Давайте спроектируем систему для турагентства с учетом указанных функций: **заказчик выбирает и оплачивает тур**, а **турагент определяет тур как "горящий" и устанавливает скидки для постоянных клиентов**. На основе этого создадим UML-диаграммы для иллюстрации структуры и поведения системы.

**Типы и их взаимодействия**

**Основные сущности системы:**

1. **Пользователь**:
   * Роли:
     + **Заказчик**: выбирает тур, бронирует его и оплачивает.
     + **Турагент**: создает туры, устанавливает скидки и помечает тур как "горящий".
2. **Тур**:
   * Типы туров: отдых, экскурсия, шоппинг.
   * Свойства: название, описание, цена, статус ("горящий"), размер скидки.
3. **Заказ**:
   * Свойства: выбранный тур, пользователь, дата бронирования, статус оплаты.
4. **Оплата**:
   * Свойства: сумма, статус оплаты.

**Основные взаимодействия:**

* Заказчик регистрируется и авторизуется в системе.
* Заказчик просматривает доступные туры, фильтрует их и выбирает тур.
* После выбора тура заказчик создает заказ и оплачивает его.
* Турагент добавляет новые туры, отмечает их как "горящие" и устанавливает скидки для постоянных клиентов.

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

**Варианты использования:**

* **Регистрация/Авторизация**: Заказчик или турагент входят в систему.
* **Просмотр туров**: Заказчик просматривает доступные туры.
* **Выбор тура**: Заказчик выбирает тур и добавляет его в заказ.
* **Оплата тура**: Заказчик оплачивает тур.
* **Создание тура**: Турагент добавляет новый тур в систему.
* **Пометка "горящий"**: Турагент отмечает тур как "горящий".
* **Управление скидками**: Турагент назначает скидки для постоянных клиентов.

**Диаграмма:**

* Акторы: **Заказчик** и **Турагент**.
* Варианты использования:
  + Заказчик: *Регистрация*, *Авторизация*, *Просмотр туров*, *Выбор тура*, *Оплата тура*.
  + Турагент: *Создание тура*, *Пометка "горящий"*, *Управление скидками*.
* Отношения:
  + **Включение** (include): "Выбор тура" включает "Просмотр туров".
  + **Расширение** (extend): "Оплата тура" может расширять "Выбор тура" (включает не только выбор, но и оплату).

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

**Основные классы:**

1. **User (Пользователь)**:
   * Свойства: id, name, email, password, role (Заказчик или Турагент).
   * Методы: register(), login().
2. **Tour (Тур)**:
   * Свойства: id, name, description, price, isHot, discount.
   * Методы: createTour(), setHotTour(), applyDiscount().
3. **Order (Заказ)**:
   * Свойства: id, tour, user, orderDate, paymentStatus.
   * Методы: createOrder(), payOrder().
4. **Payment (Оплата)**:
   * Свойства: id, order, amount, status.
   * Методы: processPayment().

**Диаграмма классов**:

User

+ id: int

+ name: string

+ email: string

+ password: string

+ role: string

+ register()

+ login()

Tour

+ id: int

+ name: string

+ description: string

+ price: float

+ isHot: bool

+ discount: float

+ createTour()

+ setHotTour()

+ applyDiscount()

Order

+ id: int

+ tour: Tour

+ user: User

+ orderDate: date

+ paymentStatus: string

+ createOrder()

+ payOrder()

Payment

+ id: int

+ order: Order

+ amount: float

+ status: string

+ processPayment()

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

**Сценарий: Бронирование и оплата тура**

1. Заказчик входит в систему.
2. Заказчик просматривает доступные туры.
3. Выбирает тур и создает заказ.
4. Оплачивает заказ.
5. Система обновляет статус оплаты.

**Диаграмма последовательности:**

1. **Акторы**: Заказчик, Система, Турагент.
2. Основные объекты: User, Tour, Order, Payment.

**4. Диаграмма деятельности (Activity Diagram)**

**Сценарий: Управление туром (для турагента)**

1. Турагент входит в систему.
2. Добавляет новый тур.
3. Помечает тур как "горящий".
4. Устанавливает скидку для постоянных клиентов.

**Диаграмма деятельности**:

* Действия:
  + Авторизация.
  + Добавление тура.
  + Установка параметра "горящий".
  + Применение скидки.

**5. Диаграмма компонентов (Component Diagram)**

**Компоненты системы:**

* **Frontend**:
  + Веб-приложение для заказчиков и турагентов.
* **Backend**:
  + Сервисы: управление пользователями, турами, заказами, оплатами.
* **Database**:
  + Хранение данных пользователей, туров, заказов.

**Диаграмма компонентов:**

* Связи:
  + Frontend → Backend: взаимодействие через API.
  + Backend → Database: операции CRUD.

Если нужно, я могу уточнить любую диаграмму или построить их в виде графических изображений!