

Міністерство освіти і науки України  
НТУ «Дніпровська політехніка»  
Кафедра системного аналізу та управління



Звіт з лабораторної роботи № 5  
З дисципліни «Поглиблене програмування в середовищі Java»

Виконав:  
ст. гр. 122-21-2  
Помазан Микита  
Максимович

Дніпро  
2025

## Лабораторна робота: JDBC

Мета роботи: Створити базу даних студентів у будь-якому сервері баз даних, реалізувати підключення через JDBC та розробити програму для отримання інформації про студентів, народжених у вибраному місяці.

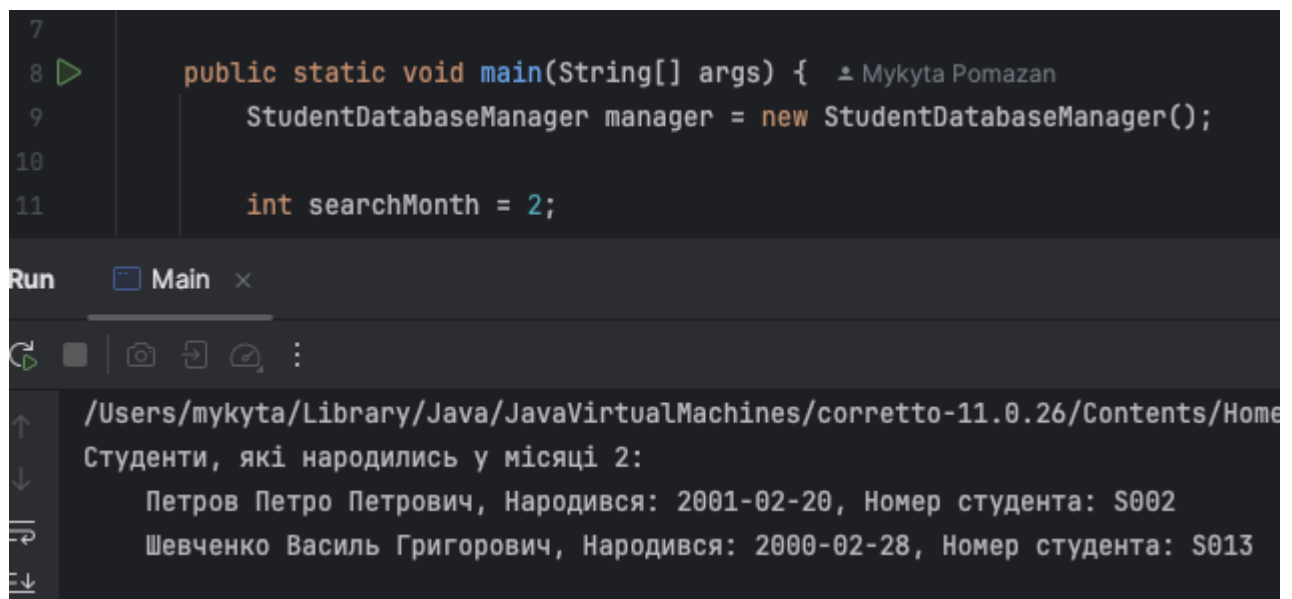
### Хід роботи:

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
- SQL запит буде сформований згідно запиту 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.



The screenshot shows a Java IDE with a code editor and a run console. The code in the editor is as follows:

```
7  
8 public static void main(String[] args) {  
9     StudentDatabaseManager manager = new StudentDatabaseManager();  
10  
11     int searchMonth = 2;
```

The run console shows the output of the program:

```
Run Main x  
/Users/мыкыта/Library/Java/JavaVirtualMachines/corretto-11.0.26/Contents/Home  
Студенти, які народились у місяці 2:  
Петров Петро Петрович, Народився: 2001-02-20, Номер студента: S002  
Шевченко Василь Григорович, Народився: 2000-02-28, Номер студента: S013
```

Рис. 1 - Результат виконання програми

	id [PK] integer	last_name character varying (50)	first_name character varying (50)	middle_name character varying (50)	birth_date date	record_book_number character varying (20)
1	1	Іванов	Іван	Іванович	2000-01-15	S001
2	2	Петров	Петро	Петрович	2001-02-20	S002
3	3	Сидоров	Олександр	Сергійович	2002-03-10	S003
4	4	Мельник	Олег	Ігорович	2000-04-25	S004
5	5	Ковальчук	Марина	Василівна	2001-05-18	S005
6	6	Ткаченко	Юрій	Андрійович	2002-07-05	S006
7	7	Гриценко	Ольга	Петрівна	2000-08-30	S007
8	8	Дорошенко	Олексій	Віталійович	2001-09-22	S008
9	9	Лисенко	Катерина	Володимирівна	2002-10-12	S009
10	10	Романенко	Антон	Максимович	2000-11-03	S010
11	11	Зайцев	Ігор	Олександрович	2001-12-19	S011
12	12	Кравченко	Наталія	Дмитрівна	2002-01-07	S012
13	13	Шевченко	Василь	Григорович	2000-02-28	S013
14	14	Бондаренко	Артем	Євгенович	2001-03-15	S014
15	15	Мороз	Андрій	Богданович	2002-04-20	S015
16	16	Савченко	Людмила	Степанівна	2000-05-29	S016
17	17	Гаврилюк	Олександра	Михайлівна	2001-07-17	S017
18	18	Козак	Дмитро	Ростиславович	2002-08-23	S018
19	19	Федорчук	Ірина	Василівна	2000-09-09	S019
20	20	Даниленко	Максим	Юрійович	2001-10-31	S020

Рис. 2 - База даних

Вибір принципу фільтрації на сервері через SQL-запит дозволяє забезпечити високу ефективність програми, оскільки сервер бази даних оптимізує виконання запитів і використовує індекси для швидкого пошуку. Це зменшує навантаження на клієнтську машину, оскільки всі операції фільтрації виконуються на сервері, а клієнт отримує лише необхідні дані. Такий підхід також підвищує масштабованість програми в майбутньому, оскільки з ростом кількості користувачів навантаження на клієнтські пристрої залишається мінімальним.

Недоліки цього підходу можуть включати ускладнення самих SQL-запитів, оскільки вони можуть ставати важкими для розуміння та підтримки. Крім того, програма стає залежною від серверної частини, і якщо сервер має проблеми з продуктивністю або доступом, це може вплинути на всю систему.

Код програми:

[https://github.com/mykytapomazan/java-basic/tree/LR\\_5](https://github.com/mykytapomazan/java-basic/tree/LR_5)

або:

## Створення BD:

```
CREATE

DATABASE students_db;

\c

students_db;

CREATE TABLE students

(

    id                SERIAL 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

);

INSERT INTO students (last_name, first_name, middle_name, birth_date, record_book_number)

VALUES ('Іванов', 'Іван', 'Іванович', '2000-01-15', 'S001'),

       ('Петров', 'Петро', 'Петрович', '2001-02-20', 'S002'),

       ('Сидоров', 'Олександр', 'Сергійович', '2002-03-10', 'S003'),

       ('Мельник', 'Олег', 'Ігорович', '2000-04-25', 'S004'),

       ('Ковальчук', 'Марина', 'Василівна', '2001-05-18', 'S005'),

       ('Ткаченко', 'Юрій', 'Андрійович', '2002-07-05', 'S006'),

       ('Гриценко', 'Ольга', 'Петрівна', '2000-08-30', 'S007'),

       ('Дорошенко', 'Олексій', 'Віталійович', '2001-09-22', 'S008'),

       ('Лисенко', 'Катерина', 'Володимирівна', '2002-10-12', 'S009'),

       ('Романенко', 'Антон', 'Максимович', '2000-11-03', 'S010'),

       ('Зайцев', 'Ігор', 'Олександрович', '2001-12-19', 'S011'),

       ('Кравченко', 'Наталія', 'Дмитрівна', '2002-01-07', 'S012'),

       ('Шевченко', 'Василь', 'Тригорович', '2000-02-28', 'S013'),

       ('Бондаренко', 'Артем', 'Євгенович', '2001-03-15', 'S014'),

       ('Мороз', 'Андрій', 'Богданович', '2002-04-20', 'S015'),

       ('Савченко', 'Людмила', 'Степанівна', '2000-05-29', 'S016'),
```

```
        ('Гаврилюк', 'Олександра', 'Михайлівна', '2001-07-17', 'S017'),  
  
        ('Козак', 'Дмитро', 'Ростиславович', '2002-08-23', 'S018'),  
  
        ('Федорчук', 'Ірина', 'Василівна', '2000-09-09', 'S019'),  
  
        ('Даниленко', 'Максим', 'Юрійович', '2001-10-31', 'S020');
```

## DatabaseConnector

```
package database;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
  
public class DatabaseConnector {  
  
    private static final String URL = "jdbc:postgresql://localhost:5432/students_db";  
  
    private static final String USER = "mykyta";  
  
    private static final String PASSWORD = "";  
  
    public static Connection getConnection() throws SQLException {  
  
        return DriverManager.getConnection(URL, USER, PASSWORD);  
  
    }  
}
```

## StudentDatabaseManager

```
package database;  
  
import model.Student;  
  
import java.sql.*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class StudentDatabaseManager {  
  
    public List<Student> getStudentsByBirthMonth(int month) {  
  
        List<Student> students = new ArrayList<>();  
  
        String query = "SELECT last_name, first_name, middle_name, birth_date, record_book_number " +
```

```

        "FROM students WHERE EXTRACT(MONTH FROM birth_date) = ?";

    try (Connection connection = DatabaseConnector.getConnection();

        PreparedStatement preparedStatement = connection.prepareStatement(query)) {

        preparedStatement.setInt(1, month);

        ResultSet resultSet = preparedStatement.executeQuery();

        while (resultSet.next()) {

            String lastName = resultSet.getString("last_name");

            String firstName = resultSet.getString("first_name");

            String middleName = resultSet.getString("middle_name");

            String birthDate = resultSet.getString("birth_date");

            String recordBookNumber = resultSet.getString("record_book_number");

            students.add(new Student(lastName, firstName, middleName, birthDate, recordBookNumber));

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return students;

}
}

```

## Student

```

package model;

import java.util.Objects;

public class Student {

    private String lastName;

    private String firstName;

    private String middleName;

    private String birthDate;

    private String recordBookNumber;

```

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

        this.lastName = lastName;

        this.firstName = firstName;

        this.middleName = middleName;

        this.birthDate = birthDate;

        this.recordBookNumber = recordBookNumber;

    }

    public String getBirthDate() {

        return birthDate;

    }

    @Override

    public String toString() {

        return lastName + " " + firstName + " " + middleName + ", Народився: " + birthDate + ", Номер
студента: " + 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 Objects.equals(lastName, student.lastName) && Objects.equals(firstName,
student.firstName) &&

            Objects.equals(middleName, student.middleName) && Objects.equals(birthDate,
student.birthDate) &&

            Objects.equals(recordBookNumber, student.recordBookNumber);

    }

    @Override

    public int hashCode() {

        return Objects.hash(lastName, firstName, middleName, birthDate, recordBookNumber);

    }

}
```

```
}
```

## Main:

```
import database.StudentDatabaseManager;

import model.Student;

import java.util.List;

public class Main {

    public static void main(String[] args) {

        StudentDatabaseManager manager = new StudentDatabaseManager();

        int searchMonth = 2;

        List<Student> students = manager.getStudentsByBirthMonth(searchMonth);

        if (students.isEmpty()) {

            System.out.println("Немає студентів, які народились у цьому місяці.");

        } else {

            System.out.println("Студенти, які народились у місяці " + searchMonth + ":");

            for (Student student : students) {

                System.out.println("\t" + student);

            }

        }

    }

}
```

Висновок: Створив базу даних студентів у будь-якому сервері баз даних, реалізувати підключення через JDBC та розробити програму для отримання інформації про студентів, народжених у вибраному місяці.