Junit

Leonardo Bueno Prado Gomes

RM82829

import br.com.fiap.compliance.avaliacao.main.JulgamentoPrisioneiro;  
import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
  
public class JulgamentoPrisioneiroTest {  
  
 @Test  
 public void calculaPenaTest() {  
  
 JulgamentoPrisioneiro julgamentoPrisioneiro = new JulgamentoPrisioneiro();  
  
 *assertEquals*(0, julgamentoPrisioneiro.calculaPena("Inocente", "Inocente"));  
 *assertEquals*(5, julgamentoPrisioneiro.calculaPena("Culpado", "Culpado"));  
 *assertEquals*(10, julgamentoPrisioneiro.calculaPena("Culpado", "Inocente"));  
 *assertEquals*(1, julgamentoPrisioneiro.calculaPena("Inocente", "Culpado"));  
 }  
}

Não foi possível executar os testes acima devido a um erro de compilação, já que a classe “Resposta”, não existe

A solução dada foi alterar o tipo dos parmêmetros do método “consultaPena” para “String” e “String”, assim como as comparações com a classe inexistente tiveram que ser substituidas por comparações do tipo “equals” da classe String.

Ao executar os tests novamente o resultado foi o seguinte:

org.opentest4j.AssertionFailedError:

Expected :0

Actual :11

A solução foi corrigir os valores das constantes definidas na classe “JulgamentoPrisioneiro”.

Ao executar novamente o resultado foi:

org.opentest4j.AssertionFailedError:

Expected :0

Actual :1

A solução foi corrigir os retornos dentro dos condicionais, que não estavam de acordo com a definição original.

Ao rodar os testes novamente, todos passaram. Resultado:

Process finished with exit code 0.

Segue versão final da 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.equals("Culpado") ) {  
 if (respostaPrisioneiroB.equals("Culpado")) {  
 return PENA\_CONDENACAO\_MUTUA;  
 } else {  
 return PENA\_CONDENACAO\_INDIVIDUAL;  
 }  
 } else {  
 if (respostaPrisioneiroB.equals("Culpado")) {  
 return PENA\_CONDENACAO\_CUMPLICES;  
 } else {  
 return PENA\_INOCENCIA;  
 }  
 }  
 }  
}

Versão final dos testes:

public class JulgamentoPrisioneiroTest {

private static JulgamentoPrisioneiro julgamentoPrisioneiro;

@BeforeAll

public static void init() {

julgamentoPrisioneiro = new JulgamentoPrisioneiro();

}

@Test

public void calculaPenaInocencia() {

assertEquals(0, julgamentoPrisioneiro.calculaPena("Inocente", "Inocente"));

}

@Test

public void calculaPenaMutua() {

assertEquals(5, julgamentoPrisioneiro.calculaPena("Culpado", "Culpado"));

}

@Test

public void calculaPenaIndividual() {

assertEquals(10, julgamentoPrisioneiro.calculaPena("Culpado", "Inocente"));

}

@Test

public void calculaPenaCumplices() {

assertEquals(1, julgamentoPrisioneiro.calculaPena("Inocente", "Culpado"));

}

}