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](#_Toc147776658)

[1 Разработка диаграмм 4](#_Toc147776659)

[1.1 Диаграмма компонентов 4](#_Toc147776660)

[1.2 Диаграмма развёртывания 5](#_Toc147776661)

[1.3 Диаграмма объектов 7](#_Toc147776662)

[1.4 Диаграмма связей 8](#_Toc147776663)

[1.5 Диаграмма пакетов 9](#_Toc147776664)

[1.6 Диаграмма времени 10](#_Toc147776665)

[1.7 Диаграмма профиля 10](#_Toc147776666)

[Заключение 12](#_Toc147776667)

[Список используемых источников 13](#_Toc147776668)

# 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.

Целью лабораторной работы №4 спроектировать 7 предложенных диаграмм UML:

– диаграмму компонентов (Component Diagram);

– диаграмму развертывания (Deployment Diagram);

– диаграмму объектов (Object Diagram);

– диаграмму  связей (Relationship Diagram);

– диаграмму пакетов (Package Diagram);

– диаграмму времени (Timing Diagram);

– диаграмму профиля (Profile Diagram).

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

# 1 Разработка диаграмм

## Диаграмма компонентов

Диаграмма компонентов [3] – это один из элементов языка моделирования UML, который позволяет визуализировать организацию компонентов системы и зависимости между ними. Она используется для моделирования компонентов, которые помогают сделать функциональные возможности системы. Компонентами могут быть программные компоненты, такие как база данных или пользовательский интерфейс; или аппаратные компоненты, такие как схема, микросхема или устройство; или бизнес-подразделение, такое как поставщик, платежная ведомость или доставка. Диаграмма компонентов показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами.

Диаграммы компонентов используются для визуализации организации компонентов системы и зависимостей между ними. Они позволяют получить высокоуровневое представление о компонентах системы.

Компонентами могут быть программные компоненты, такие как база данных или пользовательский интерфейс; или аппаратные компоненты, такие как схема, микросхема или устройство; или бизнес-подразделение, такое как поставщик, платежная ведомость или доставка.

На диаграмме компонентов можно использовать следующие символы:

1 Компонент: прямоугольник со значком компонента и стереотипом компонента. Стереотип компонента обычно используется над именем компонента, чтобы не перепутать форму со значком класса. Компонент может быть представлен прямоугольником со значком компонента в правом верхнем углу и названием компонента. Или прямоугольником со стереотипом компонента (текст <<компонент>>).

2 Предоставляемый интерфейс и требуемый интерфейс: интерфейсы на диаграмме показывают, как компоненты соединены друг с другом и взаимодействуют друг с другом. Соединитель сборки позволяет соединить требуемый интерфейс компонента (представленный полукругом и сплошной линией) с предусмотренным интерфейсом (представленный окружностью и сплошной линией) другого компонента. Это показывает, что один компонент предоставляет услугу, которая требуется другому.

3 Порт: используется, когда компонент делегирует интерфейсы внутреннему классу.

4 Зависимости: стрелка зависимостей используется для показа отношений между двумя компонентами.

На рисунке 1 изображена диаграмма компонентов.

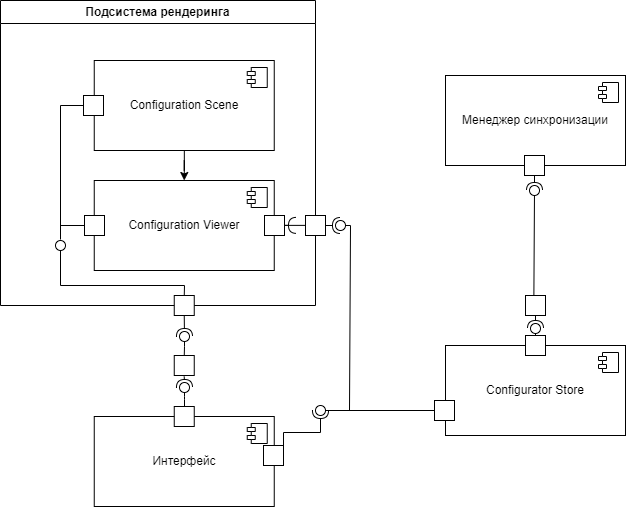


Рисунок 1 – Диаграмма обзора взаимодействия

Диаграмма компонентов – это, фактически, список артефактов, из которых состоит моделируемая система, с указанием некоторых отношений между ними.

## 1.2 Диаграмма развёртывания

Диаграмма развертывания [4] – это один из элементов языка моделирования UML, который позволяет визуализировать физическое развертывание компонентов системы и их взаимодействие. Она используется для моделирования физической архитектуры системы, показывая, как компоненты системы развертываются на устройствах и как они взаимодействуют друг с другом.

Диаграмма развертывания состоит из узлов и связей между ними. Узлы представляют физические устройства, на которых размещаются компоненты системы, такие как серверы, компьютеры, мобильные устройства и т.д. Связи между узлами показывают, как компоненты связаны друг с другом и как они обмениваются данными.

На диаграмме развертывания можно использовать следующие символы:

1 Узел: прямоугольник со значком узла и названием узла. Узел представляет физическое устройство, на котором размещаются компоненты системы.

2 Компонент: прямоугольник со значком компонента и названием компонента. Компонент представляет программный модуль или часть системы.

3 Связь: линия, которая показывает связь между двумя узлами или между узлом и компонентом. Связь может быть направленной или двунаправленной.

4 Артефакт: прямоугольник со значком артефакта и названием артефакта. Артефакт представляет файл или другой объект, который может быть развернут на узле.

На рисунке 2 изображена схематичная диаграмма развёртывания разрабатываемого приложения. Ввиду того, что оно представляет собой классическое веб приложение, схема является вполне ожидаемой.

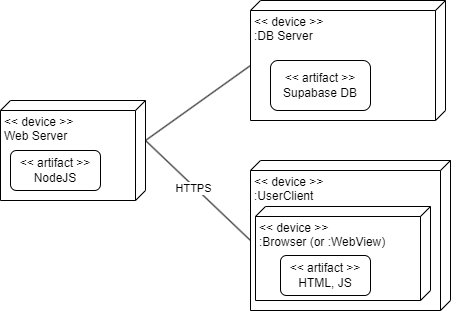


Рисунок 2 – Диаграмма развёртывания приложения

Чтобы усложнить диаграмму, также можно включить в нее разные составляющие веб-сервера и показать намного детальнее, как работает Javascript на UserClient.

## 1.3 Диаграмма объектов

Диаграмма объектов [5] – это один из элементов языка моделирования UML, который позволяет визуализировать объекты, которые используются в системе, и их взаимодействие. Она используется для моделирования структуры объектов в системе и отношений между ними.

Диаграмма объектов состоит из объектов и связей между ними. Объекты представляют экземпляры классов, которые используются в системе. Связи между объектами показывают, как объекты взаимодействуют друг с другом и как они обмениваются данными.

На диаграмме объектов можно использовать следующие символы:

– объект: прямоугольник со значком объекта и названием объекта. Объект представляет экземпляр класса, который используется в системе;

– связь: линия, которая показывает связь между двумя объектами. Связь может быть направленной или двунаправленной;

– роль: текст, который указывает на роль, которую играет объект в связи;

– мультипликатор: число, которое указывает на количество экземпляров класса, которые могут быть связаны с другим классом.

На рисунке 3 изображена диаграмма объектов.

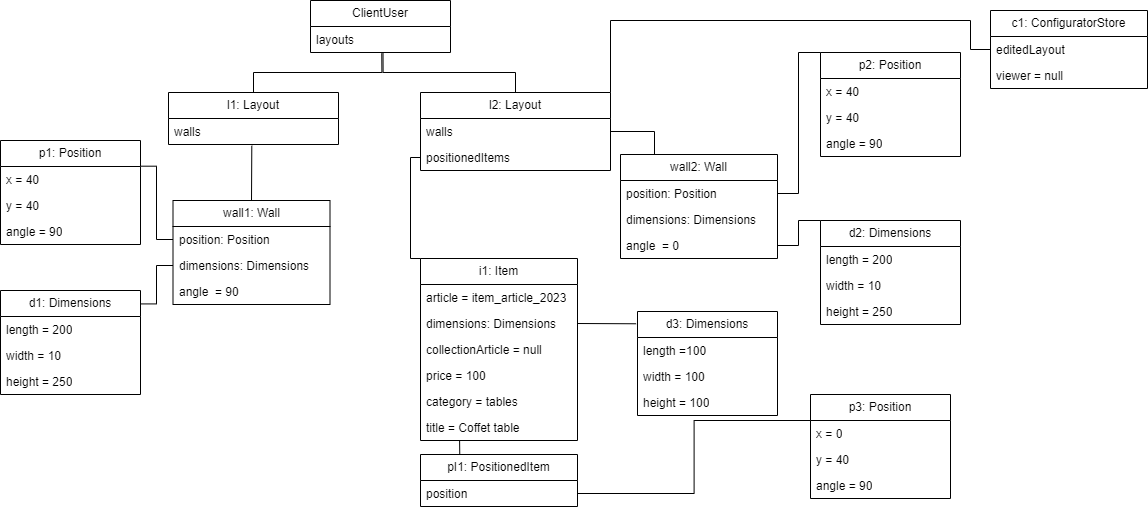


Рисунок 3 – Диаграмма объектов

Диаграмма объектов описывает конкретные экземпляры объектов и напрямую соотносится с диаграммой классов, которая даёт общее представление о конфигурации системы. Она используется для документирования структур данных и создания статических снимков состояний объектов принимая во внимание реальные экземпляры или прототипы. Динамику поведения объектов обычно изображают в виде последовательности таких диаграмм.

## 1.4 Диаграмма связей

Диаграмма связей [6] – это один из элементов языка моделирования UML, который позволяет визуализировать отношения между объектами в системе. Она используется для моделирования структуры системы, показывая, как объекты взаимодействуют друг с другом и как они обмениваются данными.

Диаграмма связей состоит из объектов и связей между ними. Объекты представляют экземпляры классов, которые используются в системе. Связи между объектами показывают, как объекты взаимодействуют друг с другом и как они обмениваются данными.

На диаграмме связей можно использовать следующие символы:

– объект: прямоугольник со значком объекта и названием объекта. Объект представляет экземпляр класса, который используется в системе;

– связь: линия, которая показывает связь между двумя объектами. Связь может быть направленной или двунаправленной;

– роль: текст, который указывает на роль, которую играет объект в связи;

– мультипликатор: число, которое указывает на количество экземпляров класса, которые могут быть связаны с другим классом.

На рисунке 4 изображена диаграмма связей.

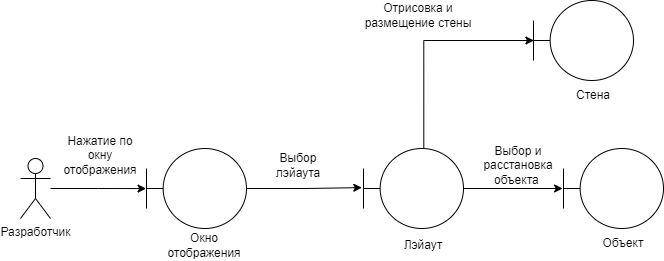


Рисунок 4 – Диаграмма связей

Диаграмма имеет схематичный вид, может расширяться при необходимости.

## 1.5 Диаграмма пакетов

Диаграмма пакетов [7] – это один из элементов языка моделирования UML, который позволяет визуализировать организацию пакетов в системе и зависимости между ними. Пакеты используются для группировки элементов модели и предоставления пространства имен для сгруппированных элементов. Пакет может содержать другие пакеты, что обеспечивает иерархическую организацию пакетов.

Диаграмма пакетов состоит из пакетов и связей между ними. Пакеты представляют собой группы элементов модели, которые могут быть классами, объектами, компонентами, узлами, экземплярами узлов и т.д24. Связи между пакетами показывают зависимости между ними и могут быть направленными или двунаправленными.

На диаграмме пакетов можно использовать следующие символы:

1 Пакет: прямоугольник со значком пакета и названием пакета. Пакет представляет группу элементов модели.

2 Связь: линия, которая показывает связь между двумя пакетами. Связь может быть направленной или двунаправленной.

3 Импортирование пакета: отношение между импортирующим пространством имен и пакетом, указывающим на то, что импортирующее пространство имен добавляет имена членов пакета в свое собственное пространство имен.

4 Слияние пакета: направленное отношение между двумя пакетами, которое указывает на то, что содержимое двух пакетов должно быть объединено.

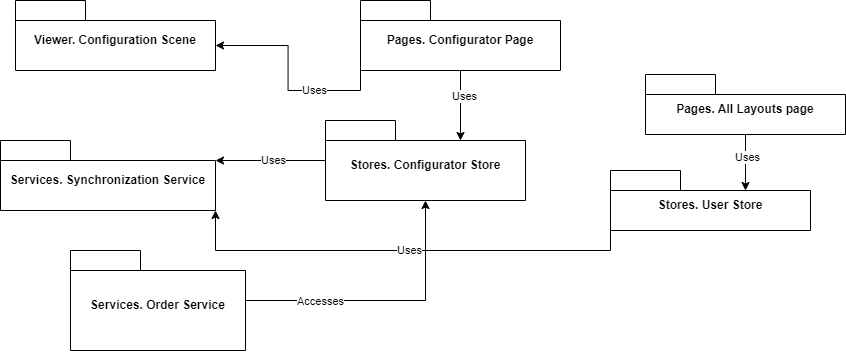


Рисунок 5 – Диаграмма пакетов

Таким образом можно видеть пакеты, как наборы схожих по семантике модулей системы.

## 1.6 Диаграмма времени

Временная диаграмма (Timing Diagram) [8] – это один из 14 типов диаграмм UML и относится к категории диаграмм моделирования поведения. Она используется для отображения изменений состояний объектов во времени и показывает, как объекты взаимодействуют друг с другом во времени. Во времени могут взаимодействовать не только объекты в приложении, но и, например, что актуально для веб-приложении, процесс рендеринга всей веб-страницы.

На рисунке для примера изображена диаграмма времени приложения, отображающая ход рендеренга приложения в браузере. Зная чётко процесс, время на каждом этапе, можно очень неплохо оптимизировать работу веб-приложения за счёт переносов изменения визуальной части на более высокие уровни и использования других техник, описания которых не входят в цели данной лабораторной работы.

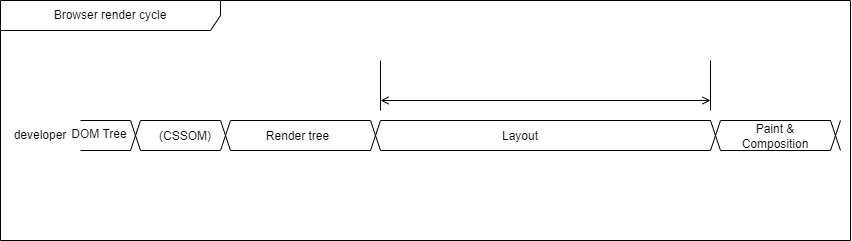


Рисунок 6 – Диаграмма времени рендеринга веб-приложения

Временные диаграммы используются для изучения поведения процессов или объектов в течение заданного периода времени.

## 1.7 Диаграмма профиля

Профильная диаграмма [9] – это структурная диаграмма в языке UML, которая позволяет расширять и настраивать модели UML для конкретных областей и платформ. Она предоставляет общий механизм расширения для настройки моделей UML для определенных областей и платформ. Профильная диаграмма определяется с помощью стереотипов, определений теговых значений и ограничений, которые применяются к конкретным элементам модели, таким как классы, атрибуты, операции и действия. Профиль – это коллекция таких расширений, которые вместе настраивают UML для определенной области (например, авиакосмической, медицинской, финансовой) или платформы (J2EE, .NET). На рисунке 7 изображена диаграмма профиля.

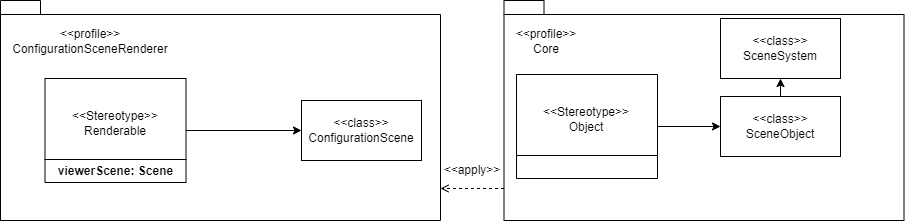


Рисунок 7 – Диаграмма профиля

Диаграмма профиля позволяет расширить стандартную нотацию UML путем создания новых элементов и отношений, которые отражают специфические требования и особенности предметной области проекта. Это особенно полезно, когда нотация UML не обладает достаточным набором символов, чтобы описать все аспекты системы.

Примером использования диаграммы профиля может быть разработка модели для конкретной отрасли, например, финансового сектора или здравоохранения. В этих отраслях могут быть свои уникальные аспекты и правила, которые необходимо учесть при моделировании системы. Диаграмма профиля позволяет вам создавать новые символы, отношения и правила, которые соответствуют этим особенностям.

# Заключение

В ходе выполнения лабораторной работы были изучили различные виды диаграмм в языке UML, которые позволяют моделировать структуру и поведение программного обеспечения. Были разработали диаграммы компонентов, развертывания, объектов, связей, пакетов, времени и профиля для разрабатываемого проекта, использованы стандартные элементы и нотации UML для представления различных аспектов ПО.

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: 08.10.2023

[2] СТП БГУИР 2017 [Электронный ресурс]. – Режим доступа: https://library.bsuir.by/ru/-standarty-po-diplomnomu-proektirovaniyu. – Дата доступа: 08.10.2023

[3] Диаграмма компонентов [Электронный ресурс]. – Режим доступа: https://www.smartdraw.com/component-diagram. – Дата доступа: 08.10.2023

[4] Диаграмма развертывания [Электронный ресурс]. – Режим доступа: https://www.lucidchart.com/pages/ru/uml-deployment-diagram. – Дата доступа: 08.10.2023

[5] Диаграмма объектов [Электронный ресурс]. – Режим доступа: https://www.lucidchart.com/pages/uml-object-diagram. – Дата доступа: 09.10.2023

[6] Диаграмма связей [Электронный ресурс]. – Режим доступа: https://www.lucidchart.com/pages/templates/entity-relationship-diagram-example-uml-notation. – Дата доступа: 09.10.2023

[7] Диаграмма пакетов [Электронный ресурс]. – Режим доступа: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/. – Дата доступа: 09.10.2023

[8] Диаграмма времени [Электронный ресурс]. – Режим доступа: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-timing-diagram/. – Дата доступа: 09.10.2023

[9] Диаграмма профиля [Электронный ресурс]. – Режим доступа: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-profile-diagram/. – Дата доступа: 09.10.2023