



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота
№7
Шаблон «Стратегія»

Виконали
Студенти групи ІА-23:
Ковальський К.В.

Перевірив:
Колеснік В.М.

Київ 2022

Хід роботи:

1. Повторити теоретичні відомості
2. Провести рефакторинг свого коду з лабораторної роботи №6
 - використати шаблон «Стратегія»
 - дотримуватись принципів SOLID
3. Додати реалізацію ще одного алгоритму сортування на свій вибір
 - крім `Array.sort()`
4. Відповісти на контрольні запитання

```
5 usages 3 implementations
interface Sorting {
    4 usages 3 implementations
    long[] sortSomething(long[] array);
    1 usage 3 implementations
    void printResult(long[] array);
}
```

```
import java.util.Arrays;
1 usage
public class Selection_sort implements Sorting{
    4 usages
    @Override
    public long[] sortSomething(long[] array) {
        for (int maxIndex = 0; maxIndex < array.length; maxIndex++) {
            long maxValue = array[maxIndex];

            int maxIndexLocal = maxIndex;
            long maxValueLocal = maxValue;
            for (int i = maxIndexLocal + 1; i < array.length; i++) {
                if (array[i] > array[maxIndexLocal]) {
                    maxValueLocal = array[i];
                    maxIndexLocal = i;
                }
            }
            array[maxIndex] = maxValueLocal;
            array[maxIndexLocal] = maxValue;
        }
        return array;
    }
    1 usage
    @Override
    public void printResult(long[] array) {
        String result = Arrays.toString(sortSomething(array));
        System.out.println("Відсортований масив: " + result);
    }
}
```

```

import java.util.Arrays;
1 usage
public class Insertion_sort implements Sorting{
    4 usages
    @Override
    public long[] sortSomething(long[] array) {
        for (int i = 1; i < array.length; i++) {
            long temVar = array[i];
            int location = i - 1;
            while (location >= 0 && array[location] < temVar) {
                array[location + 1] = array[location];
                location -= 1;
            }
            array[location + 1] = temVar;
        }
        return array;
    }
    1 usage
    @Override
    public void printResult(long[] array) {
        String result = Arrays.toString(sortSomething(array));
        System.out.println("Відсортований масив: " + result);
    }
}

```

```

import java.util.Arrays;
1 usage
public class Bubble_sort implements Sorting{
    4 usages
    @Override
    public long[] sortSomething(long[] array) {
        for (int i = 0; i < array.length - 1; i++) {
            for (int j = 0; j < array.length - 1; j++) {
                if (array[j] < array[j + 1]) {
                    long temVar = array[j + 1];
                    array[j + 1] = array[j];
                    array[j] = temVar;
                }
            }
        }
        return array;
    }
    1 usage
    @Override
    public void printResult(long[] array) {
        String result = Arrays.toString(sortSomething(array));
        System.out.println("Відсортований масив: " + result);
    }
}

```

```

2 usages
public class Sort {
    3 usages
    Sorting typeOfSorting;
    2 usages
    long[] array;
    3 usages
    public void setTypeOfSorting(Sorting typeOfSorting) {
        this.typeOfSorting = typeOfSorting;
    }
    3 usages
    public void setArray(long[] array) {
        this.array = array;
    }
    3 usages
    public void executeSorting() {
        long[] result = typeOfSorting.sortSomething(array);
        typeOfSorting.printResult(result);
    }
}

```

```

public class Test_sort {
    public static void main(String[] args) {
        Sort firstMachine = new Sort();

        firstMachine.setTypeOfSorting(new Selection_sort());
        firstMachine.setArray(new long[]{642795, 458702, 564709, 475802, 542987, 514532, 785489});
        firstMachine.executeSorting();

        firstMachine.setTypeOfSorting(new Insertion_sort());
        firstMachine.setArray(new long[]{642795, 458702, 564709, 475802, 542987, 514532, 785489});
        firstMachine.executeSorting();

        firstMachine.setTypeOfSorting(new Bubble_sort());
        firstMachine.setArray(new long[]{642795, 458702, 564709, 475802, 542987, 514532, 785489});
        firstMachine.executeSorting();
    }
}

```

Висновок: Виконавши цю лабораторну роботу, навчилися робити рефакторинг коду, додавши додатковий метод сортування – Bubble sort. Виконали завдання дотримавшись принципів SOLID та використали шаблон проектування «Стратегія».