

Техническое задание

Создание программы на языке Java «Аналитика для формулы расчёта процента от стоимости билета на концерт при его возврате»

Оглавление

1. Введение.....	3
1.1. Назначение системы	3
2. Обзор системы	3
2.1. Содержание системы	3
2.2. Функции системы.....	3
2.3. Характеристики пользователей	3
3. Системные требования	4
3.1. Функциональные требования	4
3.2. Требования к юзабилити	4
3.3. Интерфейс системы	5
3.4. Требования к безопасности.....	5
3.4. Управление информацией.....	5
3.4. Политика и правила	5
4. Тестирование и проверка (список необходимых тестов, которые отражают зеркально раздел 3)	6

1. Введение

1.1. Назначение системы

В случае отказа от посещения мероприятия по причинам, не связанным с отменой/переносом мероприятия пользователь имеет возможность вернуть приобретенный билет.

Система должна подсчитать количество средств, которое необходимо вернуть пользователю в зависимости от сроков возврата билета.

2. Обзор системы

2.1. Содержание системы

Билет пользователя должен быть создан как отдельный класс. Обработка стоимости возврата должна происходить отдельно и включать в себя метод для определения типа билета и непосредственно подсчета стоимости возврата.

2.2. Функции системы

Система должна хранить данные о билетах. После ввода данных пользователя их необходимо обработать, изменить и вывести результат.

После вычисления суммы возврата пользователю нужно сообщить, какое количество средств он получит с точностью до сотых.

2.3. Характеристики пользователей

Программа рассчитана на все слои общества. Стоит организовать максимально понятный для пользователя интерфейс.

3. Системные требования

3.1. Функциональные требования

Программа должна иметь защиту от ошибок и работать без внешнего вмешательства. Необходимо проработать всевозможные варианты действий пользователей и предусмотреть ввод неверных данных.

Обрабатывать дату и время стоит в формате “уууу-ММ-dd НН:mm”.

После введения всех данных нужно определить как сам тип билета, так и в какой промежуток времени был совершен возврат:

- Раннее 1 месяца – 100% суммы
- От 10 дней до 1 месяца – 90% суммы
- От 7 до 3 дней – 80% суммы
- От 3 до 2 дней – 70% суммы
- За 1 день – 60% суммы
- В день окончания действия билет (ДОДБ) от 24 до 12 часов - 50% суммы
- В ДОДБ от 12 до 5 часов – 40% суммы
- В ДОДБ от 5 до 3 часов – 35% суммы
- В ДОДБ от 3 до 2 часов – 30% суммы
- В ДОДБ за 1 час – 20% суммы
- В течении двух часов после ДОДБ – 10% суммы

3.2. Требования к юзабилити

Для работы программы пользователь должен ввести данные своего билета. Спустя некоторое время получить ответ в виде суммы возврата.

3.3. Интерфейс системы

Не требуется разработка графического интерфейса. Пользователь взаимодействует напрямую с консолью и файлами Java.

3.4. Требования к безопасности

Программа не должна нести вред здоровью пользователей и окружающей среды, вызывать катаклизмы и роботизированные восстания.

3.4. Управление информацией

Класс билет должен содержать в себе такие поля как: тип билета, описание, имя владельца, стоимость, дату и время совершения покупки, дату и время окончания действия билета.

Типы билетов должны храниться как перечисление. Для каждого типа билета указан свой коэффициент стоимости билета. Коэффициент учитывается после определения промежутка возврата билета:

- Билет на концерт – 0.7
- Билет в кинотеатр – 0.95
- Билет на самолет – 0.6
- Билет на автобус – 0.9
- Билет на поезд – 0.8

3.4. Политика и правила

Данная программа не должна нести в себе оскорбляющий характер или же каким-либо задевающий чувства, принципы и веру пользователей. Не предназначена для использования в коммерческих целях.

4. Тестирование и проверка (список необходимых тестов, которые отражают зеркально раздел 3)

Необходимо составить тест-кейсы в виде автотестов JUnit для функции расчёта процента от стоимости билета при возврате. Потребуется не менее 5 тестов с позитивным сценарием, не менее 5 тестов с негативным сценарием и 5 ручных тестов.

Нужно составить тест-сюит, состоящий из одного или нескольких тест-кейсов. Оформить их в виде текстового документа.