INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA UNIVERSIDADE DE SÃO PAULO

TRABALHO DE CONCLUSÃO DE CURSO - TESTE DE SOFTWARE EM JAVA - PROGRAMA DE VERÃO IME-USP

Aluno: Luciano Augusto Campagnoli da Silva
Docente: Alexandre Locci Martins

São Paulo
de janeiro de 2020

Introdução

O documento em questão se refere ao projeto de conclusão do curso **Teste de Programa em Java**, ministrado pelo Professor Alexandre Locci Martins no **XLIX Programa de Verão IME-USP**. O projeto foi desenvolvido na linguagem **Java**, utilizando a **Eclipse IDE**, hospedado em um repositório remoto do **GitHub**. Além disso, foi submetido a testes unitários elaborados com o *framework* **JUnit**. Também analisou-se a cobertura dos casos de testes unitários com o *plug-in* **Eclemma**.

Descrição do projeto e casos de teste

Ele implementa as funcionalidades especificadas em documento fornecido no portal da disciplina. A primeira delas consiste na ordenação de um arranjo de códigos numéricos; a segunda retorna o primeiro nome de um funcionário dado o seu nome completo; e a terceira reproduz uma calculadora com operações aritméticas básicas. Deve-se observar que a saída da funcionalidade de calculadora é um dado no tipo ***String***, e não em um tipo numérico. Essa implementação é explicada pelo retorno de mensagens em linguagem natural quando a entrada não era fornecida no formato esperado. Portanto, para facilitar a elaboração do código, mesmo que a operação fosse realizada com sucesso, sua saída era exportada pela função como uma cadeia de caracteres, permitindo a criação de uma única função que contemplasse todos os casos esperados e inesperados.

O desenvolvimento da aplicação seguiu os princípios do padrão **Desenvolvimento Guiado por Testes** (em inglês, ***Test Driven Development***, ou **TDD**), que determina que se elabore primeiramente os casos de teste unitários e, baseando-se neles, seja realizada a implementação posteriormente. A malha de teste (em outras palavras, o conjunto de casos de teste utilizados) é explicada e detalhada a seguir, separada segundo as funcionalidades do programa.

Funcionalidade para ordenar códigos de *array*

Entrada	Saída Esperada	Motivo para caso de teste
{89, 45, 61, 22}	{22, 45, 61, 89}	*Array* de tamanho **n > 1**
{22}	{22}	*Array* de tamanho **n = 1** (singularidade)
{}	{}	*Array* de tamanho **n = 0** (singularidade)

Tabela 1: malha de testes para a funcionalidade de ordenação de códigos.

Funcionalidade para retirar primeiro nome de nome completo

Entrada	Saída Esperada	Motivo para caso de teste
Luciano Augusto Campagnoli da Silva	Luciano	Nome completo com mais de uma palavra
Ronaldo	Ronaldo	Nome de apenas uma palavra (singularidade)
	Insira um nome!!	Nome vazio (singularidade)

Tabela 2: malha de testes para a funcionalidade de extração de primeiro nome.

Funcionalidade de calculadora

Entrada	Saída Esperada	Motivo para caso de teste
<2, +, 1>	3	Operação de adição simples
<7, -, 2>	5	Operação de subtração simples
<9, *, 8>	72	Operação de multiplicação simples
<32, /, 8>	4	Operação de divisão simples
<5, /, 2>	2.5	Operação de divisão com resultado não inteiro
<32, /, 0>	Erro, não se pode dividir por 0!!	Divisão por 0 (singularidade)
<32, (, 2>	Operação inválida. Tente novamente!	Caractere não reconhecido para operações (singularidade)

Tabela 3: malha de testes para a função de calculadora.

Com os casos de teste acima definidos, foi verificada a cobertura de código por eles realizada. Os resultados são expostos na tabela abaixo:

Medida de cobertura	Cobertos(as)	Não cobertos(as)	Total	Cobertura (%)
Instruções	239	0	239	100
Ramos	14	0	14	100
Linhas	49	0	49	100
Métodos	12	0	12	100
Tipos	2	0	2	100
Complexidade	19	0	19	100

Tabela 4: Cobertura de código realizada com ***Eclemma***.

Conclusão

Com a realização do curso e as atividades propostas, foi possível adquirir conhecimento da área de **Teste de *Software***, uma das mais importantes do desenvolvimento de atividades em computação. O aluno pôde adquirir não apenas os conhecimentos técnicos necessários e de manuseio das ferramentas utilizadas, mas também aprendeu sobre o padrão de projeto **TDD**, que coloca os testes como diretriz do desenvolvimento de *software* e permite sua realização com mais qualidade e de forma mais rápida e eficiente.