

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



**ЗВІТ**  
**про виконання практичних робіт**  
**з дисципліни**  
**«Поглиблене програмування в середовищі Java»**  
**Лабораторна робота № 4**

Виконав:  
студент гр. 121-21-1  
Соболев Єгор Романович

Перевірив:  
Доц. Мінєєв О.С.

**Дніпро**  
**2025**

**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.

### Код програми:

```
package org.example.controller;

import org.example.model.*;

public class FacultyCreator { 1 usage
    public static Faculty createFaculty(String name, Human head) { return new Faculty(name, head); }
}

package org.example.controller;

import org.example.model.*;

public class FacultyCreator { 1 usage
    public static Faculty createFaculty(String name, Human head) { return new Faculty(name, head); }
}
```

```
package org.example.controller;

import org.example.model.*;

class GroupCreator { no usages
@ > public static Group createGroup(String name, Human head) { return new Group(name, head); }
}

package org.example.controller;

import org.example.model.*;

class StudentCreator { no usages
@ public static Student createStudent(String firstName, String lastName, String middleName, Sex gender) { no usages
    return new Student(firstName, lastName, middleName, gender);
}
}

package org.example.controller;

import org.example.model.*;

public class UniversityCreator { 1 usage
@ > public static University createUniversity(String name, Human head) { return new University(name, head); }
}

Lab4\...\FacultyCreator.java  Lab4\...\GroupCreator.java  Lab4\...\StudentCreator.java  Lab4\...\UniversityCreator.java  Lab4\...

package org.example.main;

> import ...
> public class Run {
> > public static void main(String[] args) { createTypicalUniversity(); }

    public static void createTypicalUniversity() { 1 usage
        Human rector = new Human( firstName: "Іван", lastName: "Петров", middleName: "Олексійович", Sex.MALE) {};
        University university = UniversityCreator.createUniversity( name: "Національний університет", rector);

        Human dean = new Human( firstName: "Марія", lastName: "Іванова", middleName: "Сергіївна", Sex.FEMALE) {};
        Faculty faculty = FacultyCreator.createFaculty( name: "Факультет інформатики", dean);
        university.addFaculty(faculty);

        System.out.println("Університет створено!");
    }
}
```

```

public class Department { 8 usages
    private String name; 4 usages
    private Human head; 1 usage
    private List<Group> groups; 5 usages

    public Department(String name, Human head) { 2 usages
        this.name = name;
        this.head = head;
        this.groups = new ArrayList<>();
    }

    public void addGroup(Group group) { groups.add(group); }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Department that = (Department) o;
        return Objects.equals(name, that.name) &&
            Objects.equals(groups, that.groups);
    }

    @Override
    public int hashCode() { return Objects.hash(name, groups); }
}

```

```

{
  "name": "Тестовий університет",
  "faculties": [
    {
      "name": "Факультет 1",
      "departments": [
        {
          "name": "Кафедра 1",
          "groups": [
            {
              "name": "Група 1",
              "students": [
                {
                  "firstName": "Студент1",
                  "lastName": "Прізвище",
                  "middleName": "Ім\у0027я",
                  "gender": "MALE"
                },
                {
                  "firstName": "Студент2",
                  "lastName": "Прізвище",
                  "middleName": "Ім\у0027я",
                  "gender": "MALE"
                }
              ]
            }
          ]
        },
        {
          "name": "Група 2",
          "students": [
            {
              "firstName": "Студент1",
              "lastName": "Прізвище",
              "middleName": "Ім\у0027я",
              "gender": "MALE"
            },
            {
              "firstName": "Студент2",
              "lastName": "Прізвище"
            }
          ]
        }
      ]
    }
  ]
}

```

## Результат роботи програми:

```

0 ms  ✓ Tests passed: 1 of 1 test - 150 ms
0 ms  ✓ Університет успішно збережено у файл: university.json
      ✓ Університет успішно завантажено з файлу: university.json

```