МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет «Дніпровська політехніка»

Изображение выглядит как текст

Автоматически созданное описание

Кафедра ПЗКС

ЗВІТ

з лабораторної роботи №5

з дисципліни «Поглиблене програмування в середовищі Java»

Виконав: ст. гр. 122-21-3

Пак Даніїл Євгенович

Перевірив: 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. Створити нову гілку для роботи з проектом – «LR\_5».

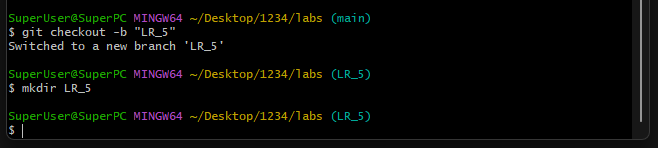
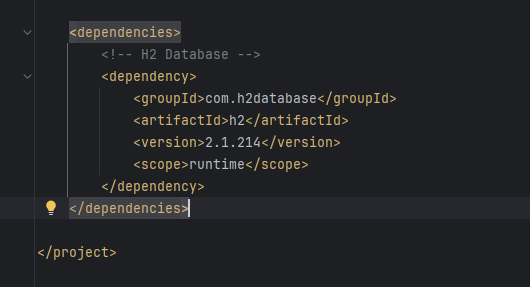


Рис. 1. Створення нової гілки проекту «LR\_5»

1. Виконати підключення залежності для роботи з базою даних.

Рис. 2. Залежність h2 в файлі «pom.xml»

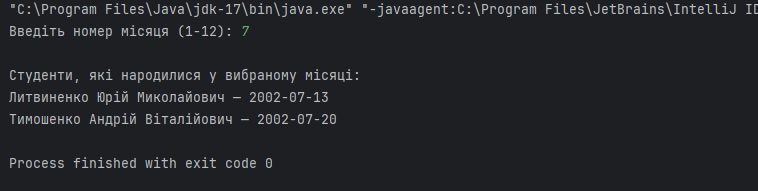


Рис. 3. Результат роботи програми

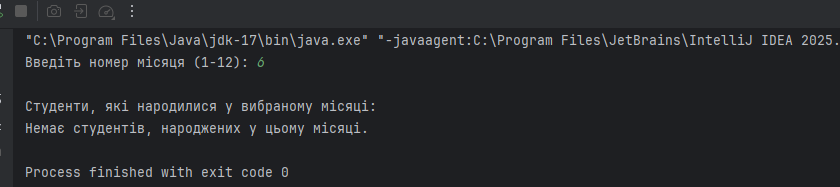


Рис. 4. Результат пошуку студентів шостого місяця народження

4. Виконати злиття гілки LR\_5 з основною - main

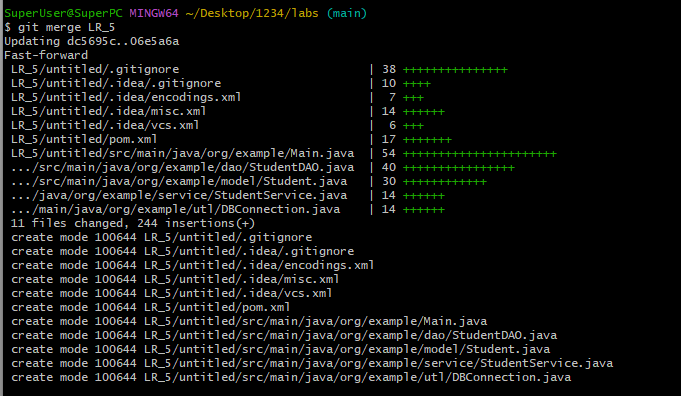


Рис. 5. Результат злиття гілок

**Код програми**

***//* Лістинг «Student.java»**

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 + ")";

}

**// Лістинг «DBManager.java»**

package org.example.database;

import org.example.model.Student;

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 = "";

public static void initDatabase() {

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

Statement statement = connection.createStatement()) {

String sql = "RUNSCRIPT FROM 'src/main/resources/database/schema.sql'";

statement.execute(sql);

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

} catch (SQLException e) {

e.printStackTrace();

}

}

// Запит до БД для отримання студентів, народжених у певному місяці

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;

}

}

**Лістинг «db.sql»**

CREATE DATABASE IF NOT EXISTS studentdb;

USE studentdb;

CREATE TABLE IF NOT EXISTS student (

id INT PRIMARY KEY AUTO\_INCREMENT,

surname VARCHAR(50) NOT NULL,

name VARCHAR(50) NOT NULL,

patronymic VARCHAR(50),

birth\_date DATE NOT NULL,

record\_book\_number VARCHAR(20) NOT NULL

);

INSERT INTO student (surname, name, patronymic, birth\_date, record\_book\_number) VALUES

('Іванов', 'Петро', 'Іванович', '2001-01-15', 'RB001'),

('Петренко', 'Олег', 'Васильович', '2002-02-17', 'RB002'),

('Ковальчук', 'Анна', 'Миколаївна', '2001-03-20', 'RB003'),

('Сидоренко', 'Людмила', 'Олександрівна', '2000-04-10', 'RB004'),

('Дорошенко', 'Ірина', 'Іванівна', '2001-05-05', 'RB005'),

('Захаренко', 'Віталій', 'Григорович', '2002-07-22', 'RB006'),

('Мельник', 'Олексій', 'Ігорович', '2000-08-01', 'RB007'),

('Кравець', 'Тетяна', 'Сергіївна', '2002-09-13', 'RB008'),

('Гончар', 'Юрій', 'Васильович', '2001-10-11', 'RB009'),

('Шевченко', 'Артем', 'Олексійович', '2000-11-02', 'RB010'),

('Бондар', 'Марія', 'Петрівна', '2001-12-19', 'RB011'),

('Литвин', 'Андрій', 'Іванович', '2002-01-25', 'RB012'),

('Ткаченко', 'Ігор', 'Богданович', '2002-02-28', 'RB013'),

('Романенко', 'Світлана', 'Юріївна', '2000-03-05', 'RB014'),

('Назаренко', 'Євген', 'Михайлович', '2001-04-14', 'RB015'),

('Федорчук', 'Леся', 'Олегівна', '2001-05-21', 'RB016'),

('Мазур', 'Інна', 'Романівна', '2002-07-30', 'RB017'),

('Олійник', 'Дмитро', 'Іванович', '2000-08-09', 'RB018'),

('Семенюк', 'Наталя', 'Валентинівна', '2001-09-06', 'RB019'),

('Хоменко', 'Сергій', 'Павлович', '2000-10-18', 'RB020');

**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(); // Ініціалізація бази даних

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); // Виводимо інформацію про студентів

}

}

}

**Висновок**

Реалізовано базу даних студентів. Для пошуку студентів, народжених у певному місяці, застосований підхід виконує фільтрацію на стороні SQL. Цей метод зменшує обсяг даних, що передаються між клієнтом і сервером, та забезпечує продуктивність при роботі з великими обсягами даних.