Documentação de Teste JUnit

Gabriel Bertocco Biancardi

RM: 83607

Classe original:

Classe JulgamentoPrisioneiro {

int PENA\_INOCENCIA = 0;

int PENA\_CONDENACAO\_MUTUA = 5;

int PENA\_CONDENACAO\_INDIVIDUAL = 10;

int PENA\_CONDENACAO\_CUMPLICES = 1;

int calculaPena(String respostaPrisioneiroA, String respostaPrisioneiroB) {

se (respostaPrisioneiroA = “Culpado”) {

se (respostaPrisioneiroB = “Culpado”) {

retorna PENA\_CONDENACAO\_MUTUA;

} senao {

retorna PENA\_CONDENACAO\_INDIVIDUAL;

}

} senao {

se (respostaPrisioneiroB = “Culpado”) {

retorna PENA\_CONDENACAO\_CUMPLICES;

} senao {

retorna PENA\_INOCENCIA;

}

}

}

}

Classe a ser testada:

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;

}

}

}

}

1. Primeiro erro encontrado:

Faltou o enum Resposta para que o método calcula possa funcionar corretamente:

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;

}

}

}

}

1. Classe de teste criada e método para validação criada:

@Test

**void** testaInocenciaIndividual() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

DilemaPrisioneiro dp = **new** DilemaPrisioneiro();

Assert.*assertNotNull*(dp);

**int** penaSuspeitoA = dp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

**int** penaSuspeitoB = dp.calculaPena(respostaSuspeitoB, respostaSuspeitoA);

Assert.*assertEquals*(0, penaSuspeitoA);

Assert.*assertEquals*(10, penaSuspeitoB);

}

O teste valida o caso do suspeito A ser inocente e o B receber condenação individual. Como visto, o teste aponta que o número para o suspeito está errado, com o erro java.lang.AssertionError: expected:<0> but was:<13>, ou seja, recebeu o valor 13, porém o correto seria ter o valor 0.

Classe corrigida:

public class JulgamentoPrisioneiro {

private int PENA\_INOCENCIA = 0;

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;

}

}

}

}

1. Segunda rodada de testes

@Test

**void** testaMutua() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

Resposta respostaSuspeitoB = Resposta.***DELACAO***;

DilemaPrisioneiro dp = **new** DilemaPrisioneiro();

Assert.*assertNotNull*(dp);

**int** penaSuspeitoA = dp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

Assert.*assertEquals*(5, penaSuspeitoA);

}

O teste valida o caso do suspeito A e o B recebem condenação mútua. Como visto, o teste aponta que o número para o suspeito está errado, com o erro java.lang.AssertionError: expected:<5> but was:<15>, ou seja, recebeu o valor 13, porém o correto seria ter o valor 0.

Classe 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 = 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;

}

}

}

}

1. Testando Pena Cumplices:

@Test

**void** testaCumplices() {

Resposta respostaSuspeitoA = Resposta.***NEGACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

DilemaPrisioneiro dp = **new** DilemaPrisioneiro();

Assert.*assertNotNull*(dp);

**int** penaSuspeitoA = dp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

Assert.*assertEquals*(11, penaSuspeitoA);

}

O teste valida o caso do suspeito A e o B recebem condenação cumplices. Como visto, o teste aponta que o número está correto, não apontando nenhum erro.

Testes finalizados

Classe final:

**package** br.com.fiap.nac;

**public** **class** DilemaPrisioneiro {

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

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

**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;

}

}

}

}

Código JUnit final:

**class** TesteDilema {

@Test

**void** testaInocenciaIndividual() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

DilemaPrisioneiro dp = **new** DilemaPrisioneiro();

Assert.*assertNotNull*(dp);

**int** penaSuspeitoA = dp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

**int** penaSuspeitoB = dp.calculaPena(respostaSuspeitoB, respostaSuspeitoA);

Assert.*assertEquals*(0, penaSuspeitoA);

Assert.*assertEquals*(10, penaSuspeitoB);

}

@Test

**void** testaMutua() {

Resposta respostaSuspeitoA = Resposta.***DELACAO***;

Resposta respostaSuspeitoB = Resposta.***DELACAO***;

DilemaPrisioneiro dp = **new** DilemaPrisioneiro();

Assert.*assertNotNull*(dp);

**int** penaSuspeitoA = dp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

Assert.*assertEquals*(5, penaSuspeitoA);

}

@Test

**void** testaCumplices() {

Resposta respostaSuspeitoA = Resposta.***NEGACAO***;

Resposta respostaSuspeitoB = Resposta.***NEGACAO***;

DilemaPrisioneiro dp = **new** DilemaPrisioneiro();

Assert.*assertNotNull*(dp);

**int** penaSuspeitoA = dp.calculaPena(respostaSuspeitoA, respostaSuspeitoB);

Assert.*assertEquals*(11, penaSuspeitoA);

}

}