**Jeferson Barreto Nascimento**

**RM 79557 – 3SIR**

**Etapa de Teste 1**

**JulgamentoPrisioneiro.java**

**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** **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;

}

}

}

}

**JulgamentoPrisioneiroTest.java**

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

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTest {

@Test

void testJulgamento() {

fail("Not yet implemented");

}

}

**Observações:** Erro encontrado na classe de implementação pela ausência de declaração do *enum* Resposta.

**Etapa de Teste 2**

JulgamentoPrisioneiro.java

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;

}

}

}

}

JulgamentoPrisioneiroTest.java

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

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTest {

@Test

void testJulgamento() {

fail("Not yet implemented");

}

}

Observações: Correção do erro de implementação com a adição da enum Resposta.

**Etapa de Teste 3**

JulgamentoPrisioneiro.java

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;

}

}

}

}

JulgamentoPrisioneiroTest.java

import static org.junit.Assert.assertNotNull;

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

import org.junit.jupiter.api.Test;

import JulgamentoPrisioneiro.Resposta;

class JulgamentoPrisioneiroTest {

@Test

void testJulgamento() {

Resposta respSuspeitoA = Resposta.DELACAO;

Resposta respSuspeitoB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

assertNotNull(jp);

int penaSuspeitoA = jp.calculaPena(respSuspeitoA, respSuspeitoB);

int penaSuspeitoB = jp.calculaPena(respSuspeitoB, respSuspeitoA);

assertEquals(5, penaSuspeitoA);

assertEquals(5, penaSuspeitoB);

}

}

Observações: Erro na importação do enum Resposta e da utilização no teste.

**Etapa de Teste 4**

JulgamentoPrisioneiro.java

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;

}

}

}

}

Resposta.java

public enum Resposta {

NEGACAO, DELACAO

}

JulgamentoPrisioneiroTest.java

import static org.junit.Assert.assertNotNull;

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

import org.junit.jupiter.api.Test;

import JulgamentoPrisioneiro.Resposta;

class JulgamentoPrisioneiroTest {

@Test

void testJulgamento() {

Resposta respSuspeitoA = Resposta.DELACAO;

Resposta respSuspeitoB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

assertNotNull(jp);

int penaSuspeitoA = jp.calculaPena(respSuspeitoA, respSuspeitoB);

int penaSuspeitoB = jp.calculaPena(respSuspeitoB, respSuspeitoA);

assertEquals(5, penaSuspeitoA);

assertEquals(5, penaSuspeitoB);

}

}

Observações: Adição da classe Resposta.java contendo o enum público para utilização na classe de implementação e de teste.

**Etapa de Teste 5**

JulgamentoPrisioneiro.java

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 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;

}

}

}

}

Resposta.java

public enum Resposta {

NEGACAO, DELACAO

}

JulgamentoPrisioneiroTest.java

import static org.junit.Assert.assertNotNull;

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

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTest {

@Test

public void testJulgamento() {

Resposta respSuspeitoA = Resposta.DELACAO;

Resposta respSuspeitoB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

assertNotNull(jp);

int penaSuspeitoA = jp.calculaPena(respSuspeitoA, respSuspeitoB);

int penaSuspeitoB = jp.calculaPena(respSuspeitoB, respSuspeitoA);

assertEquals(5, penaSuspeitoA);

assertEquals(5, penaSuspeitoB);

}

}

Observações: Remoção do enum Resposta na classe de implementação (JulgamentoPrisioneiro.java) para utilização do correspondente que consta na classe (Resposta.java)

**Etapa de Teste 6**

JulgamentoPrisioneiro.java

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 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;

}

}

}

}

Resposta.java

public enum Resposta {

NEGACAO, DELACAO

}

JulgamentoPrisioneiroTest.java

import static org.junit.Assert.assertNotNull;

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

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTest {

@Test

public void testJulgamento() {

Resposta respSuspeitoA = Resposta.DELACAO;

Resposta respSuspeitoB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

assertNotNull(jp);

int penaSuspeitoA = jp.calculaPena(respSuspeitoA, respSuspeitoB);

int penaSuspeitoB = jp.calculaPena(respSuspeitoB, respSuspeitoA);

assertEquals(15, penaSuspeitoA);

assertEquals(15, penaSuspeitoB);

}

}

Observações: Teste de execução simulando situação de acerto realizado com sucesso.

(penaSuspeitoA==15 && penaSuspeitoB==15) = Acerto

Etapa de Teste 7

JulgamentoPrisioneiro.java

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 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;

}

}

}

}

Resposta.java

public enum Resposta {

NEGACAO, DELACAO

}

JulgamentoPrisioneiroTest.java

import static org.junit.Assert.assertNotNull;

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

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTest {

@Test

public void testJulgamento() {

Resposta respSuspeitoA = Resposta.DELACAO;

Resposta respSuspeitoB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

assertNotNull(jp);

int penaSuspeitoA = jp.calculaPena(respSuspeitoA, respSuspeitoB);

int penaSuspeitoB = jp.calculaPena(respSuspeitoB, respSuspeitoA);

assertEquals(10, penaSuspeitoA);

assertEquals(15, penaSuspeitoB);

}

}

Observações: Teste de execução simulando situação de erro realizado com sucesso.

(penaSuspeitoA!=15 && penaSuspeitoB==15) = ERRO

Resultado Final: Alterações e testes realizados com sucesso em aplicações de dados positivos e negativos na simulação