Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Технологии разработки программного обеспечения»

**ОТЧЕТ**

к лабораторной работе

на тему:

**«Проектирование и разработка ПО на языке UML»**

|  |  |
| --- | --- |
|  | Выполнил:  Слуцкий Никита Сергеевич,  студент группы 053505 |
|  | Проверил: Гриценко Никита Юрьевич, ассистент каф. Информатики |

Минск 2023

СОДЕРЖАНИЕ

[Введение 3](#_Toc147668195)

[1 Разработка диаграмм 4](#_Toc147668196)

[1.1 Диаграмма обзора (Interaction overview diagram) 4](#_Toc147668197)

[1.2 Диаграмма вариантов использования 5](#_Toc147668198)

[1.3 Диаграмма последовательности 6](#_Toc147668199)

[1.4 Диаграмма активности 7](#_Toc147668200)

[1.5 Диаграмма состояний 8](#_Toc147668201)

[1.6 Диаграмма классов 9](#_Toc147668202)

[Заключение 11](#_Toc147668203)

[Список используемых источников 12](#_Toc147668204)

# Введение

Проектирование диаграмм UML является важным этапом в разработке программного обеспечения. UML (Unified Modeling Language) [1] – это стандартизированный язык моделирования, который позволяет создавать визуальные модели для систем с интенсивным использованием программного обеспечения.

В UML 2.2 существует 14 типов диаграмм, которые делятся на две категории: 7 типов диаграмм представляют структурную информацию, а еще 7 представляют общие типы диаграмм UML для моделирования поведения, включая четыре, которые представляют различные аспекты взаимодействия.

Каждая диаграмма UML из 14 типов диаграмм UML предоставляет большой набор конструкций и обозначений, которые охватывают различные потребности для большинства проектов разработки программного обеспечения

Целью лабораторной работы №3 спроектировать 6 предложенных диаграмм UML:

* диаграмму обзора (Overview Diagram);
* диаграмму вариантов использования (Use Case Diagram);
* диаграмму последовательностей (Sequence Diagram);
* диаграмму активности (Activity Diagram);
* диаграмму состояний(State Diagram);
* диаграмму классов (Class Diagram).

Также целью ставится подготовить отчёт о проделанной работе, оформленный по стандарту предприятия БГУИР 2017 [2].

# 1 Разработка диаграмм

## Диаграмма обзора (Interaction overview diagram)

Диаграмма обзора (Overview Diagram) [3] – это диаграмма, которая показывает общую структуру системы и ее основные компоненты. Она является одной из диаграмм структуры UML и используется для представления общей архитектуры системы. Диаграмма обзора может быть использована для описания системы на высоком уровне, чтобы понять ее структуру и взаимодействие между компонентами.

Диаграмма обзора может содержать несколько компонентов, которые могут быть связаны друг с другом. Компоненты могут быть представлены в виде прямоугольников, а связи между ними – в виде стрелок. Диаграмма обзора может также содержать описание каждого компонента и его связей с другими компонентами. Примерная диаграмма обзора для разрабатываемого приложения отображена на рисунке 1.

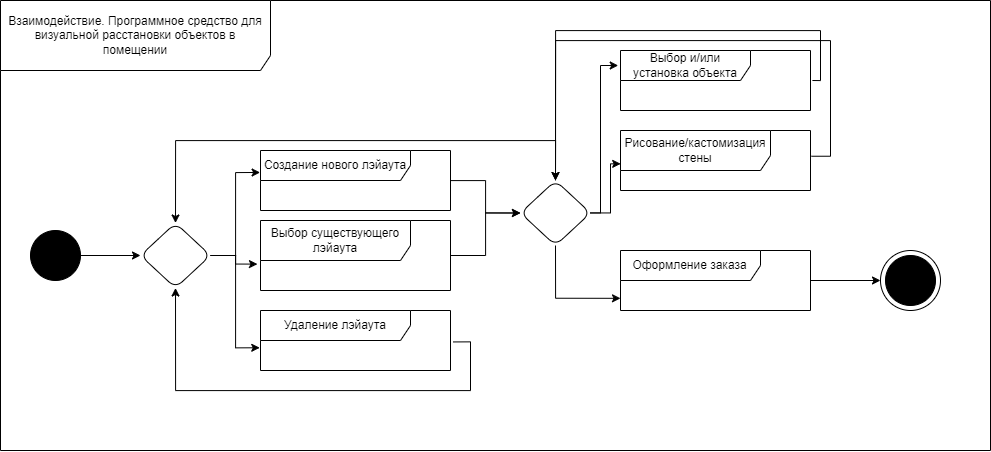


Рисунок 1 – Диаграмма обзора взаимодействия

Эти диаграммы наглядно описывают движение потока управления между взаимодействующими узлами. К последним относятся начальные узлы, конечные узлы потока или деятельности, а также узлы решений, слияния, разделения и соединения.

## 1.2 Диаграмма вариантов использования

Диаграмма вариантов использования (Use Case Diagram) [4] – это диаграмма, которая описывает функциональность системы с точки зрения ее пользователей. Она является одной из диаграмм поведения UML и используется для моделирования функциональных требований к системе. Диаграмма вариантов использования показывает, как различные типы пользователей (актеры) взаимодействуют с системой, чтобы выполнить определенные задачи.

Диаграмма вариантов использования состоит из актеров, вариантов использования и связей между ними. Актеры представляют различные типы пользователей, которые могут взаимодействовать с системой. Варианты использования описывают, как каждый актер использует систему для выполнения своих задач. Связи между актерами и вариантами использования показывают, как каждый актер использует каждый вариант использования 2.

Для создания диаграммы вариантов использования используются следующие элементы:

1 Актеры: пользователи системы, которые могут взаимодействовать с ней.

2 Варианты использования: описывают, как каждый актер использует систему для выполнения своих задач.

3 Связи: показывают, как каждый актер использует каждый вариант использования.

На рисунке 2 изображена диаграмма вариантов использования.

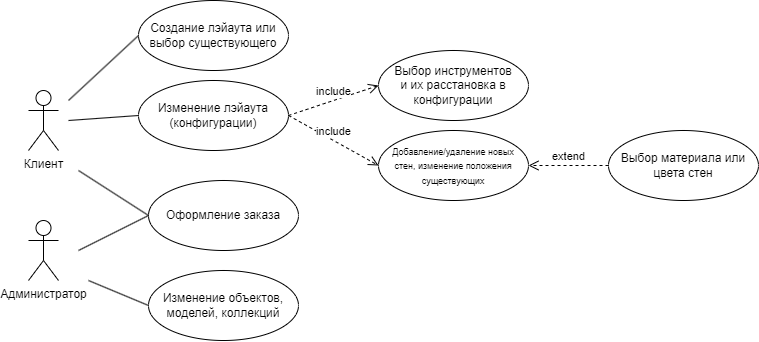


Рисунок 2 – Диаграмма вариантов использования

Для создания диаграммы вариантов использования можно использовать специальные графические символы и связи. Эти символы и связи помогают создать эффективную диаграмму, которая может быть легко понята другими членами команды.

## 1.3 Диаграмма последовательности

Диаграмма последовательности (Sequence Diagram) [5] – это диаграмма взаимодействия, которая показывает порядок взаимодействия между объектами в рамках одного сценария использования. Она описывает динамическое сотрудничество между несколькими объектами, показывая временной порядок, в котором сообщения передаются между ними. Диаграмма последовательности ориентирована на время и визуально показывает порядок взаимодействия, используя вертикальную ось диаграммы для представления времени, когда и какие сообщения отправляются.

Для создания диаграммы последовательности используются следующие элементы:

1 Объекты: представляют собой экземпляры классов, которые участвуют во взаимодействии.

2 Линии жизни: представляют собой вертикальные линии, которые представляют время жизни объекта.

3 Сообщения: представляют собой стрелки, которые указывают на передачу сообщений между объектами.

4 Активации: представляют собой прямоугольники, которые указывают на период времени, когда объект выполняет операцию.

На рисунке 3 представлена диаграмма последовательности действий со стороны клиента.

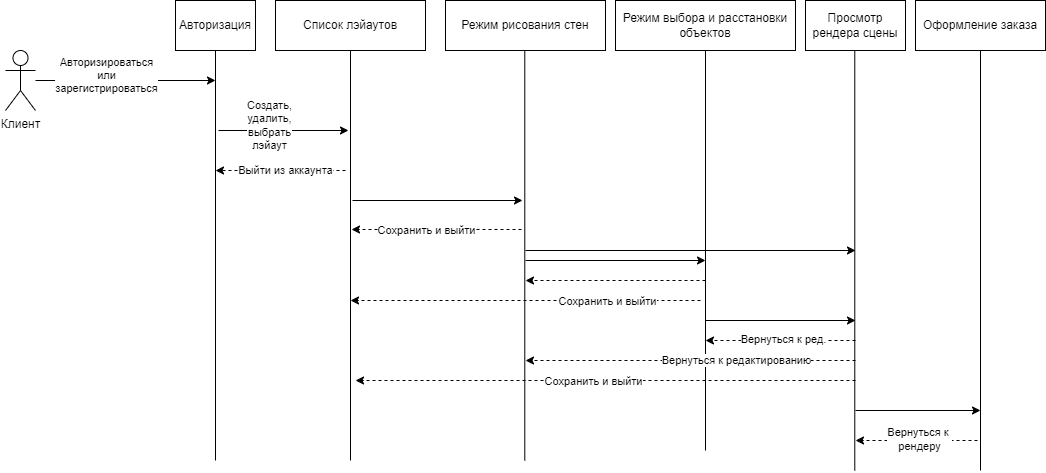


Рисунок 3 – Диаграмма последовательности

Диаграмма последовательности может использоваться для моделирования различных аспектов системы, таких как процесс обработки заказов или процесс регистрации пользователя.

Диаграммы последовательностей используются для уточнения диаграмм прецедентов, более детального описания логики сценариев использования.

Диаграммы последовательностей обычно содержат объекты, которые взаимодействуют в рамках сценария, сообщения, которыми они обмениваются, и возвращаемые результаты, связанные с сообщениями. Впрочем, часто возвращаемые результаты обозначают лишь в том случае, если это не очевидно из контекста.

## 1.4 Диаграмма активности

Диаграмма активности (Activity Diagram) [6] – это диаграмма поведения UML, которая показывает последовательность действий и поток управления от одного действия к другому.

Она описывает рабочий процесс пошаговых действий системы и иллюстрирует поток деятельности через систему. Диаграмма активности UML позволяет более детально визуализировать конкретный случай использования и может быть использована для отображения потока событий в бизнес-процессе.

Для создания диаграммы активности используются следующие элементы:

1 Действия: представляют собой шаги в рабочем процессе, которые выполняются пользователями или программным обеспечением.

2 Решения: показывают условные ветвления в потоке управления, которые могут привести к различным действиям.

3 Связи: показывают поток управления между действиями.

Для создания диаграммы активности можно использовать специальные графические символы и связи. Эти символы и связи помогают создать эффективную диаграмму, которая может быть легко понята другими членами команды.

На рисунке 4 изображена диаграмма активности взаимодействия клиента и сервера.

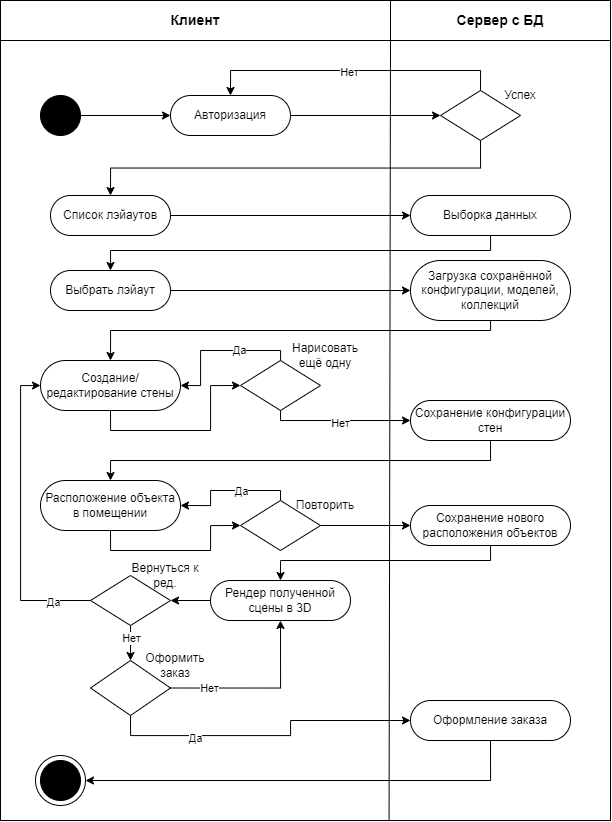


Рисунок 4 – Диаграмма активности

## 1.5 Диаграмма состояний

Диаграмма состояний (State Diagram) [7] – это диаграмма, которая показывает различные состояния, в которых может находиться объект, и переходы между этими состояниями.

Она является одной из диаграмм поведения UML и используется для моделирования поведения объектов в системе.

Для создания диаграммы состояний используются следующие элементы:

– состояния: представляют собой фазы или условия объекта;

– события: представляют собой внешние события, которые могут изменить состояние объекта;

– переходы: показывают, как объект переходит из одного состояния в другое;

– действия: представляют собой действия, которые выполняются при переходе из одного состояния в другое.

На рисунке 5 изображена примерная диаграмма состояний разрабатываемого приложения.

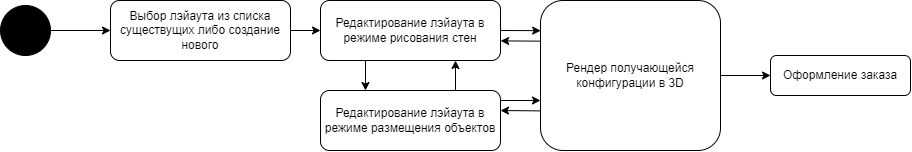


Рисунок 5 – Диаграмма состояний

Диаграмма состояний может использоваться для моделирования различных аспектов системы, таких как процесс обработки заказов или процесс регистрации пользователя.

## 1.6 Диаграмма классов

Диаграмма классов (Class Diagram) [8] – это диаграмма статической структуры, которая описывает структуру системы, показывая ее классы, их атрибуты, операции (или методы) и отношения между объектами. Диаграмма классов является одним из шести типов структурных диаграмм UML.

Для создания диаграммы классов используются следующие элементы:

– классы: представляют собой шаблоны для создания объектов, описывают свойства и методы объектов;

– атрибуты: представляют собой переменные, которые хранят данные объекта;

– операции: представляют собой функции, которые могут быть выполнены объектом;

– отношения: показывают связи между классами;

Диаграмма классов может использоваться для моделирования различных аспектов системы, таких как процесс обработки заказов или процесс регистрации пользователя.

Для создания диаграммы классов можно использовать специальные графические символы и связи. Эти символы и связи помогают создать эффективную диаграмму, которая может быть легко понята другими членами команды. На рисунке 6 представлена диаграмма классов.

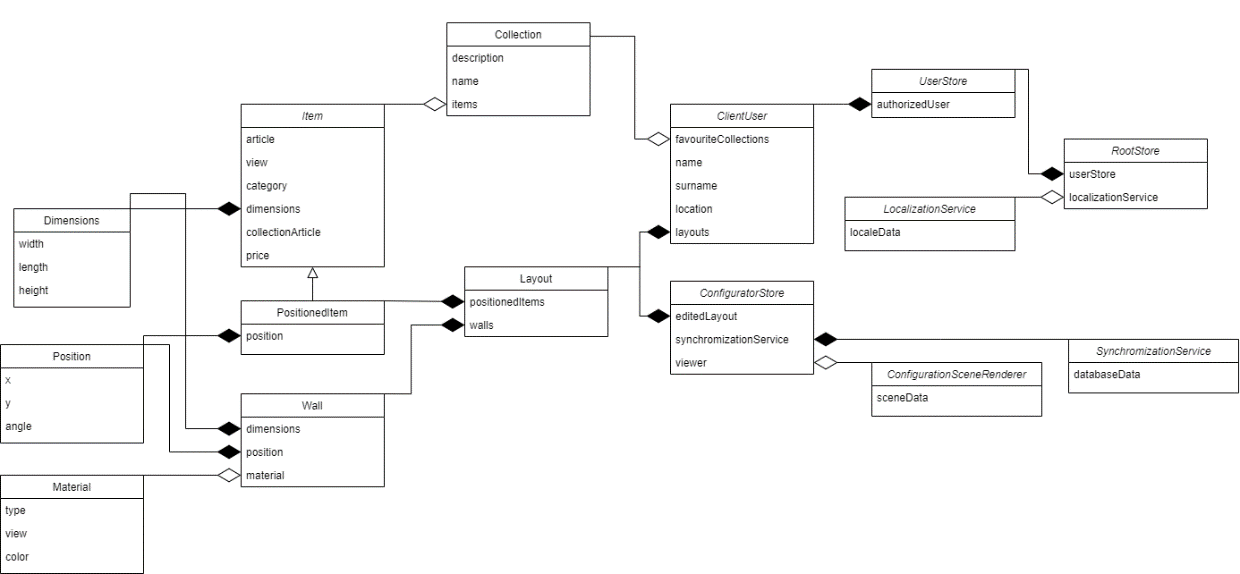


Рисунок 6 – Диаграмма классов

Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы (классов, типов и т. п.) Она содержит в себе также некоторые элементы поведения (например, операции), однако их динамика должна быть отражена на диаграммах других. Для удобства восприятия диаграмму классов можно также дополнить представлением пакетов, включая вложенные.

# Заключение

Каждая диаграмма UML имеет свои особенности и может использоваться для моделирования различных аспектов системы. Диаграмма обзора показывает общую структуру системы и ее основные компоненты. Диаграмма вариантов использования описывает функциональность системы с точки зрения ее пользователей. Диаграмма последовательностей описывает порядок взаимодействия между объектами в рамках одного сценария использования. Диаграмма активности показывает последовательность действий и поток управления от одного действия к другому. Диаграмма состояний показывает различные состояния, в которых может находиться объект, и переходы между этими состояниями. Диаграмма классов описывает структуру системы, показывая ее классы, их атрибуты, операции (или методы) и отношения между объектами.

Каждая диаграмма UML из 14 типов диаграмм UML предоставляет большой набор конструкций и обозначений, которые охватывают различные потребности для большинства проектов разработки программного обеспечения.

Были спроектированы схемы указанных диаграмм и подготовлен отчёт. Цели лабораторной работы можно считать достигнутыми.

# Список используемых источников

[1] UML-диаграммы [Электронный ресурс]. – Режим доступа: https://www.cybermedian.com/ru/a-comprehensive-guide-to-14-types-of-uml-diagram/. – Дата доступа: 06.10.2023

[2] СТП БГУИР 2017 [Электронный ресурс]. – Режим доступа: https://library.bsuir.by/ru/-standarty-po-diplomnomu-proektirovaniyu. – Дата доступа: 06.10.2023

[3] Диаграмма обзора [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/738428/. – Дата доступа: 06.10.2023

[4] Диаграмма вариантов использования [Электронный ресурс]. – Режим доступа: https://www.lucidchart.com/pages/uml-use-case-diagram. – Дата доступа: 07.10.2023

[5] Диаграмма последовательности [Электронный ресурс]. – Режим доступа: https://blog.visual-paradigm.com/ru/everything-you-need-to-know-about-sequence-diagrams/. – Дата доступа: 07.10.2023

[6] Диаграмма активности [Электронный ресурс]. – Режим доступа: https://www.lucidchart.com/pages/uml-activity-diagram. – Дата доступа: 08.10.2023

[7] Диаграмма состояний [Электронный ресурс]. – Режим доступа: http://it-gost.ru/articles/view\_articles/97. – Дата доступа: 08.10.2023

[8] Диаграмма классов UML [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/511798/. – Дата доступа: 08.10.2023