

Лабораторна робота №4

JUnit. Json

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. Не змішувати ці роботи не в якому разі.

Код програми JsonManager.java

```
package org.KostiuchenkoD.Lab4.JSON;

import com.google.gson.*;
import org.KostiuchenkoD.Lab4.Model.*;

import java.io.*;
import java.io.*;
import java.io.*;
import java.lang.reflect.Type;

public class JSONManager {

    private static final Gson gson = new GsonBuilder()
        .setPrettyPrinting()
        .registerTypeAdapter(Human.class, new HumanAdapter())
        .create();

    public static void saveToFile(University university, String filename)
        throws IOException {

        try (FileWriter writer = new FileWriter(filename)) {

            gson.toJson(university, writer);

        }

    }

}
```

```

    public static University loadFromFile(String filename) throws IOException
    {

        try (FileReader reader = new FileReader(filename)) {

            return gson.fromJson(reader, University.class);

        }

    }

    private static class HumanAdapter implements JsonSerializer<Human>,
JsonDeserializer<Human> {

        @Override
        public JsonElement serialize(Human src, Type typeOfSrc,
JsonSerializationContext context) {

            JsonObject jsonObject = context.serialize(src).getAsJsonObject();

            jsonObject.addProperty("type", src.getClass().getSimpleName());
            return jsonObject;

        }

        @Override
        public Human deserialize(JsonElement json, Type typeOfT,
JsonDeserializationContext context) throws JsonParseException {

            JsonObject jsonObject = json.getAsJsonObject();

            String type = jsonObject.get("type").getString();

            Class<? extends Human> clazz = "Student".equals(type) ?
Student.class : Human.class;

            return context.deserialize(json, clazz);

        }

    }
}

```

JUnit_Test.java

```

package org.KostiuchenkoD.Lab4.test;

import org.KostiuchenkoD.Lab4.JSON.JSONManager;
import org.KostiuchenkoD.Lab4.Model.*;
import org.junit.jupiter.api.Test;

import java.io.IOException;

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

public class JUnit_test {

    @Test
    public void testJsonSerialization() throws IOException {

```

```

University oldUniversity = new University("Test University", new
Student("Ivan", "Petrenko", "Ivanovych", Human.Sex.MALE, "S90"));

Faculty faculty1 = new Faculty("Faculty of Science", new
Student("Danylo", "Evans", "Mykhailovych", Human.Sex.MALE, "S91"));
Faculty faculty2 = new Faculty("Faculty of Humanities", new
Student("Sofia", "Clark", "Elizavetivna", Human.Sex.FEMALE, "S92"));

Department department11 = new Department("Mathematical Modeling", new
Student("Iryna", "Zakharova", "Oleksiiyivna", Human.Sex.FEMALE, "S93"));
Department department12 = new Department("Statistics and Data
Analysis", new Student("Maksym", "Sergeyenko", "Hryhorovych", Human.Sex.MALE,
"S94"));
Department department21 = new Department("Optimization and Operations
Research", new Student("Olena", "Kravchenko", "Vasylivna", Human.Sex.FEMALE,
"S95"));
Department department22 = new Department("Computer Modeling", new
Student("Vadym", "Polyakov", "Andriyovych", Human.Sex.MALE, "S96"));

Group group111 = new Group("CS-101", new Student("Ivan", "Petrenko",
"Ivanovych", Human.Sex.MALE, "S97"));
Group group112 = new Group("CS-102", new Student("Olena",
"Sidorenko", "Ivanivna", Human.Sex.FEMALE, "S98"));
Group group121 = new Group("CS-103", new Student("Dmytro", "Brown",
"Robertovych", Human.Sex.MALE, "S99"));
Group group122 = new Group("CS-104", new Student("Yuliia", "Johnson",
"Rose", Human.Sex.FEMALE, "S100"));
Group group211 = new Group("CS-105", new Student("Mykhailo",
"Williams", "Yakovych", Human.Sex.MALE, "S101"));
Group group212 = new Group("CS-106", new Student("Sofia", "Miller",
"Annivna", Human.Sex.FEMALE, "S102"));
Group group221 = new Group("CS-107", new Student("Danylo", "Taylor",
"Thomasovych", Human.Sex.MALE, "S103"));
Group group222 = new Group("CS-108", new Student("Olha", "Anderson",
"Grace", Human.Sex.FEMALE, "S104"));

group111.addStudent(new Student("Charlie", "Johnson", "Davidovych",
Human.Sex.MALE, "S105"));
group111.addStudent(new Student("Yevheniia", "Williams", "Sofiiivna",
Human.Sex.FEMALE, "S106"));
group112.addStudent(new Student("Leonid", "Brown", "Yakovych",
Human.Sex.MALE, "S107"));
group112.addStudent(new Student("Olha", "Jones", "Grace",
Human.Sex.FEMALE, "S108"));
group121.addStudent(new Student("Noah", "Garcia", "Henry",
Human.Sex.MALE, "S109"));
group121.addStudent(new Student("Anna", "Martinez", "Isabella",
Human.Sex.FEMALE, "S110"));
group122.addStudent(new Student("Viktor", "Davis", "Benjamin",
Human.Sex.MALE, "S111"));
group122.addStudent(new Student("Sofia", "Rodriguez", "Charlotte",
Human.Sex.FEMALE, "S112"));
group211.addStudent(new Student("Yaroslav", "Lopez", "Elijah",
Human.Sex.MALE, "S113"));

```

```

        group211.addStudent(new Student("Mariia", "Hernandez", "Scarlett",
Human.Sex.FEMALE, "S114"));
        group212.addStudent(new Student("Bohdan", "Gonzalez", "Lucas",
Human.Sex.MALE, "S115"));
        group212.addStudent(new Student("Khrystyna", "Wilson", "Avery",
Human.Sex.FEMALE, "S116"));
        group221.addStudent(new Student("Elijah", "Anderson", "Danylovyh",
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("Hanna", "Moore", "Lillian",
Human.Sex.FEMALE, "S120"));

        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);

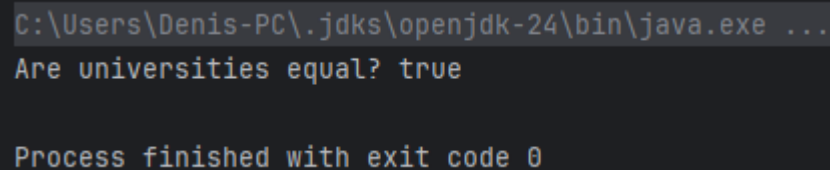
        JSONManager.saveToFile(oldUniversity, "university.json");

        University newUniversity =
JSONManager.loadFromFile("university.json");

        System.out.println("Are universities equal? " +
oldUniversity.equals(newUniversity));

        assertEquals(oldUniversity, newUniversity);
    }
}

```



```

C:\Users\Denis-PC\jdk\openjdk-24\bin\java.exe ...
Are universities equal? true

Process finished with exit code 0

```

Рис.1 - результат