



Java → Basic syntax and simple programs → Methods → [Functional decomposition](#)

2144 users solved this problem. Latest completion was about 3 hours ago.

Functional decomposition → Roots of quadratic equation

 Medium  ⌚ 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:

```
1 public static void findRoots(double a, double b, double c) {
2     // the equation is ax^2 + bx + c = 0
3     double discriminant = b * b - 4 * a * c;
4     if (discriminant < 0) {
5         System.out.println("No real roots!");
6     } else if (discriminant == 0) {
7         double x = -b / (2 * a);
8         System.out.println("x = " + x);
9     } else {
10
11         double x1 = (-b + Math.pow(discriminant, 0.5)) / (2 * a);
12
13         double x2 = (-b - Math.pow(discriminant, 0.5)) / (2 * a);
14
15         System.out.println("x1 = " + x1);
16
17         System.out.println("x2 = " + x2);
18     }
19 }
20 }
```

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

```
1 public static double calculateDiscriminant(double a, double b, double c) {
2     return b * b - 4 * a * c;
3 }
4
5 public static void calculateRoots(double a, double b, double c, double discriminant) {
6     double x1 = (-b + Math.pow(discriminant, 0.5)) / (2 * a);
7     double x2 = (-b - Math.pow(discriminant, 0.5)) / (2 * a);
8     if (x1 == x2) {
9         System.out.println("x = " + x1);
10
11     } else {
12
13         System.out.println("x1 = " + x1);
14
15         System.out.println("x2 = " + x2);
16     }
17 }
18 }
```

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)

```
1 public static void main(String[] args) {
2
3     Scanner scanner = new Scanner(System.in);
4     int a = scanner.nextInt();
5     int b = scanner.nextInt();
6     int c = scanner.nextInt();
7
8     double discriminant = calculateDiscriminant(a, b, c);
9
10
11     if (discriminant < 0) {
12
13         System.out.println("No real roots!");
14
15     } else if (discriminant == 0) {
16
17         calculateRoots(a, b, c, discriminant);
18
19     } else {
20
21         calculateRoots(a, b, c, discriminant);
22
23     }
24 }
```

2)

```
1 public static void main(String[] args) {
2
3     Scanner scanner = new Scanner(System.in);
4     int a = scanner.nextInt();
5     int b = scanner.nextInt();
6     int c = scanner.nextInt();
7
8     double discriminant = calculateDiscriminant(a, b, c);
9
10
11     calculateRoots(a, b, c, discriminant);
12
13 }
```

3)

```
1 public static void main(String[] args) {
2
3     Scanner scanner = new Scanner(System.in);
4     int a = scanner.nextInt();
5     int b = scanner.nextInt();
6     int c = scanner.nextInt();
7
8     double discriminant = calculateDiscriminant(a, b, c);
9
10
11     if (discriminant < 0) {
12
13         System.out.println("No real roots!");
14
15     } else {
16
17         calculateRoots(a, b, c, discriminant);
18
19     }
20 }
```

[Report a typo](#)

✓ Select one option from the list

- ☐ 1
- ☐ 2
- ☐ 3

✓ Correct.

170 users liked this problem. 9 didn't like it. What about you?



Continue

This content was created 3 months ago and updated 5 days ago. [Share your feedback below in comments to help us improve it!](#)

[Comments \(4\)](#) [Hints \(0\)](#) [Useful links \(0\)](#) [Show discussion](#)