

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



Лабораторна робота №4

з дисципліни:

«Поглиблене програмування Java»

Виконав:

студент групи 123-21-1

Зеленський Д. І.

Перевірив:

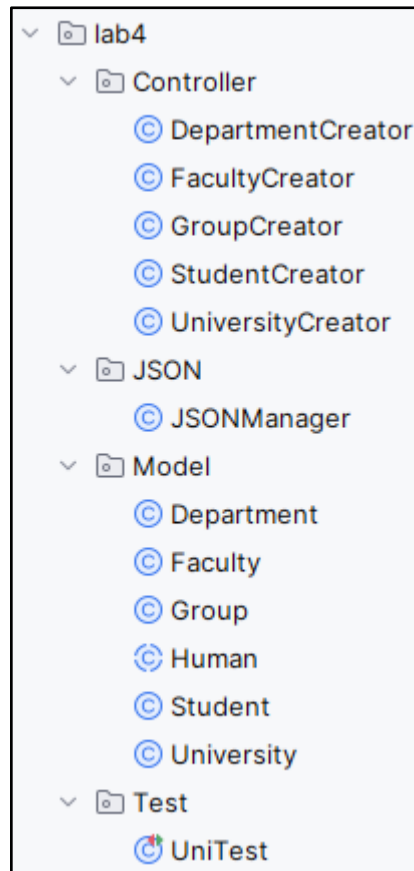
Мінєєв О.С.

м. Дніпро

2025

Лабораторна робота №4

JSON



JSONManager.java

```
package org.green.lab4.JSON;

import com.google.gson.*;
import org.green.lab4.Model.*;

import java.io.File;
import java.io.FileWriter;
import java.io.FileReader;
import java.io.IOException;
import java.lang.reflect.Type;

public class JSONManager {
    // Ініціалізація об'єкта Gson з налаштуваннями для форматування та
    // адаптера для серіалізації/десеріалізації класу Human
    private static final Gson gson = new GsonBuilder()
        .setPrettyPrinting() // Додає красиве форматування для виводу
        JSON
        .registerTypeAdapter(Human.class, new HumanAdapter()) //
        Регіструємо власний адаптер для класу Human
        .create(); // Створюємо об'єкт Gson з вказаними налаштуваннями

    // Метод для серіалізації об'єкта University в JSON формат та запису в
    // файл
}
```

```

    public static void saveToFile(University university, String filename)
    throws IOException {
        // Використовуємо FileWriter для запису в файл
        try (FileWriter writer = new FileWriter(filename)) {
            // Перетворюємо об'єкт university в JSON і записуємо в файл
            gson.toJson(university, writer);
        }
    }

    // Метод для десеріалізації JSON з файлу в об'єкт University
    public static University loadFromFile(String filename) throws
    IOException {
        // Використовуємо FileReader для зчитування з файлу
        try (FileReader reader = new FileReader(filename)) {
            // Перетворюємо JSON з файлу в об'єкт типу University
            return gson.fromJson(reader, University.class);
        }
    }

    // Внутрішній клас, який реалізує адаптер для серіалізації та
    десеріалізації класу Human
    private static class HumanAdapter implements JsonSerializer<Human>,
    JsonDeserializer<Human> {

        // Метод для серіалізації об'єкта Human в JSON
        @Override
        public JsonElement serialize(Human src, Type typeOfSrc,
        JsonSerializationContext context) {
            // Перетворюємо об'єкт в JSON об'єкт
            JsonObject jsonObject =
            context.serialize(src).getAsJsonObject();
            // Додаємо додаткове поле "type", яке вказує на реальний клас
            об'єкта
            jsonObject.addProperty("type", src.getClass().getSimpleName());
            return jsonObject;
        }

        // Метод для десеріалізації JSON в об'єкт Human
        @Override
        public Human deserialize(JsonElement json, Type typeOfT,
        JsonDeserializationContext context) throws JsonParseException {
            // Отримуємо JSON об'єкт
            JsonObject jsonObject = json.getAsJsonObject();
            // Отримуємо значення поля "type" для визначення реального класу
            String type = jsonObject.get("type").getString();
            // Вибір класу в залежності від значення "type"
            Class<? extends Human> clazz = "Student".equals(type) ?
            Student.class : Human.class;
            // Десеріалізуємо JSON в об'єкт відповідного класу
            return context.deserialize(jsonObject, clazz);
        }
    }
}

```

UniTest.java

```
package org.green.lab4.Test;

import org.green.lab4.JSON.JSONManager;
import org.green.lab4.Model.*;
import org.junit.jupiter.api.Test;

import java.io.IOException;

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

public class UniTest {
    // Тестування серіалізації і десеріалізації об'єкта University в/з JSON
    // формату
    @Test
    public void testJsonSerialization() throws IOException {
        // Створення університету з назвою "Test University" та студентом
        // "John Doe"
        University oldUniversity = new University("Test University", new
        Student("John", "Doe", "Michael", Human.Sex.MALE, "S90"));

        // Створення факультету "Faculty of Science" та студента "Alice
        // Smith"
        Faculty faculty1 = new Faculty("Faculty of Science", new
        Student("Daniel", "Evans", "Michael", Human.Sex.MALE, "S91"));
        Faculty faculty2 = new Faculty("Faculty of Humanities", new
        Student("Sophia", "Clark", "Elizabeth", Human.Sex.FEMALE, "S92"));

        // Створення кафедр
        Department department11 = new Department("Mathematical Modeling",
        new Student("Irina", "Zakharova", "Alexeevna", Human.Sex.FEMALE, "S93"));
        Department department12 = new Department("Statistics and Data
        Analysis", new Student("Maxim", "Sergeyenko", "Grigorievich",
        Human.Sex.MALE, "S94"));
        Department department21 = new Department("Optimization and
        Operations Research", new Student("Elena", "Kravchenko", "Vasilyevna",
        Human.Sex.FEMALE, "S95"));
        Department department22 = new Department("Computer Modeling", new
        Student("Vadim", "Polyakov", "Andreevich", Human.Sex.MALE, "S96"));

        // Створення груп
        Group group111 = new Group("CS-101", new Student("John", "Doe",
        "Michael", Human.Sex.MALE, "S97"));
        Group group112 = new Group("CS-102", new Student("Alice", "Smith",
        "Jane", Human.Sex.FEMALE, "S98"));
        Group group121 = new Group("CS-103", new Student("David", "Brown",
        "Robert", Human.Sex.MALE, "S99"));
        Group group122 = new Group("CS-104", new Student("Emma", "Johnson",
        "Rose", Human.Sex.FEMALE, "S100"));
        Group group211 = new Group("CS-105", new Student("Michael",
        "Williams", "James", Human.Sex.MALE, "S101"));
        Group group212 = new Group("CS-106", new Student("Sophia", "Miller",
        "Anne", Human.Sex.FEMALE, "S102"));
        Group group221 = new Group("CS-107", new Student("Daniel", "Taylor",
        "Thomas", Human.Sex.MALE, "S103"));
        Group group222 = new Group("CS-108", new Student("Olivia",
        "Anderson", "Grace", Human.Sex.FEMALE, "S104"));

        // Додавання студентів до групи
        group111.addStudent(new Student("Charlie", "Johnson", "David",
        Human.Sex.MALE, "S105"));
        group111.addStudent(new Student("Emma", "Williams", "Sophia",
```

```

Human.Sex.FEMALE, "S106"));
    group112.addStudent(new Student("Liam", "Brown", "James",
Human.Sex.MALE, "S107"));
    group112.addStudent(new Student("Olivia", "Jones", "Grace",
Human.Sex.FEMALE, "S108"));
    group121.addStudent(new Student("Noah", "Garcia", "Henry",
Human.Sex.MALE, "S109"));
    group121.addStudent(new Student("Ava", "Martinez", "Isabella",
Human.Sex.FEMALE, "S110"));
    group122.addStudent(new Student("William", "Davis", "Benjamin",
Human.Sex.MALE, "S111"));
    group122.addStudent(new Student("Sophia", "Rodriguez", "Charlotte",
Human.Sex.FEMALE, "S112"));
    group211.addStudent(new Student("James", "Lopez", "Elijah",
Human.Sex.MALE, "S113"));
    group211.addStudent(new Student("Mia", "Hernandez", "Scarlett",
Human.Sex.FEMALE, "S114"));
    group212.addStudent(new Student("Benjamin", "Gonzalez", "Lucas",
Human.Sex.MALE, "S115"));
    group212.addStudent(new Student("Charlotte", "Wilson", "Avery",
Human.Sex.FEMALE, "S116"));
    group221.addStudent(new Student("Elijah", "Anderson", "Daniel",
Human.Sex.MALE, "S117"));
    group221.addStudent(new Student("Amelia", "Thomas", "Evelyn",
Human.Sex.FEMALE, "S118"));
    group222.addStudent(new Student("Lucas", "Taylor", "Jack",
Human.Sex.MALE, "S119"));
    group222.addStudent(new Student("Harper", "Moore", "Lillian",
Human.Sex.FEMALE, "S120"));

    // Додавання групи до кафедри, кафедри до факультету та факультету
до університету
    department11.addGroup(group111);
    department11.addGroup(group112);
    department12.addGroup(group121);
    department12.addGroup(group122);
    department21.addGroup(group211);
    department21.addGroup(group212);
    department22.addGroup(group221);
    department22.addGroup(group222);

    faculty1.addDepartment(department11);
    faculty1.addDepartment(department12);
    faculty2.addDepartment(department21);
    faculty2.addDepartment(department22);

    oldUniversity.addFaculty(faculty1);
    oldUniversity.addFaculty(faculty2);

    // Збереження об'єкта oldUniversity у файл "university.json" у JSON
форматі
    JSONManager.saveToFile(oldUniversity, "university.json");
    // Завантаження університету з JSON файлу в об'єкт newUniversity
    University newUniversity =
JSONManager.loadFromFile("university.json");
    // Перевірка чи рівні об'єкти oldUniversity і newUniversity
    System.out.println("Are universities equal? " +
oldUniversity.equals(newUniversity));
    // Використовуємо JUnit assertEquals для порівняння об'єктів
    // Якщо об'єкти однакові, тест пройде успішно
    assertEquals(oldUniversity, newUniversity);
}
}

```

Результат

```
{
  "name": "Test University",
  "head": {
    "studentId": "S90",
    "firstName": "John",
    "lastName": "Doe",
    "patronymic": "Michael",
    "sex": "MALE",
    "type": "Student"
  },
  "faculties": [
    {
      "name": "Faculty of Science",
      "head": {
        "studentId": "S91",
        "firstName": "Daniel",
        "lastName": "Evans",
        "patronymic": "Michael",
        "sex": "MALE",
        "type": "Student"
      },
      "departments": [
        {
          "name": "Mathematical Modeling",
          "head": {
            "studentId": "S93",
            "firstName": "Irina",
            "lastName": "Zakharova",
            "patronymic": "Alexeevna",
            "sex": "FEMALE",
            "type": "Student"
          },
          "groups": [
            {
              "name": "CS-101",
              "head": {
                "studentId": "S97",
                "firstName": "John",
                "lastName": "Doe",
                "patronymic": "Michael",
                "sex": "MALE",
                "type": "Student"
              },
              "students": [
                {
                  "studentId": "S105",
                  "firstName": "Charlie",
                  "lastName": "Johnson",
                  "patronymic": "David",
                  "sex": "MALE"
                },
                {
                  "studentId": "S106",
                  "firstName": "Emma",
                  "lastName": "Williams",
                  "patronymic": "Sophia",
                  "sex": "FEMALE"
                }
              ]
            },
            {
              "name": "CS-102",
```

```

        "head": {
            "studentId": "S98",
            "firstName": "Alice",
            "lastName": "Smith",
            "patronymic": "Jane",
            "sex": "FEMALE",
            "type": "Student"
        },
        "students": [
            {
                "studentId": "S107",
                "firstName": "Liam",
                "lastName": "Brown",
                "patronymic": "James",
                "sex": "MALE"
            },
            {
                "studentId": "S108",
                "firstName": "Olivia",
                "lastName": "Jones",
                "patronymic": "Grace",
                "sex": "FEMALE"
            }
        ]
    },
    {
        "name": "Statistics and Data Analysis",
        "head": {
            "studentId": "S94",
            "firstName": "Maxim",
            "lastName": "Sergeyenko",
            "patronymic": "Grigorievich",
            "sex": "MALE",
            "type": "Student"
        },
        "groups": [
            {
                "name": "CS-103",
                "head": {
                    "studentId": "S99",
                    "firstName": "David",
                    "lastName": "Brown",
                    "patronymic": "Robert",
                    "sex": "MALE",
                    "type": "Student"
                },
                "students": [
                    {
                        "studentId": "S109",
                        "firstName": "Noah",
                        "lastName": "Garcia",
                        "patronymic": "Henry",
                        "sex": "MALE"
                    },
                    {
                        "studentId": "S110",
                        "firstName": "Ava",
                        "lastName": "Martinez",
                        "patronymic": "Isabella",
                        "sex": "FEMALE"
                    }
                ]
            }
        ]
    }
]

```

```

    },
    {
      "name": "CS-104",
      "head": {
        "studentId": "S100",
        "firstName": "Emma",
        "lastName": "Johnson",
        "patronymic": "Rose",
        "sex": "FEMALE",
        "type": "Student"
      },
      "students": [
        {
          "studentId": "S111",
          "firstName": "William",
          "lastName": "Davis",
          "patronymic": "Benjamin",
          "sex": "MALE"
        },
        {
          "studentId": "S112",
          "firstName": "Sophia",
          "lastName": "Rodriguez",
          "patronymic": "Charlotte",
          "sex": "FEMALE"
        }
      ]
    }
  ]
},
{
  "name": "Faculty of Humanities",
  "head": {
    "studentId": "S92",
    "firstName": "Sophia",
    "lastName": "Clark",
    "patronymic": "Elizabeth",
    "sex": "FEMALE",
    "type": "Student"
  },
  "departments": [
    {
      "name": "Optimization and Operations Research",
      "head": {
        "studentId": "S95",
        "firstName": "Elena",
        "lastName": "Kravchenko",
        "patronymic": "Vasilyevna",
        "sex": "FEMALE",
        "type": "Student"
      },
      "groups": [
        {
          "name": "CS-105",
          "head": {
            "studentId": "S101",
            "firstName": "Michael",
            "lastName": "Williams",
            "patronymic": "James",
            "sex": "MALE",
            "type": "Student"
          }
        }
      ]
    }
  ]
}

```



```

        "students": [
            {
                "studentId": "S113",
                "firstName": "James",
                "lastName": "Lopez",
                "patronymic": "Elijah",
                "sex": "MALE"
            },
            {
                "studentId": "S114",
                "firstName": "Mia",
                "lastName": "Hernandez",
                "patronymic": "Scarlett",
                "sex": "FEMALE"
            }
        ]
    },
    {
        "name": "CS-106",
        "head": {
            "studentId": "S102",
            "firstName": "Sophia",
            "lastName": "Miller",
            "patronymic": "Anne",
            "sex": "FEMALE",
            "type": "Student"
        },
        "students": [
            {
                "studentId": "S115",
                "firstName": "Benjamin",
                "lastName": "Gonzalez",
                "patronymic": "Lucas",
                "sex": "MALE"
            },
            {
                "studentId": "S116",
                "firstName": "Charlotte",
                "lastName": "Wilson",
                "patronymic": "Avery",
                "sex": "FEMALE"
            }
        ]
    }
],
{
    "name": "Computer Modeling",
    "head": {
        "studentId": "S96",
        "firstName": "Vadim",
        "lastName": "Polyakov",
        "patronymic": "Andreevich",
        "sex": "MALE",
        "type": "Student"
    },
    "groups": [
        {
            "name": "CS-107",
            "head": {
                "studentId": "S103",
                "firstName": "Daniel",
                "lastName": "Taylor",
                "patronymic": "Thomas",

```

```

        "sex": "MALE",
        "type": "Student"
    },
    "students": [
        {
            "studentId": "S117",
            "firstName": "Elijah",
            "lastName": "Anderson",
            "patronymic": "Daniel",
            "sex": "MALE"
        },
        {
            "studentId": "S118",
            "firstName": "Amelia",
            "lastName": "Thomas",
            "patronymic": "Evelyn",
            "sex": "FEMALE"
        }
    ]
},
{
    "name": "CS-108",
    "head": {
        "studentId": "S104",
        "firstName": "Olivia",
        "lastName": "Anderson",
        "patronymic": "Grace",
        "sex": "FEMALE",
        "type": "Student"
    },
    "students": [
        {
            "studentId": "S119",
            "firstName": "Lucas",
            "lastName": "Taylor",
            "patronymic": "Jack",
            "sex": "MALE"
        },
        {
            "studentId": "S120",
            "firstName": "Harper",
            "lastName": "Moore",
            "patronymic": "Lillian",
            "sex": "FEMALE"
        }
    ]
}
]
}
]
}
]
}
]
}
]
}
}

```