***Nome: Vinícius Pereira de Andrade Santos RM: 79937***

Parte 1 – Criar JUNIT

Foi desenvolvida a seguinte JUNIT com base no algoritmo que me foi fornecido:

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

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.Test;

import junit.framework.TestCase;

import org.junit.jupiter.api.Test;

import junit.framework.TestCase;

class JUNITteste extends TestCase {

*@Test*

void testeCondenacaoMutua() {

String respostaPrisioneiroA = "Culpado";

String respostaPrisioneiroB = "Culpado";

JulgamentoPrisioneiro julg = new JulgamentoPrisioneiro();

int resultuadoEsperado = 5;

int resultadoReal = julg.calculaPena(respostaPrisioneiroA,respostaPrisioneiroB);

*assertEquals*(resultuadoEsperado, resultadoReal);

}

*@Test*

void testeCondenacaoIndividual() {

String respostaPrisioneiroA = "Culpado";

String respostaPrisioneiroB = "Inocente";

JulgamentoPrisioneiro julg = new JulgamentoPrisioneiro();

int resultuadoEsperado = 10;

int resultadoReal = julg.calculaPena(respostaPrisioneiroA,respostaPrisioneiroB);

*assertEquals*(resultuadoEsperado, resultadoReal);

}

*@Test*

void testeCondenacaoCumplices() {

String respostaPrisioneiroA = "Inocente";

String respostaPrisioneiroB = "Culpado";

JulgamentoPrisioneiro julg = new JulgamentoPrisioneiro();

int resultuadoEsperado = 1;

int resultadoReal = julg.calculaPena(respostaPrisioneiroA,respostaPrisioneiroB);

*assertEquals*(resultuadoEsperado, resultadoReal);

}

*@Test*

void testeInocencia() {

String respostaPrisioneiroA = "Inocente";

String respostaPrisioneiroB = "Inocente";

JulgamentoPrisioneiro julg = new JulgamentoPrisioneiro();

int resultuadoEsperado = 0;

int resultadoReal = julg.calculaPena(respostaPrisioneiroA,respostaPrisioneiroB);

*assertEquals*(resultuadoEsperado, resultadoReal);

}

}

Parte 2 – Criar arquivo Java

Assim que foi criado o seguinte arquivo Java:

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) {

} else {

return PENA\_INOCENCIA;

}

} else {

if (respostaPrisioneiroB == Resposta.DELACAO) {

return PENA\_CONDENACAO\_INDIVIDUAL;

} else {

return PENA\_CONDENACAO\_CUMPLICES;

}

}

}

}

a JUNIT já relatou o seguinte erro:

java.lang.Error: Unresolved compilation problem:

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

Decidi fazer a seguinte alteração no tipo da resposta, de “Resposta” para “String”, da seguinte forma:

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(String respostaPrisioneiroA, String respostaPrisioneiroB) {

if (respostaPrisioneiroA == "Culpado") {

if (respostaPrisioneiroB == "Culpado") {

} else {

return PENA\_INOCENCIA;

}

} else {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_INDIVIDUAL;

} else {

return PENA\_CONDENACAO\_CUMPLICES;

}

}

}

}

PARTE 3 – Testes

Continuando a testar com a JUNIT, chegamos no seguinte erro:

“java.lang.Error: Unresolved compilation problem:

This method must return a result of type int

”

Podemos observar que falta um último return no código java, por isso a JUNIT não roda corretamente. Será feita a seguinte alteração:

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(String respostaPrisioneiroA, String respostaPrisioneiroB) {

if (respostaPrisioneiroA == "Culpado") {

if (respostaPrisioneiroB == "Culpado") {

} else {

return PENA\_INOCENCIA;

}

} else {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_INDIVIDUAL;

} else {

return PENA\_CONDENACAO\_CUMPLICES;

}

}

return PENA\_CONDENACAO\_MUTUA;

}

}

PARTE 4 – Testes

Rodando novamente a JUNIT temos o seguinte erro no “testeCondenacaoMutual”

“junit.framework.AssertionFailedError: expected:<5> but was:<15>”

Isso se deve ao fato de o retorno na classe java estar errado. Arrumando, temos o seguinte código java:

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(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;

}

}

}

}

PARTE 5 – TESTES

Rodando novamente a JUNIT temos o seguinte erro no “testeCondenacaoCumplices”

“junit.framework.AssertionFailedError: expected:<1> but was:<10>”

O código está diferente nesse quesito, o retorno da variável PENA\_CONDENACAO\_INDIVIDUAL é 10. Resolvi fazer a alteração na JUNIT, deixando-a dessa forma:

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

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.Test;

import junit.framework.TestCase;

import org.junit.jupiter.api.Test;

import junit.framework.TestCase;

class JUNITteste extends TestCase {

@Test

void testeCondenacaoMutua() {

String respostaPrisioneiroA = "Culpado";

String respostaPrisioneiroB = "Culpado";

JulgamentoPrisioneiro julg = new JulgamentoPrisioneiro();

int resultuadoEsperado = 5;

int resultadoReal = julg.calculaPena(respostaPrisioneiroA,respostaPrisioneiroB);

assertEquals(resultuadoEsperado, resultadoReal);

}

@Test

void testeCondenacaoIndividual() {

String respostaPrisioneiroA = "Culpado";

String respostaPrisioneiroB = "Inocente";

JulgamentoPrisioneiro julg = new JulgamentoPrisioneiro();

int resultuadoEsperado = 10;

int resultadoReal = julg.calculaPena(respostaPrisioneiroA,respostaPrisioneiroB);

assertEquals(resultuadoEsperado, resultadoReal);

}

@Test

void testeCondenacaoCumplices() {

String respostaPrisioneiroA = "Inocente";

String respostaPrisioneiroB = "Culpado";

JulgamentoPrisioneiro julg = new JulgamentoPrisioneiro();

int resultuadoEsperado = 10;

int resultadoReal = julg.calculaPena(respostaPrisioneiroA,respostaPrisioneiroB);

assertEquals(resultuadoEsperado, resultadoReal);

}

@Test

void testeInocencia() {

String respostaPrisioneiroA = "Inocente";

String respostaPrisioneiroB = "Inocente";

JulgamentoPrisioneiro julg = new JulgamentoPrisioneiro();

int resultuadoEsperado = 0;

int resultadoReal = julg.calculaPena(respostaPrisioneiroA,respostaPrisioneiroB);

assertEquals(resultuadoEsperado, resultadoReal);

}

}

PARTE 6 – TESTES NOVOS

Rodando novamente a JUNIT temos o seguinte erro no “testeCondenacaoMutua”

“junit.framework.AssertionFailedError: expected:<5> but was:<15>”

Isso se deve ao fato dos “returns” de PENA\_INOCENCIA e PENA\_CONDENACAO\_CUMPLICES estarem invertidos no arquivo Java. Arrumando o código, temos:

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(String respostaPrisioneiroA, String respostaPrisioneiroB) {

if (respostaPrisioneiroA == "Culpado") {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_MUTUA;

} else {

return PENA\_CONDENACAO\_CUMPLICES;

}

} else {

if (respostaPrisioneiroB == "Culpado") {

return PENA\_CONDENACAO\_INDIVIDUAL;

} else {

return PENA\_INOCENCIA;

}

}

}

}

PARTE 7 – TESTES NOVOS

Rodando novamente a JUNIT temos o seguinte erro no “testeCondenacaoIndividual”

“junit.framework.AssertionFailedError: expected:<10> but was:<11>”

Isso se deve ao fato dos “returns” de PENA\_CONDENACAO\_INDIVIDUAL e PENA\_CONDENACAO\_CUMPLICES estarem invertidos no arquivo Java. Arrumando o código, temos:

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(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;

}

}

}

}