



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

**Лабораторна робота №7**  
із дисципліни «Основи програмування»  
**Тема: «Шаблон «Стратегія»**

Виконали:  
Студенти групи ІА-24  
Гуменюк К.Е.  
Тильна.М.С .  
Любченко.І.М

Перевірив:  
Колеснік Валерій Миколайович

**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

**Код**

```
import java.util.Arrays;
public class Bubble implements Sorting {
    @Override
    public void sort(short[] array) {
        System.out.println("Array before bubbleSorting:" +
Arrays.toString(array));
        for (int i = 0; i < array.length - 1; i++) {
            for (int j = 0; j < array.length - i - 1; j++) {
                if (array[j + 1] > array[j]) {
                    short swap = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = swap;
                }
            }
        }
        if (array.length == 0) {
            throw new IllegalArgumentException("Нічого сортувати");
        }
        System.out.println("Array after bubbleSorting:" +
Arrays.toString(array));
    }
}
```

```
import java.util.Arrays;
public class Insertion implements Sorting {

    @Override
    public void sort(short[] array) {
        System.out.println("Array before insertionSorting:" +
Arrays.toString(array));
        if (array.length == 0) {
            throw new IllegalArgumentException("Нічого сортувати");
        }
        int j;
        for (int i = 1; i < array.length; i++) {
            short swap = array[i];
            for (j = i; (j > 0) && (array[j - 1] < swap); j--) {
                array[j] = array[j - 1];
            }
            array[j] = swap;
        }
    }
}
```

```

        System.out.println("Array after insertionSorting:" +
Arrays.toString(array));
    }
}

```

```

import java.util.Arrays;
public class Selection implements Sorting {

    @Override
    public void sort(short[] array) {
        System.out.println("Array before selectionSorting:" +
Arrays.toString(array));
        for (int i = 0; i < array.length; i++) {
            int pos = i;
            short min = array[i];
            for (int j = i + 1; j < array.length; j++) {
                if (array[j] < min) {
                    pos = j;
                    min = array[j];
                }
            }
            array[pos] = array[i];
            array[i] = min;
        }
        if (array.length == 0) {
            throw new IllegalArgumentException("Нічого сортувати");
        }
        System.out.println("Array after selectionSorting:" +
Arrays.toString(array));
    }
}

```

```

interface Sorting {
    default void sort(short[] array) {

    }
}

```

```

public class SortingArray {
    public static void main(String[] args) {
        Strategy sortingNow = new Strategy();
        short[] a = {-1, -2, -3, 12, -5, 1};
        sortingNow.setStrategy(new Selection());
        sortingNow.executeStrategy(a);
        short[] b = {11, 12, 17, -12, 0};
        sortingNow.setStrategy(new Bubble());
        sortingNow.executeStrategy(b);
        short[] c = {-23, -3, -23, -54, -11, 21};
        sortingNow.setStrategy(new Insertion());
        sortingNow.executeStrategy(c);
    }
}

```

```
}  
}
```

```
public class Strategy {  
    Sorting strategy;  
  
    public void setStrategy(Sorting strategy) {  
        this.strategy = strategy;  
    }  
  
    public void executeStrategy(short[] array) {  
        strategy.sort(array);  
    }  
}
```

### Результат

```
"C:\Program Files\Java\jdk-17.0.5\bin\java.exe" "-javaagent:C:\P  
Array before selectionSorting:[-1, -2, -3, 12, -5, 1]  
Array after selectionSorting:[12, 1, -1, -2, -3, -5]  
Array before bubbleSorting:[11, 12, 17, -12, 0]  
Array after bubbleSorting:[17, 12, 11, 0, -12]  
Array before insertionSorting:[-23, -3, -23, -54, -11, 21]  
Array after insertionSorting:[21, -3, -11, -23, -23, -54]  
  
Process finished with exit code 0
```

**Висновок:** на цій лабораторній роботі ми ознайомились з шаблоном «Стратегія» та принципами SOLID. Використовуючи їх провели рефакторинг свого коду. Також додали реалізацію ще одного алгоритму сортування.