TRABALHO DE FERRAMENTAS DEVOPS

Curso: 3º Período de Sistemas Para a Internet

Professora: Iara Carnevale

Alunos:

★ Daniel Mesquita Oliveira RAº 14044
★ Felipe Gustavo RAº 13663
★ Gabriel Capoia RAº 13630
★ João Gabryel RAº 13459

Técnica utilizada para os Testes

A técnica que utilizamos para os testes foi o teste de caixa preta que é o teste que utiliza conjunto de dados, esse teste foi utilizado para testar os métodos sem se preocupar com a codificação e sim, se preocupar com o funcionamento deles. Utilizamos testes positivos e testes negativos para compor os testes, utilizamos esses testes, pois eles possibilitam que analisemos os resultados assertivamente conforme as requisições, utilizamos classes de equivalência para a validação desses métodos.

Métodos Utilizados para os testes

```
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.fail;

public class Math_14044_13663_13630_13459 {


    public static int adicionar(int a, int b) {
        return a + b;
    }

    public static int subtrair(int a, int b) {
        return a - b;
    }

    public static int multiplicar(int a, int b) {
        return a * b;
    }

    public static int dividir(int a, int b) {
        return a / b;
    }
}
```

Código dos Testes

```
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

public class Math_14044_13663_13630_13459_Test {

    @Test
    public void testAdicionar1() {
        int resultado = Math_14044_13663_13630_13459.adicionar(2, 3);
        assertEquals(5, resultado);
        imprimirResultado("Resultado adicionar1: ", resultado);
    }

    @Test
    public void testAdicionar2() {
        int resultado = Math_14044_13663_13630_13459.adicionar(-3, 2);
        assertEquals(-1, resultado);
        imprimirResultado("Resultado adicionar2: ", resultado);
    }

    @Test
    public void testAdicionar3() {
        int resultado = Math_14044_13663_13630_13459.adicionar(-2, -3);
        assertEquals(-5, resultado);
        imprimirResultado("Resultado adicionar3: ", resultado);
    }

    @Test
    public void testAdicionar4() {
        int resultado = Math_14044_13663_13630_13459.adicionar(0, 3);
        assertEquals(3, resultado);
        imprimirResultado("Resultado adicionar4: ", resultado);
    }

    @Test
    public void testAdicionar5() {
        int resultado = Math_14044_13663_13630_13459.adicionar(0, -3);
        assertEquals(-3, resultado);
        imprimirResultado("Resultado adicionar5: ", resultado);
    }

    @Test
```

```
public void testNegativoAdd() {
    int resultado = Math_14044_13663_13630_13459.adicionar(Integer.parseInt("a"), 1);
}

@Test
public void testSubtrair1() {
    int resultado = Math_14044_13663_13630_13459.subtrair(3, 2);
    assertEquals(1, resultado);
    imprimirResultado("Resultado subtrair1: ", resultado);
}

@Test
public void testSubtrair2() {
    int resultado = Math_14044_13663_13630_13459.subtrair(-3, 2);
    assertEquals(-5, resultado);
    imprimirResultado("Resultado subtrair2: ", resultado);
}

@Test
public void testSubtrair3() {
    int resultado = Math_14044_13663_13630_13459.subtrair(-2, -3);
    assertEquals(1, resultado);
    imprimirResultado("Resultado subtrair3: ", resultado);
}

@Test
public void testSubtrair4() {
    int resultado = Math_14044_13663_13630_13459.subtrair(0, 3);
    assertEquals(-3, resultado);
    imprimirResultado("Resultado subtrair4: ", resultado);
}

@Test
public void testSubtrair5() {
    int resultado = Math_14044_13663_13630_13459.subtrair(0, -3);
    assertEquals(3, resultado);
    imprimirResultado("Resultado subtrair5: ", resultado);
}

@Test
public void testNegativoSub() {
    int resultado = Math_14044_13663_13630_13459.subtrair(Integer.parseInt("a"), 1);
    }
```

```
@Test
public void testMultiplicar1() {
    int resultado = Math_14044_13663_13630_13459.multiplicar(2, 3);
    assertEquals(6, resultado);
    imprimirResultado("Resultado multiplicar1: ", resultado);
}

@Test
public void testMultiplicar2() {
    int resultado = Math_14044_13663_13630_13459.multiplicar(-2, 3);
    assertEquals(-6, resultado);
    imprimirResultado("Resultado multiplicar2: ", resultado);
}

@Test
public void testMultiplicar3() {
    int resultado = Math_14044_13663_13630_13459.multiplicar(-2, -3);
    assertEquals(6, resultado);
    imprimirResultado("Resultado multiplicar3: ", resultado);
}

@Test
public void testMultiplicar4() {
    int resultado = Math_14044_13663_13630_13459.multiplicar(0, 3);
    assertEquals(0, resultado);
    imprimirResultado("Resultado multiplicar4: ", resultado);
}

@Test
public void testMultiplicar5() {
    int resultado = Math_14044_13663_13630_13459.multiplicar(0, -3);
    assertEquals(0, resultado);
    imprimirResultado("Resultado multiplicar5: ", resultado);
}

@Test
public void testNegativoMulti() {
    int resultado = Math_14044_13663_13630_13459.multiplicar(Integer.parseInt("a"), 2);
}

@Test
public void testDividir1() {
    int resultado = Math_14044_13663_13630_13459.dividir(6, 3);
    assertEquals(2, resultado);
```

```
    imprimirResultado("Resultado dividir1: ", resultado);

}

@Test
public void testDividir2() {
    int resultado = Math_14044_13663_13630_13459.dividir(6, -3);
    assertEquals(-2, resultado);
    imprimirResultado("Resultado dividir2: ", resultado);

}

@Test
public void testDividir3() {
    int resultado = Math_14044_13663_13630_13459.dividir(-6, -3);
    assertEquals(2, resultado);
    imprimirResultado("Resultado dividir3: ", resultado);

}

@Test
public void testDividir4() {
    int resultado = Math_14044_13663_13630_13459.dividir(0, 3);
    assertEquals(0, resultado);
    imprimirResultado("Resultado dividir4: ", resultado);

}

@Test
public void testDividir5() {
    int resultado = Math_14044_13663_13630_13459.dividir(0, -3);
    assertEquals(0, resultado);
    imprimirResultado("Resultado dividir5: ", resultado);

}

@Test
public void testNegativoDiv() {
    int resultado = Math_14044_13663_13630_13459.dividir(1, 0);
    assertEquals(0, resultado);

}
```

```
    @Test
    public void testNegativoDiv2() {
        int resultado = Math_14044_13663_13630_13459.dividir(Integer.parseInt("a"), 0);
    }

    private void imprimirResultado(String mensagem, int resultado) {
        System.out.println(mensagem + ": " + resultado);
    }
}
```

Tabelas de Testes

SOMA

Testes Positivos		
ENTRADA		SAIDA
2	3	5
-3	2	-1
-2	-3	-5
0	3	3
0	-3	-3

Testes Negativos		
ENTRADA		SAIDA
-1	"a"	erro

SUBTRAÇÃO

Testes Positivos		
ENTRADA		SAIDA
3	2	1
-3	2	-5
-2	-3	1
0	3	-3
0	-3	3

Testes Negativos		
ENTRADA		SAIDA
"a"	1	erro

MULTIPLICAÇÃO

Testes Positivos		
ENTRADA		SAIDA
2	3	6
-2	3	-6
-2	-3	6
0	3	0
0	-3	0
0	0	0

Testes Negativos		
ENTRADA		SAIDA
-1	a	erro

DIVISÃO

Testes Positivos		
ENTRADA		SAIDA
6	3	2
6	-3	-2
-6	-3	2
0	3	0
0	-3	0

Testes Negativos		
ENTRADA		SAIDA
-1	a	erro
1	0	#DIV/0!
0	0	#DIV/0!