

**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 13579

**Преподаватель:**

Тюрин Иван Николаевич

**Выполнил:**

Алхимовици Арсений

**Группа:**

P3310

Санкт-Петербург  
2025 г.

## Задание

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

```
public class HumanBeing {
    private long id; //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.time.LocalDateTime creationDate; //Поле не может быть null,
    Значение этого поля должно генерироваться автоматически
    private boolean realHero;
    private Boolean hasToothpick; //Поле может быть null
    private Car car; //Поле не может быть null
    private Mood mood; //Поле может быть null
    private float impactSpeed;
    private String soundtrackName; //Поле не может быть null
    private WeaponType weaponType; //Поле не может быть null
}
public class Coordinates {
    private Float x; //Поле не может быть null
    private Float y; //Максимальное значение поля: 818, Поле не может быть null
}
public class Car {
    private String name; //Поле может быть null
    private Boolean cool; //Поле не может быть null
}
public enum Mood {
    SADNESS,
    SORROW,
    GLOOM,
    CALM,
    RAGE;
}
public enum WeaponType {
    HAMMER,
    AXE,
    RIFLE,
    MACHINE_GUN;
}
```

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 по неполному 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, операция должна быть отменена, пользователю нужно сообщить о невозможности удаления объекта. 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:

Удалить один (любой) объект, значение поля `impactSpeed` которого эквивалентно заданному.

Рассчитать сумму значений поля `impactSpeed` для всех объектов.

Вернуть массив объектов, значение поля `name` которых содержит заданную подстроку.

Удалить всех героев без зубочисток.

Пересадить всех героев, не имеющих автомобиля на красные "Lada Kalina".

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.

### HumanBeing Management

Special Operations

Create Human

Create Car

Humans

ID	Name	Coords	Hero	Toothpick	Car	Impact	Soundtrack	Weapon	Actions
6	21	21.0,21.0	No	Yes	Toyota Camry	21.0	22	HAMMER	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Рисунок 1: Пользовательский интерфейс разработанного приложения

## Исходный код

<https://github.com/senya-2011/is-course>

# UML-диаграммы

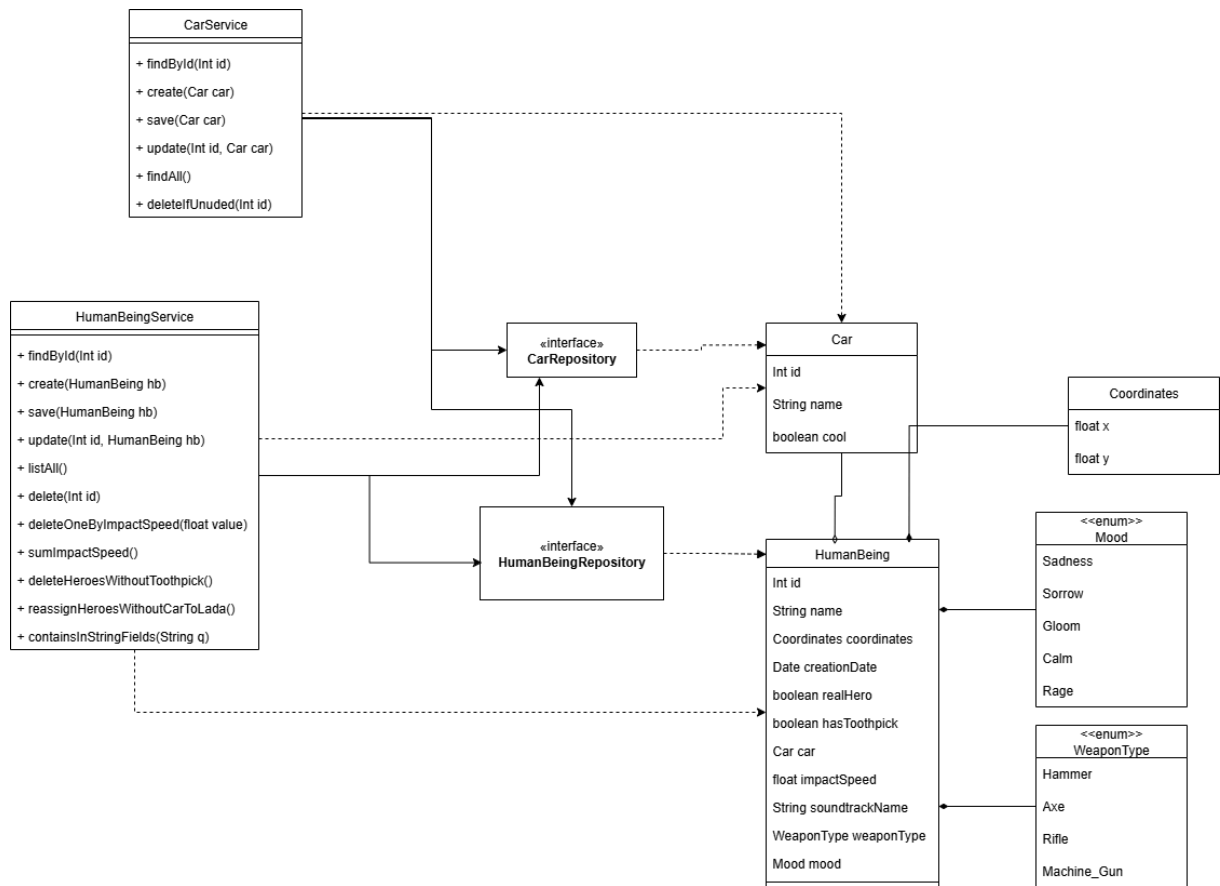


Рисунок 2: UML-диаграмма классов

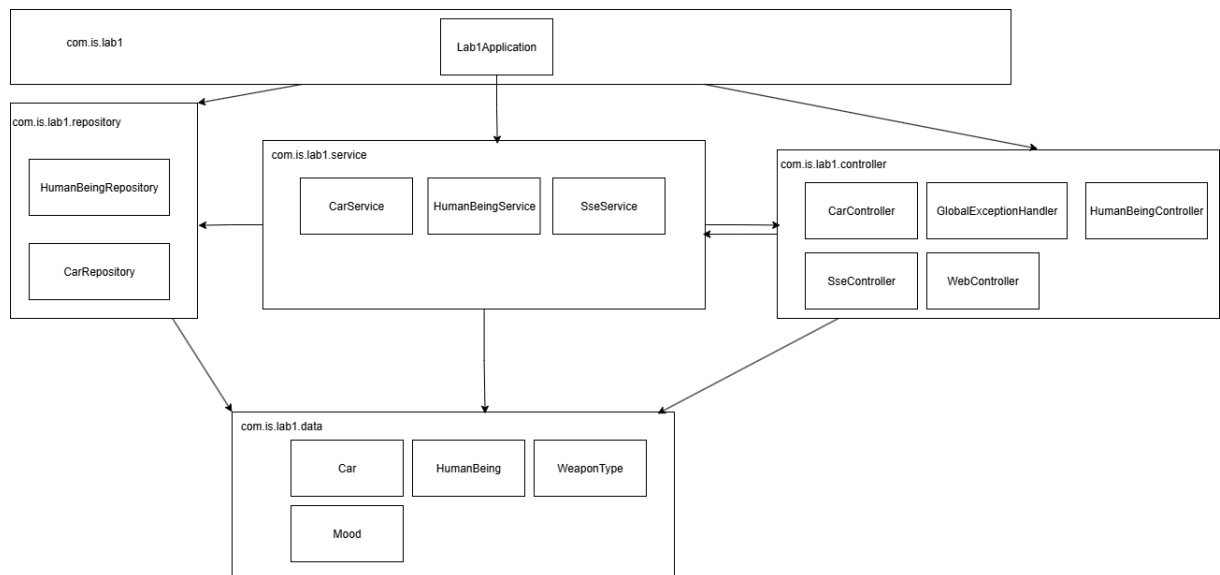


Рисунок 3: UML-диаграмма пакетов разработанного приложения

## **Выводы по работе**

В ходе выполнения лабораторной работы была спроектирована и реализована информационная система для управления сущностью HumanBeing с использованием Spring Boot, JPA/Hibernate и PostgreSQL. Архитектура с разделением на уровни представления, бизнес-логики и доступа к данным повышает сопровождаемость решения. Дополнительно реализована поддержка real-time уведомлений через Server-Sent Events. Система включает CRUD операции, поиск, валидацию данных и обеспечена тестами. Таким образом, цели лабораторной работы достигнуты, получены практические навыки моделирования, проектирования и реализации серверной ИС на базе Spring Boot, JPA/Hibernate и PostgreSQL.