

Программирование

Домашнее задание 1

Модель кинотеатра

Общее описание:

В первом домашнем задании по нашему курсу вам необходимо будет разработать простую модель бронирования и покупки билетов в кинотеатре. Разумеется, поскольку мы пока не касались вопросов визуального интерфейса, сделать вам все это нужно будет в рамках консольного приложения и простого линейного ввода-вывода.

Практически все современные кинотеатры обладают онлайн-платформой (сайтом / приложением), на которой вы можете в электронном виде выбрать места и купить билеты на понравившийся вам фильм. Модель такой платформы и предстоит вам реализовать.

Входные данные:

В упрощенной модели, которую мы рассматриваем в рамках первого ДЗ, предполагается, что в кинотеатре идет всего 1 фильм в день. Как следствие, вам необходимо хранить данные о размере зала (будем считать, что это прямоугольник), стоимости билета на каждое из мест, а также данные о заполняемости зала по мере покупки билетов на сеанс. Изначально, разумеется, все места свободны (не выкуплены).

Таким образом, исходно мы получаем следующий набор данных:

- В первой строке два числа: N и M , где N – количество рядов в зале, а M – количество мест в одном ряду
- В последующих N строках вводится матрица размера $M \times N$, где каждое число — это стоимость соответствующего места в зале.

Пример ввода данных вы можете видеть ниже:

```
Размер зала:
4 5

Стоимости билетов в зале:
250 300 400 300 250
300 500 800 500 300
300 400 500 400 300
250 350 500 350 250
```

1. Задание 1. Хранение данных (1 балл):

Реализуйте программно возможность считывать и хранить данные о стоимости билетов в зале для любого вводимого зала.

2. Задание 2. Пользовательский интерфейс, покупка билетов (3 балла):

В рамках модели пользовательского интерфейса вы должны:

- Запросить данные о балансе пользователя (количество денег на счете пользователя – для простоты считаем, что сервис может обращаться к таким данным).
- Вывести пользователю список доступных (некупленных) мест в кинотеатре. Список выводится в виде матрицы с описанием обозначений, используемых для указания купленных и некупленных мест (см. пример ниже)

```
Доступные места (0 - место доступно; x - место выкуплено):
0 0 0 0 0
0 x 0 0 0
0 0 x x 0
0 0 0 0 0
```

- Попросить пользователя выбрать места, которые он хочет выкупить. Проверить, что места свободны. Проверить, что у пользователя достаточно средств для покупки билетов на выбранные места. Осуществить транзакцию с внесением всех необходимых изменений в данные

3. Задание 3. Внутренний интерфейс, аналитическая система, внесение изменений в цены (2,5 балла):

В рамках модели внутреннего интерфейса (интерфейса, доступного администратору кинотеатра) вы должны:

- Осуществлять вход во внутренний интерфейс по паролю (для простоты и на первый раз пароль можно захардкодить – прописать его вручную в коде программы)
- Иметь возможность подсчитать количество проданных мест, количество свободных мест, текущую выручку по всем проданным билетам на сеанс
- Иметь возможность изменить цены на билеты (не потеряв при этом информацию о том, какая выручка по проданным билетам была, исходя из старых цен).

4. Дополнительные баллы (на оценку отлично):

Дополнительные баллы за это ДЗ можно получить, только в случае полного выполнения основного блока (задания 1 – 3).

Дополнительные баллы можно получить за любые из нижеприведенных заданий, но, в отличие от основной части, каждое задание оценивается бинарно: либо дополнительное задание полностью выполнено (и тогда за него выставляется балл, указанный в скобках), либо задание считается невыполненным, и за него ничего не выставляется.

Вы можете выполнить любое количество дополнительных заданий, но получить оценку выше 10 нельзя.

Дополнительные задания:

- **(0.5 балла)** Оригинальная визуализация в рамках консоли. Изучите возможности консольного приложения и способы кастомизации консоли и примените их в рамках вашей программы.
- **(1.5 балла)** Имплементация любой дополнительной логики. Придумайте что-то интересное и своё с точки зрения реализации дополнительных функций в рамках модели кинотеатра.

- **(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. Например, если пользователь в строке, где нужно ввести размер зала, вводит что-то не то, попросите его ввести размер еще раз.
- **(1.5 балла)** Разумная имплементация ООП. Внедрите в вашу программу классы и/или другие компоненты объектно-ориентированного подхода так, чтобы это упрощало и делало более удобной вашу программу. Простое создание неиспользуемых или бесполезных с точки зрения эффективности программы классов – не засчитывается

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.

Дедлайн сдачи работы:

Сдача работы осуществляется до **23:59 6 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.