

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники Направление подготовки
09.03.04 Программная инженерия Дисциплина «Информационные системы»

ОТЧЕТ

По лабораторной работе

Вариант 16161

Преподаватель:

Тюрин Иван Николаевич

Выполнил:

Ильин Никита

Группа:

P3310

Санкт-Петербург
2025 г.

Задание

Реализовать информационную систему, которая позволяет взаимодействовать с объектами класса `Organization`, описание которого приведено ниже:

```
public class Organization {
    private Integer id; //Поле не может быть null, Значение поля должно быть
    больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно
    генерироваться автоматически
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Coordinates coordinates; //Поле не может быть null
    private java.util.Date creationDate; //Поле не может быть null, Значение
    этого поля должно генерироваться автоматически
    private Address officialAddress; //Поле не может быть null
    private float annualTurnover; //Значение поля должно быть больше 0
    private Long employeesCount; //Поле не может быть null, Значение поля должно
    быть больше 0
    private double rating; //Значение поля должно быть больше 0
    private String fullName; //Строка не может быть пустой, Поле не может быть
    null
    private OrganizationType type; //Поле может быть null
    private Address postalAddress; //Поле не может быть null
}
public class Coordinates {
    private int x; //Значение поля должно быть больше -524
    private Float y; //Максимальное значение поля: 476, Поле не может быть null
}
public class Address {
    private String street; //Строка не может быть пустой, Поле не может быть null
    private String zipCode; //Поле может быть null
}
public class Location {
    private double x;
    private Double y; //Поле не может быть null
    private String name; //Длина строки не должна быть больше 492, Поле не может
    быть null
}
public enum OrganizationType {
    COMMERCIAL,
    PUBLIC,
    TRUST,
    PRIVATE_LIMITED_COMPANY;
}
```

Разработанная система должна удовлетворять следующим требованиям:

Основное назначение информационной системы - управление объектами, созданными на основе заданного в варианте класса.

Необходимо, чтобы с помощью системы можно было выполнить следующие операции с объектами: создание нового объекта, получение информации об объекте по ИД, обновление объекта (модификация его атрибутов), удаление объекта. Операции должны осуществляться в отдельных окнах (интерфейсах) приложения. При получении информации об объекте класса должна также выводиться информация о связанных с ним объектах.

При создании объекта класса необходимо дать пользователю возможность связать новый объект с объектами вспомогательных классов, которые могут быть связаны с

созданным объектом и уже есть в системе.

Выполнение операций по управлению объектами должно осуществляться на серверной части (не на клиенте), изменения должны синхронизироваться с базой данных.

На главном экране системы должен выводиться список текущих объектов в виде таблицы (каждый атрибут объекта - отдельная колонка в таблице). При отображении таблицы должна использоваться пагинация (если все объекты не помещаются на одном экране). Нужно обеспечить возможность фильтровать/сортировать строки таблицы, которые показывают объекты (по значениям любой из строковых колонок). Фильтрация элементов должна производиться только по полному совпадению.

Переход к обновлению (модификации) объекта должен быть возможен из таблицы с общим списком объектов и из области с визуализацией объекта (при ее реализации).

При добавлении/удалении/изменении объекта, он должен автоматически появиться/исчезнуть/измениться в интерфейсах у других пользователей, авторизованных в системе.

Если при удалении объекта с ним связан другой объект, связанные объекты должны удаляться.

Пользователи должны иметь возможность просмотра всех объектов. Для модификации объекта должно открываться отдельное диалоговое окно. При вводе некорректных значений в поля объекта должны появляться информативные сообщения о соответствующих ошибках.

В системе должен быть реализован отдельный пользовательский интерфейс для выполнения специальных операций над объектами:

Вернуть количество объектов, значение поля rating которых равно заданному.

Вернуть массив объектов, значение поля fullName которых начинается с заданной подстроки.

Вернуть массив объектов, значение поля fullName которых больше заданного.

Вывести 5 организаций с максимальным годовым оборотом

Найти среднее количество сотрудников для 10 крупнейших организаций по годовому обороту

Представленные операции должны быть реализованы в рамках компонентов бизнес-логики приложения без прямого использования функций и процедур БД.

Особенности хранения объектов, которые должны быть реализованы в системе:

Организовать хранение данных об объектах в реляционной СУБД (PostgreSQL). Каждый объект, с которым работает ИС, должен быть сохранен в базе данных.

Все требования к полям класса (указанные в виде комментариев к описанию классов) должны быть выполнены на уровне ORM и БД.

Для генерации поля id использовать средства базы данных.

Для подключения к БД на кафедральном сервере использовать хост pg, имя базы данных - studs, имя пользователя/пароль совпадают с таковыми для подключения к серверу.

При создании системы нужно учитывать следующие особенности организации взаимодействия с пользователем:

Система должна реагировать на некорректный пользовательский ввод, ограничивая ввод недопустимых значений и информируя пользователей о причине ошибки.

Переходы между различными логически обособленными частями системы должны осуществляться с помощью меню.

При добавлении/удалении/изменении объекта, он должен автоматически появиться/исчезнуть/измениться на области у всех других клиентов.

При разработке ИС должны учитываться следующие требования:

В качестве основы для реализации ИС необходимо использовать Spring MVC.
Для создания уровня хранения необходимо использовать JPA + Hibernate.
Разные уровни приложения должны быть отделены друг от друга, разные логические части ИС должны находиться в отдельных компонентах.
Содержание отчёта:

Текст задания.

UML-диаграммы классов и пакетов разработанного приложения.

Исходный код системы или ссылка на репозиторий с исходным кодом.

Выводы по работе.

Вопросы к защите лабораторной работы:

Шаблоны проектирования и архитектурные шаблоны.

Платформа Jakarta EE. Виды компонентов.

Jakarta EE. Управляемые бины. CDI-бины.

Концепция ORM. Библиотеки ORM Hibernate и EclipseLink. Особенности, API, сходства и отличия.

Технология Jakarta Persistence. Особенности, API, интеграция с ORM-провайдерами.

Технология Jakarta Data.

Платформа Spring. Сходства и отличия с Java EE.

Spring Boot.

Spring Data.

Исходный код

https://github.com/MrTheFall/study_is

UML-диаграммы

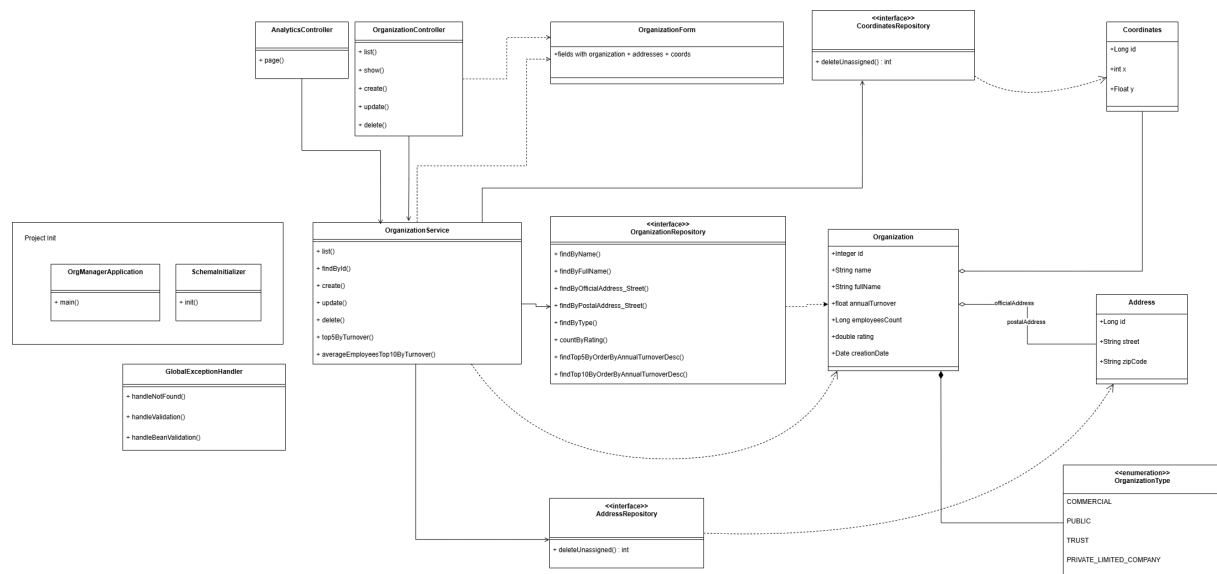


Рисунок 1: UML-диаграмма классов

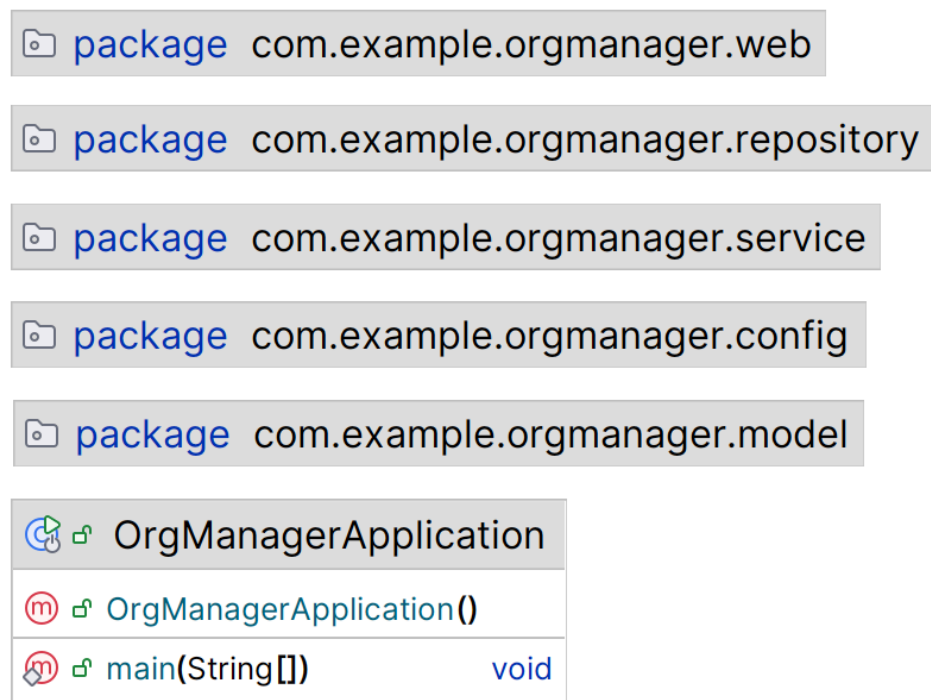


Рисунок 2: UML-диаграмма пакетов разработанного приложения

Выводы по работе

В ходе выполнения лабораторной работы по дисциплине «Информационные системы» была спроектирована и реализована информационная система для управления сущностью Organization.

На стороне сервера реализованы операции CRUD, проверка бизнес-правил и валидация атрибутов, а также синхронизация состояния с реляционной СУБД PostgreSQL через JPA/Hibernate с генерацией идентификаторов на уровне базы данных.

Пользовательский интерфейс обеспечивает просмотр объектов в табличном виде с пагинацией, сортировкой и фильтрацией по строковым полям, переход к просмотру и редактированию выбранного объекта. Дополнительно реализованы специализированные операции бизнес-логики без прямого использования процедур БД: подсчёт по rating, выборка по префиксу/сравнению поля fullName, выборка топ-5 по годовому обороту и вычисление среднего числа сотрудников для топ-10 организаций.

Архитектура с разделением на уровни представления, бизнес-логики и доступа к данным повышает сопровождаемость и расширяемость решения. Таким образом, цели лабораторной работы достигнуты, получены практические навыки моделирования, проектирования и реализации серверной ИС на базе Spring MVC, JPA/Hibernate и PostgreSQL.