



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:

Волошин Вадім

Воронюк Євгеній

Савонік Назар

Перевірів:

Колеснік В. М.

Київ 2022

Хід роботи:

Провести рефакторинг свого коду з лабораторної роботи №6:

- використати шаблон «Стратегія».
- дотримуватись принципів SOLID.

Додати реалізацію ще одного алгоритму сортування на свій вибір

- крім `Array.sort()`.

interface Strategy:

```
1 5 usages 3 implementations  
1 1 ↓ public interface Strategy {  
2 1 usage 3 implementations  
2 1 ↓ void sort(short[] array);  
3 }
```

class Array:

```
1 6 usages  
1 public class Array {  
2 2 usages  
2 private Strategy strategy;  
3  
4 3 usages  
4 public Array(Strategy strategy) { this.strategy = strategy; }  
7  
8 3 usages  
8 public void sort(short[] array) {  
9 strategy.sort(array);  
10 }  
11  
12 3 usages  
12 @ public void toString(short[] array) {  
13 for (short i : array) {  
14 System.out.print(i + " ");  
15 }  
16 }  
17 }
```

class MainSorting:

```
1  ► public class MainSorting {
2  ►  ► public static void main(String[] args) {
3      Array insertion = new Array(new InsertionSorting());
4      short[] insertionArray = {12, 45, 23, 22, 43};
5      System.out.println("Insertion sort method: ");
6      insertion.sort(insertionArray);
7      insertion.toString(insertionArray);
8      System.out.println();
9
10     Array selection = new Array(new SelectionSorting());
11     short[] selectionArray = {12, 45, 23, 22, 43};
12     System.out.println("Selection sort method: ");
13     selection.sort(selectionArray);
14     selection.toString(selectionArray);
15     System.out.println();
16
17     Array bubble = new Array(new BubbleSorting());
18     short[] bubbleArray = {12, 45, 23, 22, 43};
19     System.out.println("Bubble sort method: ");
20     bubble.sort(bubbleArray);
21     bubble.toString(bubbleArray);
22     System.out.println();
23
24     }
25 }
```

class InsertionSorting:

```
1  1 usage
   public class InsertionSorting implements Strategy {
2      1 usage
3      1 usage
4      @Override
5      public void sort(short[] array) {
6          for (int i = 1; i < array.length; i++) {
7              int current = array[i];
8              int j = i - 1;
9              while (j >= 0 && current > array[j]) {
10                 array[j + 1] = array[j];
11                 j--;
12             }
13             array[j + 1] = (short) current;
14         }
15     }
16 }
```

class SelectionSorting:

```
1 usage
1 public class SelectionSorting implements Strategy{
2     1 usage
3     @Override
4     public void sort(short[] array){
5         for (int i = 0; i < array.length; i++) {
6             int min = i;
7             for (int j = i + 1; j < array.length; j++) {
8                 if (array[i] < array[j]) {
9                     min = j;
10                }
11            }
12            short temp = array[i];
13            array[i] = array[min];
14            array[min] = temp;
15        }
16    }
```

class BubbleSorting:

```
1 usage
1 public class BubbleSorting implements Strategy {
2     1 usage
3     @Override
4     public void sort(short[] array) {
5         for (int i = 0; i < array.length; i++) {
6             for (int j = 0; j < array.length - 1; j++) {
7                 if (array[j] < array[j + 1]) {
8                     short temp = array[j + 1];
9                     array[j + 1] = array[j];
10                    array[j] = temp;
11                }
12            }
13        }
14    }
```

Висновок: ми провели рефакторинг алгоритмів сортування InsertionSort, SelectionSort з лабораторної роботи 6, додали реалізацію алгоритму BubbleSort. У даній лабораторній роботі ми використали шаблон «Стратегія», ознайомилися із принципами SOLID. Також ознайомилися із патернами проектування, із термінами «component», «dependency», «coupling» «cohesion»; із залежностями, явним і неявним використанням принципів залежностей.