

□ Лабораторная работа №5

Задание на

лабораторную работу

Лабораторная работа является продолжением лабораторных работ. Требуется создать и подключить базу данных для хранения результатов запросов, ведения логов. Студенты самостоятельно выбирают технологии для подключения базы данных к коду приложения ([рекомендуется фреймворки для работы с базой данных Hibernate или Spring Data, рекомендуемая СУБД PostgreSQL](#)), а также сами распределяют между собой задачи (примерно поровну) и решают, когда делать «коммиты». В работе может встретиться очень много однотипных задач и подзадач – чтобы избежать дублирования кода, рекомендуется создавать новые методы и/или классы. Также настоятельно рекомендуется пользоваться таким изобретением человечества, как сеть Интернет,

поскольку задачи являются типовыми и где-то уже наверняка были реализованы. Все скрипты запросов должны располагаться в папки resources/sql. Для однотипных интерфейсов создаем отдельные папки типам

интерфейсов.

[Перед началом разработки требуется продумать оптимальную модель базы данных, что в ней будет лежать, как будут взаимосвязаны объекты и как будет осуществляться доступ к ним.](#)

Задание 1

Требуется реализовать интерфейсы для доступа к таблицам, созданным в базе данных. Для каждого типа объектов лежащих базе

данных требуется создать [классы модели данных](#), для всех данных передающихся внутри программы реализовать [DTO](#) или [DAO](#). Реализовать сервисы для извлечения данных из базы.

```
@Entity

public class Student {

    @Id
    @GeneratedValue(strategy=GenerationType.
        AUTO)

    private Long id;

    private String name;
```

```
        // getters and setters  
    }  
}
```

Пример класса модели данных

```
public class CreateProductRequest {  
    private String name;  
    private Double price;  
}
```

```
public class ProductResponse {  
    private Long id;  
    private String name;  
    private Double price;  
}
```

Пример DTO класса.

Задание 2

Реализация CRUD + операций

Необходимо разработать

сервисы для взаимодействия с базой данных.

Создать классы для

взаимодействия с базой данных, в них должны быть реализованы методы для

выполнения CRUD операций для каждого класса модели данных.

public MathFunctionEntity

create(MathFunctionEntity ent) - для добавления новых записей в базу данных. Метод должен принимать объект реализующий интерфейс

MathFunction и добавляться в базу данных

public MathFunctionEntity

read(String query) - для получения данных из

базы данных. Метод должен принимать SQL и возвращать сущность.

public MathFunctionEntity

update(MathFunctionEntity ent) - метод для обновление

математической функции в базе данных

public void

delete(MathFunctionEntity ent) - метод для удаления

сущности в базе данных

Примеры методов

Написать запросы по добавлению/удалению/изменению данных в базе данных с помощью ранее реализованных классов. Реализовать систему поиска по данным (поиск в глубину, поиск в ширину, поиск по иерархии, в случае если это позволяет модель данных, а

также одиночный, множественный, с сортировкой по полям).

Покрыть все написанные в лабораторной работе классы и запросы тестами, требуется создать скрипты с тестовыми данными, которые будут добавляться в базу на момент запуска и удаляться после выполнения.