**Nome: Mateus Marques Fernandes**

**Rm: 80872**

1. **Caso de teste da Pena do Inocente**

**class** JulgamentoPrisioneiroTest {

@Test

**public** **void** testeCalcularPenaInocente(){

JulgamentoPrisioneiro prisioneiro = **new** JulgamentoPrisioneiro();

**int** pena = prisioneiro.calculaPena("Inocente", "Inocente");

*assertEquals*(0, pena);

}

}

**Resultado do teste:**

- java.lang.Error: Unresolved compilation problem.

- Erro de compilação pela falta da Classe Resposta, que não consta no algoritmo onde a mesma tem que ser substituída por ‘String’.

1. **Caso de teste da Pena Mutua**

@Test

**public** **void** testeCalcularPenaMutua() {

JulgamentoPrisioneiro prisioneiro = **new** JulgamentoPrisioneiro();

**int** pena = prisioneiro.calculaPena("Culpado", "Culpado");

*assertEquals*(5, pena);

}

**Resultado do teste:**

- org.opentest4j.AssertionFailedError: expected: <5> but was: <15>

- Erro devido valores das variáveis de retorno estarem diferente do algoritmo as mesmas foram ajustada.

1. **Caso de teste da Pena Individual**

@Test

**void** testeCalcularPenaIndividual() {

JulgamentoPrisioneiro prisioneiro = **new** JulgamentoPrisioneiro();

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

*assertEquals*(10, pena);

}

**Resultado do teste:**

**-** Teste realizado com sucesso.

1. **Caso de teste da Pena Cumplices**

@Test

**void** testeCalcularPenaCumplices() {

JulgamentoPrisioneiro prisioneiro = **new** JulgamentoPrisioneiro();

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

*assertEquals*(1, pena);

}

**Resultado do teste:**

**-** Teste realizado com sucesso.

**Versão final do JUNIT**

**class** JulgamentoPrisioneiroTest {

@Test

**public** **void** testeCalcularPenaInocente(){

JulgamentoPrisioneiro prisioneiro = **new** JulgamentoPrisioneiro();

**int** pena = prisioneiro.calculaPena("Inocente", "Inocente");

*assertEquals*(0, pena);

}

@Test

**public** **void** testeCalcularPenaMutua() {

JulgamentoPrisioneiro prisioneiro = **new** JulgamentoPrisioneiro();

**int** pena = prisioneiro.calculaPena("Culpado", "Culpado");

*assertEquals*(5, pena);

}

@Test

**void** testeCalcularPenaIndividual() {

JulgamentoPrisioneiro prisioneiro = **new** JulgamentoPrisioneiro();

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

*assertEquals*(10, pena);

}

@Test

**void** testeCalcularPenaCumplices() {

JulgamentoPrisioneiro prisioneiro = **new** JulgamentoPrisioneiro();

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

*assertEquals*(1, pena);

}

}

**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 == "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;

}

}

}

}