Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«**Национальный исследовательский университет ИТМО**»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа **№1**

по дисциплине «Информационные системы»

Вариант: **409522**

**Преподаватель:**   
Николаев Владимир Вячеславович

**Выполнил:**

Саранча Павел Александрович

**Группа:** Р3309

Санкт-Петербург, 2025

# Текст задания

**public** **class** City {

**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** Integer area; //Значение поля должно быть больше 0, Поле не может быть null

**private** Integer population; //Значение поля должно быть больше 0, Поле не может быть null

**private** java.time.LocalDateTime establishmentDate;

**private** Boolean capital; //Поле может быть null

**private** Long metersAboveSeaLevel;

**private** Long carCode; //Значение поля должно быть больше 0, Максимальное значение поля: 1000, Поле может быть null

**private** Climate climate; //Поле может быть null

**private** StandardOfLiving standardOfLiving; //Поле может быть null

**private** Human governor; //Поле может быть null

}

**public** **class** Coordinates {

**private** float x;

**private** long y; //Значение поля должно быть больше -959

}

**public** **class** Human {

**private** float height; //Значение поля должно быть больше 0

}

**public** **enum** Climate {

   RAIN\_FOREST,

   TROPICAL\_SAVANNA,

   OCEANIC;

}

**public** **enum** StandardOfLiving {

   HIGH,

   LOW,

   VERY\_LOW;

}

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 объетов 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 все объекты, значение поля climate которого эквивалентно заданному.
* Рассчитать среднее значение поля metersAboveSeaLevel для всех объектов.
* Вернуть массив уникальных значений поля carCode по всем объектам.
* Рассчитать длину маршрута до города с наибольшей площадью
* Рассчитать длину маршрута от точки с координатами (0,0,0) до города с максимальным населением

Представленные операции должны быть реализованы в качестве функций БД, которые необходимо вызывать из уровня бизнес-логики приложения.

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.

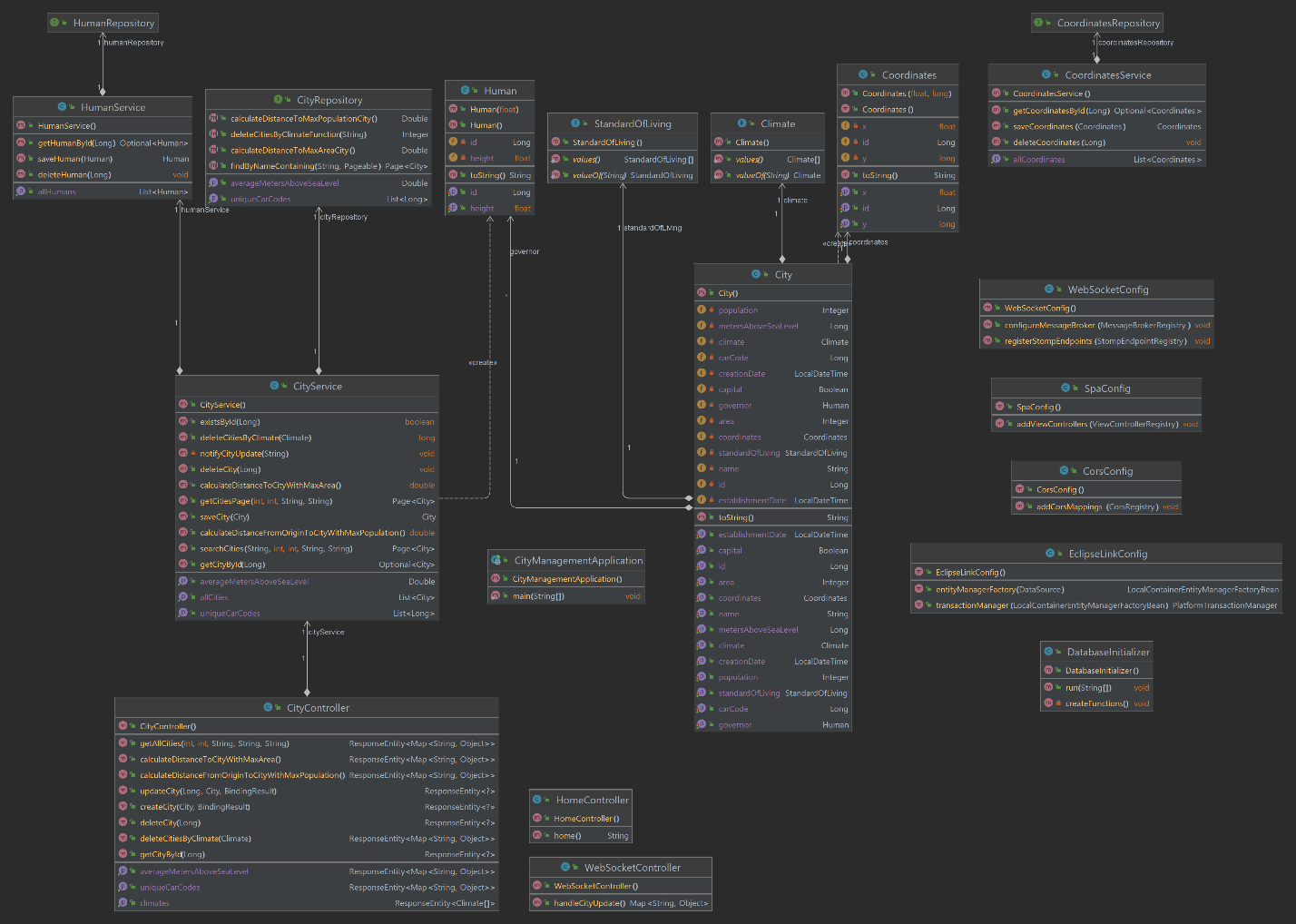
При создании системы нужно учитывать следующие особенности организации взаимодействия с пользователем:

* 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 EclipseLink.
* Разные уровни приложения должны быть отделены друг от друга, разные логические части ИС должны находиться в отдельных компонентах.

# UML-диаграммы классов и пакетов разработанного приложения.



# Cсылка на репозиторий с исходным кодом.

<https://github.com/PaulLocust/is-lab1>

# Выводы по работе.

Реализовал информационную систему, которая позволяет взаимодействовать с объектами класса City. Попробовал разработку на Spring, ещё раз потрогал шаблон MVC, использовал в работе ORM EclipseLink, использовал в работе автоматическое переключение на режимы локальной и продовой разработки с помощью spring-dotenv, реализовал удобный фронтенд. В дополнение к заданию, оформил Swagger API документацию.