
```

### 1. Задание 1. Считывание и хранение данных с помощью ООП (3 балла):

Реализуйте программную модель считывания и хранения всех вводимых данных из предыдущего раздела «Входные данные». Модель обязательно должна содержать в себе все логически соответствующие предметной области классы, верно следовать концепциям и правилам имплементации ООП, обладать четкой и понятной структурой.

Указание: вам потребуется не менее двух, а, вероятнее, три или четыре, различных класса (в зависимости от проектировки). Подумайте над предметной областью. При необходимости визуализации, нарисуйте схему взаимодействия объектов разных типов между собой.

## 2. Задание 2. Пользовательский интерфейс, покупка билетов (2 балла):

В рамках модели пользовательского интерфейса вы должны иметь возможность имплементировать все функции пользователей из первого домашнего задания (с учетом новой схемы хранения данных), а также следующие новые функции:

- Пополнение баланса пользователя. В рамках одной пользовательской сессии, пользователь вводит в самом начале из консоли свой текущий баланс, а позднее, в течение сессии, имеет возможность его пополнять.
- Возможность покупки билета на любой фильм, в любой зал, на любое время в любой сеанс. Внимание: не давайте пользователю купить билет и не отображайте в списке доступных сеансов, сеансы, которые к моменту пользовательской сессии остались во времени в прошлом. Например, не показывайте сеанс на 21:00 29 января, если сейчас 22:00 29 января (программа берет эти значения по текущей дате и времени компьютера).
- Сохраняйте данные обо всех купленных пользователем билетах в ходе его пользовательской сессии. Будьте готовы по запросу пользователя показать ему список купленных им билетов, отсортированных по дате сеанса.
- В самом начале пользовательской сессии (прямо перед или непосредственно после запроса баланса) считайте данные о ФИО пользователя. После покупки билетов пользователем, присвойте каждому купленному билету метку с ФИО этого пользователя. Это понадобится для аналитики администратора.

Примечание: для простоты вы можете пока не создавать класс пользователя и ограничиться присвоением строковой метки с фамилией-именем для каждого купленного билета. В будущем, мы будем усложнять данную модель.

### **3. Задание 3. Внутренний интерфейс, аналитическая система (2 балла):**

В рамках модели интерфейса администратора вы должны иметь возможность имплементировать все функции администратора из первого домашнего задания (с учетом новой схемы хранения данных), а также следующие новые функции:

- Возможность добавить/удалить/изменить любые данные о кинотеатре (список фильмов, информацию о фильмах, залы, информацию о залах, сеансы). Не подлежат изменению лишь размеры залов: они такие, как их задают при инициализации.
- Расширенная аналитика по продажам. Возможность выводить информацию о загрузженности залов (числе проданных мест, числе свободных мест), общей выручке по проданным билетам со следующими фильтрами (одним или несколько):
  - на один конкретный фильм
  - в одном конкретном зале
  - на одном конкретном сеансе
  - за один конкретный промежуток времени (с момента  $x$  по момент  $y$ )
  - в какой-то период дня (в любой день указанного периода со времени  $x$  до времени  $y$ )
  - за весь период по всем фильмам
  - на фильмы с одним конкретным возрастным рейтингом

Вы можете также придумать и другие фильтры аналитики по данным!

- Основы системы клиентской аналитики. Возможность выводить информацию (ФИО) о наиболее лояльных клиентах: клиентах, купивших наибольшее количество билетов; клиентах, купивших билеты на наибольшее количество разных сеансов; клиентах, потративших наибольшую сумму денег (Топ-1, Топ-3, Топ-5).

#### **4. Дополнительные баллы (на оценку отлично):**

Дополнительные баллы за это ДЗ можно получить по той же схеме, что и в ДЗ-1. За всеми деталями обращайтесь к пункту 4 («Дополнительные баллы») задания к ДЗ-1.

Дополнительные задания на дополнительные баллы для ДЗ-2:

- **(1.5 балла)** Имплементация дополнительной логики объектно-ориентированного подхода. Внимание: логика должна соответствовать по уровню сложности реализованному в заданиях 2 и 3 функционалу. В случае, недостаточной сложности/оригинальности вашего функционала, данное задание не засчитывается.
- **(1.5 балла)** Программная обработка исключений. Вы должны убедиться, что ни при каком вводе пользователя с клавиатуры в консоль не происходит аварийного завершения программы. Предусмотрите все варианты, где пользователь может что-то неверно ввести.
- **(1.5 балла)** Файловый ввод-вывод. Осуществляйте чтение всех исходных входных данных из локального файла, а не из консоли. По завершении цикла программы осуществляйте запись (измененных) данных обратно в файл.

#### **5. Штрафная политика:**

За несамостоятельность решения (невозможность защитить написанный код) студенту выставляется 0 баллов.

За любой плагиат, копирование между несколькими работами, за все работы выставляется 0 баллов.

Кроме того, предусмотрено понижение баллов в случаях, если:

- Код написан в плохом стиле: не соблюдаются отступы, не соблюдается стиль названия переменных (например все переменные называются буквами a, b, c), другие осложняющие понимание кода ситуации.

- Программа не работает циклично или некорректно осуществляется выход из цикла работы программы.
- Код написан единым блоком в методе Main, не происходит смыслового разделения и дискретизации кода (методы не используются совсем или же частично используются, но основная логика все равно пишется в Main).
- Программа реализует не весь функционал или реализует его не для всех входных данных. Происходят ошибки при определенных запросах. Программа неверно подсчитывает / хранит те или иные данные. Прочие смысловые ошибки.
- Концептуальные ошибки при имплементации и/или использовании концепций объектно-ориентированного программирования. Непонимание этих концепций или ошибочное понимание. Бесполезное (не выполняющее прямого назначения) использование классов в программе.

#### **Дедлайн сдачи работы:**

Сдача работы осуществляется до **23:59 20 февраля**.

Каждые сутки просрочки равносильны штрафу в 2 балла, но получить за ДЗ меньше 0 баллов нельзя. Таким образом, через 5 суток сдавать ДЗ бессмысленно.