##### 

##### @startuml

##### left to right direction

##### actor Музыкант

##### actor Лейбл

##### actor Перекупщик

##### actor Покупатель

##### actor Администратор

##### rectangle "Система продажи музыкальной продукции" {

##### Музыкант -- (Регистрация)

##### Музыкант -- (Создание товара)

##### Музыкант -- (Управление товарами)

##### Лейбл -- (Регистрация)

##### Лейбл -- (Добавление товаров от авторов)

##### Лейбл -- (Управление заказами)

##### Перекупщик -- (Регистрация)

##### Перекупщик -- (Выбор товаров для перепродажи)

##### Перекупщик -- (Установка цены)

##### Покупатель -- (Поиск товаров)

##### Покупатель -- (Просмотр товаров)

##### Покупатель -- (Оформление заказа)

##### Покупатель -- (Подтверждение доставки)

##### Администратор -- (Модерация контента)

##### Администратор -- (Управление пользователями)

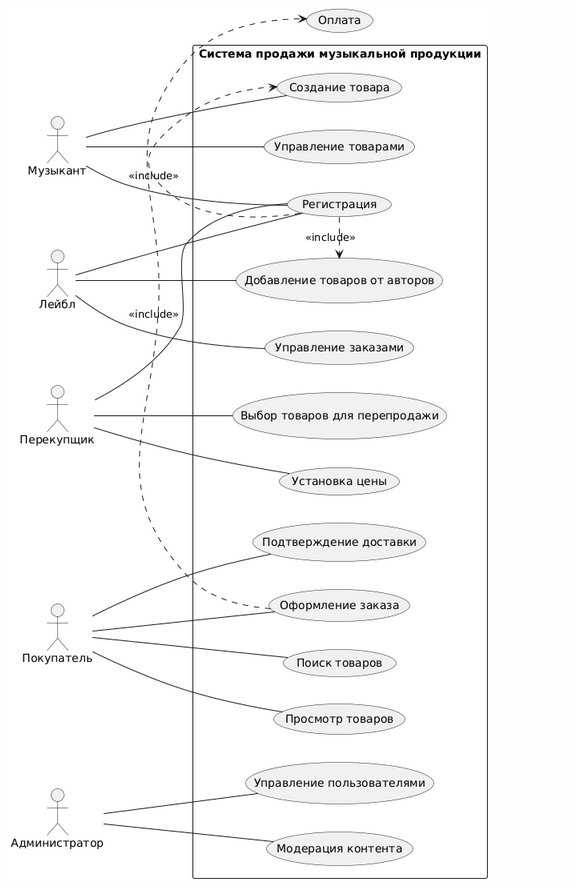
##### }

##### (Регистрация) .> (Создание товара) : <<include>>

##### (Регистрация) .> (Добавление товаров от авторов) : <<include>>

##### (Оформление заказа) .> (Оплата) : <<include>>

##### @enduml



##### 

##### @startuml

##### class User {

##### +int id

##### +str username

##### +str email

##### +str password

##### +str role

##### +bool is\_reseller

##### +register()

##### +login()

##### }

##### class Product {

##### +int id

##### +str name

##### +str description

##### +float price

##### +str genre

##### +str seller\_type

##### +bool is\_original

##### +upload()

##### +edit()

##### }

##### class ResellerProduct {

##### +int id

##### +Product product

##### +User reseller

##### +float reseller\_price

##### }

##### class Order {

##### +int id

##### +int user\_id

##### +int product\_id

##### +str status

##### +str delivery\_status

##### +date created\_at

##### +place\_order()

##### +confirm\_payment()

##### }

##### class Payment {

##### +int id

##### +int order\_id

##### +float amount

##### +process\_payment()

##### +refund()

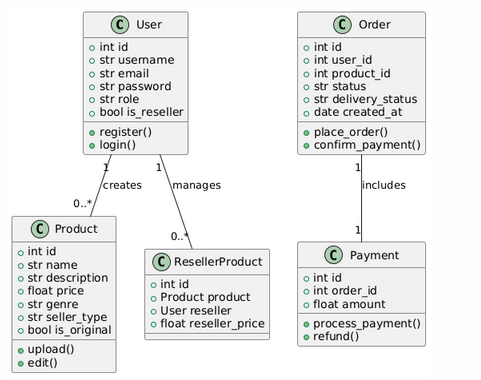
##### }

##### User "1" -- "0..\*" Product : creates

##### User "1" -- "0..\*" ResellerProduct : manages

##### Order "1" -- "1" Payment : includes

##### @enduml



#### ****C. Sequence Diagram (Диаграмма последовательности)****

**Назначение:** Показывает шаги процесса оформления заказа и оплаты.

**Код PlantUML:**

##### @startuml

##### actor Покупатель

##### participant "Система" as System

##### participant "Корзина" as Cart

##### participant "Платежный шлюз" as PaymentGateway

##### participant "Продавец" as Seller

##### Покупатель -> System : Выбрать товар

##### System -> Cart : Добавить товар

##### Покупатель -> System : Оформить заказ

##### System -> PaymentGateway : Обработать оплату

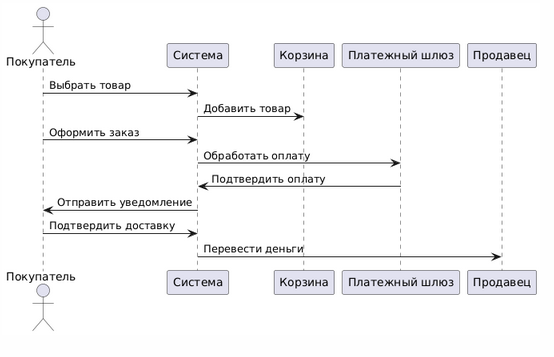
##### PaymentGateway -> System : Подтвердить оплату

##### System -> Покупатель : Отправить уведомление

##### Покупатель -> System : Подтвердить доставку

##### System -> Seller : Перевести деньги

##### @enduml



### ***1. Use Case Diagram (Диаграмма вариантов использования)***

#### ****Что это такое?****

* Это диаграмма, которая показывает, как пользователи взаимодействуют с системой.
* Она описывает основные функции системы через "варианты использования" (use cases).

#### ****Зачем нужна эта схема?****

* Помогает понять, какие действия могут выполнять разные типы пользователей (музыканты, лейблы, перекупщики, покупатели, администраторы).
* Показывает, какие функции доступны для каждой роли.

#### ****Описание схемы:****

1. **Актеры (пользователи):** 
   * **Музыкант:** Может регистрироваться, создавать товары (музыка, мерч), управлять своими товарами.
   * **Лейбл:** Может регистрироваться, добавлять товары от разных авторов, управлять заказами.
   * **Перекупщик:** Может регистрироваться, выбирать товары из каталога для перепродажи, устанавливать свою цену.
   * **Покупатель:** Может искать товары, просматривать их, оформлять заказы, подтверждать доставку.
   * **Администратор:** Может модерировать контент и управлять пользователями.
2. **Варианты использования:** 
   * **Регистрация/Авторизация:** Все пользователи могут зарегистрироваться и войти в систему.
   * **Создание товара:** Доступно только музыкантам и лейблам.
   * **Выбор товаров для перепродажи:** Доступно только перекупщикам.
   * **Оформление заказа:** Покупатель выбирает товар, добавляет его в корзину и оформляет заказ.
   * **Подтверждение доставки:** Покупатель подтверждает, что получил товар.
   * **Модерация контента:** Администратор проверяет товары и управляет пользователями.
3. **Связи между вариантами использования:** 
   * Например, "Регистрация" включает в себя возможность "Создания товара" для музыкантов и лейблов.
   * "Оформление заказа" включает в себя "Оплату".

#### ****Как объяснить учителю?****

* "Эта диаграмма показывает, кто может что делать в системе. Например, музыканты могут создавать товары, а перекупщики — только перепродавать их. Покупатели могут оформлять заказы, а администраторы следят за порядком."

***2. Class Diagram (Диаграмма классов)***

#### ****Что это такое?****

* Это диаграмма, которая показывает структуру системы на уровне классов.
* Классы — это основные объекты системы (например, пользователи, товары, заказы), а связи между ними показывают, как они взаимодействуют.

#### ****Зачем нужна эта схема?****

* Помогает понять, как организованы данные в системе.
* Показывает, какие поля и методы есть у каждого класса.
* Упрощает процесс написания кода.

#### ****Описание схемы:****

1. **Основные классы:** 
   * **User (Пользователь):** 
     + Поля: id, username, email, password, role (роль пользователя), is\_reseller (является ли перекупщиком).
     + Методы: register() (регистрация), login() (вход в систему).
   * **Product (Товар):** 
     + Поля: id, name (название), description (описание), price (цена), genre (жанр), seller\_type (тип продавца), is\_original (является ли товар оригинальным).
     + Методы: upload() (загрузка товара), edit() (редактирование).
   * **ResellerProduct (Товары перекупщика):** 
     + Поля: product (ссылка на оригинальный товар), reseller (ссылка на перекупщика), reseller\_price (цена перекупщика).
   * **Order (Заказ):** 
     + Поля: id, user\_id (покупатель), product\_id (товар), status (статус заказа), delivery\_status (статус доставки).
     + Методы: place\_order() (оформление заказа), confirm\_payment() (подтверждение оплаты).
   * **Payment (Платеж):** 
     + Поля: id, order\_id (заказ), amount (сумма), payment\_status (статус платежа).
     + Методы: process\_payment() (обработка платежа), refund() (возврат денег).
2. **Связи между классами:** 
   * Один пользователь (User) может создать много товаров (Product).
   * Один заказ (Order) связан с одним платежом (Payment).
   * Перекупщик (User) может управлять несколькими товарами (ResellerProduct).

#### ****Как объяснить учителю?****

* "Эта диаграмма показывает, как устроены данные в системе. Например, у нас есть класс User (пользователь), который может быть музыкантом, лейблом или перекупщиком. У каждого пользователя есть свои товары (Product). Если пользователь — перекупщик, он может создавать ResellerProduct с собственной ценой. Также есть классы Order (заказ) и Payment (платеж), которые связаны между собой."

### ***3. Sequence Diagram (Диаграмма последовательности)***

#### ****Что это такое?****

* Это диаграмма, которая показывает шаги выполнения конкретного действия в системе.
* Она описывает взаимодействие между пользователями и компонентами системы во времени.

#### ****Зачем нужна эта схема?****

* Помогает понять, как происходит процесс оформления заказа и оплаты.
* Показывает последовательность действий, чтобы избежать ошибок при разработке.

#### ****Описание схемы:****

1. **Участники:** 
   * **Покупатель:** Инициирует процесс оформления заказа.
   * **Система:** Обрабатывает запросы покупателя.
   * **Корзина:** Хранит выбранные товары.
   * **Платежный шлюз:** Обрабатывает платеж.
   * **Продавец:** Получает деньги после подтверждения доставки.
2. **Шаги процесса:** 
   * Покупатель выбирает товар и добавляет его в корзину.
   * Покупатель оформляет заказ.
   * Система отправляет запрос на обработку платежа в платежный шлюз.
   * Платежный шлюз подтверждает оплату.
   * Система уведомляет покупателя об успешной оплате.
   * Покупатель подтверждает доставку товара.
   * Система переводит деньги продавцу.

#### ****Как объяснить учителю?****

* "Эта диаграмма показывает, как происходит процесс оформления заказа. Например, покупатель выбирает товар, добавляет его в корзину и оформляет заказ. После этого система обрабатывает платеж через платежный шлюз. Когда покупатель подтверждает доставку, деньги переводятся продавцу. Это помогает нам понять, как работают все компоненты системы."

### **Итог**

1. **Use Case Diagram:** 
   * Показывает, кто может что делать в системе.
   * Пример: "Музыканты могут создавать товары, а перекупщики — только перепродавать."
2. **Class Diagram:** 
   * Показывает структуру данных (классы и их связи).
   * Пример: "Класс User связан с классом Product, потому что пользователи создают товары."
3. **Sequence Diagram:** 
   * Показывает последовательность действий при оформлении заказа.
   * Пример: "Покупатель выбирает товар, оплачивает его, подтверждает доставку, и деньги переводятся продавцу."