



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
Призвіще:
Шкарніков Антон,
Кармазіна Анастасія,
Сотніков Олексій.

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

Результат виконання роботи:**Код файлу \ISortStrategy.java**

```
public interface ISortStrategy {  
    void Sort(float[] array);  
}
```

Код файлу \BubbleSort.java

```
public class BubbleSort implements ISortStrategy {  
    public void Sort(float[] 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]){  
                    float swap = array[j];  
                    array[j] = array[j+1];  
                    array[j+1] = swap;  
                }  
            }  
        }  
        System.out.println("Відбулося сортування бульбашкою");  
    }  
}
```

Код файлу \SelectionSort.java

```
public class SelectionSort implements ISortStrategy {
    public void Sort(float[] array){
        for (int i = 0; i < array.length-1; i++) {
            int min_idx = i;
            for (int j = i+1; j < array.length; j++)
                if (array[j] > array[min_idx])
                    min_idx = j;

            float temp = array[min_idx];
            array[min_idx] = array[i];
            array[i] = temp;
        }
        System.out.println("Відбулося сортування вибором");
    }
}
```

Код файлу \InsertionSort.java

```
public class InsertionSort implements ISortStrategy {
    public void Sort(float[] array) {
        for (int i = 0; i < array.length; i++){
            float temp = array[i];
            int j = i - 1;
            while (j >= 0 && array[j] < temp) {
                array[j + 1] = array[j];
                j--;
            }
            array[j + 1] = temp;
        }
        System.out.println("Відбулося сортування Включенням");
    }
}
```

Код файлу \Context.java

```
import java.util.Arrays;

public class Context {
    private ISortStrategy sortStrategy;

    Context() {

    }

    // Стратегію сортування передаємо для контексту через конструктор
    public Context(ISortStrategy sortStrategy) {
        this.sortStrategy = sortStrategy;
    }

    // Залишаємо можливість змінювати стратегію під час виконання
    програми
    public void SetStrategy(ISortStrategy strategy) {
        this.sortStrategy = strategy;
    }

    // Сортуюємо використовуючи вибрану стратегію
    public void UseSortStrategy() {
        float[] array = { 1.5f, 6.0f, 3.3f, 15.8f, 29.2f, 4.1f, 8.7f };
        this.sortStrategy.Sort(array);
        System.out.println(Arrays.toString(array));
    }
}
```

Код файлу \Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Context context = new Context();  
  
        System.out.println("Застосуємо стратегію сортування Вибором");  
        context.SetStrategy(new SelectionSort());  
        context.UseSortStrategy();  
  
        System.out.println();  
  
        System.out.println("Застосуємо стратегію сортування Бульбашкою");  
        context.SetStrategy(new BubbleSort());  
        context.UseSortStrategy();  
  
        System.out.println();  
  
        System.out.println("Застосуємо стратегію сортування Включенням");  
        context.SetStrategy(new InsertionSort());  
        context.UseSortStrategy();  
    }  
}
```

Результат виконання програми:

```
Застосуємо стратегію сортування Вибором  
Відбулося сортування вибором  
[29.2, 15.8, 8.7, 6.0, 4.1, 3.3, 1.5]  
  
Застосуємо стратегію сортування Бульбашкою  
Відбулося сортування бульбашкою  
[29.2, 15.8, 8.7, 6.0, 4.1, 3.3, 1.5]  
  
Застосуємо стратегію сортування Включенням  
Відбулося сортування Включенням  
[29.2, 15.8, 8.7, 6.0, 4.1, 3.3, 1.5]
```

Висновок: у даній лабораторній роботі ми провели рефакторинг свого коду з попередньої лабораторної роботи, додали ще один алгоритм сортування. Ознайомилися з шаблоном “Стратегія” та принципами SOLID.

Рефакторинг — це контрольований процес покращення коду, без написання нової функціональності.

Патерн проектування — це типовий спосіб вирішення певної проблеми, що часто зустрічається при проектуванні архітектури програм.

SOLID — це аббревіатура складена з перших літер п'яти базових принципів об'єктно-орієнтованого програмування та дизайну запропонована Робертом Мартіном.

Принципи SOLID використовують для дизайну та розробки таких програмних систем, які, з великою ймовірністю, зможуть тривалий час розвиватися, розширятися і підтримуватися.