Nome: Yago Henrique Rebello

RM: 78756

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

import org.junit.jupiter.api.Test;

class Prisioneiro {

@Test

public void testCondenacaoMutua() {

String respostaPrisioneiroA = "Culpado";

String respostaPrisioneiroB = "Culpado";

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

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

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

assertEquals(5, prisioneiroA);

assertEquals(5, prisioneiroB);

}

@Test

public void testCondenacaoIndidual() {

String respostaPrisioneiroA = "Culpado";

String respostaPrisioneiroB = "Inocente";

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

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

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

assertEquals(10, prisioneiroA);

assertEquals(0, prisioneiroB);

}

@Test

public void testCondenacaoCumplice() {

String respostaPrisioneiroA = "Culpado";

String respostaPrisioneiroB = "Culpado";

JulgamentoPrisioneiro jp = new JulgamentoPrisioneiro();

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

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

assertEquals(5, prisioneiroA);

assertEquals(1, prisioneiroB);

}

}

2)Rodei o primeiro teste, ele deu o seguinte erro:

The method calculaPena(Resposta, Resposta) from tthe type JulgamnetoPrisioneiro refers to the missing type Resposta.

Classe utilizada:

**public** **class** JulgamentoPrisioneiro {

**private** **int** PENA\_INOCENCIA = 13;

**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;

}

}

}

}

3) Classe concertada, mas continua exibindo o seguinte erro:

The method calculaPena(JulgamentoPrisioneiro.Resposta, JulgamentoPrisioneiro.Resposta) in the type JulgamnetoPrisioneiro is not applicable for the arguments (String, String)

Classe arrumada:

**public** **class** JulgamentoPrisioneiro {

**private** **int** PENA\_INOCENCIA = 13;

**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;

}

}

}

4)Classe de teste corrigida, mas exibindo valores errados por conta da classe fornecida pelo professor

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

**import** org.junit.Assert;

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

**import** JulgamentoPrisioneiro.Resposta;

**class** Prisioneiro {

@Test

**public** **void** testCondenacaoMutua() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.DELACAO;

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

Assert.*assertNotNull*(jp);

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

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

*assertEquals*(5, prisioneiroA);

*assertEquals*(5, prisioneiroB);

}

@Test

**public** **void** testCondenacaoIndidual() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.DELACAO;

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

Assert.*assertNotNull*(jp);

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

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

*assertEquals*(10, prisioneiroA);

*assertEquals*(0, prisioneiroB);

}

@Test

**public** **void** testCondenacaoCumplice() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.DELACAO;

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

Assert.*assertNotNull*(jp);

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

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

*assertEquals*(5, prisioneiroA);

*assertEquals*(1, prisioneiroB);

}

}

5) Classe fornecida pelo professor corrigida

**public** **class** JulgamentoPrisioneiro {

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

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

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

**private** **int** PENA\_CONDENACAO\_CUMPLICES = 1;

**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 de teste correta :

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

**import** org.junit.Assert;

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

**import** JulgamentoPrisioneiro.Resposta;

**class** Prisioneiro {

@Test

**public** **void** testCondenacaoMutua() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.DELACAO;

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

Assert.*assertNotNull*(jp);

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

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

*assertEquals*(5, prisioneiroA);

*assertEquals*(5, prisioneiroB);

}

@Test

**public** **void** testCondenacaoIndidual() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.DELACAO;

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

Assert.*assertNotNull*(jp);

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

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

*assertEquals*(10, prisioneiroA);

*assertEquals*(0, prisioneiroB);

}

@Test

**public** **void** testCondenacaoCumplice() {

Resposta respostaSuspeitoA = Resposta.DELACAO;

Resposta respostaSuspeitoB = Resposta.DELACAO;

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

Assert.*assertNotNull*(jp);

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

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

*assertEquals*(5, prisioneiroA);

*assertEquals*(1, prisioneiroB);

}

}