



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

з дисципліни «Програмування-1. Основи програмування»

Тема «Шаблон «Стратегія»»

Виконали
студенти групи ІА-23:
Тюска А.Ю.
Хохол М. В.
Пожар Д.Ю.

Перевірив:
Колеснік В. М.

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()`

Головний клас Main

```
public class Main {
    public static void main(String[] args) {
        byte[] firstArray = {4, -1, 11, 0, 7};
        byte[] secondArray = {5, 4, 3, 6, 1};
        byte[] thirdArray = {10, -3, 1, 5, -4};

        Sorter bubbleRow = new Sorter(new BubbleSort(), new RowPrint());
        Sorter selectionColumn = new Sorter(new SelectionSort(), new
ColumnPrint());
        Sorter insertionRow = new Sorter(new InsertionSort(), new RowPrint());

        bubbleRow.execute(firstArray);
        selectionColumn.execute(secondArray);
        insertionRow.execute(thirdArray);
    }
}
```

Клас Sorter – відповідає за створення й реалізацію необхідного сортування та вивід його у консоль

```
public class Sorter {
    ISort sorter;
    IPrint printer;

    public Sorter(ISort sorter, IPrint printer) {
        this.sorter = sorter;
        this.printer = printer;
    }

    public void execute(byte[] arr) {
        sorter.sort(arr);
        printer.print(arr);
    }
}
```

Інтерфейс ISort – вказує, що у класах, що наслідують даний інтерфейс повинно бути реалізовано метод sort

```
public interface ISort {
    void sort(byte[] arr);
}
```

Клас BubbleSort – сортує масив “бульбашкою”

```
public class BubbleSort implements ISort {
    @Override
    public void sort(byte[] arr) {
        for (int i = 0; i < arr.length; i++) {
            for (int k = 0; k < arr.length - 1; k++) {
                if (arr[k] > arr[k + 1]) {
                    byte temp = arr[k + 1];
                    arr[k + 1] = arr[k];
                    arr[k] = temp;
                }
            }
        }
    }
}
```

Клас InsertionSort – сортує масив “включенням”

```
public class InsertionSort implements ISort {
    @Override
    public void sort(byte[] arr) {
        for (int i = 0; i < arr.length; i++) {

            byte temp = arr[i];
            int j = i - 1;

            for ( ; (j > -1) && (arr[j] > temp); j--) {
                arr[j + 1] = arr[j];
            }

            arr[j + 1] = temp;
        }
    }
}
```

Клас SelectionSort – сортує масив “вибором”

```
public class SelectionSort implements ISort {
    @Override
    public void sort(byte[] arr) {
        for (int i = 0; i < arr.length - 1; i++) {
            int index = i;

            for (int j = i + 1; j < arr.length; j++) {
                if (arr[index] > arr[j]) {
                    index = j;
                }
            }

            byte smallestValue = arr[index];
            arr[index] = arr[i];
            arr[i] = smallestValue;
        }
    }
}
```

Інтерфейс IPrint - вказує, що у класах, що наслідують даний інтерфейс повинно бути реалізовано метод print

```
public interface IPrint {  
    void print(byte[] arr);  
}
```

Клас RowPrint – виводить у рядок елементи масиву

```
public class RowPrint implements IPrint {  
    public void print(byte[] arr) {  
        for (int value : arr) {  
            System.out.print(value + " ");  
        }  
  
        System.out.println();  
    }  
}
```

Клас ColumnPrint – виводить у стовпчик елементи масиву

```
public class ColumnPrint implements IPrint {  
    public void print(byte[] arr) {  
        for (int value : arr) {  
            System.out.println(value);  
        }  
    }  
}
```

Результат:

```
-1 0 4 7 11  
1  
3  
4  
5  
6  
-4 -3 1 5 10
```

Висновок

Виконавши лабораторну роботу, ми зробили рефакторинг коду, додавши ще один метод сортування – Insertion sort, та реалізували завдання, дотримуючись принципів SOLID та використовуючи шаблон проектування “Стратегія”.