Міністерство освіти і науки України
НТУ «Дніпровська політехніка»
Кафедра програмного забезпечення комп’ютерних систем

Звіт з лабораторної роботи №4

З дисципліни «Поглиблене програмування Java»

Виконала:
студентка групи 122-21-2
Таран Ірина Віталіївна
Перевірив:
доцент Мінєєв Олександр Сергійович

Дніпро
2025

ЛАБОРАТОРНА РОБОТА №4

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 проекті 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.

Хід роботи:

Лістинг конфігураційного файлу pom.xml:

pom.xml (lab4)

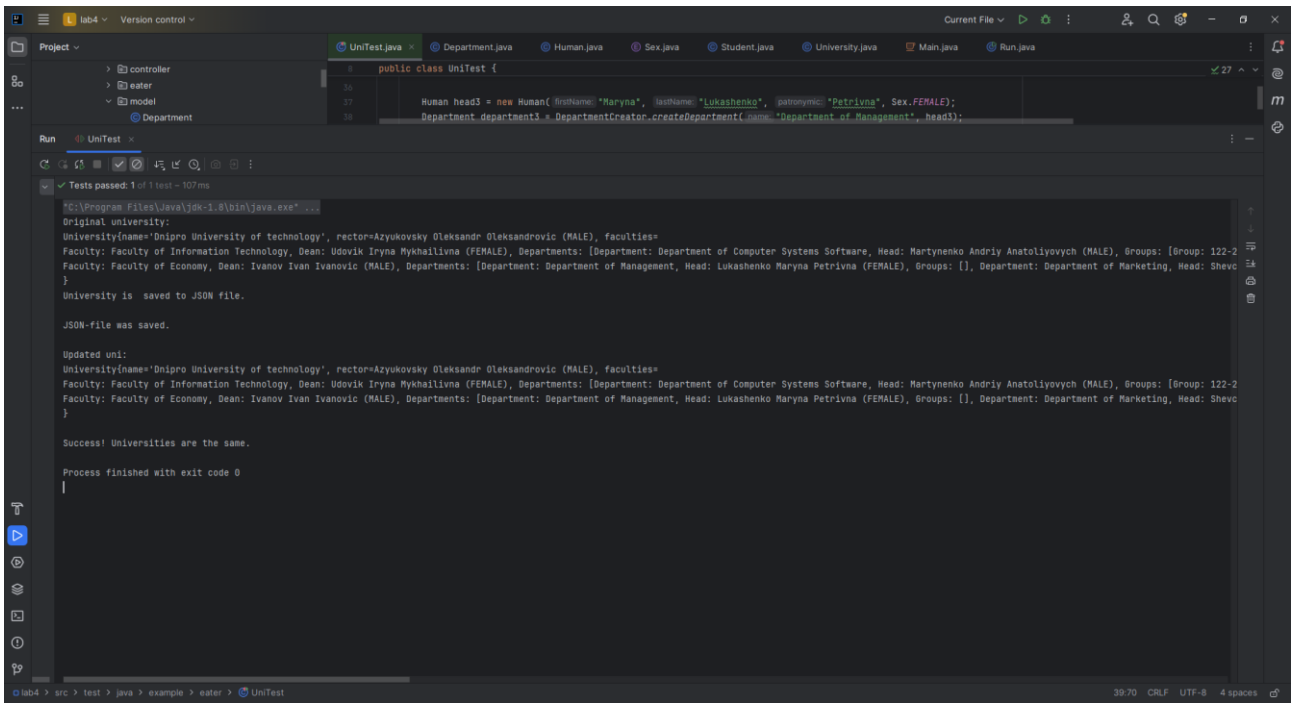
```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>lab4</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>

        <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
        <dependency>
            <groupId>com.google.code.gson</groupId>
            <artifactId>gson</artifactId>
            <version>2.8.8</version>
        </dependency>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-api</artifactId>
            <version>5.8.1</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-engine</artifactId>
            <version>5.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

</project>
```

Лістинг UniTest.java:

```
package example.eater;

import example.controller.*;
import example.model.*;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class UniTest {
    @Test
    public void testUniversityJsonSerialization() {

        Human rector = new Human("Oleksandr", "Azyukovsky", "Oleksandrovic",
Sex.MALE);
        University oldUniversity = UniversityCreator.createUniversity(
                "Dnipro University of technology",
                rector
```

```
        );


        Human dean1 = new Human("Iryna", "Udovik", "Mykhailivna", Sex.FEMALE);
        Faculty faculty1 = FacultyCreator.createFaculty("Faculty of Information
Technology", dean1);
        UniversityCreator.addFacultyToUniversity(oldUniversity, faculty1);

        Human dean2 = new Human("Ivan", "Ivanov", "Ivanovic", Sex.MALE);
        Faculty faculty2 = FacultyCreator.createFaculty("Faculty of Economy",
dean2);
        UniversityCreator.addFacultyToUniversity(oldUniversity, faculty2);


        Human head1 = new Human("Andriy", "Martynenko","Anatoliyovych", Sex.MALE);
        Department department1 = DepartmentCreator.createDepartment("Department of
Computer Systems Software", head1);
        FacultyCreator.addDepartmentToFaculty(faculty1, department1);

        Human head2 = new Human("Mykola", "Shvets", "Ivanovic", Sex.MALE);
        Department department2 = DepartmentCreator.createDepartment("Department of
Business Informatics", head2);
        FacultyCreator.addDepartmentToFaculty(faculty1, department2);


        Human head3 = new Human("Maryna", "Lukashenko", "Petrivna", Sex.FEMALE);
        Department department3 = DepartmentCreator.createDepartment("Department of
Management", head3);
        FacultyCreator.addDepartmentToFaculty(faculty2, department3);

        Human head4 = new Human("Pavlo", "Shevchenko", "Ivanovic", Sex.MALE);
        Department department4 = DepartmentCreator.createDepartment("Department of
Marketing", head4);
        FacultyCreator.addDepartmentToFaculty(faculty2, department4);


        Human groupHead1 = new Human("Sophia","Sholoiko",  "Andriyivna",
Sex.FEMALE);
        Group group1 = GroupCreator.createGroup("122-21-2", groupHead1);
        DepartmentCreator.addGroupToDepartment(department1, group1);

        Human groupHead2 = new Human("Savchenko","Ihor",  "Anatoliyovych",
Sex.MALE);
        Group group2 = GroupCreator.createGroup("122-21-3", groupHead2);
        DepartmentCreator.addGroupToDepartment(department1, group2);


        Human groupHead3 = new Human("Hanna", "Koval", "Vitaliivna", Sex.FEMALE);
        Group group3 = GroupCreator.createGroup("073-21-1", groupHead3);
        DepartmentCreator.addGroupToDepartment(department2, group3);

        Human groupHead4 = new Human("Daryna", "Levina", "Andriyivna", Sex.FEMALE);
        Group group4 = GroupCreator.createGroup("073-21-2", groupHead4);
        DepartmentCreator.addGroupToDepartment(department2, group4);


        Student student1 = StudentCreator.createStudent("Taran","Iryna",
"Vitaliivna", Sex.FEMALE);
        GroupCreator.addStudentToGroup(group1, student1);

        Student student2 = StudentCreator.createStudent("Alekseenko", "Kateryna",
"Andriyivna", Sex.FEMALE);
```

```
        GroupCreator.addStudentToGroup(group1, student2);


        Student student3 = StudentCreator.createStudent("Kosach", "Hlib",
"Anatoliyovych", Sex.MALE);
        GroupCreator.addStudentToGroup(group2, student3);

        Student student4 = StudentCreator.createStudent("Pototskiy", "Ivan",
"Oleksandrovic", Sex.MALE);
        GroupCreator.addStudentToGroup(group2, student4);

        // Serializing into JSON format
        System.out.println("Original university:");
        System.out.println(oldUniversity);
        JsonManager.saveToJson(oldUniversity);
        System.out.println("\nJSON-file was saved.");

        // Deserializing into JSON format
        University newUniversity = JsonManager.loadFromJson();
        System.out.println("\nUpdated uni:");
        System.out.println(newUniversity);


        assertNotNull(newUniversity, "Error: newUniversity is null");
        assertEquals(oldUniversity, newUniversity, "Universities do not match");
        assertEquals(oldUniversity.toString(), newUniversity.toString(), "String
representations of universities do not match");


        assertEquals(2, newUniversity.getFaculties().size(), "The number of
faculties does not match.");
        assertEquals(2,
newUniversity.getFaculties().get(0).getDepartments().size(), "The number of
departments at the first faculty does not match.");
        assertEquals(2,
newUniversity.getFaculties().get(0).getDepartments().get(0).getGroups().size(),
"The number of groups at the first department does not match.");
        assertEquals(2,
newUniversity.getFaculties().get(0).getDepartments().get(0).getGroups().get(0).getS
tudents().size(), "The number of students in the first group does not match");

        System.out.println("\nSuccess! Universities are the same.");
    }


}
```


lab4
Version control
Current File
Project
controller
eater
model
Department
UniTest.java
Department.java
Human.java
Sex.java
Student.java
University.java
Main.java
Run.java
public class UniTest {
Human head3 = new Human(firstName: "Maryna", lastName: "Lukashenko", patronymic: "Petrivna", Sex.FEMALE);
Department department3 = DepartmentCreator.createDepartment(name: "Department of Management", head3);
Run
UniTest
Tests passed: 1 of 1 test – 107 ms
"C:\Program Files\Java\jdk-1.8\bin\java.exe" ...
Original university:
University{name='Dnipro University of technology', rector=Azyukovsky Oleksandr Oleksandrovic (MALE), faculties=
Faculty: Faculty of Information Technology, Dean: Udovik Iryna Mykhailivna (FEMALE), Departments: [Department: Department of Computer Systems Software, Head: Martynenko Andriy Anatoliyovych (MALE), Groups: [Group: 122-2
Faculty: Faculty of Economy, Dean: Ivanov Ivan Ivanovic (MALE), Departments: [Department: Department of Management, Head: Lukashenko Maryna Petrivna (FEMALE), Groups: [], Department: Department of Marketing, Head: Shevc
}
University is saved to JSON file.
JSON-file was saved.
Updated uni:
University{name='Dnipro University of technology', rector=Azyukovsky Oleksandr Oleksandrovic (MALE), faculties=
Faculty: Faculty of Information Technology, Dean: Udovik Iryna Mykhailivna (FEMALE), Departments: [Department: Department of Computer Systems Software, Head: Martynenko Andriy Anatoliyovych (MALE), Groups: [Group: 122-2
Faculty: Faculty of Economy, Dean: Ivanov Ivan Ivanovic (MALE), Departments: [Department: Department of Management, Head: Lukashenko Maryna Petrivna (FEMALE), Groups: [], Department: Department of Marketing, Head: Shevc
}
Success! Universities are the same.
Process finished with exit code 0
lab4 › src › test › java › example › eater › UniTest
39:70
CRLF
UTF-8
4 spaces