

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



**ЗВІТ**  
**про виконання лабораторної роботи**  
**з дисципліни**  
**«Поглиблене програмування в середовищі java»**  
**Лабораторна робота № 4**

**Виконав:**

студент гр. 126-21-1

Мовлик Дмитро Дмитрович

**Прийняв:**

Доцент

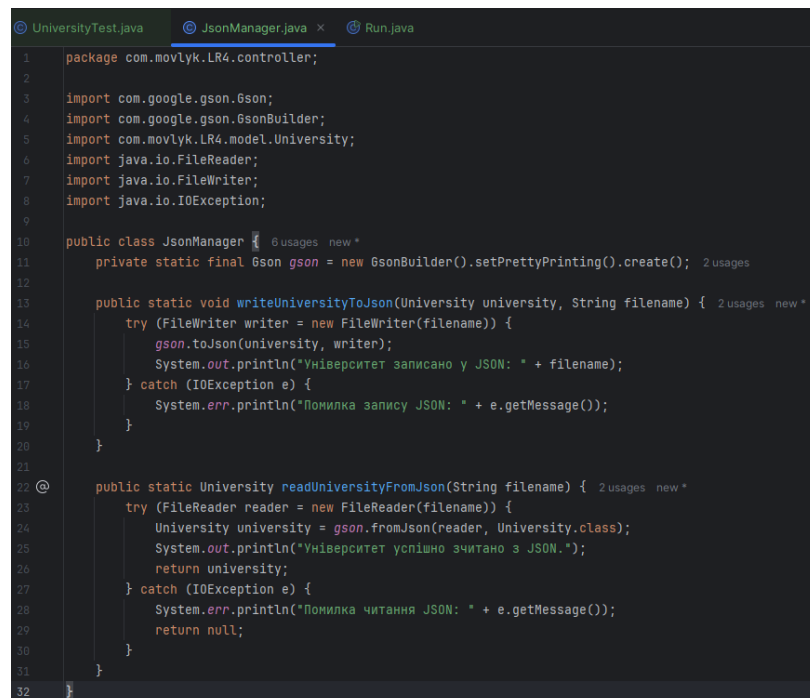
Мінеєв Олександр Сергійович

**Дніпро**  
**2025**

## Хід роботи

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.



```
1 package com.movlyk.LR4.controller;
2
3 import com.google.gson.Gson;
4 import com.google.gson.GsonBuilder;
5 import com.movlyk.LR4.model.University;
6 import java.io.FileReader;
7 import java.io.FileWriter;
8 import java.io.IOException;
9
10 public class JsonManager {
11     private static final Gson gson = new GsonBuilder().setPrettyPrinting().create();
12
13     public static void writeUniversityToJson(University university, String filename) {
14         try (FileWriter writer = new FileWriter(filename)) {
15             gson.toJson(university, writer);
16             System.out.println("Університет записано у JSON: " + filename);
17         } catch (IOException e) {
18             System.err.println("Помилка запису JSON: " + e.getMessage());
19         }
20     }
21
22     public static University readUniversityFromJson(String filename) {
23         try (FileReader reader = new FileReader(filename)) {
24             University university = gson.fromJson(reader, University.class);
25             System.out.println("Університет успішно зчитано з JSON.");
26             return university;
27         } catch (IOException e) {
28             System.err.println("Помилка читання JSON: " + e.getMessage());
29             return null;
30         }
31     }
32 }
```

Рис.1 – розроблений для запису та зчитування, клас *JsonManager*

```

1 package com.movlyk.LR4.test;
2
3 import com.movlyk.LR4.controller.JsonManager;
4 import com.movlyk.LR4.model.*;
5 import org.testng.annotations.Test;
6
7 import static org.junit.jupiter.api.Assertions.*;
8
9 class UniversityTest {
10
11     @Test
12     void testWriteAndReadUniversityFromJson() {
13         Human rector = new Human( firstName: "Іван", lastName: "Петров", middleName: "Михайлович", Sex.MALE);
14         University oldUniversity = new University( name: "Київський університет", rector);
15
16         Faculty faculty = new Faculty( name: "Факультет Комп'ютерних Наук");
17         Group group = new Group( name: "Група 101");
18         Student student1 = new Student( firstName: "Олег", lastName: "Іванов", middleName: "Степанович", Sex.MALE, group: "Група 101");
19         Student student2 = new Student( firstName: "Марія", lastName: "Петренко", middleName: "Олександрівна", Sex.FEMALE, group: "Група
20
21         group.addStudent(student1);
22         group.addStudent(student2);
23         faculty.addGroup(group);
24         oldUniversity.addFaculty(faculty);
25
26         String filename = "university_test.json";
27         JsonManager.writeUniversityToJson(oldUniversity, filename);
28         University newUniversity = JsonManager.readUniversityFromJson(filename);
29
30         assertNotNull(newUniversity, message: "Файл JSON не зчитався, об'єкт newUniversity == null");
31
32         assertEquals(oldUniversity.getName(), newUniversity.getName(), message: "Назва університетів не співпадають");
33         assertEquals(oldUniversity.getRector(), newUniversity.getRector(), message: "Ректори не співпадають");
34         assertEquals(oldUniversity.getFaculties().size(), newUniversity.getFaculties().size(), message: "Кількість факультетів не с
35
36         for (int i = 0; i < oldUniversity.getFaculties().size(); i++) {
37             assertEquals(oldUniversity.getFaculties().get(i).getName(), newUniversity.getFaculties().get(i).getName(), message: "Фа
38         }
39     }
40 }

```

**Рис.2** – Код JUnit тестування

Далі по роботі показано частину того, що було виведено в university.json (Рис.3)

```

1 {
2   "name": "Київський університет",
3   "rector": {
4     "firstName": "Іван",
5     "lastName": "Петров",
6     "middleName": "Михайлович",
7     "gender": "MALE"
8   },
9   "faculties": [
10    {
11      "name": "Факультет Комп'ютерних Наук",
12      "groups": [
13        {
14          "name": "Група 101",
15          "students": [
16            {
17              "group": "Група 101",
18              "firstName": "Олег",
19              "lastName": "Іванов",
20              "middleName": "Степанович",
21              "gender": "MALE"
22            },
23            {
24              "group": "Група 101",
25              "firstName": "Марія",
26              "lastName": "Петренко",
27              "middleName": "Олександрівна",
28              "gender": "FEMALE"
29            }
30          ]
31        }
32      ]
33    }
34  ]
35 }

```

**Рис.3** – Вміст файлу university.json

**Висновок:** у ході практичної роботи було додано можливість збереження університету у форматі JSON, його запис у файл, та зчитування з файлу та відновлення об'єкта. Було реалізовано клас `JsonManager` для роботи з JSON. Також розроблено Junit тест, який перевіряє коректність збереження та відновлення університету шляхом порівняння об'єктів за методом `equals`.