

Информационные системы

- [Задания к лабораторным работам](#)
- [Задание на курсовую работу](#)
- [Документация к СУБД PostgreSQL](#)

Лабораторные работы

Лабораторная работа #1

Введите вариант:

Внимание! У разных вариантов разный текст задания!

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

```
public class BookCreature {
    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 long age; //Значение поля должно быть больше 0
    private BookCreatureType creatureType; //Поле не может быть null
    private MagicCity creatureLocation; //Поле может быть null
    private Double attackLevel; //Значение поля должно быть больше 0, Поле не может быть null
    private Ring ring; //Поле не может быть null
}
public class Coordinates {
    private Long x; //Поле не может быть null
    private float y;
}
public class MagicCity {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private int area; //Значение поля должно быть больше 0
    private long population; //Значение поля должно быть больше 0
    private java.time.LocalDateTime establishmentDate;
    private BookCreatureType governor; //Поле может быть null
    private boolean capital;
    private Float populationDensity; //Значение поля должно быть больше 0
}
public class Ring {
    private String name; //Поле не может быть null, Строка не может быть пустой
    private long weight; //Значение поля должно быть больше 0
}
public enum BookCreatureType {
    HOBBIT,
    ELF,
    HUMAN,
    GOLLUM;
}
```

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**.

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 быть связаны с другим объектом (по выбору пользователя), а изначальный объект удален.
- Пользователю системы должен быть предоставлен интерфейс для авторизации/регистрации нового пользователя. У каждого пользователя должен быть один пароль. Требования к паролю: пароль должен быть уникален. В системе предполагается использование следующих видов пользователей (ролей): обычные пользователи и администраторы. Если в системе уже создан хотя бы один администратор, зарегистрировать нового администратора можно только при одобрении одним из существующих администраторов (у администратора должен быть реализован интерфейс со списком заявок и возможностью их одобрения).
- Редактировать и удалять объекты могут только пользователи, которые их создали, и администраторы (администраторы могут удалять все объекты).
- Зарегистрированные пользователи должны иметь возможность просмотра всех объектов, но модифицировать (обновлять) могут только принадлежащие им (объект принадлежит пользователю, если он его создал). 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 ring которого эквивалентно заданному.
- Вернуть один (любой) объект, значение поля age которого является минимальным.
- Вернуть массив объектов, значение поля attackLevel которых больше заданного.
- Совершить обмен кольцами (указать id персонажей).
- Переместить всех хоббитов с кольцами в Мордор.

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.
- Пароли при хранении хэшировать алгоритмом MD5.
- При хранении объектов сохранять информацию о пользователе, который создал этот объект, а также фиксировать даты и пользователей, которые обновляли и изменяли объекты. Для хранения информации о пользователях и об изменениях объектов нужно продумать и реализовать соответствующие таблицы.
- Таблицы БД, не отображающие заданные классы объектов, должны содержать необходимые связи с другими таблицами и соответствовать ЗНФ.
- 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.
- Во всех интерфейсах системы должно быть реализовано отображение информации о текущем пользователе (кто авторизован) и предоставляться возможность изменить текущего пользователя.
- [Опциональное задание - +2 балл] В отдельном окне ИС должна осуществляться визуализация объектов коллекции. При визуализации использовать данные о координатах и размерах объекта. Объекты от разных

- В качестве основы для реализации ИС необходимо использовать Java EE, Managed Beans.
- 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.

Содержание отчёта:

1. Текст задания.
2. UML-диаграммы классов и пакетов разработанного приложения.
3. Исходный код системы или ссылка на репозиторий с исходным кодом.
4. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Шаблоны проектирования и архитектурные шаблоны.
2. Платформа Jakarta EE. Виды компонентов.
3. Jakarta EE. Управляемые бины. CDI-бины.
4. Концепция ORM. Библиотеки ORM Hibernate и EclipseLink. Особенности, API, сходства и отличия.
5. Технология Jakarta Persistence. Особенности, API, интеграция с ORM-провайдерами.
6. Технология Jakarta Data.
7. Платформа Spring. Сходства и отличия с Java EE.
8. Spring Boot.
9. Spring Data.

