

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 №1**

**Построение диаграммы Вариантов использования и диаграммы  
Последовательности и генерация кода**

Преподаватель

\_\_\_\_\_ Р. Р. Абзалимов

Исполнитель

\_\_\_\_\_ Л. Д. Слепцов

Оценка: \_\_\_\_\_

« \_\_\_\_ » \_\_\_\_\_ 2025 г.

Москва

202

**Цель работы:**

Изучение объектно-ориентированного анализа и моделирования бизнес-процессов в исследуемой предметной области с помощью языка UML.

**Тема:** Охота.

**Ход работы:****Корзина**

Диаграмма вариантов использования (use case diagram) — диаграмма, на которой изображаются отношения между акторами и вариантами использования (прецедентами). Это исходное концептуальное представление или концептуальная модель системы в процессе её проектирования и разработки.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

На рисунке 1 изображена диаграмма вариантов использования корзины интернет-магазина для охоты HuntPlanet.

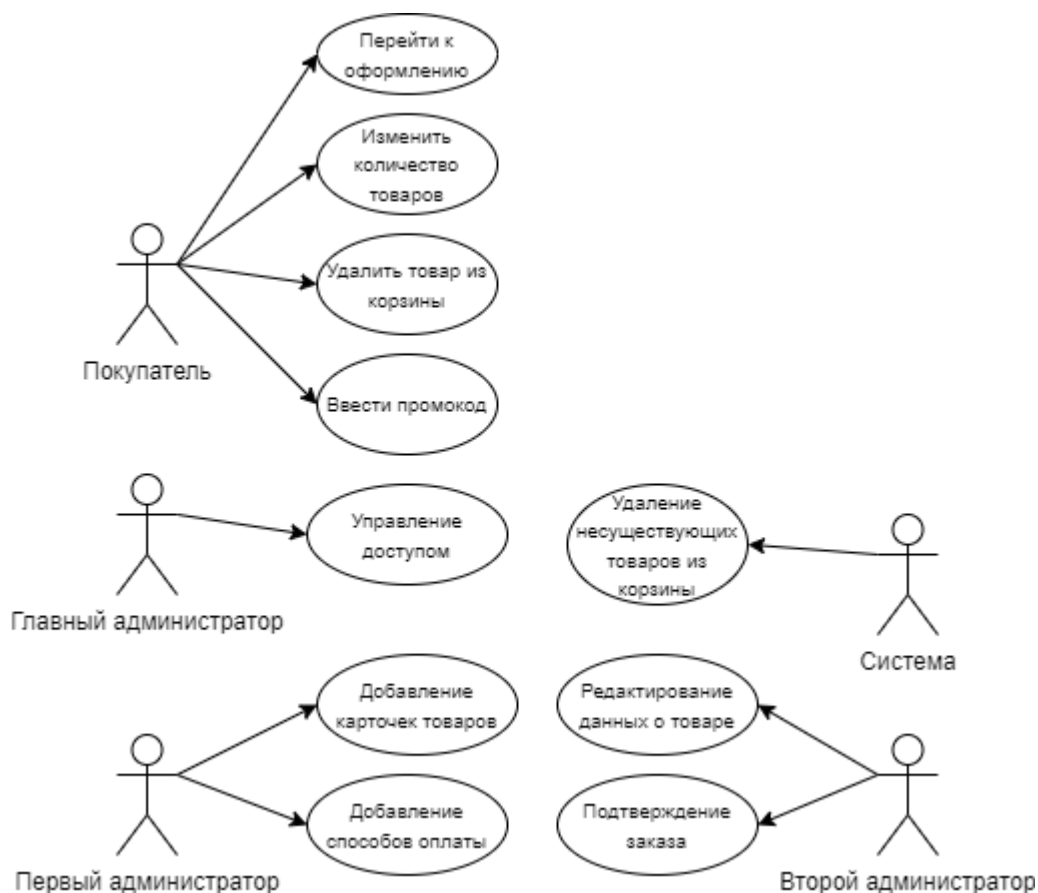


Рисунок 1 – диаграмма вариантов использования

Диаграмма деятельности — это UML-диаграмма, на которой показаны действия. Она представляет собой графическое представление рабочих процессов поэтапных действий с поддержкой выбора, итерации и параллелизма.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

На рисунке 2 изображена диаграмма деятельности корзины интернет-магазина Aliexpress.

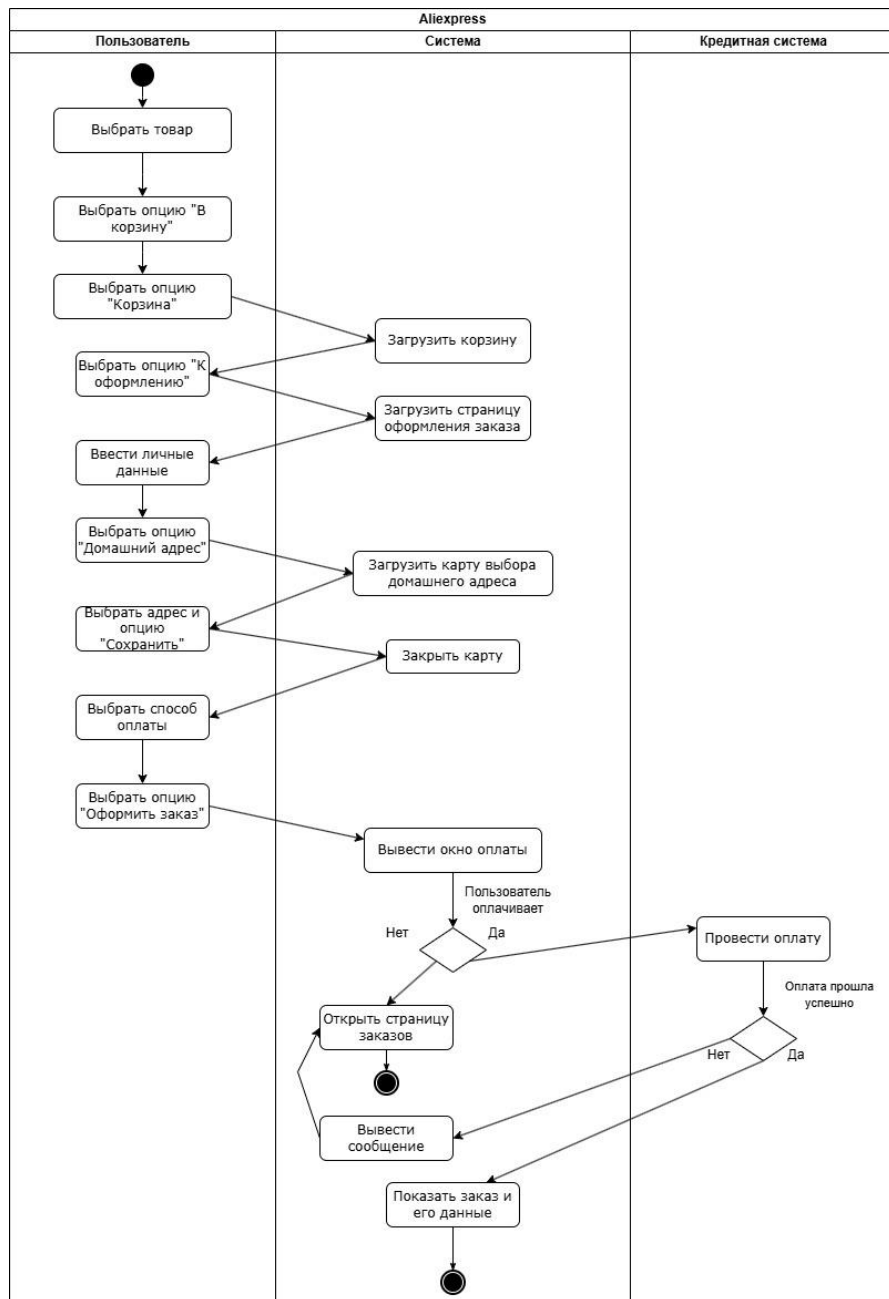


Рисунок 2 – диаграмма деятельности корзины интернет-магазина AliExpress

На рисунке 3 изображена диаграмма деятельности корзины интернет-магазина Wildberries.

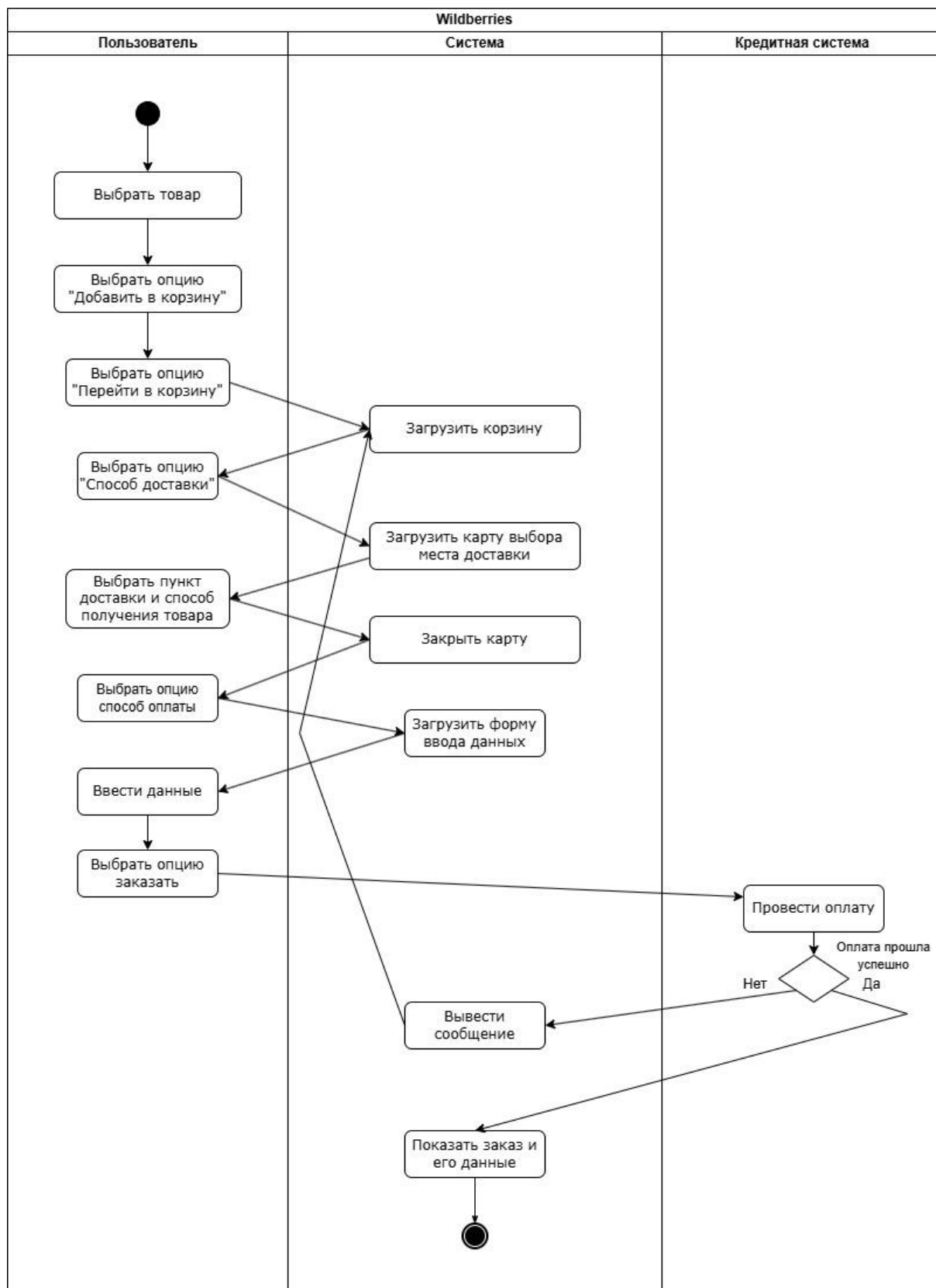


Рисунок 3 – диаграмма деятельности корзины интернет-магазина Wildberries

На рисунке 4 изображена диаграмма деятельности корзины интернет-магазина OZON.

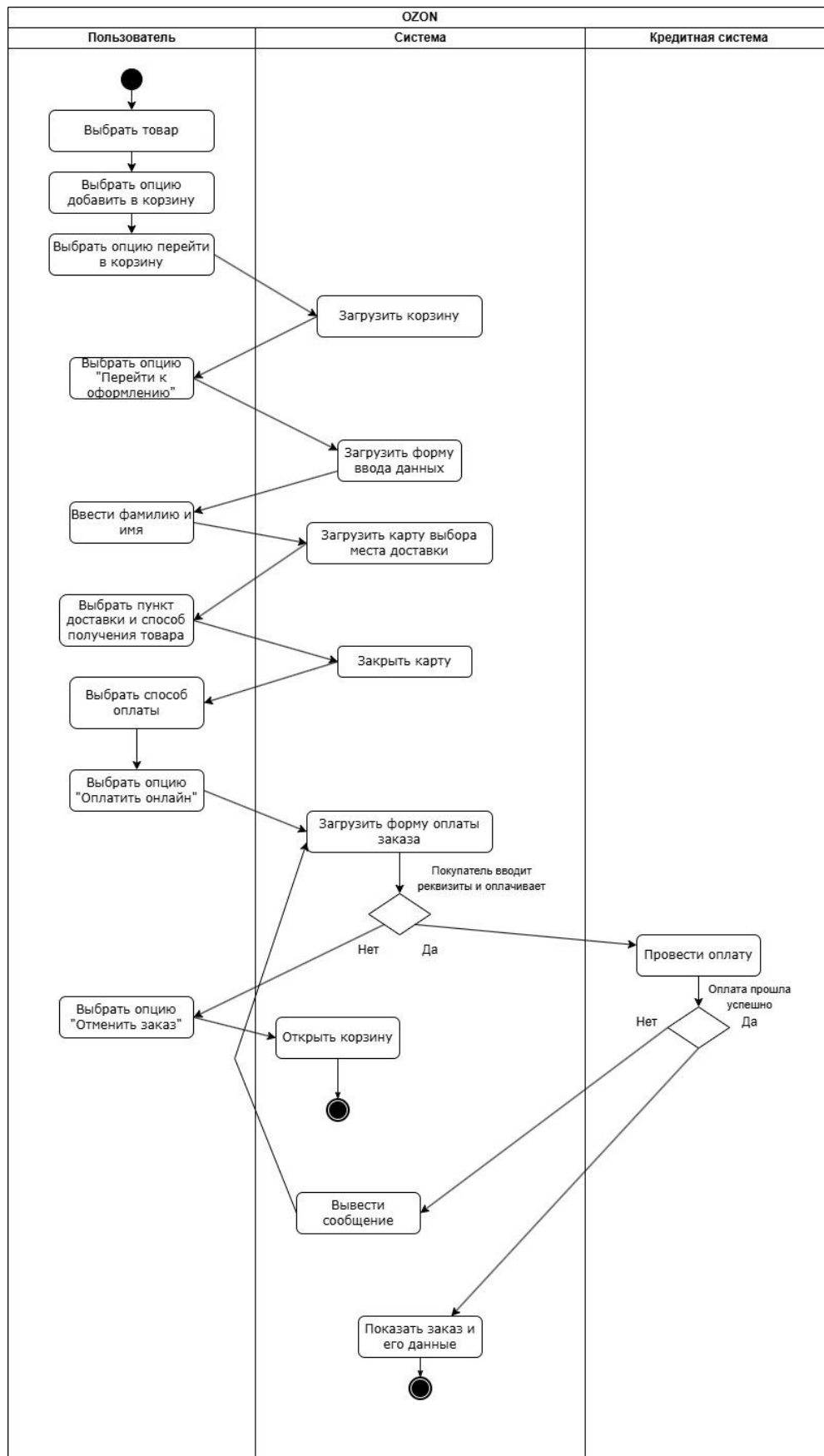


Рисунок 4 – диаграмма деятельности корзины интернет-магазина OZON

На рисунке 5 изображена диаграмма деятельности корзины интернет-магазина для охоты HuntPlanet.

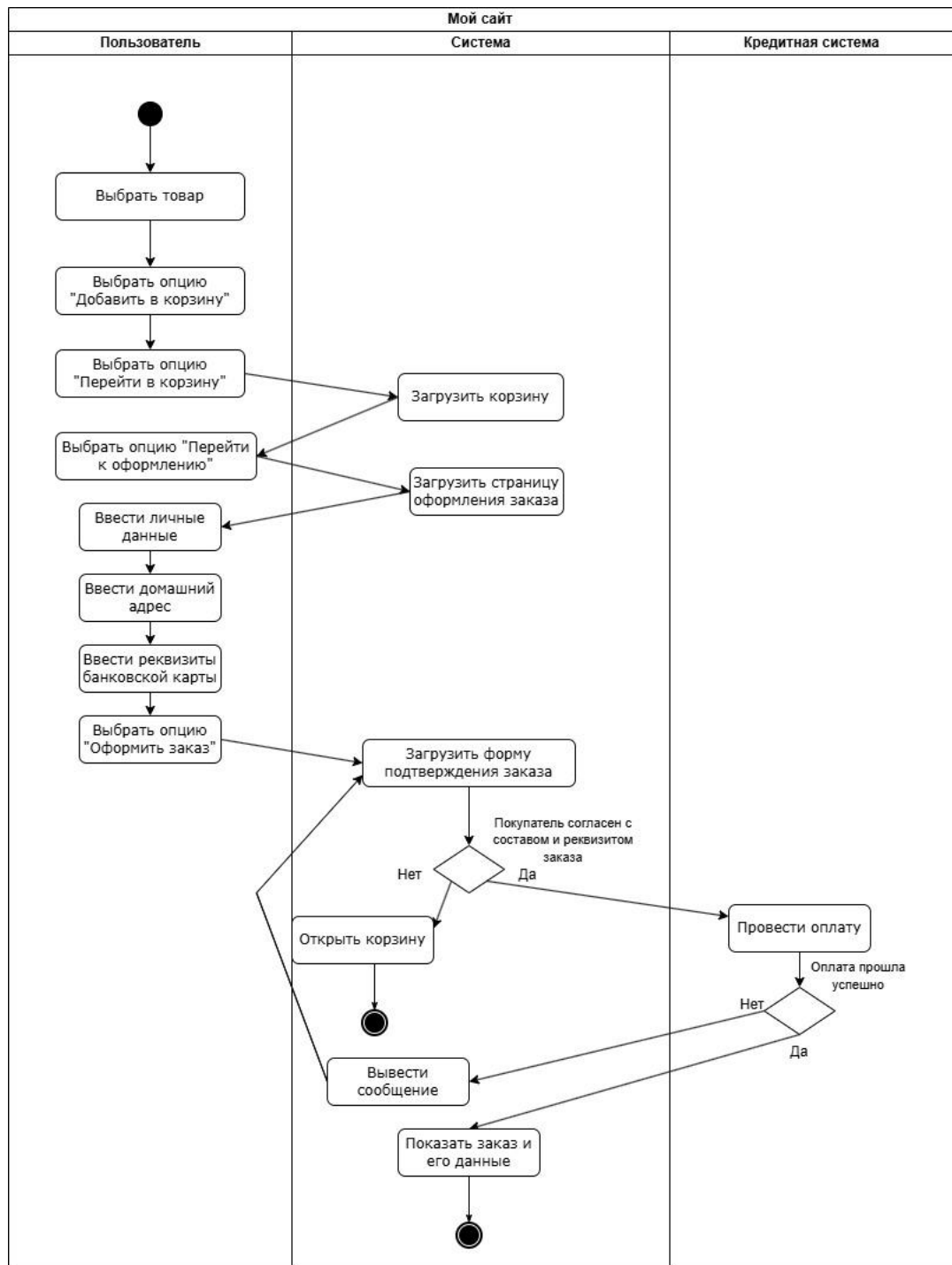


Рисунок 5 – диаграмма деятельности корзины интернет-магазина для охоты HuntPlanet

На рисунке 6 изображена корзина интернет-магазина AliExpress, где зелёными рамками выделены позаимствованные элементы.

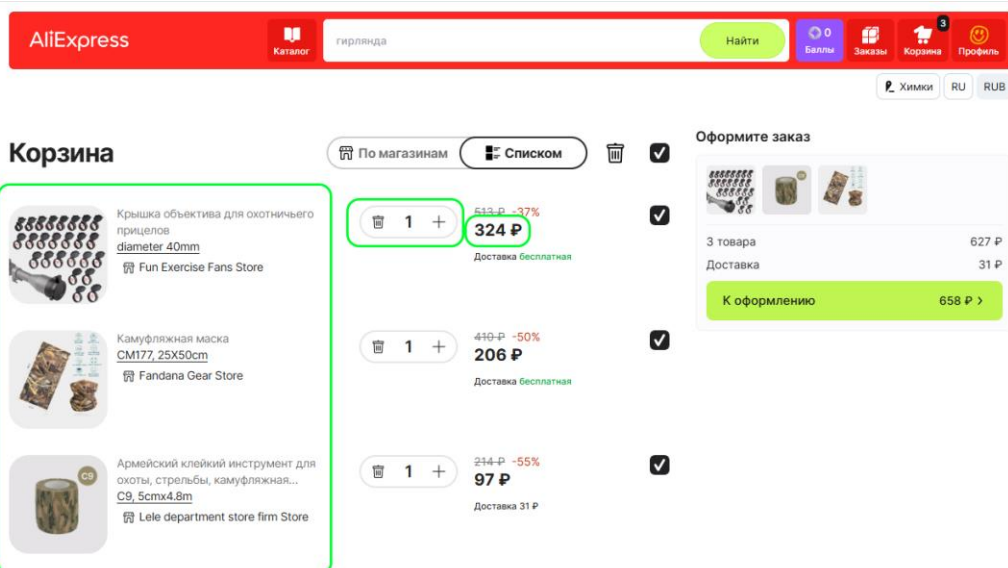


Рисунок 6 – корзина интернет-магазина AliExpress

На рисунке 7 изображена корзина интернет-магазина Wildberries, где зелёными рамками выделены позаимствованные элементы.

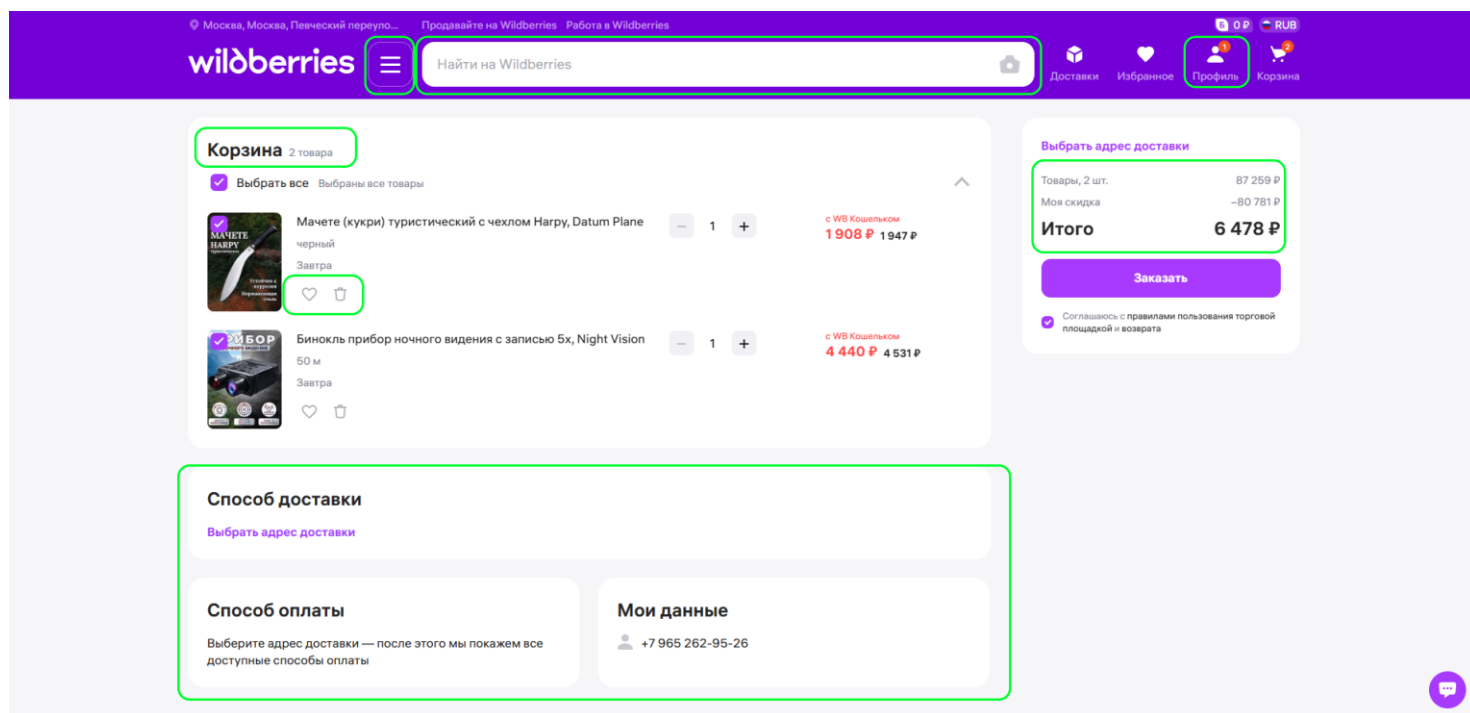


Рисунок 7 – корзина интернет-магазина Wildberries

На рисунке 8 изображена корзина интернет-магазина OZON, где зелёными рамками выделены позаимствованные элементы.



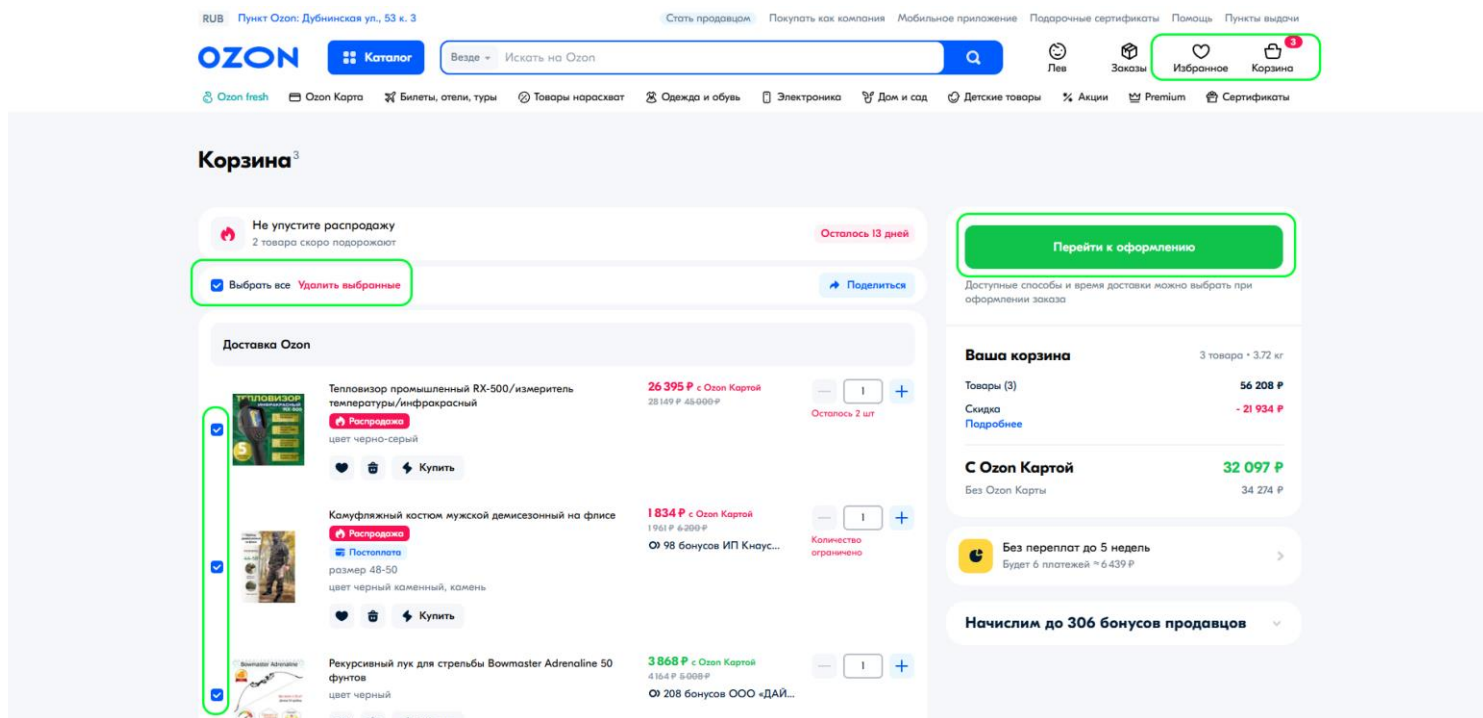


Рисунок 8 – корзина интернет-магазина OZON

На рисунке 9 изображена корзина интернет-магазина для охоты HuntPlanet.

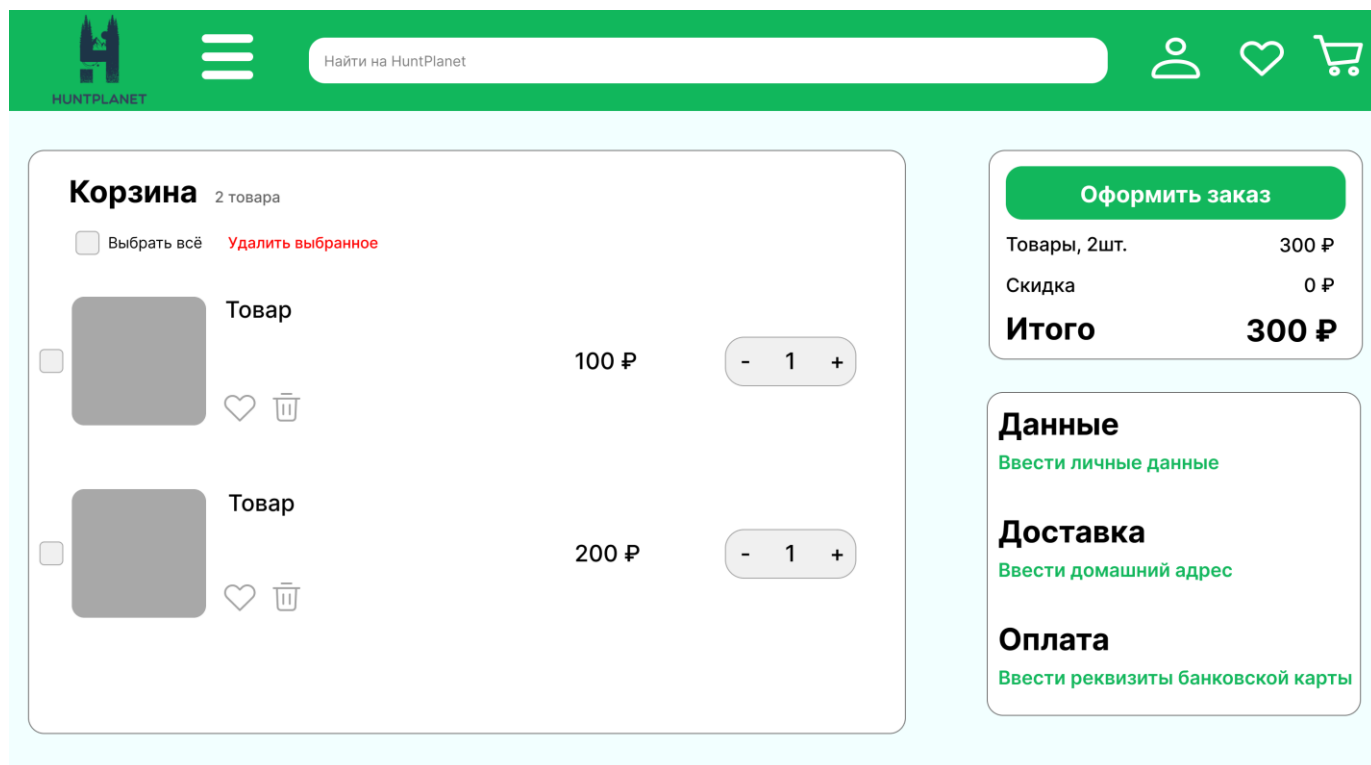


Рисунок 9 – корзина интернет-магазина для охоты HuntPlanet

Вопросы:

1. Что такое UML и каковы его основные компоненты?

UML (Unified Modeling Language) представляет собой стандартный графический язык для концептуального, визуального и документированного описания систем. Он используется для визуализации, спецификации, создания и документирования артефактов программного обеспечения. UML помогает разработчикам и другим заинтересованным сторонам лучше понять структуру и поведение системы.

Структурные диаграммы:

- Классовая диаграмма — отображает классы и связи между ними.
- Компонентная диаграмма — показывает компоненты системы и их взаимосвязи.
- Диаграмма развертывания — описывает физическую реализацию системы, включая узлы и их связи.
- Объектная диаграмма — отображает экземпляры классов и их связи на момент выполнения.
- Диаграмма пакетов — показывает организацию модели на уровне пакетов.

Поведенческие диаграммы:

- Диаграмма прецедентов — отражает взаимодействие пользователей и системы, описывая функции, доступные пользователю.
- Диаграмма последовательностей — показывает, как объекты взаимодействуют во времени.
- Диаграмма коммуникации — отображает взаимосвязь между объектами и их взаимодействия.
- Диаграмма состояний — описывает состояния объектов и переходы между ними.
- Диаграмма действий — представляет поток управления между действиями, описывающими логику бизнес-процессов.

## **2. Каковы основные этапы объектно-ориентированного анализа и моделирования бизнес-процессов?**

Процесс моделирования — это структурированный подход к созданию абстрактного представления системы. Он включает несколько ключевых шагов:

- Определение цели моделирования

На этом этапе необходимо четко определить, для чего создается модель. Это может быть анализ требований, проектирование системы или документация. Понимание цели помогает сосредоточиться на важных аспектах модели.

- Сбор требований

Здесь следует собрать все необходимые требования от заинтересованных сторон, включая пользователей, заказчиков и технический персонал. Это могут быть функциональные и нефункциональные требования, которые будут определять, что должно быть включено в модель.

- Определение границ модели

Нужно определить, что именно будет включено в модель, а что остается вне её. Это помогает фокусироваться на главных аспектах и избегать избыточности.

- Создание абстракций

На этом этапе создаются абстрактные представления (модели) системы, используя различные диаграммы и нотации, такие как UML. Это может включать структурные диаграммы для отображения классов и их связей, а также поведенческие диаграммы для описания взаимодействия.

- Валидация модели

После создания модели очень важно проверить её на соответствие требованиям. Это может быть осуществлено путем рецензий, обсуждений с заинтересованными сторонами или тестирования модели на примерах.

- Документация

На этом этапе необходимо задокументировать модель и объяснить её составные части. Документация помогает другим участникам проекта понять модель и использовать её в будущем.

– Итерации и доработка

Моделирование — это итеративный процесс. На основании отзывов и новой информации модель может постоянно обновляться и уточняться. Это позволяет улучшить качество модели и повысить её соответствие требованиям.

### **3. Что такое диаграмма вариантов использования и какова ее роль в проектировании системы?**

Диаграмма вариантов использования — это важный инструмент в визуализации требований пользователей, который позволяет понять, как различные пользователи (или «актеры») взаимодействуют с системой. Визуализация этих взаимодействий помогает разработчикам и заинтересованным сторонам легче осознать функционал системы. Основные аспекты, как диаграмма вариантов использования помогает в этом процессе:

– Идентификация актеров

Диаграмма показывает, кто будет использовать систему. Актеры могут быть как пользователями, так и другими системами. Это позволяет четко определить, кто заинтересован в функционале системы и какие роли они играют.

– Представление вариантов использования

Каждый вариант использования описывает конкретную функцию системы с точки зрения пользователя. Это помогает выявить, что именно пользователи ожидают от системы и какие задачи они хотят выполнять.

– Визуальная ясность

Диаграмма вариантов использования использует простые графические элементы, что делает информацию более доступной и легкой для восприятия. Это позволяет всем заинтересованным сторонам (пользователям, аналитикам и разработчикам) быстро понять основные функции системы.

– Определение границ системы

Диаграмма помогает четко обозначить границы системы, показывая, что включено в функциональность, а что находится за её пределами. Это важно для управления ожиданиями и разработки сроков.

- Улучшение коммуникации

Визуализация требований с помощью диаграммы вариантов использования облегчает общение между всеми участниками проекта. Это помогает избежать недопонимания и обеспечивает более продуктивные обсуждения.

- Поддержка анализа и проектирования

Диаграмма может быть использована как основа для дальнейшего анализа и проектирования системы. С её помощью можно легко определить, какие из вариантов использования требуют более детального описания, или какие сценарии требуют особого внимания.

#### **4. Что такое диаграмма деятельности и как она отличается от диаграммы вариантов использования?**

Диаграмма деятельности — это тип модельной диаграммы, которая используется для описания динамического поведения систем, процессов или выполнения задач. Это полезный инструмент в различных ситуациях. Несколько случаев, когда целесообразно использовать диаграмму деятельности:

- Моделирование бизнес-процессов

Диаграмма деятельности помогает визуализировать и оптимизировать бизнес-процессы, показывая последовательность задач и действия, которые необходимо выполнить для достижения определенной цели.

- Уточнение потоков управления

Когда необходимо прояснить потоки управления в системе, диаграмма деятельности помогает отображать, как различные действия и решения связаны друг с другом. Это важно для понимания логики работы системы.

- Управление сложными системами

В сложных системах, где имеется множество элементов и взаимосвязей, диаграмма деятельности помогает упорядочить действия и понять, как они взаимодействуют. Это способствует выявлению узких мест и потенциальных проблем.

- Обеспечение требований к программному обеспечению

Создание диаграммы деятельности на этапе сбора требований помогает разработчикам и аналитикам более четко увидеть, какие действия должны выполняться пользователями и системе, что может повысить точность выполнения требований.

- Документация процессов

Диаграмма деятельности может служить эффективным документированием, обеспечивая четкое представление о процессах и действиях, что важно для обучения новых сотрудников или при передаче проектов между командами.

- Анализ альтернативных сценариев

При разработке системы полезно анализировать разные сценарии выполнения задач. Диаграммы деятельности могут помочь визуализировать альтернативные пути, позволяя выбирать оптимальный подход.

- Определение ошибок и недочетов

Визуальное представление рабочего процесса помогает выявить ошибки или недочеты в процессе, что может улучшить его эффективность и качество.

## **5. Как построить диаграмму последовательности и какие элементы она включает?**

На диаграмме последовательности представляются следующие основные объекты:

- Активные объекты (или инстансы)

Эти объекты представляют собой сущности, которые участвуют в взаимодействии, такие как:

- Пользователи (люди или их роли).

- Программы или компоненты системы.
- Внешние системы или службы.
- Сообщения

Сообщения определяют взаимодействие между объектами. Они могут включать:

- Вызовы методов: обозначают запрос на выполнение действия у другого объекта (например, "objectA.method()").
- Ответы: показывают возвращаемое значение или подтверждение выполнения операции (например, "return value").
- Возврат: указывают на окончание вызова метода и возвращение к предыдущему объекту.

- Сигналы

Это особый тип сообщений, который используется для уведомления объектов о происходящих событиях (например, "signalEvent")

- Условия и альтернативные Flows

Иногда диаграмма может включать условные сообщения, указывающие на разные пути выполнения (например, "if condition").

- Время

Каждый объект на диаграмме имеет горизонтальную линию времени, представляющую его существование в момент выполнения. Временные метки указывают на последовательность сообщений.

- Создание и уничтожение объектов

Диаграмма может также показывать создание новых объектов (например, используя специальный символ) и уничтожение (например, "destroy").

Сообщения должны быть представлены четко и однозначно:

- Указание на вызов метода: сообщение, указывающее на вызов метода другого объекта, должно показывать, какой метод вызывается и на каком объекте.

- Линии ответа: нужно показывать линии, которые удостоверяют взаимодействие и возврат значений обратно.

- Синхронные и асинхронные сообщения: различать синхронные (ожидающие ответа) и асинхронные (не ожидают ответа) сообщения.

## **6. Какие преимущества предоставляет использование CASE-средств для разработчиков?**

Использование CASE-средств (Computer-Aided Software Engineering) предоставляет разработчикам множество преимуществ:

- Увеличение производительности

CASE-средства автоматизируют рутинные задачи, что позволяет разработчикам сосредоточиться на более сложных аспектах проекта. Это ведет к более быстрой реализации и разработке программного обеспечения.

- Улучшение качества программного обеспечения

Использование визуальных инструментов и стандартов помогает минимизировать ошибки в проектировании и кодировании, что способствует созданию более качественного продукта.

- Моделирование и визуализация

CASE-средства предоставляют возможность моделировать и визуализировать архитектуру системы, что позволяет лучше понять структуру и взаимодействие компонентов.

- Стандартизация процессов

CASE-средства способствуют введению стандартов в процесс разработки, упрощая понимание и поддержку проекта, а также облегчают обучение новых сотрудников.

- Упрощение документирования

Многие CASE-инструменты автоматически генерируют документацию на основе моделей, что значительно упрощает процесс документирования и обеспечивает его актуальность.

- Поддержка совместной работы



CASE-средства часто имеют функции для совместной работы, что позволяет командам легче координировать свои действия и обмениваться информацией.

- Упрощение изменений и управления рисками

Использование визуальных моделей и диаграмм упрощает процесс внесения изменений в проект, позволяя легче управлять рисками, связанными с изменениями в требованиях или дизайне.

- Интеграция с другими инструментами

CASE-средства часто интегрируются с другими инструментами разработки, такими как системы контроля версий или инструменты для тестирования, что улучшает общий процесс разработки.

- Снижение затрат

Увеличение производительности, улучшение качества и автоматизация процессов могут привести к снижению затрат на разработку и поддержку программного обеспечения.

## **7. Что такое синхронизация в диаграммах деятельности и как она визуализируется?**

Синхронизация в диаграммах деятельности (activity diagrams) — это процесс, который позволяет координировать выполнение параллельных действий или потоков. В контексте бизнес-процессов и разработки программного обеспечения синхронизация важна для управления временем выполнения нескольких процессов, чтобы они могли взаимодействовать и обмениваться данными.

В диаграммах деятельности синхронизация визуализируется с помощью следующих элементов:

- Разделяющие линии (Forks)

Разделяющие линии используются для обозначения точки, в которой поток управления разделяется на несколько параллельных потоков. Они представляют собой горизонтальные или вертикальные линии, которые расходятся на две или более ветви.

- Слияния (Joins)

Слияние используется для обозначения точки, в которой параллельные потоки снова объединяются в один поток управления. Оно также отображается в виде горизонтальной или вертикальной линии.

- Параллельные действия

В диаграмме могут быть видны два или более действий, выполняемых одновременно, предоставляя наглядное представление о синхронизации в процессе.

## **8. Как можно оценить качество сгенерированного кода, полученного из диаграмм UML?**

Оценка качества сгенерированного кода из диаграмм UML можно проводить с помощью различных критериев и методов. Вот несколько подходов:

- Согласованность с диаграммами

Проверяйте, соответствует ли сгенерированный код структуре и концепциям, представленным в диаграммах UML (например, классы, последовательности, активности). Код должен отражать логику и архитектуру, обозначенные в моделях.

- Читаемость и поддерживаемость кода

Оцените, насколько код легко читаем и поддерживаем. Это включает в себя:

- Описание переменных и функций
- Использование ясных и понятных имен
- Правильное оформление и стилизация кода

- Отсутствие дублирования

Посмотрите на код на наличие дублирующихся фрагментов. Высокое качество кода предполагает, что повторяющиеся элементы должны быть вынесены в отдельные функции или классы.

- Тестируемость

Оцените, насколько легко тестировать сгенерированный код. Код должен быть структурирован таким образом, чтобы его можно было протестировать с помощью юнит-тестов и интеграционных тестов.

– Производительность

Проверьте, как эффективно работает сгенерированный код. Это можно сделать с помощью профилирования и других инструментов анализа производительности.

– Измерения качества кода

Используйте метрики, такие как:

- Коэффициент цикломатической сложности (Cyclomatic Complexity)
- Степень покрытия тестами
- Коэффициент тяжести кода (Code Churn)

– Документация

Убедитесь, что сгенерированный код имеет соответствующую документацию, объясняющую логику и структуру кода.

– Обратная связь от команды

Проведение код-ревью с участием команды может предложить разные мнения о качестве кода и выявить проблемы, которые могут быть неочевидны на первый взгляд.

## **9. Каковы основные выводы и уроки, которые можно извлечь из выполнения данной практической работы?**

Создание диаграммы вариантов использования помогает четко определить требования пользователей к системе. Это позволяет лучше понимать, какие функции необходимы для корзины, и как пользователи будут взаимодействовать с ней.

Построение диаграммы деятельности помогает проиллюстрировать рабочие процессы и последовательность действий при использовании корзины. Это позволяет обеспечить более плавный опыт для пользователя.

Диаграммы позволяют визуализировать взаимодействия между пользователями и системой. Это облегчает коммуникацию между членами

команды и заинтересованными сторонами, создавая общий взгляд на функциональность. Диаграммы служат хорошим инструментом для проверки логики бизнес-процессов и взаимодействий. На этом этапе можно выявить ошибки или недочеты еще до начала разработки.

Анализ интерфейсов корзины конкурентов позволяет создать такой интерфейс, где используется лучшее от каждого сайта, чтобы создать лучший интернет-магазин.