## **RAPPORT DE TESTS LOGICIELS-TP2**

## Réalisé par :

- Ahmed Mahfoudhi
- Mazen Issaoui
- Fares Garrach
- Hamza Sdiri

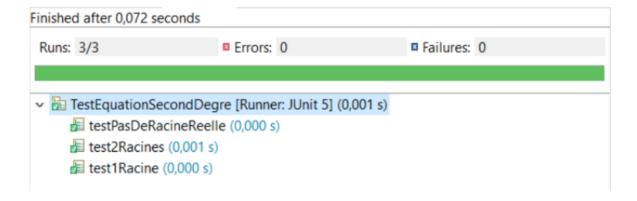
## La classe PolynomeSecondDegre

```
PolynomeSecondDegre.java ×
package tp_test2;
 3 public class PolynomeSecondDegre {
 4 private double a, b, c;
 5 private double[] racines;
 6@public PolynomeSecondDegre(double a, double b, double c) {
 7 super();
 8 this.a = a;
 9 this.b = b;
10 this.c = c;
11 }
120 public PolynomeSecondDegre() {
13 }
149 public double discriminant() {
15 return b *b - 4 * a *c;
16 }
17@ public double[] calculRacines() {
18 racines = null;
19 double discri = discriminant();
20 if (discri == 0) {
21 racines = new double[1];
22 racines[0] = -b / (2 * a);
23 } else if (discri > 0) {
24 double racineDiscriminant = Math.sqrt(discri);
25 racines = new double[2];
26 racines[0] = (-b + racineDiscriminant)/ (2 * a);
27 racines[1] = (-b - racineDiscriminant)/ (2 * a);
28 }
29 return racines;
30 }
31 }
```

Les tests de la classe PolynomeSecondDegre

```
1 package tp_test2;
4* import static org.junit.Assert.assertArrayEquals;
12
13
14 public class TestEquationSecondDegre extends TestCase{
15
160
       @Test
17
       public void test2Racines() {
18
          PolynomeSecondDegre p = new PolynomeSecondDegre(1, 1, -2);
19
          double[] expectedRacines = {1,-2};
           assertArrayEquals(expectedRacines, p.calculRacines(), θ);
20
21
       }
22
       @Test
239
       public void test1Racine() {
24
25
           PolynomeSecondDegre p = new PolynomeSecondDegre(1, -2, 1);
26
           double[] expectedRacines = {1};
27
           assertTrue(Arrays.equals(expectedRacines, p.calculRacines()));
28
       }
29
308
       @Test
       public void testPasDeRacineReelle() {
31
32
           PolynomeSecondDegre p = new PolynomeSecondDegre(1, 1, 1);
33
           double[] expectedRacines = null;
34
          assertEquals(expectedRacines, p.calculRacines());
35
       }
36
37
38 }
```

## L'exécution des tests



Dans notre cas , tous les tests sont exécutés avec succès. Cependant , Si le test échoue, JUnit stockera les informations sur l'erreur ou l'exception dans l'objet result, y compris le type d'erreur et son message. Les informations sur l'erreur peuvent être récupérées à partir de l'objet result après l'exécution du test