Functional decomposition → Roots of quadratic equation

Easy 3 minutes

A quadratic equation is an algebraic equation of degree two. It's easy to solve this equation when you know the quadratic formula.

Here is a simple program for calculating the real roots of a quadratic equation:

```
public static void findRoots(double a, double b, double c) {
    // the equation is ax^2 + bx + c = 0
    double discriminant = b * b - 4 * a * c;
    if (discriminant < 0) {
        System.out.println("No real roots!");
    } else if (discriminant == 0) {
        double x = -b / (2 * a);
        System.out.println("x = " + x);
    } else {

        double x1 = (-b + Math.pow(discriminant, 0.5)) / (2 * a);

        double x2 = (-b - Math.pow(discriminant, 0.5)) / (2 * a);

        System.out.println("x1 = " + x1);

        System.out.println("x2 = " + x2);

    }

}
```

What if we change it a bit by decomposing this code and creating additional methods? That's what we get then:

```
public static double calculateDiscriminant(double a, double b, double c) {
    return b * b - 4 * a * c;
}

public static void calculateRoots(double a, double b, double c, double discriminant) {
    double x1 = (-b + Math.pow(discriminant, 0.5)) / (2 * a);
    double x2 = (-b - Math.pow(discriminant, 0.5)) / (2 * a);
    if (x1 == x2) {
        System.out.println("x = " + x1);

    } else {

        System.out.println("x1 = " + x1);

        System.out.println("x2 = " + x2);

    }

}
```

What should the `main` method look like after decomposition?

Make sure that the method produces the correct output and is properly decomposed (no unnecessary actions are performed).

Choose the option:

1)

```
public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);
    int a = scanner.nextInt();
    int b = scanner.nextInt();
    int c = scanner.nextInt();

    double discriminant = calculateDiscriminant(a, b, c);


    if (discriminant < 0) {

        System.out.println("No real roots!");

    } else if (discriminant == 0) {

        calculateRoots(a, b, c, discriminant);

    } else {

        calculateRoots(a, b, c, discriminant);

    }

}
```

2)

```
public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);
    int a = scanner.nextInt();
    int b = scanner.nextInt();
    int c = scanner.nextInt();

    double discriminant = calculateDiscriminant(a, b, c);


    calculateRoots(a, b, c, discriminant);

}
```

3)

```
public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);
    int a = scanner.nextInt();
    int b = scanner.nextInt();
    int c = scanner.nextInt();

    double discriminant = calculateDiscriminant(a, b, c);


    if (discriminant < 0) {

        System.out.println("No real roots!");

    } else {

        calculateRoots(a, b, c, discriminant);

    }

}
```

Report a typo

Select one option from the list

- ◯ 1
- ◯ 2
- ◯ 3

✓ Correct.

191 users liked this problem. **11** didn't like it. **What about you?**

Continue

Comments (4) Hints (0) Useful links (0) Show discussion