

Національний технічний університет України  
«Київський політехнічний інститут ім. І. Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

**Лабораторна робота №4**  
**З дисципліни «Об'єктно-орієнтоване програмування»**

Виконав:  
студент групи ІО-25  
*Мамченко Д.О.*  
Номер у списку групи: 18  
Перевірив: *Алещенко О.В.*

Київ 2022р.

**Тема:** Класи в мові програмування Java.

**Мета:** Ознайомлення з класами. Використання існуючих та створення власних класів в мові Java.


**Завдання:**

Номер залікової книжки - 2518.

$$C_{11} = \text{mod}(2518, 11) = 10$$

C11	Варіант завдання
0	Визначити клас студент, який складається як мінімум з 5-и полів.
1	Визначити клас навчальний заклад, який складається як мінімум з 5-и полів.
2	Визначити клас автомобіль, який складається як мінімум з 5-и полів.
3	Визначити клас літак, який складається як мінімум з 5-и полів.
4	Визначити клас морський човен, який складається як мінімум з 5-и полів.
5	Визначити клас одяг, який складається як мінімум з 5-и полів.
6	Визначити клас косметика, який складається як мінімум з 5-и полів.
7	Визначити клас спортивний інвентар, який складається як мінімум з 5-и полів.
8	Визначити клас меблі, який складається як мінімум з 5-и полів.
9	Визначити клас студент, який складається як мінімум з 5-и полів.
10	Визначити клас студент, який складається як мінімум з 5-и полів.

**Роздруківка коду:**

 Main.java ×

```
package lab4;

import java.util.Arrays;

public class Main {

    public static void main(String[] args) {
```

```



        Student First = new Student("Михайло", "Гордун", 21,
"ІІІСА", true);
        Student Second = new Student("Дмитро", "Шарій", 19, "ТЕФ",
false);
        Student Third = new Student("Аліна", "Шарій", 18, "ЕФ",
false);
        Student Fourth = new Student("Назар", "Лоленко", 17,
"ФІОТ", true);
        Student Fifth = new Student("Олександра", "Зеленська", 18,
"ФМБ", true);

        Student[] ObjArray = new Student[] {First, Second, Third,
Fourth, Fifth};

        Arrays.sort(ObjArray, new Sorting.SortByAlphabet());
        System.out.println("\nСортування за алфавітом
(зростання):");
        for (Student Classes : ObjArray) {
            System.out.println(Classes.lastName + " " +
Classes.firstName);
        }

        Arrays.sort(ObjArray, new Sorting.SortByAge());
        System.out.println("\nСортування за віком (спадання):");
        for (int i = ObjArray.length-1; i >= 0; i--) {
            System.out.print(ObjArray[i].lastName + " ");
            System.out.print(ObjArray[i].firstName);
            System.out.println(", вік: " + ObjArray[i].age);
        }
    }
}

```

 Student.java 

```

package lab4;

public class Student {

    public String firstName;
    public String lastName;
    public int age;
    public String faculty;
    public boolean hasStipend;

    public Student(String fn, String ln, int age, String fac, boolean
wealth) {

```

```

        setValues(fn, ln, age, fac, wealth);
        getValues();
    }

    private void setValues(String fn, String ln, int age, String fac,
boolean wealth) {
        this.firstName = fn;
        this.lastName = ln;
        this.age = age;
        this.faculty = fac;
        this.hasStipend = wealth;
    }

    private void getValues() {
        System.out.println("Прізвище та ім'я: " + this.lastName + " " +
this.firstName);
        System.out.println("Вік: " + this.age);
        System.out.println("Факультет: " + this.faculty);
        System.out.print("Наявність стипендії: ");
        if (this.hasStipend)
            System.out.println("є");
        else
            System.out.println("немає");
        System.out.println("~".repeat(40));
    }
}

```

### Sorting.java

```

package lab4;

import java.util.Comparator;

public class Sorting {

    public static class SortByAlphabet implements Comparator<Student> {
        @Override
        public int compare(Student st1, Student st2) {
            if (st1.lastName.compareTo(st2.lastName) == 0) {
                return st1.firstName.compareTo(st2.firstName);
            } else {
                return st1.lastName.compareTo(st2.lastName);
            }
        }
    }
}

```

```
public static class SortByAge implements Comparator<Student> {  
    @Override  
    public int compare(Student st1, Student st2) {  
        String age1 = Integer.toString(st1.age);  
        String age2 = Integer.toString(st2.age);  
        return age1.compareTo(age2);  
    }  
}  
  
}
```

### **Висновок:**

На цій лабораторній роботі я навчився працювати з класами. Під час виконання роботи були деякі труднощі. Тяжко було зрозуміти реалізацію методів у інших класах. Ще одна проблема з якою я зіткнувся – це сортування масиву з об’єктів класу. Програма працює коректно, але є один нюанс. Якщо прізвище починається з букви “І”, то воно при сортуванні за алфавітом буде закинуте в кінець списку. Це зумовлено тим, що в таблиці кодування вся кирилиця йде раніше ніж буква “І”.