

ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ОЛЕСЯ ГОНЧАРА

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ ТА МАТЕМАТИЧНОЇ
КІБЕРНЕТИКИ

ЗВІТ ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 2

З дисципліни «Сучасні середовища програмування»

Перший (бакалаврський) рівень вищої освіти
Спеціальність 113 Прикладна математика
Освітня програма Комп'ютерне моделювання
та технології програмування

Виконавець:
Студент(ка) групи ПА–21–1
Стегайлов Володимир
Варіант №5

Постановка завдання

Застосувати патерн Компонувальник (Composite) для обчислення похідної довільної складеної функції за розрахунковими формулами $f_1(x)$ та $f_2(x)$. Використати наступні дані:

5	$f_1(x) = e^{a \cos^2 x} - b \ln^5 \sqrt{x} $ $f_2(x) = ax^3 / \sin^2(x+b)$	$a = -2, b = 3, x = 2.4$	Дата у форматі дд мм рррр
---	---	--------------------------	---------------------------

Опис логічної структури Java-програми

Патерни та класи

- Компонувальник (Composite): Інтерфейс Function з методом calculate(double x) для обчислення значення функції.
- Листок (Leaf): Класи ExpFunction, CosFunction, SinFunction, LnFunction для представлення простих функцій.
- Складений компонентувальник (Composite): Клас CompositeFunction для об'єднання простих функцій в складену функцію.

Конструктори

- Function: Базовий конструктор для всіх класів, який приймає вхідні дані.
- CompositeFunction: Конструктор для створення складеної функції з простих функцій.

Висновки за результатами роботи

У даній лабораторній роботі ми успішно застосували патерн Компонувальник для обчислення похідної складеної функції. Код був ефективно структурований та організований, що полегшує його розширення та зміну.

Додатки (код програми)

```
import java.util.Date;

import java.text.SimpleDateFormat;

import java.lang.Math;

// Інтерфейс Компонувальника

interface Function {

    double calculate(double x);

}

// Листок

class ExpFunction implements Function {

    private final double a;
```

```
public ExpFunction(double a) {  
    this.a = a;  
}  
  
@Override  
public double calculate(double x) {  
    return Math.exp(a * Math.cos(x) * Math.cos(x));  
}  
}
```

```
class CosFunction implements Function {  
    private final double x;  
  
    public CosFunction(double x) {  
        this.x = x;  
    }  
  
    @Override  
    public double calculate(double x) {  
        return Math.cos(this.x);  
    }  
}
```

```
class SinFunction implements Function {  
    private final double x;  
    private final double b;  
  
    public SinFunction(double x, double b) {  
        this.x = x;  
        this.b = b;  
    }  
}
```

```
}
```

```
@Override
```

```
public double calculate(double x) {
```

```
    return Math.sin(this.x + this.b);
```

```
}
```

```
}
```

```
class LnFunction implements Function {
```

```
    private final double x;
```

```
    public LnFunction(double x) {
```

```
        this.x = x;
```

```
    }
```

```
@Override
```

```
public double calculate(double x) {
```

```
    return Math.log(Math.sqrt(this.x)) * 5;
```

```
}
```

```
}
```

```
// Складений Компонувальник
```

```
class CompositeFunction implements Function {
```

```
    private final Function f1;
```

```
    private final Function f2;
```

```
    private final double a;
```

```
    private final double b;
```

```
    public CompositeFunction(double a, double b) {
```

```
        this.a = a;
```

```
        this.b = b;
```

```
this.f1 = new ExpFunction(a);  
this.f2 = new CompositeFunction(a, b);  
}
```

```
public CompositeFunction(Function f1, Function f2) {  
    this.f1 = f1;  
    this.f2 = f2;  
    this.a = 0;  
    this.b = 0;  
}
```

```
@Override  
public double calculate(double x) {  
    return f1.calculate(x) - Math.abs(f2.calculate(x));  
}  
}
```

// Головний клас

```
public class Main {  
    public static void main(String[] args) {  
        double a = -2;  
        double b = 3;  
        double x = 2.4;  
  
        Function f1 = new CompositeFunction(  
            new ExpFunction(a),  
            new CompositeFunction(  
                new CosFunction(x),  
                new CompositeFunction(  
                    new LnFunction(x),  
                    new Function() {
```

```

        @Override
        public double calculate(double x) {
            return Math.sqrt(x);
        }
    }
)
)
);

```

```

Function f2 = new CompositeFunction(
    new CompositeFunction(
        new Function() {
            @Override
            public double calculate(double x) {
                return a * x * x * x;
            }
        },
        new CompositeFunction(
            new SinFunction(x, b),
            new Function() {
                @Override
                public double calculate(double x) {
                    return Math.sin(x);
                }
            }
        )
    ),
    new Function() {
        @Override
        public double calculate(double x) {
            return Math.sin(x + b);
        }
    }
);

```

```
    }  
    }  
);
```

```
double result = f1.calculate(x) / f2.calculate(x);
```

```
SimpleDateFormat dateFormat = new SimpleDateFormat("dd MM yyyy");
```

```
Date date = new Date();
```

```
System.out.println("Дата: " + dateFormat.format(date));
```

```
System.out.println("Результат: " + result);
```

```
    }  
}
```