Nome: Freddy Ramos Zenteno

Rm: 81590

Classe Orignal

**package** br.com.fiap.main;

**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;

}

}

}

}

JUNIT GERADA

package br.com.fiap.main;

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

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTest {

@Test

void testCalculaPena() {

fail("Not yet implemented");

}

}

Aqui eu apliquei um cenário onde cada suspeito deu como reposta delação

**package** br.com.fiap.main;

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

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

**import** br.com.fiap.main.JulgamentoPrisioneiro.Resposta;

**class** JulgamentoPrisioneiroTest {

@Test

**void** testCalculaPenaCenario1() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

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

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(15, penaSuspeitoA);

*assertEquals*(15, penaSuspeitoB);

}

}

Aqui o segundo cenário onde o suspeito A, B tem como resposta deleção e negação respectivamente

**package** br.com.fiap.main;

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

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

**import** br.com.fiap.main.JulgamentoPrisioneiro.Resposta;

**class** JulgamentoPrisioneiroTest {

@Test

**void** testCalculaPenaCenario1() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

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

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(15, penaSuspeitoA);

*assertEquals*(15, penaSuspeitoB);

}

@Test

**void** testCalculaPenaCenario2() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(10, penaSuspeitoA);

*assertEquals*(10, penaSuspeitoB);

}

}

Aqui o segundo cenário onde o suspeito A, B tem como resposta negação e negação respectivamente

**package** br.com.fiap.main;

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

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

**import** br.com.fiap.main.JulgamentoPrisioneiro.Resposta;

**class** JulgamentoPrisioneiroTest {

@Test

**void** testCalculaPenaCenario1() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

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

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(15, penaSuspeitoA);

*assertEquals*(15, penaSuspeitoB);

}

@Test

**void** testCalculaPenaCenario2() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(10, penaSuspeitoA);

*assertEquals*(10, penaSuspeitoB);

}

@Test

**void** testCalculaPenaCenario3() {

Resposta respostaSuspeitoA = Resposta.***NEGACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(11, penaSuspeitoA);

*assertEquals*(11, penaSuspeitoB);

}

}

Aqui resultado final da classe JulgamentoPrisioneiro e JulgamentoPrisioneiroTeste

Classe:

**package** br.com.fiap.main;

**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;

}

}

}

}

Classe teste:

**package** br.com.fiap.main;

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

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

**import** br.com.fiap.main.JulgamentoPrisioneiro.Resposta;

**class** JulgamentoPrisioneiroTest {

@Test

**void** testCalculaPenaCenario1() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

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

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(15, penaSuspeitoA);

*assertEquals*(15, penaSuspeitoB);

}

@Test

**void** testCalculaPenaCenario2() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(10, penaSuspeitoA);

*assertEquals*(10, penaSuspeitoB);

}

@Test

**void** testCalculaPenaCenario3() {

Resposta respostaSuspeitoA = Resposta.***NEGACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

JulgamentoPrisioneiro dp = **new** JulgamentoPrisioneiro();

*assertNotNull*(dp);

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

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

*assertEquals*(11, penaSuspeitoA);

*assertEquals*(11, penaSuspeitoB);

}

}