Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗВІТ про виконання Лабораторної роботи з дисципліни «Програмування в середовищі Java»

Виконав:
студент гр. 124-20-1
Говоруха Д.С.
Прийняв:
викладач каф. САУ
Мінєєв О. С.

Дніпро
2023

Лабораторна робота №5

Завдання

Лабораторна робота номер 5. Jdbc

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.

Виконання

Спочатку в файлі pom.xml було прописано встановлення бази даних:

```
<dependencies>
    <dependency>
        <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>
        <version>1.4.200</version>
    </dependency>
</dependencies>
```

Потім був створений клас DBUtils для приєднання бази даних до проекту:

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

public class DBUtils {
    private static String dbURL =
"jdbc:h2:mem:lab5;INIT=RUNSCRIPT FROM 'classpath:init.sql'";
    private static String dbUsername = "sa";
    private static String dbPassword = "";

    public static Connection getConnection(){
        Connection connection = null;
        try{
            connection = DriverManager.getConnection(dbURL,
dbUsername, dbPassword);
        }catch(SQLException throwables){
            throwables.printStackTrace();
        }
        return connection;
    }
}
```

Був створений файл init.sql в якому був прописаний скрип створення схеми та таблиці. Під час кожного запуску програми він очищався та створювався знову:

```
CREATE SCHEMA IF NOT EXISTS lab5;
USE lab5;

DROP TABLE IF EXISTS students;
CREATE TABLE students (id INT PRIMARY KEY AUTO_INCREMENT, name
VARCHAR(50), lastName VARCHAR(50), fatherName VARCHAR(50),
birthday DATE, documentNumber VARCHAR(50));

INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Максим','Іванов','Олегович','1995-02-
10','ZK12345');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES
('Анастасія','Соколова','Володимирівна','1992-07-22','ZK23456');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES
```

```
('Олександр','Петренко','Сергійович','1998-11-05','ZK34567');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Юлія','Морозова','Олександрівна','1993-
09-15','ZK45678');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Дмитро','Коваленко','Ігорович','1997-07-
28','ZK56789');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Олена','Лисенко','Миколаївна','1994-04-
03','ZK67890');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Андрій','Шевченко','Іванович','1991-12-
12','ZK78901');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Ірина','Козлова','Віталіївна','1996-08-
20','ZK89012');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Віталій','Мельник','Петрович','1990-03-
07','ZK90123');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Марія','Василенко','Ігорівна','1999-05-
18','ZK01234');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Сергій','Жуков','Михайлович','1997-09-
29','ZL12345');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Наталія','Грищенко','Анатоліївна','1993-
07-08','ZL23456');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Павло','Ковальов','Олександрович','1995-
11-17','ZL34567');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES
('Катерина','Литвиненко','Володимирівна','1991-03-24','ZL45678');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Артем','Бондаренко','Сергійович','1998-
12-11','ZL56789');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Вікторія','Литвинова','Олегівна','1994-
05-02','ZL67890');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Олег','Іваненко','Миколайович','1997-09-
14','ZL78901');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Марина','Кравченко','Вікторівна','1992-
08-21','ZL89012');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Антон','Олійник','Петрович','1996-03-
06','ZL90123');
INSERT INTO students(name, lastName, fatherName, birthday,
documentNumber) VALUES ('Юлія','Ігнатенко','Андріївна','1990-05-
19','ZL01234');
```

Потім була створена клас-модель студента в якій було прописано всі данні про студента, а також метод toString(), який допомагав в подальшому виводі інформації на екран користувача:

```
public class Student {
    private int id;
    private String name;
    private String lastName;
    private String fatherName;
    private String birthday;
    private String documentNumber;

    @Override
    public String toString(){
        return "Студент "+"id:"+id+
                "/ Ім'я:"+name+
                "/ Прізвище:"+lastName+
                "/ По-батькові:"+fatherName+
                "/ День народження:"+birthday+
                "/ Залікова книжка:"+documentNumber;
    }
    public Student(){

    }

    public Student(int id, String name, String lastName, String
fatherName, String birthday, String documentNumber){
        this.id = id;
        this.name = name;
        this.lastName = lastName;
        this.fatherName = fatherName;
        this.birthday = birthday;
        this.documentNumber = documentNumber;
    }

    public int getId(){
        return id;
    }
    public void setId(int id){
        this.id = id;
    }
    public String getName(){
        return name;
    }
    public void setName(String name){
        this.name = name;
    }
    public String getLastName(){
        return lastName;
    }
    public void setLastName(String lastName){
        this.name = lastName;
    }
```

```
    public String getFatherName(){
        return fatherName;
    }
    public void setFatherName(String fatherName){
        this.name = fatherName;
    }
    public String getBirthday(){
        return birthday;
    }
    public void setBirthday(String birthday){
        this.name = birthday;
    }
    public String getDocumentNumber(){
        return documentNumber;
    }
    public void setDocumentNumber(String documentNumber){
        this.name = documentNumber;
    }
}
```

Потім було створено клас ShowStudents, в якому утворювався ліст студентів, які задовольняли запиту:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

public class ShowStudents {
    public static List<Student> getStudentData(String query){
        List<Student> students = new ArrayList<>();

        try (Connection connection = DBUtils.getConnection();
             PreparedStatement preparedStatement =
connection.prepareStatement(query)){
            ResultSet rs = preparedStatement.executeQuery();

            while (rs.next()){
                int id = rs.getInt("id");
                String name = rs.getString("name");
                String lastName = rs.getString("lastName");
                String fatherName = rs.getString("fatherName");
                String birthday = rs.getString("birthday");
                String documentNumber =
rs.getString("documentNumber");
                students.add(new Student(id, name, lastName,
fatherName, birthday, documentNumber));
            }
        }catch (SQLException throwables){
            throwables.printStackTrace();
        }
```

```
        return students;
    }
}
```

Основна програма. В ній користувача просять обрати місяць, а потім за допомогою SQL запиту створювався відбір студентів та їх подільший вивід на екран користувача:

```
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 місяць (число від 1 до 12):
");
        int month = scanner.nextInt();

        List<Student> students =
ShowStudents.getStudentData("SELECT * FROM students WHERE
MONTH(birthday) = " + month);
        for (Student student : students) {
            System.out.println(student.toString());
        }
    }
}
```

Тестування

```
Введіть місяць (число від 1 до 12): 3
Студент id:9/ Ім'я:Віталій/ Прізвище:Мельник/ По-батькові:Петрович/ День народження:1990-03-07/ Залікова книжка:ZK90123
Студент id:14/ Ім'я:Катерина/ Прізвище:Литвиненко/ По-батькові:Володимирівна/ День народження:1991-03-24/ Залікова книжка:ZL45678
Студент id:19/ Ім'я:Антон/ Прізвище:Олійник/ По-батькові:Петрович/ День народження:1996-03-06/ Залікова книжка:ZL90123
```

```
Введіть місяць (число від 1 до 12): 5
Студент id:10/ Ім'я:Марія/ Прізвище:Василенко/ По-батькові:Ігорівна/ День народження:1999-05-18/ Залікова книжка:ZK01234
Студент id:16/ Ім'я:Вікторія/ Прізвище:Литвинова/ По-батькові:Олегівна/ День народження:1994-05-02/ Залікова книжка:ZL67890
Студент id:20/ Ім'я:Юлія/ Прізвище:Ігнатенко/ По-батькові:Андріївна/ День народження:1990-05-19/ Залікова книжка:ZL01234
```

```
Введіть місяць (число від 1 до 12): 6
```

За умови, в таблиці немає студентів, в яких день народження в червні. Тому програма нікого не знайшла.

Висновки

Під час виконання лабораторної, була створена таблиця, з переліком студенів. В ній було вказані: ім’я, прізвище, по-батькові, день народження та номер залікової книжки. Було створено програму, що дозволяє користувачу вручну обрати місяць та побачити студентів з днем народження у цьому місяці. Програма була написана за усіма стандартами ООП та спроектована за принципом, що при будь-якій ситуації, буде збиратись весь перелік студента, а вже на стороні java зроблено пошук необхідного. Я обрав цей варіант через те що мені, за умови лабораторної, потрібно обирати один місяць, та виводити данні. Якби я обрав інший принцип, мені потрібно було б створювати 12 запитів, на кожен місяць, і потім обробляти їх окремо.