



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 №8
із дисципліни «Основи програмування»
Тема: «Основи ООП»

Виконали:
Студенти групи ІА-24
Зелінський І.О.
Криворучек В.С.
Трасковський Т.І.

Перевірив:
Колеснік Валерій Миколайович

Хід роботи:

1. Повторити теоретичні відомості
2. Проаналізувати предметну область завдання свого варіанту (табл.1)
3. Розробити базовий клас (відповідно до завдання можливо абстрактний клас або інтерфейс), клас-нащадок, а також допоміжні класи та/або інтерфейси за необхідністю. Відповідно до предметної області завдання передбачити відповідні методи бізнес-логіки, а також конструктори, сетери та/або гетери, методи equals() та toString().

Продемонструвати використання:

- this;
- super;
- перевантаження (overloading) та заміщення(overriding) методів;
- перевантаження (overloading) конструкторів.

| 6 | Алгоритм вирішення квадратного рівняння | Алгоритм вирішення квадратного рівняння через дискримінант | mutable |
|---|---|--|---------|
| 1 2 3 4 5 6 7 8 9 10 11 12 13 | <pre>public class MainProgramm { public static void main(String[] args) { DiscriminantMethod first = new DiscriminantMethod(4,5,6); System.out.println(first.toString()); DiscriminantMethod sec = new DiscriminantMethod(-2, 0, 6); System.out.println(sec.toString()); DiscriminantMethod third = new DiscriminantMethod(1, -3, 2); System.out.println(third.toString()); DiscriminantMethod fourth = new DiscriminantMethod(5); System.out.println(fourth.toString()); } }</pre> | <pre><terminated> MainProgramm (1) [Java Application] C:\Program Files\Java\j 4.0x^2 +5.0x +6.0 = 0 Коренів немає -2.0x^2 +6.0 = 0 x1 = -1.7320508075688772, x2 = 1.7320508075688772 1.0x^2 -3.0x +2.0 = 0 x1 = 2.0, x2 = 1.0 5.0x^2 = 0 x = 0.0</pre> | |

```

1 public class QuadraticEquation {
2     private double a;
3     private double b;
4     private double c;
5     protected double x1;
6     protected double x2;
7
8     QuadraticEquation() {
9         this.a = 1;
10        this.b = 0;
11        this.c = 0;
12    }
13    QuadraticEquation(double a, double b, double c) {
14        this.a = a;
15        this.b = b;
16        this.c = c;
17    }
18    QuadraticEquation(double a, double b) {
19        this.a = a;
20        this.b = b;
21        this.c = 0;
22    }
23    QuadraticEquation(double a) {
24        this.a = a;
25        this.b = 0;
26        this.c = 0;
27    }
28    public double getA() {
29        return a;
30    }
31    public double getB() {
32        return b;
33    }
34    public double getC() {
35        return c;
36    }
37    public void setA(double a) {
38        this.a = a;
39    }
40    public void setB(double b) {
41        this.b = b;
42    }
43    public void setC(double c) {
44        this.c = c;
45    }
46    @Override
47    public String toString() {
48        String res = a + "x^2 ";
49        if(b>0) {
50            res += "+" + b + "x";
51        } else if((b<0)){
52            res += b + "x";
53        }
54        if(c>0) {
55            res += " +" + c;
56        } else if((c<0)){
57            res += " " + c;
58        }
59        return res + " = 0";
60    }
61 }
62

```

```

1 import java.lang.Math;
2 public class DiscriminantMethod extends QuadraticEquation{
3     private double D;
4     DiscriminantMethod(double a, double b, double c){
5         super(a, b, c);
6         this.D = Math.pow(b,2) - 4 * a * c;
7     }
8     DiscriminantMethod(double a, double b){
9         super(a, b);
10        this.D = Math.pow(b,2) - 4 * a * getC();
11    }
12    DiscriminantMethod(double a){
13        super(a);
14        this.D = Math.pow(getB(),2) - 4 * a * getC();
15    }
16    DiscriminantMethod(){
17        super();
18        this.D = Math.pow(getB(),2) - 4 * getA() * getC();
19    }
20
21    public String Solve() {
22        if (D == 0) {
23            x1 = (-getB() + Math.sqrt(D)) / (2*getA());
24            return "x = " + x1;
25        }
26        if (D < 0) {
27            return "Коренів немає";
28        }
29        x1 = (-getB() + Math.sqrt(D)) / (2*getA());
30        x2 = (-getB() - Math.sqrt(D)) / (2*getA());
31        return "x1 = " + x1 + ", x2 = " + x2;
32    }
33
34    @Override
35    public String toString() {
36        return super.toString() + "\n" + Solve();
37    }
38 }

```

4. Відповісти на контрольні запитання

Висновки: на цій лаб. роботі ми вивчили основні принципи ООП, використали знання на практиці, створивши класи с основними наборами функцій та конструкторів.