Testes com JUNIT

Código Inicial

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;

}

}

}

}

JUNIT Inicial

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

import org.junit.jupiter.api.Test;

import junit.framework.Assert;

class JulgamentoPrisioneiroTeste {

@Test

void testeCalculaPena() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

int penaSuspeitoA = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

int penaSuspeitoB = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

assertEquals(15, penaSuspeitoA);

assertEquals(15, penaSuspeitoB);

}

}

Ao fazer a primeira execução do teste, a JUNIT apresentou 1 erro de compilação, portanto devemos fazer algumas correções no código.

Devemos corrigir o erro “Resposta cannot be resolved to a type/variable”.

Ou seja, devemos criar a enum class Resposta.

public enum Resposta {

DELACAO,INOCENTE

}

Agora o teste de condenação mutua está correto, não apresenta erros, portanto devemos criar o próximo teste.

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

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTeste {

// teste condenação mutua

@Test

void testeCalculaPena() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.DELACAO;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

int penaSuspeitoA = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

int penaSuspeitoB = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

assertEquals(15, penaSuspeitoA);

assertEquals(15, penaSuspeitoB);

}

// teste inocencia

@Test

void testeInocencia() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.INOCENTE;

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

int penaSuspeitoA = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

assertEquals(10, penaSuspeitoA);

}

}

Resultado do teste OK, agora devemos criar o próximo teste PENA\_CONDENACAO\_INDIVIDUAL e alterar os valores das condenações para:

**private** **int** PENA\_INOCENCIA = 0;

**private** **int** PENA\_CONDENACAO\_MUTUA = 5;

**private** **int** PENA\_CONDENACAO\_INDIVIDUAL = 10;

**private** **int** PENA\_CONDENACAO\_CUMPLICES = 11;

// teste condenaçao individual

@Test

**void** testeCondenacaoIndividual() {

Resposta respostaSuspeitoA = Resposta.***INOCENTE***;

Resposta respostaSuspeitoB = Resposta.***DELACAO***;

JulgamentoPrisioneiro jp = **new** JulgamentoPrisioneiro();

**int** penaSuspeitoA = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

**int** penaSuspeitoB = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

*assertEquals*(10, penaSuspeitoA);

*assertEquals*(10, penaSuspeitoB);

}

Resultado do teste OK, devemos criar o ultimo teste

// teste condenaçao cumplices

@Test

**void** testeCondenacaoCumplices() {

Resposta respostaSuspeitoA = Resposta.***INOCENTE***;

Resposta respostaSuspeitoB = Resposta.***INOCENTE***;

JulgamentoPrisioneiro jp = **new** JulgamentoPrisioneiro();

**int** penaSuspeitoA = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

**int** penaSuspeitoB = jp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

*assertEquals*(11, penaSuspeitoA);

*assertEquals*(11, penaSuspeitoB);

}

Teste Ok