# **TESTE JUNIT 1**

**ERRO**:

“Resposta” Não foi encontrado como uma variavel.

**CORREÇÃO**:

Criar um ENUM “Resposta”.

## **Classe JUNIT**

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.Assert;

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTeste extends JulgamentoPrisioneiro {

@Test

public void testCenario1() {

Resposta respostaPrisioneiroA = Resposta.DELACAO;

Resposta respostaPrisioneiroB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

Assert.assertNotNull(jp);

int penaPrisioneiroA = jp.calculaPena(respostaPrisioneiroA, respostaPrisioneiroB);

int penaPrisioneiroB = jp.calculaPena(respostaPrisioneiroB, respostaPrisioneiroA);

Assert.assertEquals(15, penaPrisioneiroA);

Assert.assertEquals(15, penaPrisioneiroB); }

@Test

public void testCenario2() {

Resposta respostaPrisioneiroA = Resposta.NEGACAO;

Resposta respostaPrisioneiroB = Resposta.NEGACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

Assert.assertNotNull(jp);

int penaPrisioneiroA = jp.calculaPena(respostaPrisioneiroA, respostaPrisioneiroB);

int penaPrisioneiroB = jp.calculaPena(respostaPrisioneiroB, respostaPrisioneiroA);

Assert.assertEquals(10, penaPrisioneiroA);

Assert.assertEquals(10, penaPrisioneiroB);}

public void testCenario3() {

Resposta respostaPrisioneiroA = Resposta.DELACAO;

Resposta respostaPrisioneiroB = Resposta.NEGACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

Assert.assertNotNull(jp);

int penaPrisioneiroA = jp.calculaPena(respostaPrisioneiroA, respostaPrisioneiroB);

int penaPrisioneiroB = jp.calculaPena(respostaPrisioneiroB, respostaPrisioneiroA);

Assert.assertEquals(10, penaPrisioneiroA); }

}

## **Classe**

public class JulgamentoPrisioneiro {

private int PENA\_INOCENCIA = 10;

private int PENA\_CONDENACAO\_MUTUA = 15;

private int PENA\_CONDENACAO\_INDIVIDUAL = 10;

private int PENA\_CONDENACAO\_CUMPLICES = 11;

public enum Resposta {

NEGACAO, DELACAO

}

public int calculaPena(Resposta respostaPrisioneiroA, Resposta respostaPrisioneiroB) {

if (respostaPrisioneiroA == Resposta.DELACAO) {

if (respostaPrisioneiroB == Resposta.DELACAO) {

return PENA\_CONDENACAO\_MUTUA;

} else {

return PENA\_INOCENCIA;

}

} else {

if (respostaPrisioneiroB == Resposta.DELACAO) {

return PENA\_CONDENACAO\_INDIVIDUAL;

} else {

return PENA\_CONDENACAO\_CUMPLICES; }}}}

# **TESTE JUNIT 2**

**ERRO**:

Error no assert equals no “cenário2”.

**CORREÇÃO**:

Correção do retorno do if else “pena cumplice” > “pena inocência”.

## **Classe JUNIT**

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.Assert;

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTeste extends JulgamentoPrisioneiro {

@Test

public void testCenario1() {

Resposta respostaPrisioneiroA = Resposta.DELACAO;

Resposta respostaPrisioneiroB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

Assert.assertNotNull(jp);

int penaPrisioneiroA = jp.calculaPena(respostaPrisioneiroA, respostaPrisioneiroB);

int penaPrisioneiroB = jp.calculaPena(respostaPrisioneiroB, respostaPrisioneiroA);

Assert.assertEquals(15, penaPrisioneiroA);

Assert.assertEquals(15, penaPrisioneiroB);;}

@Test

public void testCenario2() {

Resposta respostaPrisioneiroA = Resposta.NEGACAO;

Resposta respostaPrisioneiroB = Resposta.NEGACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

Assert.assertNotNull(jp);

int penaPrisioneiroA = jp.calculaPena(respostaPrisioneiroA, respostaPrisioneiroB);

int penaPrisioneiroB = jp.calculaPena(respostaPrisioneiroB, respostaPrisioneiroA);

Assert.assertEquals(10, penaPrisioneiroA);

Assert.assertEquals(10, penaPrisioneiroB);

}

public void testCenario3() {

Resposta respostaPrisioneiroA = Resposta.DELACAO;

Resposta respostaPrisioneiroB = Resposta.NEGACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

Assert.assertNotNull(jp);

int penaPrisioneiroA = jp.calculaPena(respostaPrisioneiroA, respostaPrisioneiroB);

int penaPrisioneiroB = jp.calculaPena(respostaPrisioneiroB, respostaPrisioneiroA);

Assert.assertEquals(10, penaPrisioneiroA);}}

## **Classe**

public class JulgamentoPrisioneiro {

private int PENA\_INOCENCIA = 10;

private int PENA\_CONDENACAO\_MUTUA = 15;

private int PENA\_CONDENACAO\_INDIVIDUAL = 10;

private int PENA\_CONDENACAO\_CUMPLICES = 11;

public enum Resposta {

NEGACAO, DELACAO

}

public int calculaPena(Resposta respostaPrisioneiroA, Resposta respostaPrisioneiroB) {

if (respostaPrisioneiroA == Resposta.DELACAO) {

if (respostaPrisioneiroB == Resposta.DELACAO) {

return PENA\_CONDENACAO\_MUTUA;

} else {

return PENA\_CONDENACAO\_CUMPLICES;

}

} else {

if (respostaPrisioneiroB == Resposta.DELACAO) {

return PENA\_CONDENACAO\_INDIVIDUAL;

} else {

return PENA\_INOCENCIA;

}

}

}

}

# **TESTE JUNIT 3**

**ERRO**:

Não houve erro apresentado.

**CORREÇÂO**:

Não Necessaria.

## **Classe JUNIT**

Se mantem a mesma do Teste JUNIT 2.

## **Classe**

Se mantem a mesma do Teste JUNIT 2.