

**Міністерство освіти і науки України**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**  
**«ДНІПРОВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ**  
**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  
*Кафедра інформаційних технологій та комп'ютерної інженерії*



Лабораторна робота №5

з дисципліни:  
«Базова Джава»

*Виконав:*  
студент групи 123-21-1  
Скоропад Є.В.  
*Перевірів:*  
Доцент кафедри САіУ  
Мінєєв О.С.

**Дніпро**  
**2025**

## Завдання

Створити базу даних в будь-якому сервері баз даних. Створити таблицю з переліком студентів вказати їх прізвище, ім'я, по батькові, день народження номер залікової книжки та ID.

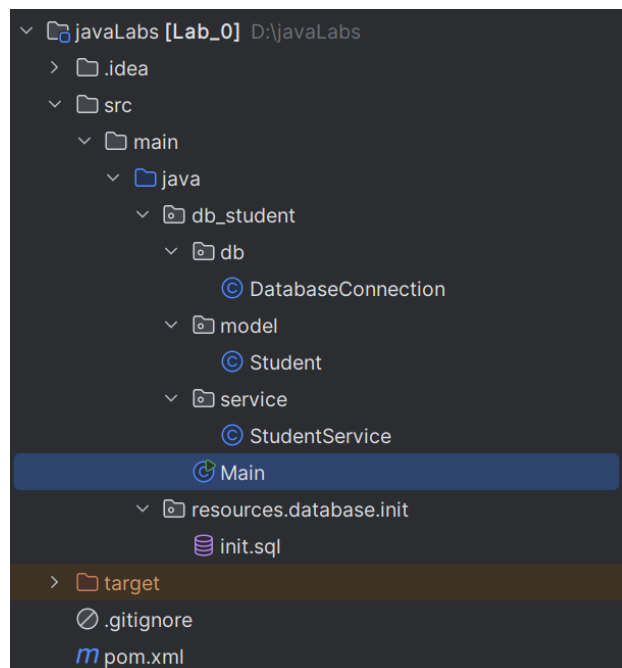
Створити програму що буде дозволяти виводити на екран інформацію про студентів які народилися в тому чи іншому місяці року. Програма повинна завдяки системі jdbc під'єднатися до вашої бази даних та робити до неї запити. Вимог до розробки бази даних немає. Програма ж має бути написана за усіма стандартами ООП. Та може бути спроектована за двох принципів:

- при будь-якій ситуації буде забиратися весь перелік студентів, а вже на стороні java буде зроблено пошук необхідного

## Налаштування для підключення до БД

The screenshot shows the 'Connection Name: BasicJava' window in Oracle SQL Developer. It has three tabs: 'Connection', 'Remote Management', and 'System Profile'. The 'Connection' tab is active, showing 'Connection Method: Standard (TCP/IP)'. Below this are three sub-tabs: 'Parameters', 'SSL', and 'Advanced'. The 'Parameters' sub-tab is active, displaying the following fields: 'Hostname' (127.0.0.1), 'Port' (3306), 'Username' (basic), 'Password' (with 'Store in Vault...' and 'Clear' buttons), and 'Default Schema' (students). To the right of each field is a descriptive text: 'Name or IP address of the server host - and TCP/IP port.', 'Name of the user to connect with.', 'The user's password. Will be requested later if it's not set.', and 'The schema to use as default schema. Leave blank to select it later.'

## Структура лабораторної



### Клас DatabaseConnection

```
package db_student.db;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://127.0.0.1:3306/students";
    private static final String USER = "basic";
    private static final String PASSWORD = "1234";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```

### Клас Student

```
package db_student.model;

import java.time.LocalDate;

public class Student {
    private int id;
    private String lastName;
    private String firstName;
    private String middleName;
    private LocalDate birthDate;
    private String recordBookNumber;

    public Student(int id, String lastName, String firstName, String middleName,
        LocalDate birthDate, String recordBookNumber) {
        this.id = id;
        this.lastName = lastName;
        this.firstName = firstName;
        this.middleName = middleName;
        this.birthDate = birthDate;
        this.recordBookNumber = recordBookNumber;
    }
}
```

```

public int getId() {
    return id;
}

public LocalDate getBirthDate() {
    return birthDate;
}

@Override
public String toString() {
    return String.format("ID: %d, ПІБ: %s %s %s, Дата народження: %s, Залікова: %s",
        id, lastName, firstName, middleName, birthDate, recordBookNumber);
}
}

```

## Клас StudentService

```

package db_student.service;

import db_student.db.DatabaseConnection;
import db_student.model.Student;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class StudentService {
    private List<Student> allStudents;

    public StudentService() {
        this.allStudents = fetchAllStudents();
    }

    public List<Student> getStudentsByMonth(int month) {
        List<Student> filteredStudents = new ArrayList<>();
        for (Student student : allStudents) {
            if (student.getBirthDate().getMonthValue() == month) {
                filteredStudents.add(student);
            }
        }
        return filteredStudents;
    }
}

```

```

    }

    private List<Student> fetchAllStudents() {
        List<Student> students = new ArrayList<>();
        String query = "SELECT id, last_name, first_name, middle_name, birth_date,
record_book_number FROM students_info";

        try (Connection conn = DatabaseConnection.getConnection();
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(query)) {

            while (rs.next()) {
                Student student = new Student(
                    rs.getInt("id"),
                    rs.getString("last_name"),
                    rs.getString("first_name"),
                    rs.getString("middle_name"),
                    rs.getDate("birth_date").toLocalDate(),
                    rs.getString("record_book_number")
                );
                students.add(student);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return students;
    }
}

```

## Клас Main

```

package db_student;

import db_student.model.Student;
import db_student.service.StudentService;

import java.util.List;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        StudentService service = new StudentService();

        while (true) {
            System.out.println("Enter month number (1-12) or 0 to exit:");
            int month = scanner.nextInt();

```

```
        if (month == 0) break;
        if (month < 1 || month > 12) {
            System.out.println("Wrong month number!");
            continue;
        }

        List<Student> students = service.getStudentsByMonth(month);
        if (students.isEmpty()) {
            System.out.println("No students born in this month have been
found.");
        } else {
            System.out.println("Known students:");
            students.forEach(System.out::println);
        }
        scanner.close();
    }
}
```