**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**

****

**ЗВІТ**

**про виконання практичних робіт**

**з дисципліни**

**«Поглиблене програмування в середовищі Java»**

**Лабораторна робота № 5**

Виконав:

студент гр. 121-21-2

Ремез Максим Валентинович

Перевірив:

Доц. Мінєєв О.С.

**Дніпро**

**2025**

**Тема:** Jdbc.

**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 6 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.

**Код програми:**

Model

Student.java

package org.example.model;  
  
import java.time.LocalDate;  
import java.util.Objects;  
  
public class Student {  
 private int id;  
 private String firstName;  
 private String lastName;  
 private String middleName;  
 private LocalDate birthDate;  
 private String recordBookNumber;  
  
 public Student(int id, String firstName, String lastName, String middleName, LocalDate birthDate, String recordBookNumber) {  
 this.id = id;  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.middleName = middleName;  
 this.birthDate = birthDate;  
 this.recordBookNumber = recordBookNumber;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public String getMiddleName() {  
 return middleName;  
 }  
  
 public LocalDate getBirthDate() {  
 return birthDate;  
 }  
  
 public String getRecordBookNumber() {  
 return recordBookNumber;  
 }  
  
 @Override  
 public String toString() {  
 return String.*format*("ID: %d, %s %s %s, Дата народження: %s, № Залікової книжки: %s",  
 id, lastName, firstName, middleName, birthDate, recordBookNumber);  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Student student = (Student) o;  
 return id == student.id && firstName.equals(student.firstName) &&  
 lastName.equals(student.lastName) && Objects.*equals*(middleName, student.middleName) &&  
 birthDate.equals(student.birthDate) && recordBookNumber.equals(student.recordBookNumber);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(id, firstName, lastName, middleName, birthDate, recordBookNumber);  
 }  
}

database

DatabaseManager.java

package org.example.database;  
  
import org.example.model.Student;  
import org.example.utils.Config;  
  
import java.sql.\*;  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DatabaseManager {  
  
 static {  
 try (Connection conn = DriverManager.*getConnection*(Config.*DB\_URL*, Config.*DB\_USER*, Config.*DB\_PASSWORD*);  
 Statement stmt = conn.createStatement()) {  
 stmt.execute("RUNSCRIPT FROM 'src/main/resources/database.sql'");  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public static List<Student> getAllStudents() {  
 List<Student> students = new ArrayList<>();  
 String query = "SELECT \* FROM students";  
  
 try (Connection conn = DriverManager.*getConnection*(Config.*DB\_URL*, Config.*DB\_USER*, Config.*DB\_PASSWORD*);  
 Statement stmt = conn.createStatement();  
 ResultSet rs = stmt.executeQuery(query)) {  
  
 while (rs.next()) {  
 students.add(new Student(  
 rs.getInt("id"),  
 rs.getString("first\_name"),  
 rs.getString("last\_name"),  
 rs.getString("middle\_name"),  
 rs.getDate("birth\_date").toLocalDate(),  
 rs.getString("record\_book\_number")  
 ));  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
  
 return students;  
 }  
  
 public static List<Student> getStudentsByMonth(int month) {  
 List<Student> students = new ArrayList<>();  
 String query = "SELECT \* FROM students WHERE MONTH(birth\_date) = ?";  
  
 try (Connection conn = DriverManager.*getConnection*(Config.*DB\_URL*, Config.*DB\_USER*, Config.*DB\_PASSWORD*);  
 PreparedStatement pstmt = conn.prepareStatement(query)) {  
  
 pstmt.setInt(1, month);  
 try (ResultSet rs = pstmt.executeQuery()) {  
 while (rs.next()) {  
 students.add(new Student(  
 rs.getInt("id"),  
 rs.getString("first\_name"),  
 rs.getString("last\_name"),  
 rs.getString("middle\_name"),  
 rs.getDate("birth\_date").toLocalDate(),  
 rs.getString("record\_book\_number")  
 ));  
 }  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
  
 return students;  
 }  
}

service

StudentService.java

package org.example.service;  
  
import org.example.database.DatabaseManager;  
import org.example.model.Student;  
  
import java.util.List;  
import java.util.stream.Collectors;  
  
public class StudentService {  
  
 public static List<Student> filterStudentsByMonthJava(int month) {  
 return DatabaseManager.*getAllStudents*()  
 .stream()  
 .filter(student -> student.getBirthDate().getMonthValue() == month)  
 .collect(Collectors.*toList*());  
 }  
}

utils

Config.java

package org.example.utils;  
  
public class Config {  
 public static final String *DB\_URL* = "jdbc:h2:./database/lab5";  
 public static final String *DB\_USER* = "sa";  
 public static final String *DB\_PASSWORD* = "";  
}

Main.java

package org.example;  
  
import org.example.database.DatabaseManager;  
import org.example.model.Student;  
import org.example.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*);  
  
 System.*out*.print("Введіть номер місяця для пошуку студентів: ");  
 int month = scanner.nextInt();  
  
 System.*out*.println("\nФільтрація на рівні SQL:");  
 List<Student> studentsFromDb = DatabaseManager.*getStudentsByMonth*(month);  
 studentsFromDb.forEach(System.*out*::println);  
  
 System.*out*.println("\nФільтрація на рівні Java:");  
 List<Student> studentsFromJava = StudentService.*filterStudentsByMonthJava*(month);  
 studentsFromJava.forEach(System.*out*::println);  
 }  
}

database.sql

DROP TABLE students;  
  
CREATE TABLE IF NOT EXISTS students (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 first\_name VARCHAR(50) NOT NULL,  
 last\_name VARCHAR(50) NOT NULL,  
 middle\_name VARCHAR(50),  
 birth\_date DATE NOT NULL,  
 record\_book\_number VARCHAR(20) UNIQUE NOT NULL  
);  
INSERT INTO students (first\_name, last\_name, middle\_name, birth\_date, record\_book\_number) VALUES  
 ('Іван', 'Петров', 'Олександрович', '2000-01-15', 'S001'),  
 ('Марія', 'Іванова', 'Сергіївна', '2001-02-20', 'S002'),  
 ('Олег', 'Сидоренко', 'Миколайович', '1999-03-10', 'S003'),  
 ('Наталія', 'Коваль', 'Ігорівна', '2002-04-05', 'S004'),  
 ('Василь', 'Гончар', 'Васильович', '2000-05-12', 'S005'),  
 ('Андрій', 'Мельник', 'Андрійович', '1998-07-25', 'S006'),  
 ('Світлана', 'Ткаченко', 'Володимирівна', '2001-08-14', 'S007'),  
 ('Юрій', 'Бондар', 'Юрійович', '1997-09-30', 'S008'),  
 ('Олександр', 'Шевченко', 'Олександрович', '1996-10-07', 'S009'),  
 ('Ганна', 'Захарова', 'Григорівна', '2003-11-18', 'S010'),  
 ('Дмитро', 'Лисенко', 'Павлович', '1999-12-23', 'S011'),  
 ('Олена', 'Романенко', 'Віталіївна', '2002-01-08', 'S012'),  
 ('Максим', 'Кравченко', 'Богданович', '1995-02-14', 'S013'),  
 ('Катерина', 'Данилюк', 'Семенівна', '1998-03-19', 'S014'),  
 ('Ігор', 'Федорчук', 'Олександрович', '2000-04-22', 'S015'),  
 ('Оксана', 'Семенова', 'Анатоліївна', '2001-05-28', 'S016'),  
 ('Богдан', 'Кузьменко', 'Ігорович', '1997-07-09', 'S017'),  
 ('Людмила', 'Павленко', 'Олегівна', '1999-08-16', 'S018'),  
 ('Роман', 'Дорошенко', 'Євгенович', '1996-09-27', 'S019'),  
 ('Тетяна', 'Мартиненко', 'Іванівна', '2003-10-11', 'S020');

**Результат роботи програми:**

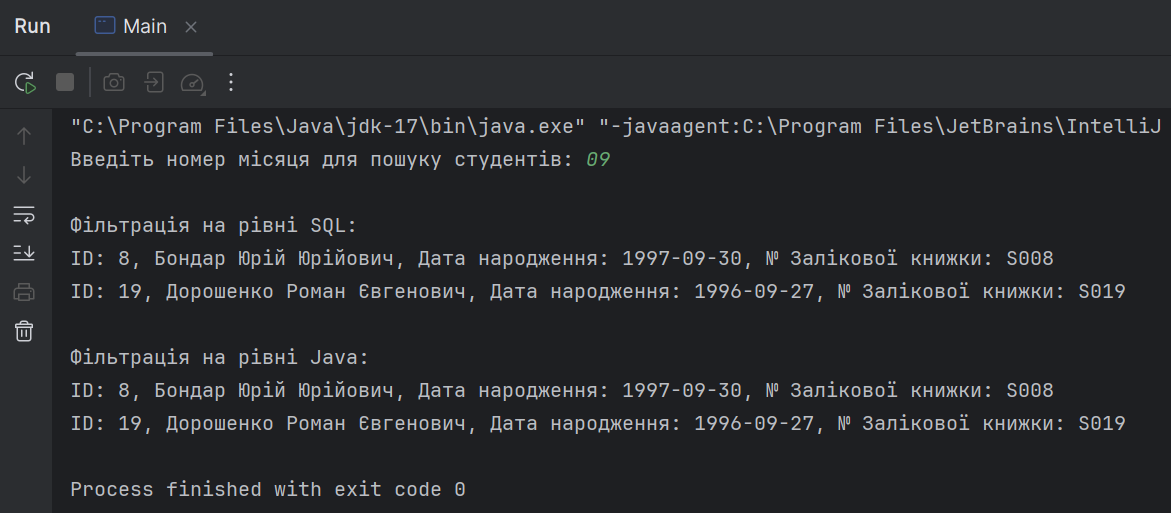


Рисунок 1 – Результат виконання коду