

Passo a passo exercício TDD

Calculadora de Strings

O método pode receber 0,1 ou 2 números e retornar sua soma.

```
6 public class CalculadoraTest {  
7  
8     @Test  
9     public void deveReceberAteDoisNumeroERetornarSuaSoma() {  
10  
11         Calculadora calculadora = new Calculadora();  
12  
13         int resultado = calculadora.soma("2,3");  
14  
15         Assert.assertEquals(5, resultado);  
16  
17     }  
18  
19 }
```

```
3 public class Calculadora {  
4  
5     public int soma(String entrada) {  
6  
7         String[] vetor = entrada.split(",");  
8  
9         int resultado = 0;  
10  
11         resultado = Integer.parseInt(vetor[0]) + Integer.parseInt(vetor[1]);  
12  
13         return resultado;  
14     }  
15  
16  
17 }
```

Para uma String Vazia (""), deve retornar zero.

```
18
19 @Test
20 public void deveRetornarZeroQuandoEntradaVazia() {
21
22     Calculadora calculadora = new Calculadora();
23
24     int resultado = calculadora.soma("");
25
26     Assert.assertEquals(0, resultado);
27
28 }
29
3 public class Calculadora {
4
5     public int soma(String entrada) {
6
7         if (entrada.equals("")) {
8             return 0;
9         }
10
11         String[] vetor = entrada.split(",");
12
13         int resultado = 0;
14
15         resultado = Integer.parseInt(vetor[0]) + Integer.parseInt(vetor[1]);
16
17         return resultado;
18     }
19
20 }
21
22 }
```

O Método deve receber uma quantidade ilimitada de números e retornar sua soma.

```
30 @Test
31 public void deveSomarEntrada() {
32
33     Calculadora calculadora = new Calculadora();
34
35     int resultado = calculadora.soma("2,3,5,8,50");
36
37     // 2+3+5+8+50 = 68
38     Assert.assertEquals(68, resultado);
39
40 }
41
```

```
3 public class Calculadora {
4
5     public int soma(String entrada) {
6
7         if (entrada.equals("")) {
8             return 0;
9         }
10
11         String[] vetor = entrada.split(",");
12
13         int resultado = 0;
14
15         for (String numero : vetor) {
16             resultado = resultado + Integer.parseInt(numero);
17         }
18
19         return resultado;
20
21     }
22
23 }
24
```

Refatoração na Classe de Teste

```
6 public class CalculadoraTest {
7
8     private Calculadora calculadora = new Calculadora();
9
10    @Test
11    public void deveRetornarZeroQuandoEntradaVazia() {
12
13        int resultado = calculadora.soma("");
14
15        Assert.assertEquals(0, resultado);
16
17    }
18
19    @Test
20    public void deveSomarEntrada() {
21
22        int resultado = calculadora.soma("2,3,5,8,50");
23
24        // 2+3+5+8+50 = 68
25        Assert.assertEquals(68, resultado);
26
27    }
28
29
30
31
32 }
33
```

Caso seja passado número negativos, uma exceção deve ser lançada.

```
29 @Test(expected=IllegalArgumentException.class)
30 public void deveGerarExcecaoQuandoNumeroNegativo() {
31
32     calculadora.soma("2,3,5,-8,50");
33
34 }
35
```

```
3 public class Calculadora {
4
5     public int soma(String entrada) {
6
7         if (entrada.equals("")) {
8             return 0;
9         }
10
11         String[] vetor = entrada.split(",");
12
13         int resultado = 0;
14
15         for (String numero : vetor) {
16
17             if(Integer.parseInt(numero) < 0) {
18                 throw new IllegalArgumentException("O numero deve ser positivo");
19             }
20
21             resultado = resultado + Integer.parseInt(numero);
22         }
23
24         return resultado;
25     }
26 }
27
```

Número maiores que 1000 devem ser ignorados.

```
36 @Test
37 public void deveSomarSomenteNumeroMenoresQueMil() {
38
39     int resultado = calculadora.soma("2000, 2,3,5,5000,8,50, 1000");
40
41     // 2+3+5+8+50 = 68
42     Assert.assertEquals(68, resultado);
43
44 }
45
46
```

```
3 public class Calculadora {
4
5     public int soma(String entrada) {
6
7         if (entrada.equals("")) {
8             return 0;
9         }
10
11         String[] vetor = entrada.split(",");
12
13         int resultado = 0;
14
15         for (String numero : vetor) {
16
17             if(Integer.parseInt(numero) < 0) {
18                 throw new IllegalArgumentException("O numero deve ser positivo");
19             }
20
21             if(Integer.parseInt(numero) >= 1000) {
22                 continue;
23             }
24
25             resultado = resultado + Integer.parseInt(numero);
26         }
27
28         return resultado;
29     }
30 }
31
32 }
```

Refatoração Final

```
3 public class Calculadora {
4
5     public int soma(String entrada) {
6
7         if (entrada.equals("")) {
8             return 0;
9         }
10
11         String[] vetor = entrada.split(",");
12
13         int resultado = 0;
14
15         for (String numero : vetor) {
16
17             int inteiro = Integer.parseInt(numero);
18
19             if(inteiro < 0) {
20                 throw new IllegalArgumentException("O numero deve ser positivo");
21             }
22
23             if(Integer.parseInt(numero) >= 1000) {
24                 continue;
25             }
26
27             resultado = resultado + inteiro;
28         }
29
30         return resultado;
31     }
32 }
33
```