МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

Факультет компьютерных технологий и прикладной математики

**Кафедра информационных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №4.1**

**ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ ПРОГРАММНОГО ПРОДУКТА: ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД**

Работу выполнил А.А. Козин

(подпись)

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность Программирование и информационные технологии

Руководитель

канд. пед. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.Ю. Добровольская

(подпись)

Краснодар

2023

# Оглавление

[Оглавление 2](#_Toc130203369)

[Введение 3](#_Toc130203370)

[Абстракция подсистемы 4](#_Toc130203371)

[Классификация абстракций 5](#_Toc130203372)

[Диаграмма классов 6](#_Toc130203373)

[Диаграммы состояний 7](#_Toc130203374)

[Диаграмма деятельности 8](#_Toc130203375)

[Диаграммы последовательности 9](#_Toc130203376)

# Введение

Цель: изучение методики объектно-ориентированного подхода программной инженерии для разработки и описания функциональности разрабатываемого программного обеспечения.

# Абстракция подсистемы

Выявим набор абстракций предметной области проектируемой ПС. Разделим выделенные в проекте абстракции на три типа:

* Абстракции сущности;
* Абстракции поведения;
* Абстракции интерфейса.

Выделить возможное поведение каждой абстракции в пределах функциональности, проектируемой ИС, представленной моделью требований UML

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракции | Тип | Описание |
| 1 | Данные о клиенте | Сущность | Паспортные данные |
| 2 | Кассир | Сущность | Пользователь системы, взаимодействующий с системой |
| 3 | Товарная накладная | Сущность | Список покупок, данные о клиенте, стоимость курса |
| 4 | Заявка о покупке | Сущность | Список покупок, ограничение по стоимости, дата |
| 5 | Запись о проведённой продаже в журнал отчёта | Поведение | Формирование записи в соответствии с проведённой операцией продажи и входных данных клиента. |

Таблица 1 – абстракция подсистемы.

# Классификация абстракций

Проведём классификацию обнаруженных абстракций по классическому подходу:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Класс | Тип | Описание |
| 1 | Человек | Сущность | Кассир |
| 2 | Предметы | Поведение | Данные о клиенте, заявка на покупку, товарная накладная. |
| 3 | События | Сущность | Запись о проведённой продаже в журнал, формирование накладной, обработка заявки |

Таблица 2 – классификация абстракций подсистемы.

# Диаграмма классов

Диаграмма классов — структурная диаграмма языка моделирования UML, которая демонстрирует общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

Эта диаграмма дает обзор программной системы путем отображения классов, атрибутов, операций и взаимосвязей.

Построим диаграмму классов UML.

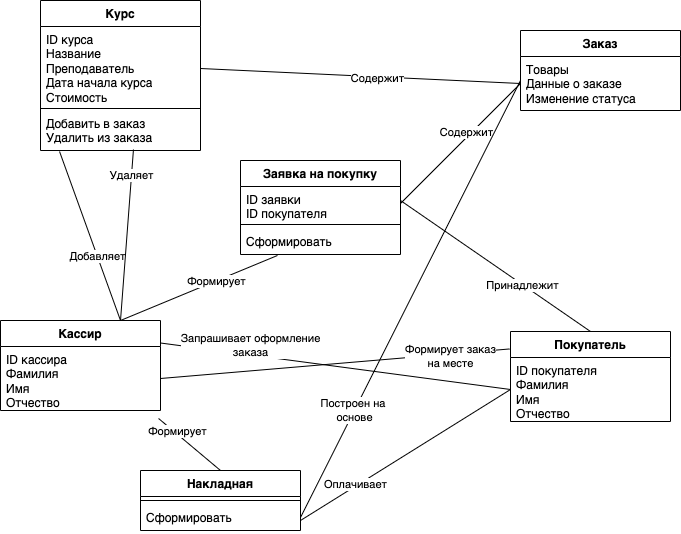


Рисунок 1 – диаграмма классов.

# Диаграммы состояний

Диаграмма состояний — это способ моделирования поведения системы, который показывает, как система может находиться в различных состояниях в зависимости от различных входных событий и условий. Она является частью языка моделирования UML, который широко используется для разработки программного обеспечения.

Диаграмма состояний представляет систему в виде набора состояний, переходов между ними и действий, которые выполняются при переходе между состояниями.

Самые изменяемые классы из построенной диаграммы классов являются «Заказ» и «Накладная», т. к. зависят от множества других факторов.

Построим диаграмму состояний на основе выбранных классов.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 2 – диаграммы состояний.

# Диаграмма деятельности

Диаграмма деятельности — это графическое представление процесса или системы, которое отображает последовательность действий, которые должны быть выполнены, чтобы достичь определенной цели. Она используется в моделировании бизнес-процессов, программном проектировании и других областях.

Диаграмма деятельности позволяет визуализировать действия, переходы и условия, связанные с выполнением процесса или системы. На диаграмме деятельности могут быть представлены как последовательные шаги, так и параллельные процессы. Она может использоваться для анализа, проектирования и документирования бизнес-процессов, а также для описания функциональных требований к программному обеспечению.

Построим диаграмму деятельности.

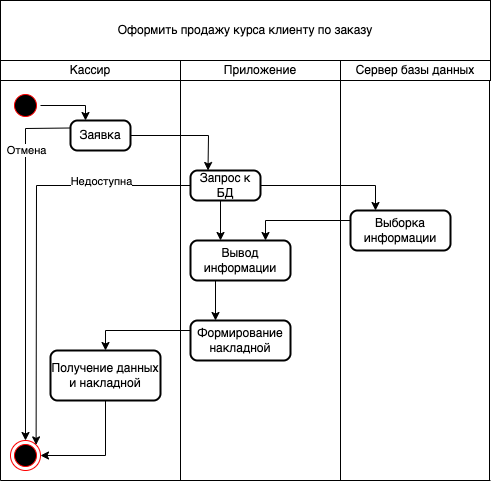


Рисунок 3 – диаграмма деятельности.

# Диаграммы последовательности

Диаграмма последовательности — это графическое представление взаимодействия объектов в системе, показывающее последовательность сообщений, которые объекты посылают друг другу в определенной ситуации или в рамках определенного сценария.

Диаграмма последовательности включает в себя вертикальные линии, которые представляют различные объекты в системе, а также стрелки, которые показывают порядок передачи сообщений между этими объектами. Каждый объект на диаграмме может иметь свою жизненную линию, на которой отображается его состояние во время выполнения операции. Она позволяет легко понять последовательность выполнения операций и выявить потенциальные проблемы в процессе взаимодействия объектов.

Для перечисленных прецедентов создадим диаграмму последовательности.

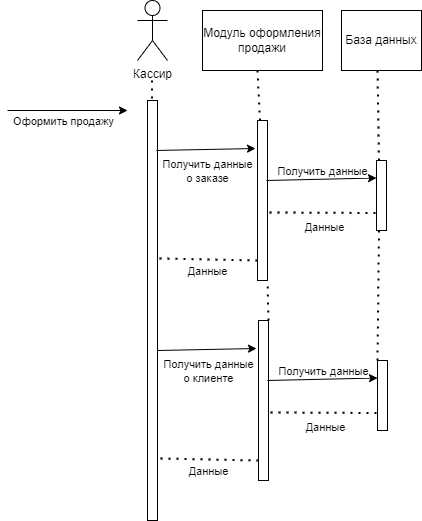


Рисунок 4 – диаграмма последовательности.