



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Лабораторна робота №8
з дисципліни «Основи програмування»
Тема: « Основи ООП »

Виконали:

студенти групи ІА-23

Каширов Д. О.

Проценко. В. І.

Шрубович Н. С.

Перевірів:

Колеснік Валерій

Київ 2022

Хід роботи:

1. Повторити теоретичні відомості
2. Проаналізувати предметну область завдання свого варіанту (табл.1)

6	Алгоритм вирішення квадратного рівняння	Алгоритм вирішення квадратного рівняння через дискримінант	mutable
---	---	--	---------

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.

4. Відповісти на контрольні запитання

```
public class Lab8 {

    public static void main(String[] args) {
        // клас QuadEquationSolve_Discr як нащадник класа QuadEquationSolve
        // використовує static block з надписом "Let's begin!"
        QuadEquationSolve_Discr qsd = new QuadEquationSolve_Discr( a: 2, b: 1, c: -4);
        printRoots(qsd);

        // демонструємо overload конструктора класа QuadEquationSolve в класі
        // QuadEquationSolve_Discr (аргумент b ініціюється нульовим значенням)
        qsd = new QuadEquationSolve_Discr( a: 1, c: -4);
        printRoots(qsd);

        // демонструємо mutable екземпляра qsd класа QuadEquationSolve_Discr
        qsd.set_a(2);
        qsd.set_b(1);
        qsd.set_c(-20);
        printRoots(qsd);
    }

    3 usages
    static void printRoots(QuadEquationSolve_Discr qsd) {
        System.out.println("Roots of a quadratic equation (a = " + qsd.get_a() + ", b = " + qsd.get_b() + ", c = "
            + qsd.get_c() + "):");
        double[] roots = qsd.solve();
        for (int i = 0; i < roots.length; i++) {
            System.out.print(roots[i] + " ");
        }
        System.out.println();
    }
}
```

1 usage 1 inheritor

```
abstract class QuadEquationSolve {  
    3 usages  
    private double a;  
    3 usages  
    private double b;  
    3 usages  
    private double c;  
  
    2 usages  
    QuadEquationSolve(double a, double b, double c) {  
        this.a = a;  
        this.b = b;  
        this.c = c;  
    }  
  
    static {  
        // static block  
        System.out.println("Let's begin!");  
    }  
  
    5 usages  
    public double get_a() { return a; }  
  
    5 usages  
    public double get_b() { return b; }  
  
    2 usages  
    public double get_c() { return c; }  
  
    1 usage  
    public void set_a(double x) { a = x; }
```

```

1 usage
public void set_b(double x) { b = x; }

1 usage
public void set_c(double x) { c = x; }

1 usage 1 implementation
public abstract double[] solve();
}

4 usages
class QuadEquationSolve_Discr extends QuadEquationSolve {
1 usage
QuadEquationSolve_Discr(double a, double b, double c) { super(a, b, c); }

// це overload
1 usage
QuadEquationSolve_Discr(double a, double c) { super(a, b: 0, c); }

1 usage
@Override
public double[] solve() {
double d = Math.pow(get_b(), 2) - 4 * get_a() * get_c();
double[] res = {};
if (d > 0) {
res = new double[] { (-get_b() + Math.sqrt(d)) / (2 * get_a()), (-get_b() - Math.sqrt(d)) / (2 * get_a()) };
} else if (d == 0) {
res = new double[] { -get_b() / (2 * get_a()) };
}
return res;
}
}

```

Вивід програми:

```

Task6_Lab8 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Progr
Let's begin!

Roots of a quadratic equation (a = 2.0, b = 1.0, c = -4.0):
1.1861406616345072 -1.6861406616345072

Roots of a quadratic equation (a = 1.0, b = 0.0, c = -4.0):
2.0 -2.0

Roots of a quadratic equation (a = 2.0, b = 1.0, c = -20.0):
2.92214438511238 -3.42214438511238

Process finished with exit code 0

```

Висновок: Під час виконання цієї лабораторної роботи ми покращили свої знання об'єктно-орієнтованого програмування, дізналися про immutable та mutable об'єкти, зрозуміли різницю між перевантаженням (overloading) та заміщенням (overriding). Вдосконалили свої знання.