

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»

Звіт

з лабораторної роботи No 2
з дисципліни

«Мова програмування Java»

Виконала

студентка групи ЗПІ_зп-41
Павлюк Р.В.

Перевірив

Орленко С.П.

Київ 2025

Я обрала Завдання 2 з лабораторної роботи 2.

Для виконання завдання було створено три класи:

- Address: Допоміжний клас для зберігання адреси (вулиця, будинок).

Містить власну реалізацію equals та hashCode.

- Student: Основний клас сутності. Реалізує логіку порівняння всіх полів, включаючи вкладений об'єкт Address.

- Main: Клас із логікою тестування сценарію.

Додатково для виконання цього завдання необхідний файл gson-2.10.1.jar — це зовнішня бібліотека від Google. Він необхідний для серіалізації - автоматичного перетворення об'єкта Java у текстовий рядок формату JSON та десеріалізації - створення нового об'єкта Java на основі даних із текстового JSON-рядка. Без цього файлу компілятор не може розпізнати клас Gson, оскільки він не є частиною стандартної бібліотеки Java (JDK).

Нижче представлені скріншоти імплементаційного коду

```
task_2 > Address.java
1
2
3 public class Address {
4     private String street;
5     private String house;
6
7     public Address(String street, String house) {
8         this.street = street;
9         this.house = house;
10    }
11
12    @Override
13    public boolean equals(Object o) {
14        if (this == o) return true;
15        if (o == null || getClass() != o.getClass()) return false;
16        Address address = (Address) o;
17        return Objects.equals(street, address.street) &&
18            Objects.equals(house, address.house);
19    }
20
21    @Override
22    public int hashCode() {
23        return Objects.hash(street, house);
24    }
25 }
```

task_2 >  Student.java

```
1  import java.util.Objects;
2
3  public class Student {
4      private String firstName;
5      private String lastName;
6      private Address address;
7
8      public Student(String firstName, String lastName, Address address) {
9          this.firstName = firstName;
10         this.lastName = lastName;
11         this.address = address;
12     }
13
14     @Override
15     public boolean equals(Object o) {
16         if (this == o) return true;
17         if (o == null || getClass() != o.getClass()) return false;
18         Student student = (Student) o;
19         return Objects.equals(firstName, student.firstName) &&
20             Objects.equals(lastName, student.lastName) &&
21             Objects.equals(address, student.address);
22     }
23
24     @Override
25     public int hashCode() {
26         return Objects.hash(firstName, lastName, address);
27     }
28 }
```

task_2 >  Main.java

```
1  import com.google.gson.Gson;
2
3  public class Main {
4      public static void main(String[] args) {
5          Gson gson = new Gson();
6
7          // а. Створюємо екземпляр Student
8          Address address = new Address("Main St", "10A");
9          Student originalStudent = new Student("Ivan", "Ivanov", address);
10
11         // б. Конвертуємо в JSON (Серіалізація)
12         String json = gson.toJson(originalStudent);
13         System.out.println("JSON representation: " + json);
14
15         // в. Конвертуємо назад в об'єкт (Десеріалізація)
16         Student deserializedStudent = gson.fromJson(json, Student.class);
17
18         // г. Перевіряємо equals-ом
19         boolean result = originalStudent.equals(deserializedStudent);
20         System.out.println("Are objects equal? " + result); // Має бути true
21     }
22 }
```

Результати компіляції та демонстрація роботи програми наведені
нижче

```
ruszlanapavliuk@POL02-0349 task_2 % javac -cp "gson-2.10.1.jar" Address.java Student.java Main.java
ruszlanapavliuk@POL02-0349 task_2 % java -cp ".:gson-2.10.1.jar" Main
JSON representation: {"firstName":"Ivan","lastName":"Ivanov","address":{"street":"Main St","house":"10A"}}
Are objects equal? true
```