**Оглавление**

[1. Техническое задание 3](#_Toc36415178)

[1.1 Введение 3](#_Toc36415179)

[1.2 Термины и сокращения 4](#_Toc36415180)

[1.3 Основание для разработки 7](#_Toc36415181)

[1.4 Назначение разработки 7](#_Toc36415182)

[1.5 Требования к программному продукту 7](#_Toc36415183)

[2. UML моделирование 8](#_Toc36415184)

[2.1 Диаграмма вариантов использования 8](#_Toc36415185)

[2.2 Диаграмма классов 10](#_Toc36415186)

[2.3 Диаграмма активности 11](#_Toc36415187)

[2.4 Диаграмма компонент 11](#_Toc36415188)

[3. Тестирование 12](#_Toc36415189)

[3.1 Тестирование DataProviderCsv 12](#_Toc36415190)

[3.1 Тестирование DataProviderJdbc 12](#_Toc36415191)

[3.1 Тестирование DataProviderXml 13](#_Toc36415192)

# **Техническое задание**

## **1.1 Введение**

Предметная область курсовой работы – почтовые услуги. Java библиотека предназначена для управления почтовыми отправлениями и включает в себя:

1. Операции с почтовыми отправлениями:
   1. Добавление отправления;
   2. Удаление отправления;
   3. Получение записи о посылке по id;
   4. Поиск посылок по номеру;
2. Операции с клиентами:
   1. Добавление клиента;
   2. Удаление клиента;
   3. Получение записи о клиенте по идентификатору;
   4. Получение статистики клиента;
3. Операции с конвертами, марками, валютами:
   1. Добавление записи;
   2. Удаление записи;
   3. Получение записи по id;

API библиотека содержит функции, предназначенные для упрощения управления почтовыми услугами. Они автоматизируют обработку данных о посылках, поиск и отслеживание отправлений.

## **1.2 Термины и сокращения**

Термины и сокращения приведены в таблице 1.

Таблица 1 – Термины и сокращения

|  |  |  |
| --- | --- | --- |
| **Термин** | **Сокращение** | **Описание** |
| Extensible Markup Language | XML | Текстовый формат, предназначенный для хранения структурированных данных и обмена информацией между программами |
|  |  | Объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине вне зависимости от компьютерной архитектуры. |
| Unified Modeling Language | UML | Язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. |
| Create, read, update, delete | CRUD | Акроним, обозначающий четыре базовые функции, используемые при работе с персистентными хранилищами данных:   * Создание; * Чтение; * Редактирование; * Удаление. |
| Extensible Markup Language | XML | Расширяемый язык разметки документов, позволяющий структурировать информацию разного типа, используя для этого произвольный набор инструкций. |
| Comma-Separated Values | CSV | Значения, разделённые запятыми — текстовый формат, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми. |
| Java database Connectivity | JDBC | Соединение с базами данных на [Java](https://ru.wikipedia.org/wiki/Java" \o "Java) — платформенно независимый промышленный стандарт взаимодействия Java-приложений с различными [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94), реализованный в виде пакета java.sql, входящего в состав [Java SE](https://ru.wikipedia.org/wiki/Java_SE" \o "Java SE). |
| Command line interface | CLI | Интерфейс командной строки — разновидность [текстового интерфейса](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) (CUI) между человеком и компьютером, в котором инструкции компьютеру даются в основном путём ввода с клавиатуры текстовых строк (команд). |
| Structured Query Language | SQL | Структурированный язык запросов — язык управления базами данных для реляционных баз данных. |
| Postgresql |  | [Свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [объектно-реляционная](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) [система управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). |
| Junit |  | [Библиотека](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) для [модульного тестирования](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) программного обеспечения на языке [Java](https://ru.wikipedia.org/wiki/Java" \o "Java). |
| Система управления базами данных | СУБД | Совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). |

## **1.3 Основание для разработки**

Основанием для разработки является выполнение курсовой работы по дисциплине «Архитектура информационных систем», результатом которой будет создание интерфейса приложения на языке Java, позволяющего выполнять основные операции для различных источников данных (создание, получение, обновление и удаление записи любого объекта предметной области), а также получение статистики клиентов и создание отметки о выдаче посылки.

## **1.4 Назначение разработки**

Назначением программы является автоматизация управления (добавление, удаление и получение информации) и получения статистики клиентов.

## **1.5 Требования к программному продукту**

Продукт должен представлять собой Java библиотеку позволяющую выполнять основные CRUD операции для различных источников данных посредством различных интерфейсов.

Необходимо реализовать возможности отслеживания посылок, хранить информацию о почтовых отправлениях, вести статистику клиентов.

Результат представлен в виде интерфейса взаимодействия с Java библиотекой.

Необходимыми источниками данных были определены CSV, XML и SQL-базы данных. В качестве СУБД используется PostgreSQL.

Пользовательский интерфейс — консольный.

Для тестирования продукта была использована библиотека JUnit4.

# **UML моделирование**

Для визуального описания объектного моделирования было выбрано UML моделирование.

## **2.1 Диаграмма вариантов использования**



Рисунок 1 — Диаграмма вариантов использования

На основании технического задания создана диаграмма вариантов использования, представленная на рисунке 1, детализация каждого варианта использования представлена в таблице 2.

Таблица 2 — Детализация вариантов использования приложения

|  |  |
| --- | --- |
| **Вариант использования** | **Описание** |
| Проверить клиента на существование. | Получение записи о клиенте по его идентификатору, проверка на существование клиента с таким идентификатором в базе. |
| Добавить денежный перевод | Создание записи о денежном переводе, внесение суммы перевода. |
| Добавить ценную посылку. | Создание записи о посылке с добавленной ценностью. |
| Добавить почтовое отправление | Добавление почтового отправления по выбранному типу. |
| Добавить клиента | Добавление нового клиента, внесение информации о клиенте. |
| Добавить запись | Возможность добавить запись об отправлении и сопутствующую информацию. |
| Получить запись по ID | Возможность получить информацию из любой записи по id. |
| Удалить запись | Возможность удалить любую запись по id. |
| Добавить письмо | Возможность добавить запись о новом письме, также добавить конверт и марку из существующих товаров. |
| Проверить марку | Получение марки по идентификатору, проверка на существование в базе. |
| Проверить конверт | Получение конверта по идентификатору, проверка на существование в базе. |
| Проверить офис почты на существование. | Получение записи об офисе почты по его идентификатору, проверка на существование офиса почты с таким идентификатором в базе. |
| Добавить посылку | Возможность добавления новой записи о посылке в пакете или коробке. Требуется указать размеры посылки и хрупкость. |
| Вывести статистику клиента | Вывод всех посылок, которые получил или отправил клиент. |
| Выдать посылку | Отметка о получении посылки. |

## **2.2 Диаграмма классов**

В соответствии с диаграммой вариантов использования была построена диаграмма классов (рис. 2).

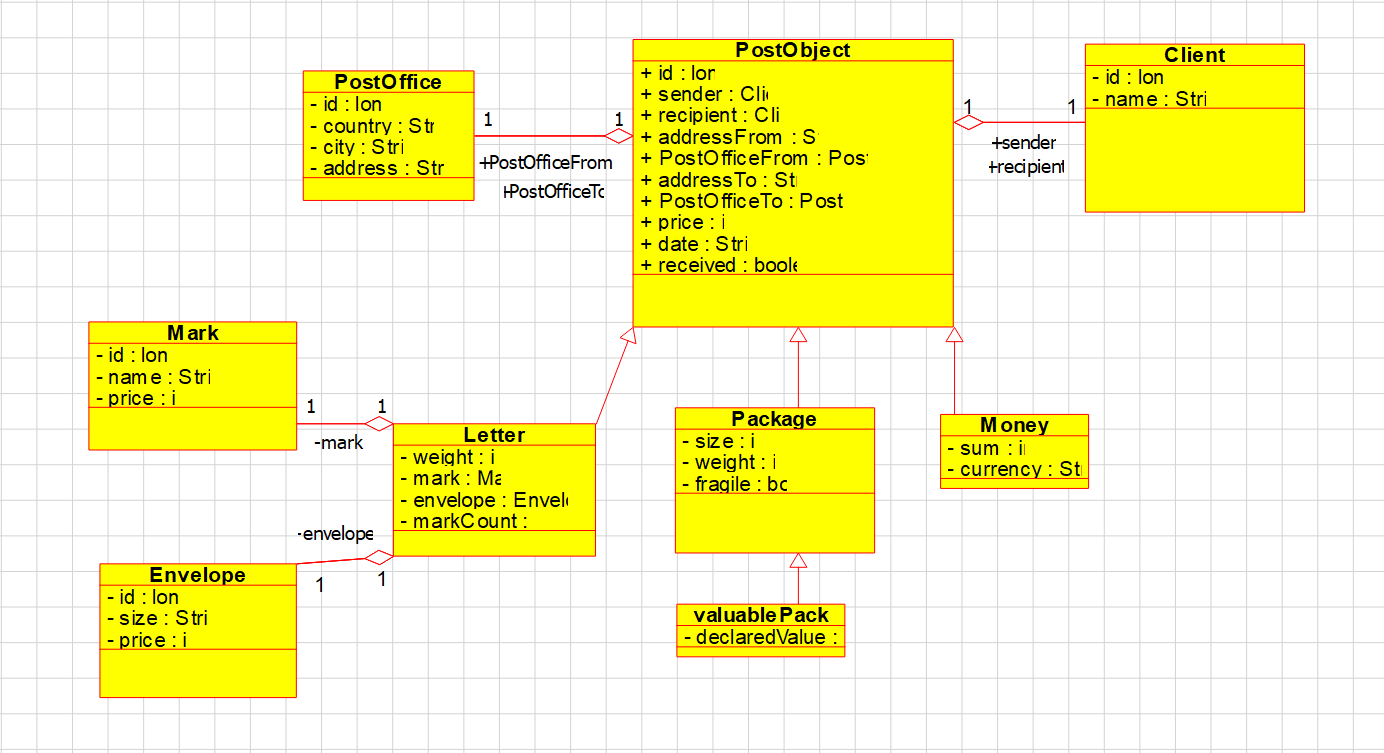


Рисунок 2 — Диаграмма классов

На диаграмме классов отражены следующие стержневые сущности:

1. Клиент (Client);
2. Почтовое отделение (PostOffice);
3. Марка (Mark)
4. Конверт (Envelope)
5. Почтовое отправление (PostObject)

Класс PostObject обобщает конкретные типы почтовых отправлений: письмо, посылка, денежный перевод.

## **2.3 Диаграмма активности**

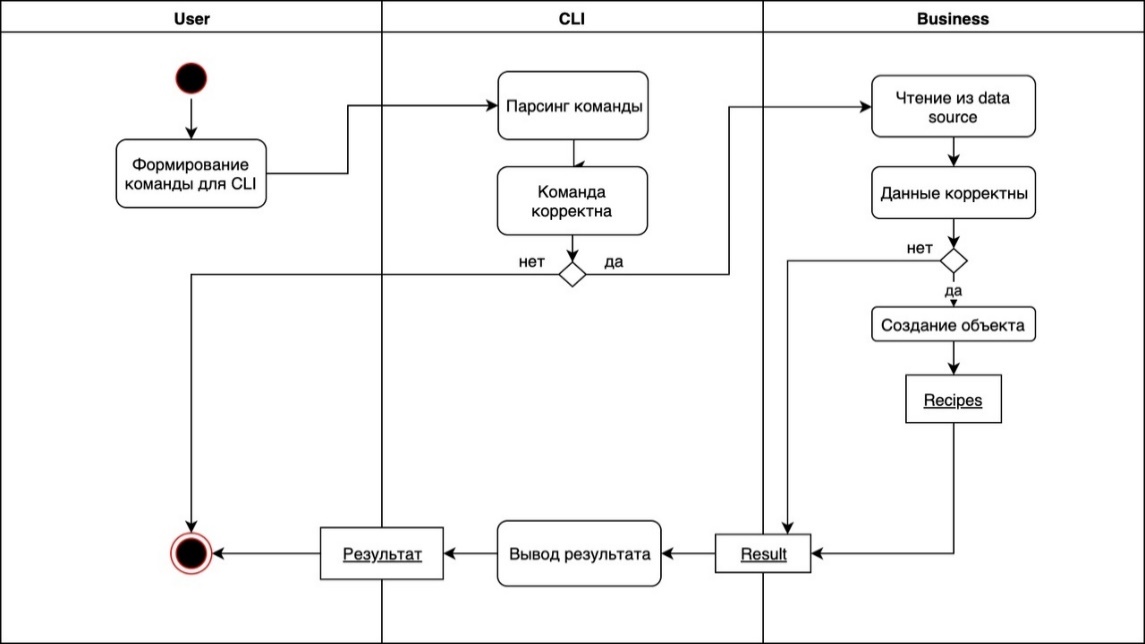


Рисунок 3 — Диаграмма активности

## **2.4 Диаграмма компонент**

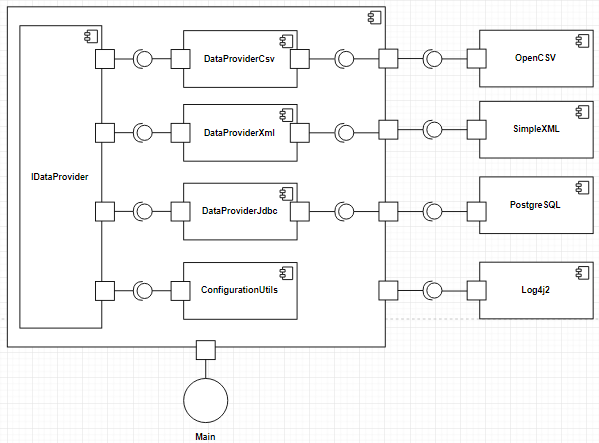


Рисунок 4 — Диаграмма компонент

# **Тестирование**

## **3.1 Тестирование DataProviderCsv**

При тестировании класса DataProviderCsv были протестированы все имплементированные методы CRUD-операций для каждой стержневой сущности, также были протестированы варианты использования.

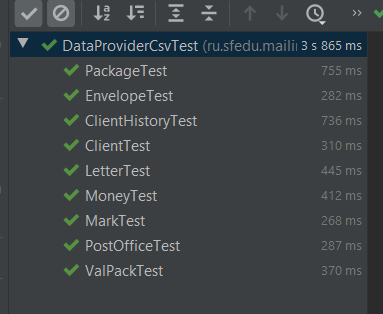


Рисунок 5 — Результаты тестирования DataProviderCsvTest

Успешно выполнены 9 тестов, время выполнения 2 секунды.

## **3.1 Тестирование DataProviderJdbc**

При тестировании класса DataProviderJdbc были протестированы все имплементированные методы CRUD-операций для каждой стержневой сущности, также были протестированы варианты использования.

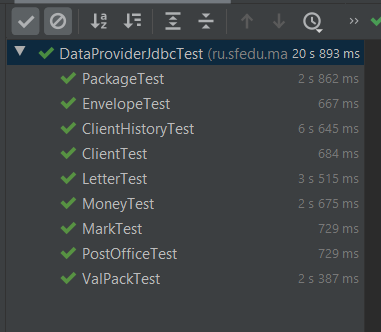


Рисунок 6 — Результаты тестирования DataProviderJdbcTest

Успешно выполнены 9 тестов, время выполнения 20 секунд.

## **3.1 Тестирование DataProviderXml**

При тестировании класса DataProviderXml были протестированы все имплементированные методы CRUD-операций для каждой стержневой сущности, также были протестированы варианты использования.

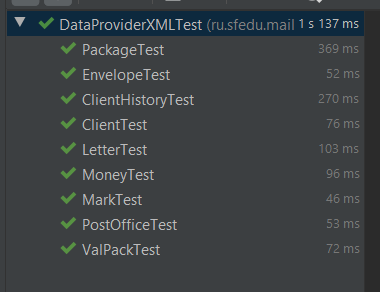


Рисунок 7 — Результаты тестирования DataProviderXmlTest

Успешно выполнены 9 тестов, время выполнения 1 секунда.