

## Домашнее задание:

### 1. Задание на выборку данных (можете использовать Statement):

- Напишите SQL-запрос, который выбирает все курсы (**name, duration, type**) с продолжительностью более 30 часов.
- Используйте JDBC для выполнения этого запроса и выведите результат на экран.

### 2. Задание на вставку данных (PreparedStatement):

- Создайте запрос для вставки нового курса в таблицу **courses**.
- Используйте JDBC для выполнения запроса и убедитесь, что новая запись добавлена.

### 3. Задание на обновление данных (PreparedStatement):

- Измените стоимость всех курсов с типом "PROGRAMMING" на 5000.
- Используйте JDBC для выполнения запроса обновления.

### 4. Задание на удаление данных (PreparedStatement):

- Удалите все курсы с продолжительностью менее 10 часов.
- Используйте JDBC для выполнения запроса удаления.

## Рекомендации:

Ссылка на дампы:

<https://drive.google.com/file/d/118XgthGRb1mhZ8VtqzUQUbT6TBBqvlM0/view?usp=sharing>

Ссылка на инструкцию по установке MySQL **Workbench**:

<https://drive.google.com/file/d/1NVdSH2mQtCV6poWJ-c33py1o1S0bUFFc/view?usp=sharing>

Ссылка на ключи для активации IntelliJ IDEA **Ultimate**:

[https://drive.google.com/file/d/1Lp-QSIBT-LF-hbRM0jPw\\_SPr6LCFZcj/view?usp=sharing](https://drive.google.com/file/d/1Lp-QSIBT-LF-hbRM0jPw_SPr6LCFZcj/view?usp=sharing)

Скриншот работы с занятием:

```

package org.example;

import java.sql.*;

public class Main {
    /*
    JDBC позволяет воздействовать с СУБД
    Основные этапы работы:
    1. Установить соединение с БД (указываем URL, имя пользователя и пароль)
    2. Выполнить SQL-запросы для работы с данными
    3. Закрытие соединений с БД.
    */
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/top";
        String user = "root";
        String pass = "sHterBen44";

        String name = "Scrum";
        int duration = 40;
        String type = "PROGRAMMING";
        String description = "Description of the new course - Scrum";
        int teacherId = 1;
        int studentsCount = 0;
        double price = 3000;
        double pricePerHour = 25.00;

        try (Connection connection = DriverManager.getConnection(url, user, pass)) {
            String insertQuery = "INSERT INTO courses " +
                "(name, duration, type, description, teacher_id, students_count, " +
                price, price_per_hour) " +
                "VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

            try (PreparedStatement preparedStatement =
                connection.prepareStatement(insertQuery)) {

                // Подставляем данные в VALUES
                preparedStatement.setString(1, name);
                preparedStatement.setInt(2, duration);
                preparedStatement.setString(3, type);
                preparedStatement.setString(4, description);
                preparedStatement.setInt(5, teacherId);
                preparedStatement.setInt(6, studentsCount);
                preparedStatement.setDouble(7, price);
                preparedStatement.setDouble(8, pricePerHour);

                int rows = preparedStatement.executeUpdate(); // выполнение запроса

                // проверка выполнения запроса (0 - ошибка, 1 - успех)
                if (rows > 0) {
                    System.out.println("Запись добавилась успешно!");
                } else {
                    System.out.println("Произошла ошибка!");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```