

Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО».

Факультет программной инженерии и компьютерной техники

Информационные системы
Лабораторная работа №1
Вариант №1733723

Выполнил
Путинцев Данил
Группа Р3307
Проверил(а)
Преподаватель: Байрамова Хумай

Санкт-Петербург 2025 год

Текст задания

Лабораторные работы

Лабораторная работа #1

Введите вариант:

Внимание! У разных вариантов разный текст задания!

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 генерироваться автоматически  
    private String name; //Поле не может быть null, Стока не может быть пустой  
    private Coordinates coordinates; //Поле не может быть null  
    private java.time.LocalDate creationDate; //Поле не может быть null, Значение этого поля генерироваться автоматически  
    private Address officialAddress; //Поле не может быть null  
    private float annualTurnover; //Значение поля должно быть больше 0  
    private long employeesCount; //Значение поля должно быть больше 0  
    private int rating; //Значение поля должно быть больше 0  
    private OrganizationType type; //Поле может быть null  
    private Address postalAddress; //Поле не может быть null  
}  
public class Coordinates {  
    private Float x; //Значение поля должно быть больше -59, Поле не может быть null  
    private double y; //Значение поля должно быть больше -5  
}  
public class Address {  
    private String street; //Длина строки не должна быть больше 181, Поле не может быть null  
    private String zipCode; //Поле не может быть null  
}
```

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 генерироваться автоматически  
  
    private String name; //Поле не может быть null, Стока не может быть пустой  
  
    private Coordinates coordinates; //Поле не может быть null  
  
    private java.time.LocalDate 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; //Значение поля должно быть больше 0  
  
    private int rating; //Значение поля должно быть больше 0  
  
    private OrganizationType type; //Поле может быть null  
  
    private Address postalAddress; //Поле не может быть null  
}  
  
public class Coordinates {  
  
    private Float x; //Значение поля должно быть больше -59, Поле не может быть null  
    private double y; //Значение поля должно быть больше -5  
}  
  
public class Address {  
  
    private String street; //Длина строки не должна быть больше 181, Поле не может быть null  
    private String zipCode; //Поле не может быть null
```

```

}

public class Location {
    private Double x; //Поле не может быть null
    private Float y; //Поле не может быть null
    private double z;
    private String name; //Поле не может быть null
}

public enum OrganizationType {
    COMMERCIAL,
    PUBLIC,
    GOVERNMENT,
    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 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, связанные объекты должны быть связаны с другим объектом (по выбору пользователя), а изначальный объект удален.
- 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:

- Рассчитать среднее значение поля rating для всех объектов.
- Вернуть массив объектов, значение поля name которых начинается с заданной подстроки.
- Вернуть массив объектов, значение поля postalAddress которых больше заданного.
- Объединить организации, создав новую и зачислив в неё всех сотрудников двух исходных. Пользователь должен указать исходные организации, имя и адрес новой организации.
- Реализовать поглощение одной организацией другой без увольнения сотрудников.

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 Jakarta EE, CDI beans.
- Для создания уровня хранения необходимо использовать Hibernate.
- 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



Рефлексия/вопросы/ответы возникающие при выполнении лабораторной работы

1. Узнал, что такое Controller. Контроллер — это компонент серверной части приложения, который обрабатывает входящие HTTP-запросы и возвращает HTTP-ответы. Его основная задача — быть мостом между клиентом и внутренней бизнес-логикой приложения.
2. Узнал, что такое CORS и как с ним бороться жить. CORS (Cross-Origin Resource Sharing) - это механизм безопасности в браузерах, который ограничивает межсайтовые HTTP-запросы.

Зачем создали CORS?

Из-за политики одинакового источника (Same-Origin Policy) - важнейшего механизма безопасности, который предотвращает:

- Кражу сессий
- Утечку конфиденциальных данных

Основные заголовки

- Access-Control-Allow-Origin - какие домены могут обращаться
- Access-Control-Allow-Methods - разрешенные HTTP-методы
- Access-Control-Allow-Headers - какие заголовки можно отправлять
- Access-Control-Allow-Credentials - разрешать ли куки/авторизацию
- Access-Control-Max-Age - кэширование CORS-настроек

3. Умею отличать DTO от DAO.

DAO (Data Access Object) — Объект доступа к данным

Назначение: Абстракция для работы с хранилищем данных (БД, файлы, внешние API).

DTO (Data Transfer Object) — Объект для передачи данных

Назначение: Перенос данных между слоями/системами без бизнес-логики.

4. Как отслеживать ssh сессии и как понять, что мы падаем и с чем прийти к провайдеру.

Симптомы "падений":

- Резкие разрывы соединения без сообщения "Connection closed"

- "Зависание" терминала без отклика
- Ошибки: Connection reset by peer, Broken pipe
- Периодическая недоступность сервера по SSH

Что показать провайдеру при проблемах

```
sudo ss -tulpn | grep ssh
```

```
sudo systemctl status sshd
```

```
sudo tail -50 /var/log/auth.log | grep sshd
```

```
sudo dmesg | grep -i ssh
```

```
mtr --report -c 10 your-server.com
```

```
ping -c 10 your-server.com
```

Исходный код системы или ссылка на репозиторий с исходным кодом

https://github.com/danp1t/ITMO/tree/main/course3/information_system/labs/lab1

Выводы по работе

Во время выполнения этой работы я узнал, что можно использовать Hibernate не только в реализации JPA, но и как самостоятельный инструмент, научился описать классы для ORM с помощью конфигов. Также в ходе реализации синхронизации между пользователями я познакомился с WebSocket.