**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**

****

**ЗВІТ**

**про виконання лабораторних робіт**

**з дисципліни**

**«Поглиблене програмування в середовищі Java»**

**Лабораторна робота № 5**

Виконала:

студентка гр. 121-21-2

Векилова Рейхана Шамсіївна

Прийняв:

Мінєєв О. С.

**Дніпро**

**2025**

**Тема:** Jdbc.

**Мета:** Навчитися працювати з базами даних у Java, використовуючи JDBC, розробити програму для виведення інформації про студентів за заданим місяцем народження.

**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 запроси.  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

 - SQL запрос буде сформований згідно запросу який зробив користувач і вже сервер управління баз даних буде вирішувати, які самі студенти народилися в тому чи іншому місяці.

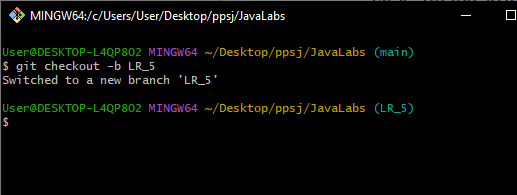
 У висновку обов'язково пояснити чому вибрали той чи інший принцип, які в нього переваги та недоліки.  Оцінка не залежить від того який сервер управління баз даних вибрали.  Перелік студентів зробити не менше 20 людей. Місяць червень зробити місяцем, коли в жодного зі студентів немає дня народження.

 SQL код створення бази даних розмістити проекті 5 лабораторної роботи  в файлі database в пекеджі resources.  Для використання цієї лабораторної роботи рекомендується активно використовувати знання отримані на дисципліні що стосуються  розробки баз даних.

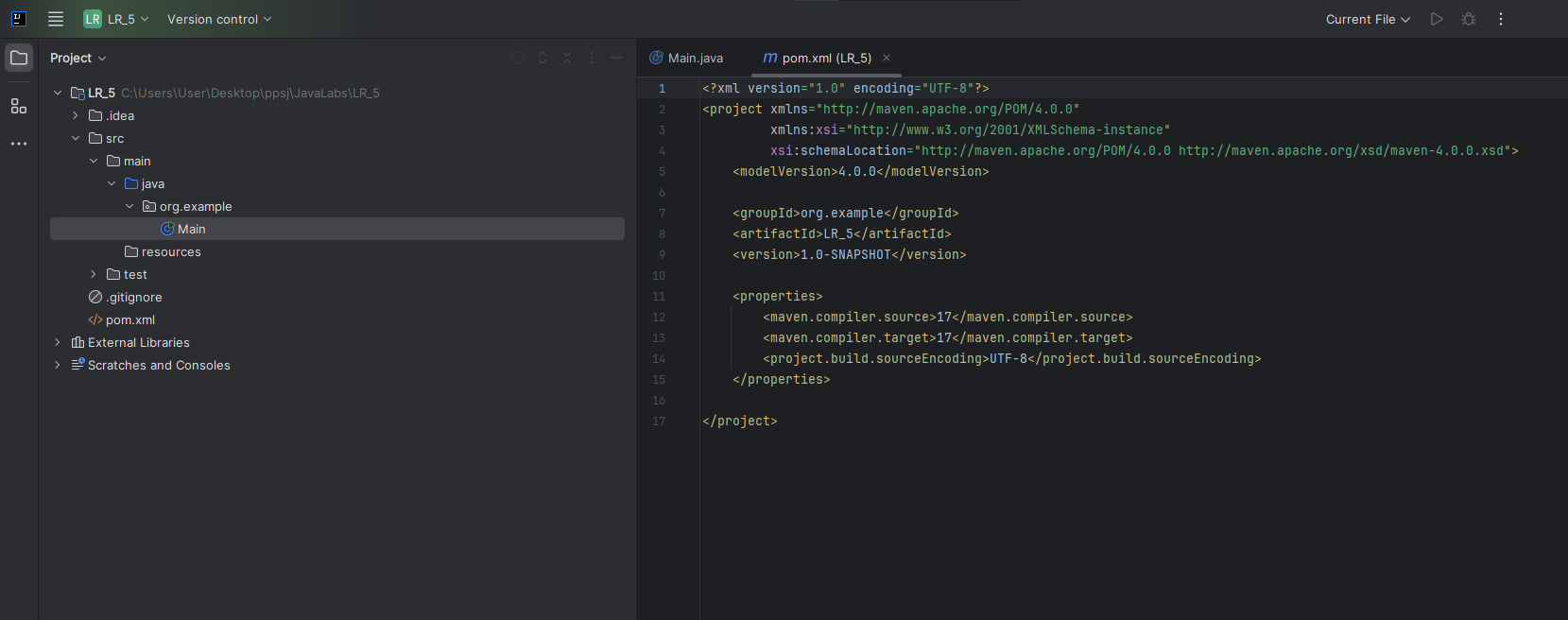
 До паперового звіту обов'язково додати принтскрин з програми  в якій ви дивитесь інформацію  вашого  сервера управління баз даних, де показати створену таблицю, її ім'я та загальні відомості бази даних, наприклад назва, ім'я, назва користувача адміністратора, пароль тощо.  Для роботи з сервером управління баз даних рекомендуємо використовувати програмне забезпечення компанії jetbrains datagrip. Або вбудовану панель користування базами даних, що міститься у середовищі intellij Idea,  яка на сьогоднішній день підтримує майже всі сервери управління баз даних.

Хід роботи

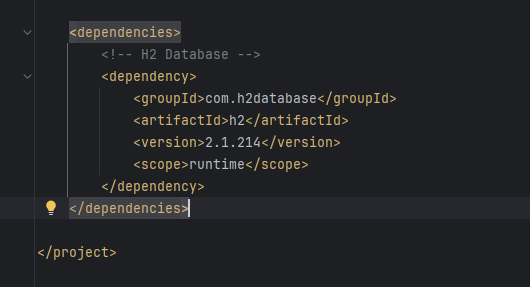
1. Створимо нову гілку для цієї лабораторної роботи (Рис. 1).

Рис. 1 — Результат виконання пункту 1

2. Створимо новий maven проєкт (Рис. 2).

Рис. 2 — Результат виконання пункту 2

3. Додамо залежності для підключення БД (Рис. 3).

Рис. 3 — Результат виконання пункту 3

4. Створимо модель Student.

*Лістинг «Student.java»*

package org.example.model;

public class Student {

private int id;

private String lastName;

private String firstName;

private String middleName;

private String birthDate;

private String recordBookNumber;

public Student(int id, String lastName, String firstName, String middleName, String birthDate, String recordBookNumber) {

this.id = id;

this.lastName = lastName;

this.firstName = firstName;

this.middleName = middleName;

this.birthDate = birthDate;

this.recordBookNumber = recordBookNumber;

}

@Override

public String toString() {

return lastName + " " + firstName + " " + (middleName != null ? middleName : "") +

" (Дата народження: " + birthDate + ")";

}

}

5. Створимо DatabaseManager.java.

*Лістинг «DatabaseManager.java»*

package org.example.database;

import org.example.model.Student;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class DatabaseManager {

private static final String URL = "jdbc:h2:./students\_db";

private static final String USER = "sa";

private static final String PASSWORD = "";

private static final String SCHEMA\_FILE = "src/main/resources/database/schema.sql";

public static void initDatabase() {

try (Connection connection = DriverManager.getConnection(URL, USER, PASSWORD);

Statement statement = connection.createStatement()) {

String sql = readSqlFile(SCHEMA\_FILE);

statement.execute(sql);

System.out.println("База даних ініціалізована з файлу schema.sql.");

} catch (SQLException | IOException e) {

e.printStackTrace();

}

}

private static String readSqlFile(String filePath) throws IOException {

StringBuilder sql = new StringBuilder();

try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {

String line;

while ((line = reader.readLine()) != null) {

sql.append(line).append("\n");

}

}

return sql.toString();

}

public static List<Student> getStudentsBornInMonth(int month) {

List<Student> students = new ArrayList<>();

String query = "SELECT \* FROM students WHERE MONTH(birth\_date) = ?";

try (Connection connection = DriverManager.getConnection(URL, USER, PASSWORD);

PreparedStatement statement = connection.prepareStatement(query)) {

statement.setInt(1, month);

ResultSet resultSet = statement.executeQuery();

while (resultSet.next()) {

students.add(new Student(

resultSet.getInt("id"),

resultSet.getString("last\_name"),

resultSet.getString("first\_name"),

resultSet.getString("middle\_name"),

resultSet.getDate("birth\_date").toString(),

resultSet.getString("record\_book\_number")

));

}

} catch (SQLException e) {

e.printStackTrace();

}

return students;

}

}

6. Створюємо файл schema.sql для ініціалізації БД.

*Лістинг «schema.sql»*

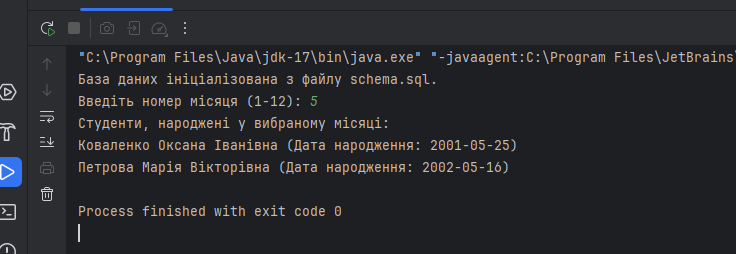
-- Створення таблиці студентів  
CREATE TABLE IF NOT EXISTS students (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 last\_name VARCHAR(50) NOT NULL,  
 first\_name VARCHAR(50) NOT NULL,  
 middle\_name VARCHAR(50),  
 birth\_date DATE NOT NULL,  
 record\_book\_number VARCHAR(20) UNIQUE NOT NULL  
 );  
  
-- Додавання тестових даних  
MERGE INTO students (last\_name, first\_name, middle\_name, birth\_date, record\_book\_number)  
 KEY (record\_book\_number)  
 VALUES  
 ('Іваненко', 'Олег', 'Сергійович', '2001-02-15', 'RB1001'),  
 ('Петров', 'Андрій', 'Миколайович', '2000-03-20', 'RB1002'),  
 ('Сидоренко', 'Марія', 'Олександрівна', '2002-04-10', 'RB1003'),  
 ('Коваленко', 'Оксана', 'Іванівна', '2001-05-25', 'RB1004'),  
 ('Петрова', 'Марія', 'Вікторівна', '2002-05-16', 'RB1005'),  
 ('Гончар', 'Віталій', 'Петрович', '2003-07-30', 'RB1006');

7. Тепер пропишемо Main.java.

Лістинг «Main.java»

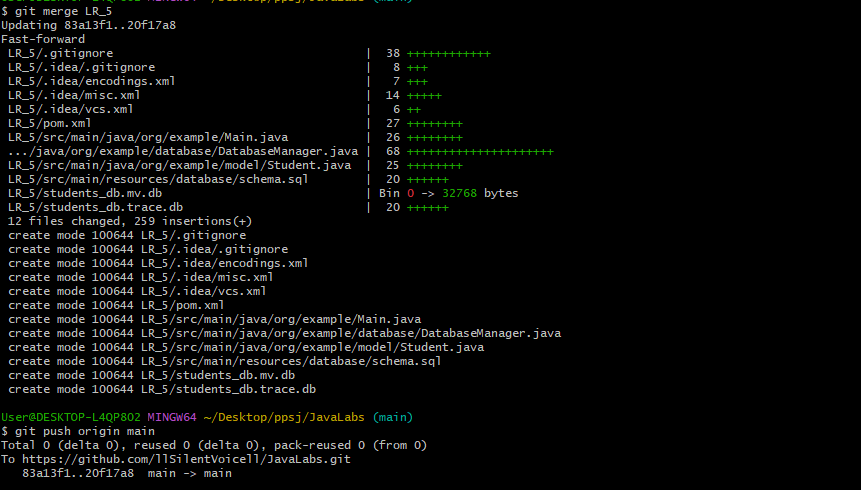
package org.example;  
  
import org.example.database.DatabaseManager;  
import org.example.model.Student;  
  
import java.util.List;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 DatabaseManager.*initDatabase*();  
  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Введіть номер місяця (1-12): ");  
 int month = scanner.nextInt();  
  
 List<Student> students = DatabaseManager.*getStudentsBornInMonth*(month);  
  
 if (students.isEmpty()) {  
 System.*out*.println("Немає студентів, народжених у цьому місяці.");  
 } else {  
 System.*out*.println("Студенти, народжені у вибраному місяці:");  
 students.forEach(System.*out*::println);  
 }  
 }  
}

8. Протестуємо проєкт (Рис. 4).

Рис. 4 — Результат роботи програми

Тестування показало працездатність розробленої програми.

8. Додаємо файли лабораторної роботи до GitHub репозиторію та мерджимо гілки (Рис. 5).

Рис. 5 — Результат виконання пункту 8

**Висновки:** у ході виконання лабораторної роботи навчилася працювати з базами даних у Java, використовуючи JDBC, розробила програму для виведення інформації про студентів за заданим місяцем народження.