NAC Governança e melhores práticas em TI

Aluno: Guilherme Godoy Freitas

Rm:77822

Classe JulgamentoPrisioneiro

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 int calculaPena(String respostaPrisioneiroA, String respostaPrisioneiroB) {

if (respostaPrisioneiroA == "Culpado") {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_MUTUA;

} else {

return PENA\_CONDENACAO\_INDIVIDUAL;

}

} else {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_CUMPLICES;

} else {

return PENA\_INOCENCIA;

}

}

}

}

JUNIT

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

import org.junit.jupiter.api.Test;

class JulgamentoPrisioneiroTeste {

*@Test*

void testCalcularPenaInocencia() {

JulgamentoPrisioneiro julgamento = new JulgamentoPrisioneiro();

int pena = julgamento.calculaPena("Inocente", "Inocente");

*assertEquals*(0, pena);

}

*@Test*

void testCalcularPenaCumplices() {

JulgamentoPrisioneiro julgamento = new JulgamentoPrisioneiro();

int pena = julgamento.calculaPena("Inocente", "Culpado");

*assertEquals*(1, pena);

}

*@Test*

void testCalcularPenaIndividual() {

JulgamentoPrisioneiro julgamento = new JulgamentoPrisioneiro();

int pena = julgamento.calculaPena("Culpado", "Inocente");

*assertEquals*(10, pena);

}

*@Test*

void testCalcularPenaMutua() {

JulgamentoPrisioneiro julgamento = new JulgamentoPrisioneiro();

int pena = julgamento.calculaPena("Culpado", "Culpado");

*assertEquals*(5, pena);

}

}

Testes Realizados:

Teste 1

Ao realizar o primeiro teste do código temos o retorno do seguinte erro:

java.lang.Error: Unresolved compilation problem:

The method calculaPena(Resposta, Resposta) from the type JulgamentoPrisioneiro refers to the missing type Resposta

at JulgamentoPrisioneiroTeste.testCalcularPenaMutua(JulgamentoPrisioneiroTeste.java:34)...

Causa: No código temos referencias que estão faltando, no caso o método calculaPena está faltando a referência do tipo Resposta

Correção:

Como na referência ele compara uma String foi alterado o formado de entrada do método, recebendo uma String e comparando também a uma String

public int calculaPena(String respostaPrisioneiroA, String respostaPrisioneiroB) {

if (respostaPrisioneiroA == "Culpado") {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_MUTUA;

} else {

return PENA\_INOCENCIA;

}

} else {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_INDIVIDUAL;

} else {

return PENA\_CONDENACAO\_CUMPLICES;

}

}

}

Teste 2

Na realização do segundo teste tivemos o retorno do seguinte erro:

org.opentest4j.AssertionFailedError: expected: <0> but was: <11>

Causa: Nas variáveis inicias de PENA\_.. os valores estavam declarados com os valores errados.

Correção:

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;

Teste 3

Ao realizar o terceiro teste tivemos o retorno do seguinte erro:

org.opentest4j.AssertionFailedError: expected: <0> but was: <1>

Causa: Como foi feita uma correção (anterior) nos valores das variáveis, foi visto que as ordens estavam invertidas no código para as penas do que foi passado pelo “Cliente”

Correção:

public int calculaPena(String respostaPrisioneiroA, String respostaPrisioneiroB) {

if (respostaPrisioneiroA == "Culpado") {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_MUTUA;

} else {

return PENA\_CONDENACAO\_INDIVIDUAL;

}

} else {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_CUMPLICES;

} else {

return PENA\_INOCENCIA;

}

}

}

}