

**Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»**

**НАВЧАЛЬНО-НАУКОВИЙ  
ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  
*Кафедра інформаційних технологій та комп'ютерної інженерії*

**ЗВІТ  
З ЛАБОРАТОРНОЇ РОБОТИ №4**

дисципліна "Веб-застосунки з Java/Spring"

Виконавець,  
студент гр. 123м-24-1 \_\_\_\_\_ А.О. Торгольський  
(підпис)

Керівник, доц. \_\_\_\_\_ О.С. Мінесв  
(підпис)

**Дніпро  
2025**

## ЗМІСТ

4	Лабораторна робота №4 .....	3
	4.1 Мета лабораторної роботи .....	3
	4.2 Завдання до лабораторної роботи .....	3
	4.3 Опис виконання лабораторної роботи .....	4
	Висновки .....	6

## **4 ЛАБОРАТОРНА РОБОТА №4**

### **СТВОРЕННЯ JUNIT ТЕСТІВ ТА ЧИТАННЯ/ЗАПИС JSON- ФАЙЛІВ**

#### **4.1 Мета лабораторної роботи**

Ознайомлення з особливостями створення JUnit тестів в Java та читання/запис даних в json-файлах.

#### **4.2 Завдання до лабораторної роботи**

Для виконання лабораторної роботи необхідно:

- створити окрему гілку під назвою lr4 від гілки master;
- розробити JUnit тест для попередньої лабораторної роботи для читання та запису json-файлів;
- залити репозиторій проєкту на GitHub.

#### **4.3 Опис виконання лабораторної роботи**

Створено гілку під назвою lr4 від гілки master (див. рисунок 3.1).

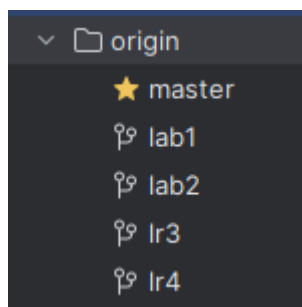


Рисунок 3.1 – Створення гілки lab2

Після чого в середовищі розробки було розроблено тест для створення комплексного об'єкту університет та запису і читання у json-файл.

Створено 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 записаний в файл у форматі json. Потім з цього файлу зчитаний та відновлений як newUniversity. В тесті порівняння newUniversity та

oldUniversity за допомогою методу equals. Для запису та зчитування університету у форматі json розроблений клас JsonManager (див. рисунок 3.2).

```
public class JsonManager { 4 usages  ▲ Andrii Torholskyi
    public void saveUniversityToJson(University university, String filePath) { 2 usages  ▲ Andrii Torholskyi
        Gson gson = new GsonBuilder()
            .setPrettyPrinting()
            .registerTypeAdapter(Human.class, new HumanAdapter())
            .create();
        try (FileWriter writer = new FileWriter(filePath)) {
            gson.toJson(university, writer);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public University loadUniversityFromJson(String filePath) { 1 usage  ▲ Andrii Torholskyi
        Gson gson = new GsonBuilder()
            .registerTypeAdapter(Human.class, new HumanAdapter())
            .create();
        try (FileReader reader = new FileReader(filePath)) {
            return gson.fromJson(reader, University.class);
        } catch (IOException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

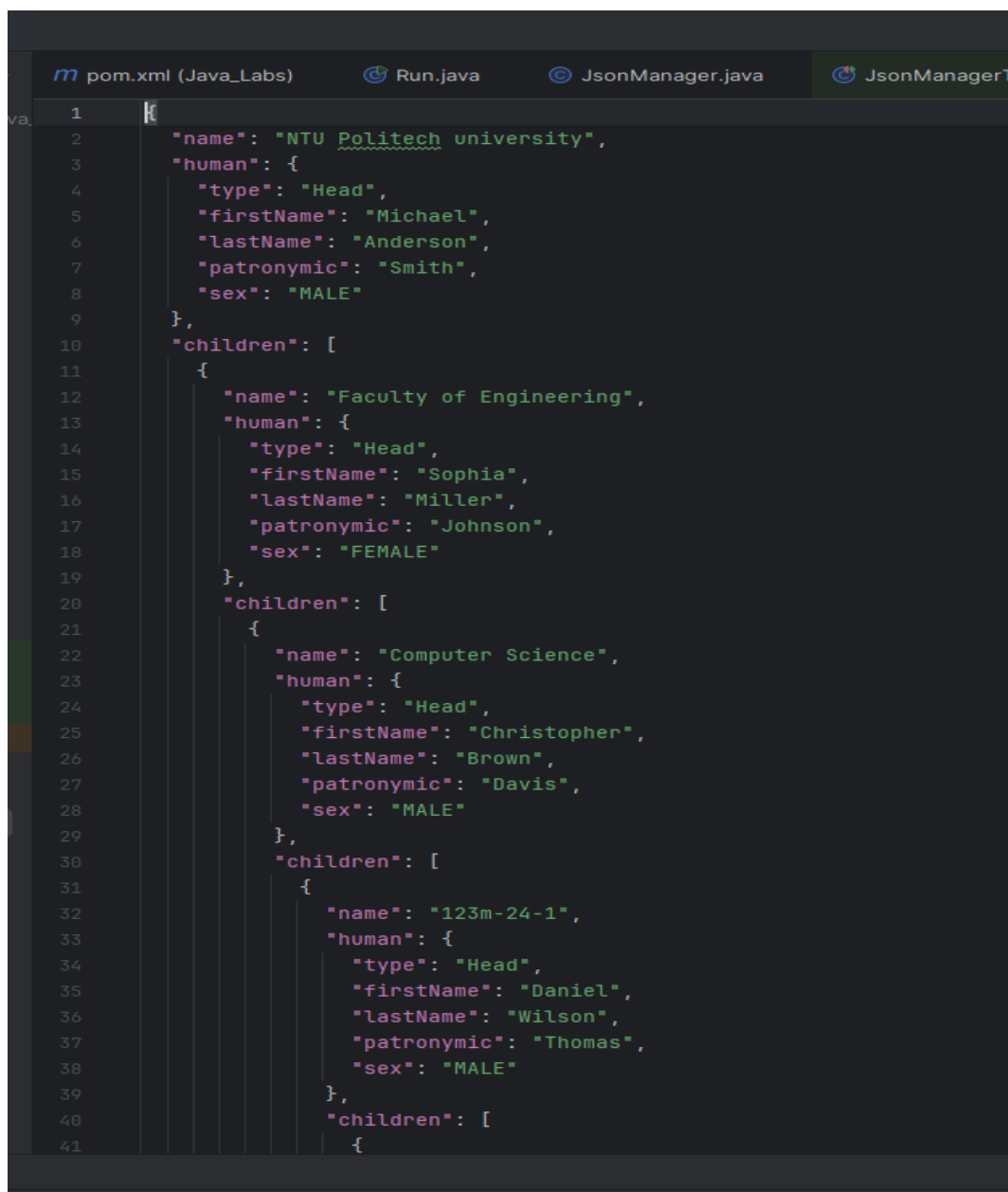
Рисунок 3.2 – Клас JsonManager

На рисунку 3.3 наведений фрагмент коду тесту.

```
12
13 class JsonManagerTest {  ▲ Andrii Torholskyi
14
15     private static final String FILE_PATH = "test_university.json"; 5 usages
16
17     @Test  ▲ Andrii Torholskyi
18     void saveUniversityToJson() {
19         JsonManager jsonManager = new JsonManager();
20         Run run = new Run();
21         University university = run.createTypicalUniversity();
22
23         jsonManager.saveUniversityToJson(university, FILE_PATH);
24
25         File file = new File(FILE_PATH);
26         assertTrue(file.exists(), message: "File should be created");
27
28         try {
29             String fileContent = new String(Files.readAllBytes(Paths.get(FILE_PATH)));
30             assertFalse(fileContent.isEmpty(), message: "File should not be empty");
31         } catch (IOException e) {
32             fail("Failed to read file: " + e.getMessage());
33         }
34     }
35
36     @Test  ▲ Andrii Torholskyi
37     void loadUniversityFromJson() {
38         JsonManager jsonManager = new JsonManager();
39         Run run = new Run();
40         University originalUniversity = run.createTypicalUniversity();
41
42         jsonManager.saveUniversityToJson(originalUniversity, FILE_PATH);
43
44         University loadedUniversity = jsonManager.loadUniversityFromJson(FILE_PATH);
45
46         assertNotNull(loadedUniversity, message: "Loaded university should not be null");
47         assertEquals(originalUniversity, loadedUniversity, message: "Loaded university should be equal to the original");
48     }
49 }
50 }
```

Рисунок 3.3 – JUnit тест

Вміст файлу test\_university.json наведено на рисунку 3.4.



```

1  {
2    "name": "NTU Politech university",
3    "human": {
4      "type": "Head",
5      "firstName": "Michael",
6      "lastName": "Anderson",
7      "patronymic": "Smith",
8      "sex": "MALE"
9    },
10   "children": [
11     {
12       "name": "Faculty of Engineering",
13       "human": {
14         "type": "Head",
15         "firstName": "Sophia",
16         "lastName": "Miller",
17         "patronymic": "Johnson",
18         "sex": "FEMALE"
19       },
20       "children": [
21         {
22           "name": "Computer Science",
23           "human": {
24             "type": "Head",
25             "firstName": "Christopher",
26             "lastName": "Brown",
27             "patronymic": "Davis",
28             "sex": "MALE"
29           },
30           "children": [
31             {
32               "name": "123m-24-1",
33               "human": {
34                 "type": "Head",
35                 "firstName": "Daniel",
36                 "lastName": "Wilson",
37                 "patronymic": "Thomas",
38                 "sex": "MALE"
39               },
40               "children": [
41                 {

```

Рисунок 3.4 – Вміст файлу test\_university.json

Після виконання всіх завдань було відправлено файли реалізації на віддалений репозиторій GitHub (див. рисунок 3.5).

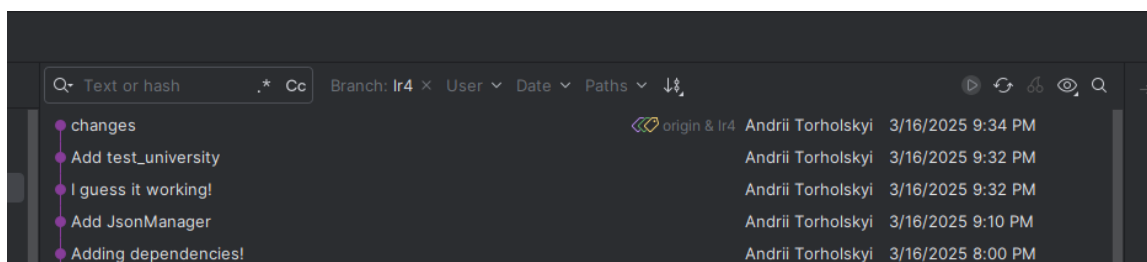


Рисунок 3.5 – Додавання файлів

## **ВИСНОВКИ**

Під час виконання лабораторної роботи були одержані навички зі створення JUnit тестів в Java та читання/запис даних в json-файлах.