

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу

Кафедра КСМ

Лабораторна робота №5
Тема “Реалізація динамічного поліморфізму”

Виконав студент
групи КІ-18-1
Чуйко Т.В.

Перевірив
Гарасимів Т.Г.

м.Івано-Франківськ
2019р.

Мета:

- вивчити принципи реалізації динамічного поліморфізму;
- закріпити на практиці знання протоколів взаємодії класів, їх властивостей і методів у ланцюзі успадкування;
- набути вміння використовувати поліморфні методи для управління поведінкою об'єктів, які належать до одного ієрархічного ланцюга успадкування.

Умова задачі:

- 1) вивчити теоретичні відомості та методичні вказівки до лабораторної роботи;
- 2) реалізувати заданий поліморфний метод та тестуючу програму;
- 3) відкомпілювати та відладити програму;
- 4) відповісти на контрольні запитання;
- 5) зробити висновки.

Загальне завдання 1: реалізувати та протестувати поліморфний метод в ієрархії класів, створений у попередній лабораторній роботі згідно з варіантом.

6	Визначення елемента за його порядковим номером
---	--

Програма мовою Java:

Клас Main:

```
public class Main {  
    public static void main(String[] args) {  
        // 1  
        ArithmeticSequence arithmeticSequence = new ArithmeticSequence(1, 4,  
10);  
        arithmeticSequence.setSize(15);  
  
        System.out.println(arithmeticSequence.getValue(1));  
        System.out.println(arithmeticSequence.getValue(4));  
        System.out.println(arithmeticSequence.getValue(9));  
        System.out.println(arithmeticSequence.getValue(15));  
  
        System.out.println(arithmeticSequence.calculateSum());  
  
        // 2  
        FibonacciSequence fibonacciSequence = new FibonacciSequence(6);  
  
        System.out.println(fibonacciSequence.getValue(1));  
        System.out.println(fibonacciSequence.getValue(3));  
        System.out.println(fibonacciSequence.getValue(6));  
  
        System.out.println(fibonacciSequence.getRatio(3));  
    }  
}
```

Клас Sequence:

```
public class Sequence {  
    protected int size;  
    protected int[] values;  
  
    public void setSize(int size) {  
        this.size = size;  
        this.values = new int[size];  
    }  
  
    public int getValue(int index) {  
        return 0;  
    }  
}
```

Клас ArithmeticSequence:

```
public class ArithmeticSequence extends Sequence {  
    private int start;  
    private int step;  
  
    public ArithmeticSequence(int start, int step, int size) {  
        super();  
        this.start = start;  
        this.step = step;  
        this.size = size;  
        this.values = new int[size];  
    }  
  
    public void setStart(int start) {  
        this.start = start;  
    }  
}
```

```

    public void setStep(int step) {
        this.step = step;
    }

    @Override
    public int getValue(int index) {
        calculateSequence();
        return values[index - 1];
    }

    public int calculateSum() {
        calculateSequence();
        int sum = 0;
        for (int value: values) {
            sum += value;
        }
        return sum;
    }

    private void calculateSequence() {
        for (int i = 0, n = 1; i < size; i++, n++) {
            values[i] = start + (n - 1) * step;
        }
    }
}

```

Клас FibonacciSequence:

```

public class FibonacciSequence extends Sequence {
    public FibonacciSequence(int size) {
        super();
        this.size = size;
        this.values = new int[size];
    }

    @Override
    public int getValue(int index) {
        calculateSequence();
        return values[index - 1];
    }

    public String getRatio(int index) {
        calculateSequence();
        String ratio = "";
        ratio += values[index - 1] + " / " + values[index - 2];
        return ratio;
    }

    private void calculateSequence() {
        values[0] = 1;
        values[1] = 1;
        for (int i = 2; i < size; i++) {
            values[i] = values[i - 1] + values[i - 2];
        }
    }
}

```

Результат:

1
13
33
57
435
1
2
8
2 / 1

Висновок: На цій лабораторній роботі я:

- вивчив принципи реалізації динамічного поліморфізму;
- закріпив на практиці знання протоколів взаємодії класів, їх властивостей і методів у ланцюзі успадкування;
- набув вміння використовувати поліморфні методи для управління поведінкою об'єктів, які належать до одного ієрархічного ланцюга успадкування.