**Курс:** Шаблоны проектирования приложений

**Тема:** Модуль 12 Диаграмма состояний, диаграмма деятельности

*Баллы:*

Задание №1 Диаграмма состояний

**Разработка системы управления процессом заказа автомобиля с использованием диаграммы состояний (State Diagram) на языке C#**

**Цель работы:**

Изучить и применить принципы построения диаграмм состояний (State Diagram) для разработки системы управления процессом заказа автомобиля (например, такси) с последующей реализацией на языке C#. Основная цель — научиться моделировать системы с разными состояниями, переключениями между ними и логикой поведения в зависимости от внешних факторов.

**Описание задания:**

Требуется разработать систему управления процессом заказа автомобиля. В процессе выполнения заказа система проходит через несколько состояний, начиная от выбора автомобиля и заканчивая завершением поездки.

**Основные состояния:**

1. **Idle** — начальное состояние, система ожидает действия пользователя.
2. **CarSelected** — пользователь выбрал автомобиль, но не подтвердил заказ.
3. **OrderConfirmed** — заказ подтвержден, автомобиль в пути.
4. **CarArrived** — автомобиль прибыл к пользователю.
5. **InTrip** — поездка начата, пользователь в автомобиле.
6. **TripCompleted** — поездка завершена, система ожидает оплату.
7. **TripCancelled** — заказ отменен.

**Дополнительные требования:**

1. Реализовать возможность отмены заказа на каждом этапе до начала поездки.
2. Реализовать обработку различных исключительных ситуаций, таких как задержка автомобиля, проблемы с оплатой и т.д.

**Этапы выполнения работы:**

**1. Проектирование диаграммы состояний (UML):**

Создайте диаграмму состояний для описания процесса заказа автомобиля. Диаграмма должна включать:

* Состояния: ожидание (Idle), выбор автомобиля (CarSelected), подтверждение заказа (OrderConfirmed), прибытие автомобиля (CarArrived), выполнение поездки (InTrip), завершение поездки (TripCompleted), отмена заказа (TripCancelled).
* Переходы между состояниями: от выбора автомобиля к подтверждению заказа, от подтверждения заказа к прибытия автомобиля и т.д.
* Обработка отмены заказа на любом этапе.

**Дополнительные задания (по желанию):**

* Реализовать обработку ошибок, например, если автомобиль задерживается или отменяется.
* Добавить возможность изменения автомобиля до подтверждения заказа.
* Реализовать систему рейтинга водителя и поездки.

Задание №2 Диаграмма вариантов использования

**Цель задания:**

Научиться проектировать **диаграммы вариантов использования** для отображения требований к системе и реализовать их ключевые элементы.

**Условие задания:**

Разработать диаграмму вариантов использования для **системы управления бронированием мероприятий**. На основе диаграммы реализовать основные функциональные элементы.

**Описание системы:**

Система позволяет пользователю бронировать и организовывать мероприятия. Существует три типа пользователей:

1. **Гость** – может просматривать список доступных мероприятий.
2. **Зарегистрированный пользователь** – может:
   * Бронировать мероприятие.
   * Отменять бронирование.
3. **Администратор** – может:
   * Управлять мероприятиями (добавлять, редактировать, удалять).
   * Просматривать все бронирования.

**Часть 1: Разработка диаграммы вариантов использования**

1. Создайте диаграмму вариантов использования для описанной системы.  
   Включите следующие элементы:
   * **Акторы:** Гость, Зарегистрированный пользователь, Администратор.
   * **Варианты использования:** Просмотр мероприятий, Бронирование мероприятия, Отмена бронирования, Управление мероприятиями, Просмотр всех бронирований.
   * Связи между акторами и вариантами использования.
   * Опционально: добавьте связь "extends" или "includes", где это необходимо.

Используйте любую удобную программу для построения UML-диаграмм, например, **Lucidchart**, **StarUML** или **Draw.io**.

**Часть 2: Реализация**

Реализуйте ключевые функции системы в виде консольного приложения на C#.

1. Создайте классы для следующих сущностей:
   * **Event** (Мероприятие) с полями: ID, Название, Дата, Место.
   * **User** (Пользователь) с полями: ID, Имя, Роль (Гость, Пользователь, Администратор).
   * **Booking** (Бронирование) с полями: ID, Пользователь, Мероприятие, Статус.
2. Реализуйте функционал:
   * Просмотр списка мероприятий.
   * Бронирование мероприятия.
   * Отмена бронирования.
   * Добавление, редактирование и удаление мероприятий (доступно только администратору).
   * Просмотр всех бронирований (доступно только администратору).
3. Создайте тестовые данные (мероприятия, пользователи) для демонстрации работы системы.

**Разработка системы управления процессом оформления заказа в интернет-магазине с использованием диаграммы деятельности (Activity Diagram) на языке C#**

**Цель работы:**

Изучить и применить принципы построения диаграмм деятельности (Activity Diagram) для моделирования процессов, а также реализовать систему управления оформлением заказа в интернет-магазине на языке C#. Основная цель — научиться строить модели бизнес-процессов с учетом возможных сценариев и реализовывать эти сценарии на практике.

**Описание задания:**

Требуется разработать систему управления процессом оформления заказа в интернет-магазине, которая проходит через несколько этапов:

1. **Выбор товаров** — покупатель добавляет товары в корзину.
2. **Оформление заказа** — покупатель заполняет данные для заказа.
3. **Оплата заказа** — покупатель оплачивает заказ через платежную систему.
4. **Подтверждение оплаты** — система проверяет, прошла ли оплата.
5. **Обработка заказа** — заказ обрабатывается на складе.
6. **Отправка заказа** — заказ отправляется на доставку.

**Основные этапы:**

1. **Выбор товаров**: покупатель выбирает товары из каталога и добавляет их в корзину.
2. **Оформление заказа**: после добавления всех товаров, покупатель вводит информацию для доставки и контактные данные.
3. **Оплата**: происходит процесс оплаты через платежную систему.
4. **Подтверждение оплаты**: проверка успешности платежа.
5. **Обработка и отправка заказа**: после подтверждения оплаты заказ обрабатывается и отправляется.

**Этапы выполнения работы:**

**1. Проектирование диаграммы деятельности (UML):**

Создайте диаграмму деятельности для описания процесса оформления заказа в интернет-магазине. Диаграмма должна включать:

* Последовательные действия: выбор товаров, оформление заказа, оплата, подтверждение и обработка заказа.
* Возможные альтернативные пути: например, если оплата не прошла или заказ отменен.
* Разветвления: проверка успешности оплаты и подтверждение заказа.

**2. Реализация на языке C#:**

Реализуйте систему управления процессом оформления заказа на языке C#, используя структуры и классы для описания логики работы. Для каждого этапа создайте соответствующие методы, которые моделируют действия пользователя и системы.

**Объяснение:**

1. **Выбор товаров**: пользователь добавляет товары в корзину с помощью метода AddProduct().
2. **Оформление заказа**: создается объект заказа с указанием данных покупателя и его корзины.
3. **Оплата**: происходит через метод Pay(), который проверяет достаточность средств.
4. **Подтверждение заказа**: если оплата прошла успешно, заказ подтверждается.
5. **Обработка заказа**: класс OrderProcessor обрабатывает заказ, отправляя его на склад и далее на доставку.

**Требования к отчету:**

1. **Диаграмма деятельности (Activity Diagram)**: отражающая основные этапы процесса и возможные ветвления.
2. **Описание системы**: краткое описание работы каждого класса и метода.
3. **Исходный код**: реализация системы на языке C#.
4. **Примеры работы программы**: сценарии выполнения программы с демонстрацией возможных ситуаций.
5. **Выводы**: анализ выполнения работы и достижение поставленных целей.

**Дополнительные задания (по желанию):**

* Реализовать возможность отмены заказа до его оплаты.
* Добавить функцию скидок и расчета итоговой стоимости с учетом промокодов.
* Добавить альтернативный путь для оплаты через разные платежные системы.
* Реализовать уведомления для покупателя о каждом этапе обработки заказа.