

Федеральное государственное образовательное бюджетное учреждение
высшего образования
**«Финансовый университет при Правительстве
Российской Федерации»
(Финансовый университет)**

Колледж информатики и программирования

ПМ.08 Разработка кода
информационных систем

Группа: ЗИСИП-622

УТВЕРЖДАЮ

Председатель предметно-цикловой
комиссии информационных систем
и программирования

_____ Т.Г. Аксёнова

« ____ » _____ 2025 г.

ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3

**Построение диаграммы Состояний и
диаграммы Классов и генерации кода**

Преподаватель

_____ Р. Р. Абзалимов

Исполнитель

_____ Л. Д. Слепцов

Оценка: _____

« ____ » _____ 2025 г.

Москва

2025

Цель работы:

Ознакомиться с методологией моделирования информационных систем на основе языка UML.

Тема: Охота.

Ход работы:

Диаграмма состояний (State Machine Diagram) — это один из видов диаграмм в UML, который используется для моделирования поведения системы или объекта в зависимости от его состояний и переходов между ними. Она показывает, как объект изменяет свои состояния в ответ на события (триггеры) и какие действия выполняются при этих переходах.

На рисунке 1 представлена диаграмма состояний корзины интернет-магазина для охоты HuntPlanet.

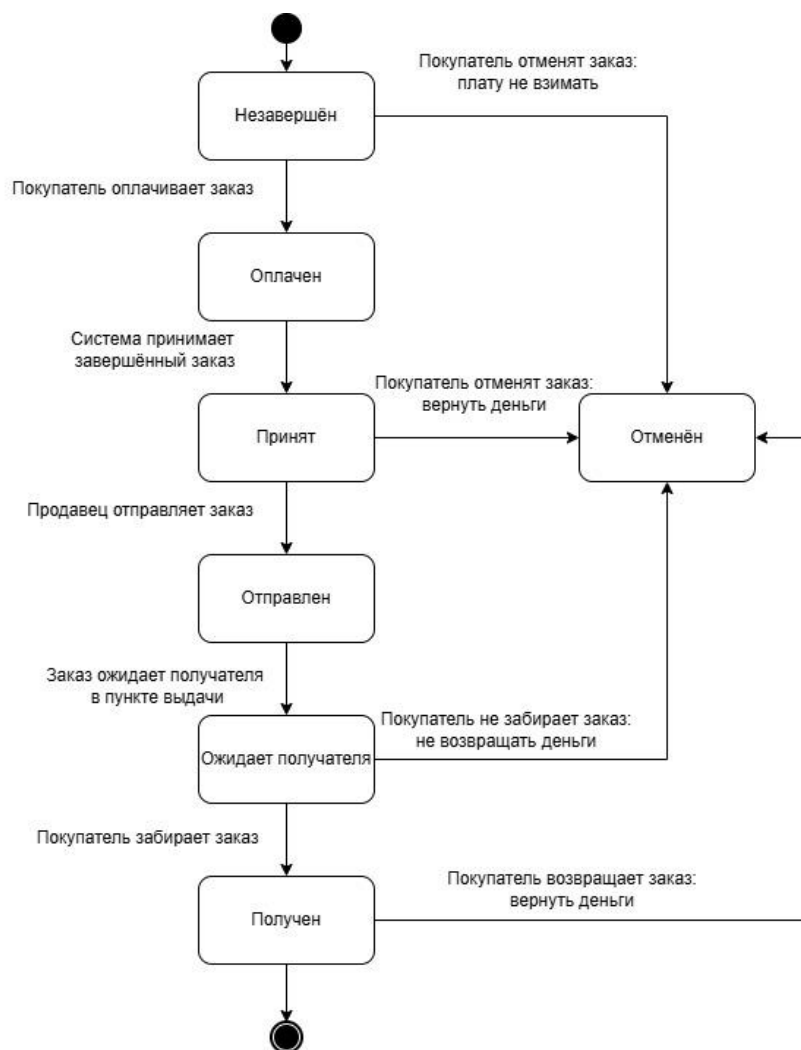


Рисунок 1 –диаграмма состояний корзины интернет-магазина для охоты
HuntPlanet

Диаграмма классов (Class Diagram) — это один из основных типов диаграмм в UML, который используется для визуализации структуры системы. Она показывает классы, их атрибуты, методы и отношения между ними. Диаграмма классов является статической, то есть она описывает структуру системы, а не её поведение.

На рисунке 2 представлена диаграмма классов корзины интернет-магазина для охоты HuntPlanet.

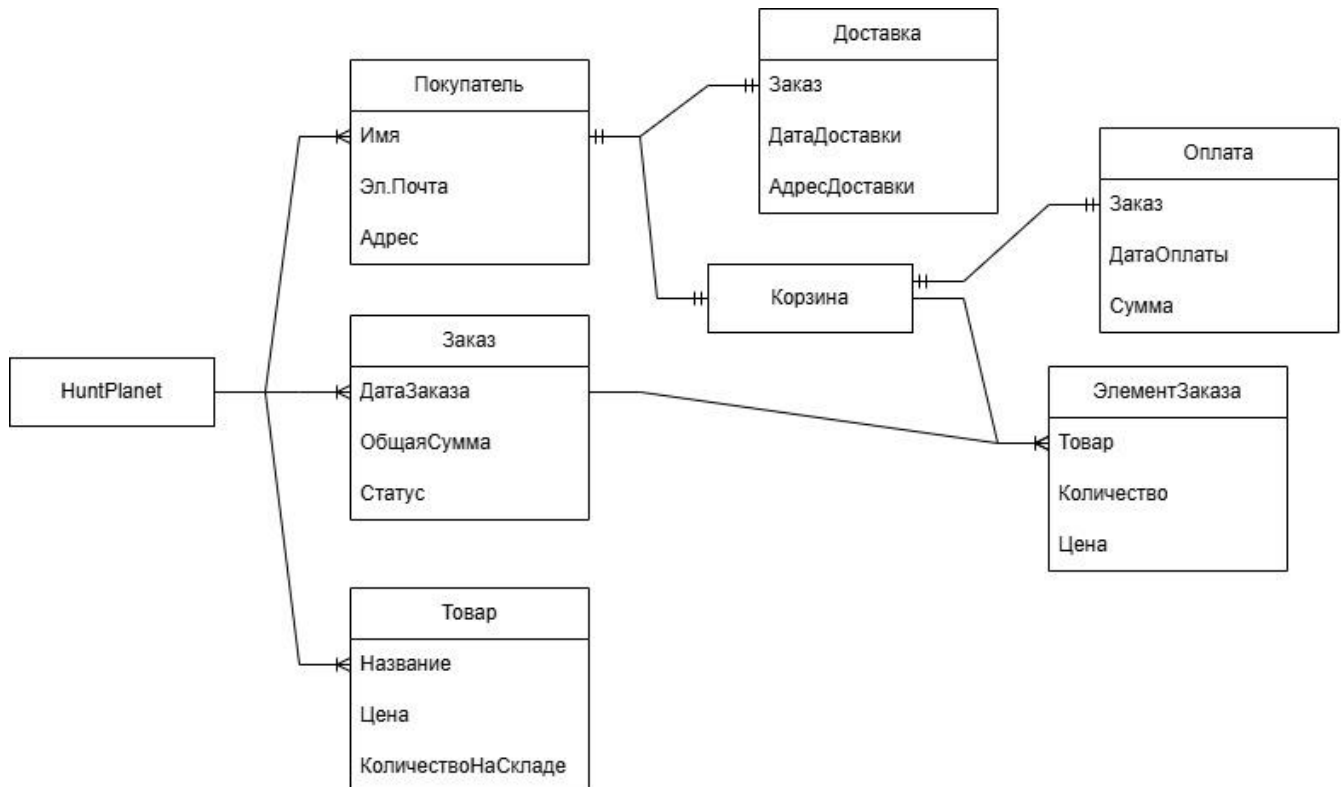


Рисунок 2 –диаграмма классов корзины интернет-магазина для охоты HuntPlanet

Вопросы:

1. Какие основные цели и задачи решает диаграмма деятельности в UML?

Диаграмма деятельности (Activity Diagram) в UML используется для моделирования бизнес-процессов, рабочих процессов и алгоритмов. Её основные цели и задачи:

- **Визуализация процессов:** Показывает последовательность действий и поток управления в системе.
- **Моделирование бизнес-процессов:** Описывает, как выполняются задачи в организации или системе.
- **Описание алгоритмов:** Используется для представления логики выполнения операций.
- **Анализ и оптимизация процессов:** Помогает выявить узкие места, избыточные шаги или возможности для улучшения.
- **Документирование процессов:** Служит наглядным документом для разработчиков, аналитиков и заказчиков.

2. Какие элементы используются в диаграмме деятельности для визуализации процессов?

Основные элементы диаграммы деятельности:

1. **Действие (Activity):** Прямоугольник с закругленными углами, представляющий шаг процесса.
2. **Переход (Transition):** Стрелка, показывающая поток управления между действиями.
3. **Начальное состояние (Initial Node):** Черный кружок, обозначающий начало процесса.
4. **Конечное состояние (Final Node):** Черный кружок с контуром, обозначающий завершение процесса.
5. **Условие (Decision Node):** Ромб, представляющий ветвление процесса на основе условия.
6. **Слияние (Merge Node):** Ромб, объединяющий несколько потоков в один.
7. **Параллельные потоки (Fork/Join):** Горизонтальные или вертикальные линии, разделяющие или объединяющие параллельные действия.

8. **Поток объектов (Object Flow):** Показывает, как объекты передаются между действиями.
9. **Дорожки (Swimlanes):** Вертикальные или горизонтальные разделы, которые группируют действия по ролям или участникам.

3. Какое значение имеют состояния и переходы в диаграмме состояний в UML?

- **Состояния (States):** Представляют конкретные условия или этапы, в которых может находиться объект. Например, заказ может быть в состоянии "Создан", "Оплачен", "Доставлен".
- **Переходы (Transitions):** Показывают, как объект переходит из одного состояния в другое. Переходы обычно вызываются событиями (например, "оплата получена") и могут сопровождаться действиями.

Состояния и переходы помогают:

- Моделировать жизненный цикл объекта.
- Описывать поведение системы в зависимости от её текущего состояния.
- Визуализировать сложные процессы с множеством условий и событий.

4. Как определяются классы и их атрибуты в диаграмме классов в UML?

- **Класс (Class):** Представляется прямоугольником, разделенным на три секции:
 1. **Имя класса:** Верхняя секция.
 2. **Атрибуты:** Средняя секция. Атрибуты описывают свойства класса (например, имя: String, возраст: int).

3. **Методы:** Нижняя секция. Методы описывают поведение класса (например, `сохранить()`, `удалить()`).

5. Какие связи между классами можно отобразить на диаграмме классов?

Основные типы связей:

1. **Ассоциация (Association):** Простая связь между классами (например, Пользователь и Заказ).
2. **Агрегация (Aggregation):** Отношение "часть-целое", где часть может существовать независимо от целого (например, Корзина и Товар).
3. **Композиция (Composition):** Более строгая форма агрегации, где часть не может существовать без целого (например, Заказ и ЭлементЗаказа).
4. **Наследование (Inheritance):** Отношение, при котором один класс наследует свойства и методы другого (например, Админ наследует Пользователь).
5. **Зависимость (Dependency):** Показывает, что один класс зависит от другого (например, Заказ зависит от Платеж).

6. Какие дополнительные элементы могут быть представлены на диаграмме классов для полноты модели?

- **Интерфейсы (Interfaces):** Описывают набор методов, которые должны быть реализованы классами.
- **Перечисления (Enumerations):** Используются для определения набора именованных значений (например, СтатусЗаказа: [Создан, Оплачен, Доставлен]).
- **Шаблоны классов (Templates):** Позволяют создавать обобщенные классы.

- **Примечания (Notes):** Добавляют пояснения к элементам диаграммы.

7. Как составить диаграмму состояний для определенного объекта или системы в UML?

1. **Определить состояния объекта:** Например, для заказа: "Создан", "Оплачен", "Доставлен".
2. **Определить события и переходы:** Например, "оплата получена" → переход из "Создан" в "Оплачен".
3. **Добавить начальное и конечное состояния.**
4. **Указать действия:** Например, при переходе в состояние "Оплачен" выполнить действие "отправить уведомление".
5. **Использовать инструменты UML** для визуализации диаграммы.

8. Какие инструменты и средства моделирования UML поддерживают построение диаграмм деятельности, состояний и классов?

- **Enterprise Architect:** Профессиональный инструмент для моделирования UML.
- **Visual Paradigm:** Поддерживает все типы диаграмм UML.
- **Lucidchart:** Онлайн-инструмент для создания диаграмм.
- **Draw.io:** Бесплатный инструмент для создания диаграмм.
- **StarUML:** Бесплатное приложение для моделирования UML.
- **IntelliJ IDEA (PlantUML):** Поддержка UML через плагины.

9. Как происходит генерация кода на основе диаграмм UML, в частности, диаграммы классов?

- **Автоматическая генерация кода:** Современные CASE-средства (например, Enterprise Architect, Visual Paradigm) позволяют

генерировать код на основе диаграмм классов. Например, класс Пользователь может быть преобразован в код на Java, C# или другом языке.

- **Шаблоны кода:** Инструменты используют шаблоны для создания классов, атрибутов, методов и связей.
- **Обратная разработка:** Возможность обновления диаграмм на основе изменений в коде.

10. Как оформить и сохранить модель в UML, созданную в CASE-средстве, и что следует указать в отчете по выполненному заданию?

- **Сохранение модели:** Модель сохраняется в формате, поддерживаемом CASE-средством (например, .ear, .xml, .uml).
- **Экспорт:** Диаграммы можно экспортировать в форматы изображений (.png, .jpg, .pdf).
- **Отчет:**
 1. **Цель модели:** Описание задачи, для которой создана модель.
 2. **Описание диаграмм:** Краткое описание каждой диаграммы.
 3. **Ответы на вопросы.**
 4. **Выводы:** Результаты анализа и рекомендации.

Вывод

Данная работа помогла глубже понять как проектирование UML-диаграмм способствует разработке эффективных и понятных систем, и как они могут быть использованы для оптимизации процесса управления корзиной в интернет-магазине.