

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 «Основи Програмування»

Тема: Шаблон «Стратегія»

Виконали:
Студенти групи ІА-23
Мозоль Владислав,
Курач Владислав,
Лядський Дмитро
Дата здачі: 01.12.2022

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

```
// Клас контексту використовує цей інтерфейс для виклику конкретної стратегії
interface Strategy {
    double[] sort(double[] arr);
}

// Реалізуємо алгоритм з використанням інтерфейсу стратегії
class selectionSort implements Strategy {

    public double[] sort(double[] arr) {
        double[] outArray = Arrays.copyOf(arr, arr.length);
        for (int i = 0; i < outArray.length-1; i++) {
            int min = i;
            for (int j = i + 1; j < outArray.length; j++)
                if (outArray[j] < outArray[min])
                    min = j;
            double result = outArray[min];
            outArray[min] = outArray[i];
            outArray[i] = result;
        }
        return outArray;
    }
}
```

```
class insertionSort implements Strategy {
    public double[] sort(double[] arr) {
        double[] outArray = Arrays.copyOf(arr, arr.length);
        for (int i = 1; i < outArray.length; i++) {
            int j = i;
            while ((j > 0) && (outArray[j - 1] > outArray[j])) {
                double buffer = outArray[j];
                outArray[j] = outArray[j - 1];
                outArray[j - 1] = buffer;
                j--;
            }
        }
        return outArray;
    }
}
```

```
class bubbleSort implements Strategy {

    public double[] sort(double[] arr) {
        double[] outArray = Arrays.copyOf(arr, arr.length);
        double temp = 0;
        for(int i=0; i < arr.length; i++){
            for(int j=1; j < (arr.length-i); j++){
                if(outArray[j-1] > outArray[j]){
                    temp = outArray[j-1];
                    outArray[j-1] = outArray[j];
                    outArray[j] = temp;
                }
            }
        }
    }
}
```

```

class Context {

    private Strategy strategy;

    // Constructor
    public Context(Strategy strategy) {

        this.strategy = strategy;
    }

    public double[] sorter(double[] arr) {

        return strategy.sort(arr);
    }
}

// Тестовий додаток
class StrategyExample {

    public static void main(String[] args) {

        Context selection;
        Context bubble;
        Context insertion;
        double[] arr = {64,8.05,0.05,0.009,25,12,22,11};
        double[] example = {7,8,5,4,2,3};
        selection = new Context(new selectionSort());
        System.out.println(Arrays.toString(selection.sorter(arr)));
        insertion = new Context(new insertionSort());
        System.out.println(Arrays.toString(insertion.sorter(example)));
        bubble = new Context(new bubbleSort());
        System.out.println(Arrays.toString(bubble.sorter(arr)));
    }
}

```

3. Додати реалізацію ще одного алгоритму сортування на свій вибір
- крім `Array.sort()`

4. Відповісти на контрольні запитання

Висновок: Отже, на цій лабораторній роботі ми ознайомились і навчились працювати зі шаблоном «Стратегія». Також дізнались що таке SOLID та його принципи. Як правильно програмувати згідно SOLID і як не порушувати ці принципи. Також дізнались що таке «Зв'язність» і «Пов'язаність».