

Relatório de Testes com JUnit

Rafael N. Copstein, Victor S. Melo

Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

{rafael.copstein, victor.melo.001}@acad.pucrs.br

Resumo. *O seguinte relatório apresenta os defeitos encontrados durante a execução dos testes unitários com JUnit. Em cada defeito é mostrada a falha observada, o nome do teste que a detectou e de que maneira foi corrigido.*

1. Relatório

1.1. Defeito 1

No primeiro defeito, o teste rodou 80%. A falha foi encontrada no teste *getChange()*, retornando a exceção *OutOfMemoryException* no método *Java.Util.Arrays.copyOf()*.

O problema foi corrigido no método *getChange()*. Este não estava decrementando o valor de troco à medida em que moedas eram devolvidas e nem removendo as moedas da máquina. Ou seja, devolvia moedas infinitamente até estourar a memória do vetor que armazena as moedas devolvidas.

1.2. Defeito 2

No segundo defeito, o teste rodou 80%. O *getChange()* falhou com as asserções, que esperavam que o valor em moedas devolvida fosse exatamente igual ao troco que deveria ser devolvido. Porém, isto ocorre apenas quando a máquina possui as moedas necessárias para devolver este troco, caso contrário, não devolve.

O problema foi resolvido com a modificação do teste para garantir que o troco seja menor (no caso de moedas insuficientes) ou igual ao valor que deve ser devolvido.

1.3. Defeito 3

No terceiro defeito, o teste rodou 83,33%. O problema foi detectado no teste *addTest()*, onde testamos os casos em que adicionamos mais moedas do que o limite permitido e o método deve jogar uma exceção.

Porém, ao inserirmos uma moeda além do limite, a exceção não estava sendo jogada. O problema ocorria pois estávamos esquecendo de alterar o valor da variável quantidade *qtd*, o que fazia com que o teste do limite nunca retornasse falso e, logo, permitisse a entrada de mais moedas do que o limite permite.