**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**



**ЗВІТ**

**про виконання лабораторних робіт**

**з дисципліни**

# «Основи програмування в середовищі java»

Виконав:

студент гр. 122-20-3

Даниш О.Д.

(П.І.Б.)

Прийняв:

доцент Мінєєв Олександр Сергійович

(П.І.Б.)

**Дніпро**

**2024**

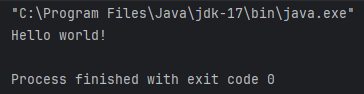
**Посилання на GitHub:** <https://github.com/ODanysh/java-labs>

**Лабораторна робота номер 0.** Hello world

**Код програми:**

package edu.ntudp.fit.danysh.lab\_0;  
  
public class Main {  
 public static void main(String[] args) {  
 System.*out*.println("Hello world!");  
 }  
}

**Результат виконання програми:**

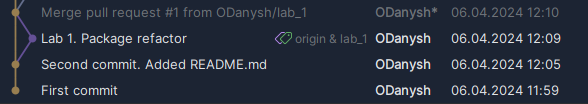
****

**Висновки:** Під час виконання лабораторної роботи булоознайомлено з можливостями мови Java.

**Лабораторна робота номер 1.** CVS. GIT

Створено репозиторій <https://github.com/ODanysh/java-labs>, створено окрему гілку під нову лабораторну, створено коміт та завантажено його на GitHub.

**Результат завантаження коміту та його злиття з основною гілкою:**

****

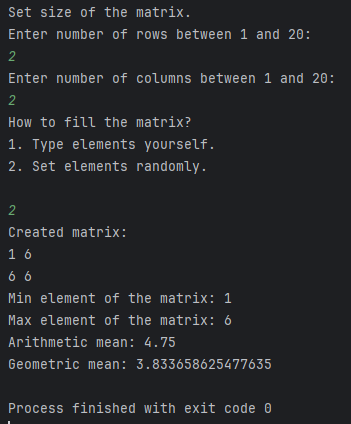
**Висновки:** Під час виконання лабораторної роботи булоознайомлено з системою контролю версій та завантажено проект на GitHub.

**Лабораторна робота номер 2.** Основи.

**Код програми:**

package edu.ntudp.fit.danysh.lab\_2;  
  
import java.util.Random;  
import java.util.Scanner;  
  
public class Main {  
 public static final int *MIN\_ROWS* = 1;  
 public static final int *MAX\_ROWS* = 20;  
 public static final int *MIN\_COLUMNS* = 1;  
 public static final int *MAX\_COLUMNS* = 20;  
 public static final int *MIN\_RANDOM\_NUMBER* = 1;  
 public static final int *MAX\_RANDOM\_NUMBER* = 10;  
 public static void main(String[] args) {  
 int[][] matrix;  
 int rows = 0, columns = 0;  
 Scanner scanner = new Scanner(System.*in*);  
 int input;  
 System.*out*.println("Set size of the matrix.");  
 while (rows < *MIN\_ROWS* || rows > *MAX\_ROWS*) {  
 System.*out*.println("Enter number of rows between "+*MIN\_ROWS*+" and "+*MAX\_ROWS*+":");  
 if (scanner.hasNextInt()) {  
 rows = scanner.nextInt();  
 if (rows < *MIN\_ROWS* || rows > *MAX\_ROWS*) {  
 System.*out*.println("You should write number between "+*MIN\_ROWS*+" and "+*MAX\_ROWS*);  
 }  
 scanner.nextLine();  
 }  
 else {  
 System.*out*.println("You should write number between "+*MIN\_ROWS*+" and "+*MAX\_ROWS*);  
 scanner.nextLine();  
 }  
 }  
 while (columns < *MIN\_COLUMNS* || columns > *MAX\_COLUMNS*) {  
 System.*out*.println("Enter number of columns between "+*MIN\_COLUMNS*+" and "+*MAX\_COLUMNS*+":");  
 if (scanner.hasNextInt()) {  
 columns = scanner.nextInt();  
 if (columns < *MIN\_COLUMNS* || columns > *MAX\_COLUMNS*) {  
 System.*out*.println("You should write number between "+*MIN\_COLUMNS*+" and "+*MAX\_COLUMNS*);  
 }  
 scanner.nextLine();  
 }  
 else {  
 System.*out*.println("You should write number between "+*MIN\_COLUMNS*+" and "+*MAX\_COLUMNS*);  
 scanner.nextLine();  
 }  
 }  
 while (true) {  
 System.*out*.println("""  
 How to fill the matrix?  
 1. Type elements yourself.  
 2. Set elements randomly."""  
 );  
 System.*out*.println();  
 if (scanner.hasNextInt()) {  
 input = scanner.nextInt();  
 if (input == 1) {  
 matrix = *getCustomMatrix*(rows, columns);  
 break;  
 }  
 else if (input == 2) {  
 matrix = *getRandomMatrix*(rows, columns);  
 break;  
 }  
 else {  
 System.*out*.println("You should choose between 1 and 2 options");  
 }  
 }  
 else {  
 System.*out*.println("You should choose between 1 and 2 options");  
 }  
 scanner.nextLine();  
 }  
 scanner.close();  
 System.*out*.println("Created matrix:");  
 *printMatrix*(matrix);  
 System.*out*.println("Min element of the matrix: " + *findMinElement*(matrix));  
 System.*out*.println("Max element of the matrix: " + *findMaxElement*(matrix));  
 System.*out*.println("Arithmetic mean: " + *findArithmeticMean*(matrix));  
 System.*out*.println("Geometric mean: " + *findGeometricMean*(matrix));  
 }  
 public static int[][] getCustomMatrix(int inputRows, int inputColumns) {  
 int[][] matrix = new int[inputRows][inputColumns];  
 Scanner scanner = new Scanner(System.*in*);  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix[0].length; j++) {  
 System.*out*.print("Row " + (i + 1) + ", Column " + (j + 1) + " element: ");  
 if (scanner.hasNextInt()) {  
 matrix[i][j] = scanner.nextInt();  
 scanner.nextLine();  
 System.*out*.println();  
 }  
 }  
 }  
 return matrix;  
 }  
 public static int[][] getRandomMatrix(int rows, int columns) {  
 int[][] matrix = new int[rows][columns];  
 int diff = *MAX\_RANDOM\_NUMBER* - *MIN\_RANDOM\_NUMBER*;  
 Random rand = new Random();  
 for (int i = 0; i < matrix.length; i++) {  
 for (int j = 0; j < matrix[0].length; j++) {  
 matrix[i][j] = rand.nextInt(diff + 1) + *MIN\_RANDOM\_NUMBER*;  
 }  
 }  
 return matrix;  
 }  
 public static void printMatrix(int[][] matrix) {  
 for (int[] ints : matrix) {  
 for (int j = 0; j < matrix[0].length; j++) {  
 System.*out*.print(ints[j] + " ");  
 }  
 System.*out*.println();  
 }  
 }  
 public static int findMinElement(int[][] matrix) {  
 int min = matrix[0][0];  
 for (int[] ints : matrix) {  
 for (int j = 0; j < matrix[0].length; j++) {  
 if (ints[j] < min) {  
 min = ints[j];  
 }  
 }  
 }  
 return min;  
 }  
 public static int findMaxElement(int[][] matrix) {  
 int max = matrix[0][0];  
 for (int[] ints : matrix) {  
 for (int j = 0; j < matrix[0].length; j++) {  
 if (ints[j] > max) {  
 max = ints[j];  
 }  
 }  
 }  
 return max;  
 }  
 public static double findArithmeticMean(int[][] matrix) {  
 double sumOfElements = 0;  
 for (int[] ints : matrix) {  
 for (int j = 0; j < matrix[0].length; j++) {  
 sumOfElements += ints[j];  
 }  
 }  
 return sumOfElements / (matrix.length \* matrix[0].length);  
 }  
 public static double findGeometricMean(int[][] matrix) {  
 int productOfElements = 1;  
 for (int[] ints : matrix) {  
 for (int j = 0; j < matrix[0].length; j++) {  
 productOfElements \*= ints[j];  
 }  
 }  
 return Math.*pow*(productOfElements, 1.0 / (double) (matrix.length \* matrix[0].length));  
 }  
}

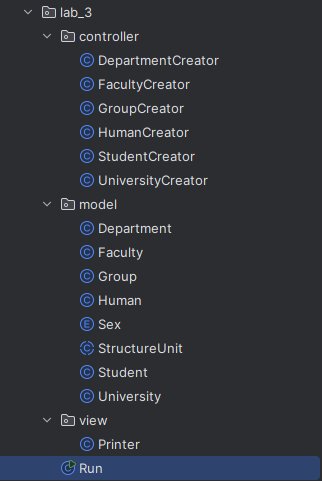
**Результат виконання програми:**



**Висновки:** Під час виконання лабораторної роботи булоознайомлено з основами мови Java, створено програму для створення та роботи з матрицями.

**Лабораторна робота номер 3.** ООП.

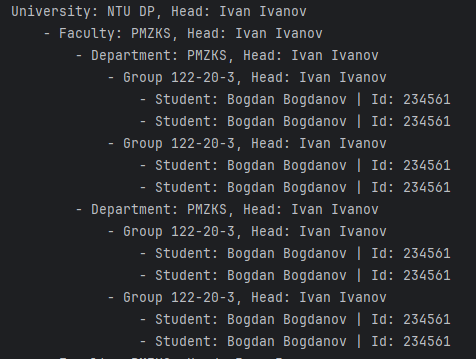
Створено модель університету:



**Код програми:**

package edu.ntudp.fit.danysh.lab\_3;  
  
import edu.ntudp.fit.danysh.lab\_3.controller.UniversityCreator;  
import edu.ntudp.fit.danysh.lab\_3.model.University;  
import edu.ntudp.fit.danysh.lab\_3.view.Printer;  
  
public class Run {  
 public static void main(String[] args) {  
 University university = new UniversityCreator().createTypicalUniversity();  
 Printer printer = new Printer();  
 printer.printUniversityTree(university);  
  
 }  
}

**Результат виконання програми:**



**Висновки:** Під час виконання лабораторної роботи булоознайомлено з можливостями ООП в мові Java.

**Лабораторна робота номер 4.** JUnit. Json

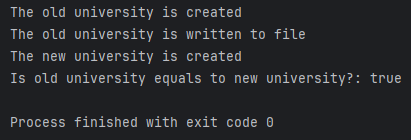
До моделі з третьої лабораторної додано Json менеджер:

package edu.ntudp.fit.danysh.lab\_4;  
  
import com.google.gson.Gson;  
import edu.ntudp.fit.danysh.lab\_4.model.University;  
  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
  
public class JsonManager {  
  
 public void writeToJson(University university, String fileName) {  
 try (FileWriter fileWriter = new FileWriter(fileName)) {  
 Gson gson = new Gson();  
 gson.toJson(university, fileWriter);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public University readFromJson(String fileName) {  
 try (FileReader fileReader = new FileReader(fileName)) {  
 Gson gson = new Gson();  
 return gson.fromJson(fileReader, University.class);  
 } catch (IOException e) {  
 e.printStackTrace();  
 return null;  
 }  
 }  
}

**Код програми:**

package edu.ntudp.fit.danysh.lab\_4;  
  
import edu.ntudp.fit.danysh.lab\_4.controller.UniversityCreator;  
import edu.ntudp.fit.danysh.lab\_4.model.University;  
import edu.ntudp.fit.danysh.lab\_4.view.Printer;  
  
public class Run {  
 public static void main(String[] args) {  
 University oldUniversity = new UniversityCreator().createTypicalUniversity();  
 System.*out*.println("The old university is created");  
 JsonManager JsonManager = new JsonManager();  
 JsonManager.writeToJson(oldUniversity,"oldUniversityJson");  
 System.*out*.println("The old university is written to file");  
 University newUniversity = JsonManager.readFromJson("oldUniversityJson");  
 System.*out*.println("The new university is created");  
 System.*out*.println("Is old university equals to new university?: " + oldUniversity.equals(newUniversity));  
 }  
}

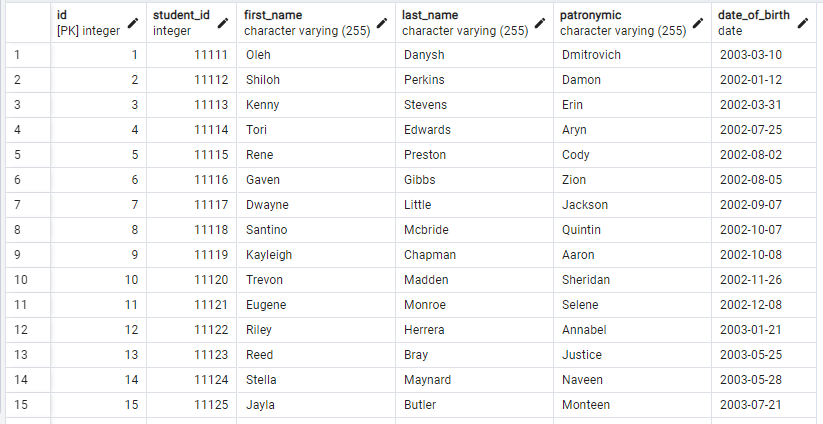
**Результат виконання програми:**



**Висновки:** Під час виконання лабораторної роботи булоознайомлено з можливостями Junit та Json.

**Лабораторна робота номер 5.** Jdbc

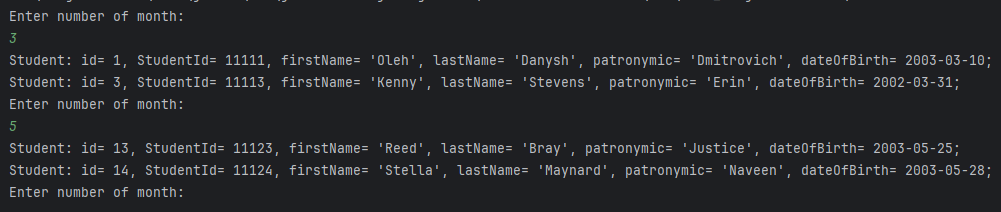
Створено базу даних на PostgreSQL та заповнено її даними зі студентами:



**Код програми:**

package edu.ntudp.fit.danysh.lab\_5;  
  
import java.sql.\*;  
import java.time.LocalDate;  
import java.util.\*;  
  
public class Run {  
 private static final String *DB\_URL* = "jdbc:postgresql://localhost:5432/JavaDB";  
 private static final String *DB\_USER* = "postgres";  
 private static final String *DB\_PASSWORD* = "admin";  
 public static void main(String[] args) {  
 try {  
 Class.*forName*("org.postgresql.Driver");  
 Connection connection = DriverManager.*getConnection*(*DB\_URL*, *DB\_USER*, *DB\_PASSWORD*);  
 Scanner scanner = new Scanner(System.*in*);  
 while(true) {  
 System.*out*.println("Enter number of month: ");  
 try {  
 int input = scanner.nextInt();  
 if (input < 1 || input > 12) { break; }  
 List<Student> students = *getStudentsByMonthOfBirth*(connection, input);  
 for (Student student : students) {  
 System.*out*.println(student);  
 }  
 }  
 catch (InputMismatchException e) {  
 System.*out*.println("You should enter number of month.");  
 scanner.nextLine();  
 }  
 }  
 connection.close();  
 }  
 catch (SQLException e) {  
 System.*out*.println("Connection failed" + e);  
 }  
 catch (ClassNotFoundException e) {  
 System.*out*.println("Class not found");  
 }  
 }  
  
 private static List<Student> getStudentsByMonthOfBirth(Connection connection, int month) throws SQLException {  
 List<Student> students = new ArrayList<>();  
 String sqlRequest = "SELECT \* FROM public.students WHERE EXTRACT(MONTH FROM date\_of\_birth) = " + month + ";";  
 PreparedStatement statement = connection.prepareStatement(sqlRequest);  
 ResultSet resultSet = statement.executeQuery();  
 while (resultSet.next()) {  
 int id = resultSet.getInt("id");  
 int student\_id = resultSet.getInt("student\_id");  
 String firstName = resultSet.getString("first\_name");  
 String lastName = resultSet.getString("last\_name");  
 String patronymic = resultSet.getString("patronymic");  
 LocalDate dateOfBirth = resultSet.getDate("date\_of\_birth").toLocalDate();  
  
 Student student = new Student(id, student\_id, firstName, lastName, patronymic, dateOfBirth);  
 students.add(student);  
 }  
 return students;  
 }  
}

**Результат виконання програми:**



**Висновки:** Під час виконання лабораторної роботи булоознайомлено з можливостями використання мови Java для роботи з базами даних.