

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет «Дніпровська політехніка»



Кафедра ПЗКС

ЗВІТ

Лабораторна робота №4

з дисципліни «Поглиблене програмування в середовищі Java»

Виконав: ст. гр. 122-21-1

Карапиш Ілля Володимирович

Перевірили:

Мінєєв О.С.

Шевченко Ю.О.

Дніпро

2025

Лабораторна робота № 4

Тема: «JUnit. JSON».

Завдання: Додати до лабораторної роботи 3 можливість запису університету у формат json, запис цього формату у файл, зчитування цього формату файлу, та створення об'єкту з текстового формату json. В проєкті повинен бути зроблений JUnit тест, який буде виглядати наступним чином: створити об'єкт університет(`oldUniversity`), в якому в кожному підрозділі маються два підрозділи нижчого рівня. Наприклад на факультеті дві кафедри, на кожній кафедрі дві групи, на кожній групі два студенти. Цей об'єкт повинен бути записаний в файл у форматі json. Потім з цього файлу зчитаний та відновлений як `newUniversity`. В тесті повинні бути порівняні `newUniversity` та `oldUniversity` за допомогою методу `equals`. Якщо все зроблено правильно то університети повинні бути еквівалентні, а метод `equals` повинен повернути `True`. Для запису та зчитування університету у форматі json повинен бути зроблений клас `JsonManager`. Для безпосереднього перетворення університету у формат json та його відновлення цього формату, можливо використання сторонніх бібліотек наприклад `Gson`, `Jackson` чи будь-яких інших.

Для початку розробки лабораторної роботи номер 4 повністю скопіювати програмний код лабораторної роботи номер 3. Не змішувати ці роботи не в якому разі.

Хід роботи

Код програми:

`JsonManager.java:`

```
package ntudp.pzks.lr4.controller;

import com.google.gson.Gson;
import ntudp.pzks.lr4.model.University;

import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
```

```

public class JsonManager {

    public static void writeUniversityToJson(University university, String filePath) {
        Gson gson = new Gson();
        try (FileWriter writer = new FileWriter(filePath)) {
            gson.toJson(university, writer);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static University readUniversityFromJson(String filePath) {
        Gson gson = new Gson();
        try (FileReader reader = new FileReader(filePath)) {
            return gson.fromJson(reader, University.class);
        } catch (IOException e) {
            e.printStackTrace();
            return null;
        }
    }
}

```

UniversityJsonTest.java:

```

import ntudp.pzks.lr4.model.*;

import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import ntudp.pzks.lr4.controller.JsonManager;

import java.io.File;
import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

public class UniversityJsonTest {
    private static final String FILE_PATH = "university.json";

    @Test
    public void testJsonSerialization() {
        University oldUniversity = createSampleUniversity();

        JsonManager.writeUniversityToJson(oldUniversity, FILE_PATH);

        University newUniversity = JsonManager.readUniversityFromJson(FILE_PATH);

        assertEquals(oldUniversity, newUniversity, "oldUniversity and newUniversity should be the same");
    }
}

```

```

}

private University createSampleUniversity() {
    Student student1 = new Student("Oleksii", "Artemov", "Artemovych", Sex.MALE, "S1");
    Student student2 = new Student("Anna", "Annenko", "Andriivna", Sex.FEMALE, "S2");
    List<Student> students = List.of(student1, student2);

    Human groupHead = new Human("Petro", "Kovalenko", "Anatoliyovych", Sex.MALE);
    Group group1 = new Group("122-21-1", groupHead, students);
    Group group2 = new Group("125-20-2", groupHead, students);

    Human departmentHead = new Human("Olha", "Shevchenko", "Petrivna", Sex.FEMALE);
    Department department1 = new Department("PZKS", departmentHead, List.of(group1,
group2));
    Department department2 = new Department("SAY", departmentHead, List.of(group1,
group2));

    Human facultyHead = new Human("Taras", "Dronov", "Illich", Sex.MALE);
    Faculty faculty1 = new Faculty("FIT", facultyHead, List.of(department1, department2));
    Faculty faculty2 = new Faculty("FE", facultyHead, List.of(department1, department2));

    Human universityHead = new Human("Nastya", "Sydorova", "Artemivna", Sex.FEMALE);
    return new University("Dnipro University of Technology", universityHead, List.of(faculty1,
faculty2));
}
}

```

Результат виконання програми:

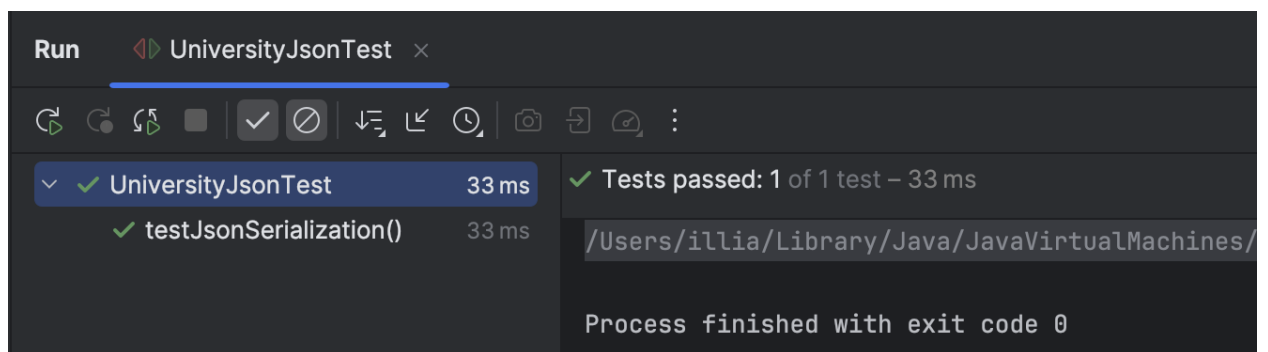


Рис. 4.1 – Результат виконання програми

Репозиторій з кодом програми: <https://github.com/illyaXD/Karapysh-javalabs>

Висновки: у ході виконання даної лабораторної роботи ми навчилися працювати з Junit та JSON у Java.