



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA - PARAÍBA

CAMPUS CAMPINA GRANDE

DISCIPLINA: TESTE DE SOFTWARE

PROFESSORA: MIRNA CARELLI OLIVEIRA MAIA

Relatório de Testes

Antonio Roberto P. de Lima Jr Matrícula: 202021250033

Aryelson Gonçalves Messias Matrícula: 202021250045

Marcelo Ribeiro da Silva Matrícula: 202021250020



Campina Grande, Dezembro de 2022.

Neste presente relatório nós optamos pela utilização da framework Java JUnit, por ser simples o bastante para testadores iniciantes, além da praticidade das ferramentas fornecidas pela IDE Eclipse, que permitem acompanhar facilmente a cobertura do código, além de checar quais casos de teste apresentaram erro e a comparação de qual o valor esperava-se e o valor real. Abaixo forneço o link dos scripts utilizados para a realização dos testes de unidade.

Link do script na plataforma Replit:

<https://replit.com/@AntonioJr3/Scripttesteequipe#sistemaJava.java>

Cobertura do código:

▼ Sistema.java	 63,0 %	415	244	659
> Sistema	 63,0 %	415	244	659

A cobertura do código foi abaixo do esperado por motivos que estarão melhores descritos ao decorrer deste relatório.

1. Informação do Plano:

O plano de testes foi realizado previamente com o intuito de realizar uma análise sob a interface com o usuário, ou seja, um teste caixa preta. O objetivo foi de primeiramente utilizar casos de teste em que seria obtido êxito para as funcionalidades, após isso visou-se realizar a análise dos valores limite de tais funcionalidades, de tal maneira a explorar os pontos especificados no documento de criação do sistema, no entanto isso não foi possível, por motivos que serão explorados após a representação dos dados de exposição abaixo mostrados:

Dados de execução:

- Total Planejado: 100% (104)
- Total executado: 42.31% (39)
- Total êxito: 66.66% (26)
- Total Falha: 33.33% (13)

Com os dados acima pode-se concluir que apenas cerca de 37.5% dos casos de teste planejados foram realizados e isso se deu pois a interface com o usuário estava inoperante. Após a criação do usuário, passo necessário para acessar o sistema, não é possível avançar, pois ao realizar o login o sistema apresenta o erro “Exception in thread "main" java.lang.StackOverflowError”. Com isso, a abordagem teve que ser alterada, de um teste caixa preta para um teste caixa branca, sendo assim, um grande prejuízo de perda de casos de teste se sucedeu.

Com o tipo de teste que se procedeu não foi possível realizar o teste de entradas indevidas, já que ao ir diretamente aos métodos se faz necessário inserir os tipos corretos. Com isso restaram todos os testes que tinham como objetivos testar os valores limites, como valores negativos para algumas operações e o valor 0 em outras.

2. Lista de requisição de mudanças(seguinte lista dos casos de teste caso haja dúvida):

RF1 - Pelo fato de não ser possível testar através de um teste caixa preta não é verificável se as exceções estão de acordo com as especificações do documento. Sugiro refazer o código de interface por inteiro e realizar a busca pela solução do erro, ou realizar as verificações dentro da classe “Sistema.java”.

RF2 - A função não testa para $A = 0$, um dos requisitos do documento do sistema. Sugiro colocar esta condição e também mudar o return, pois ao retornar null irá causar erro.

RF3 - Funciona bem. Sugiro usar menos condicionais, apenas testar se um dos ângulos é menor ou igual a zero e se a soma dos três é maior que 180° é o bastante.

RF4 - Este método está completamente incorreto, ele checka apenas se o triângulo é retângulo, ao invés de checkar se é acutângulo, obtusângulo e retângulo, além disso a equação está incorreta, o método está usando o teorema de pitágoras, quando na verdade deveria estar somente checkando os ângulos. Sugiro refazer por completo.

RF5 - Este método está fazendo o que o anterior deveria, que é classificar triângulos, e seria aqui onde deveria-se usar pitágoras, além disso os valores usados nas condicionais desse método são para ângulos.

RF6 - Este método está faltando verificar se a soma dos ângulos é igual a 360 e se eles são negativos, apenas checka se são iguais, independente do ângulo e do sinal. Sugiro adicionar essas condicionais e remover as condicionais redundantes.

RF7a - Método mudou assinatura exigida no documento do sistema, deve ser refeito. Tenta somar apenas dois lados, porém um retângulo possui quatro.

RF7b - Método correto.

RF8 - Foi criada uma variável global para realizar as operações, desta forma quando a calculadora não é capaz de realizar uma nova operação ela puramente retorna o resultado da operação anterior. Sugiro tornar a variável local.

RF9a - Método correto. Sugiro usar Math.pi para ter um valor exato do pi, ao invés de um valor aproximado.

RF9b - Método correto. Sugiro usar Math.pi para ter um valor exato do pi, ao invés de um valor aproximado.

RF10 - Método não aceita coordenadas negativas. Sugiro aceitar essas coordenadas, além disso, o método está correto.

3. Contatos:

Antonio Roberto Paulino de Lima Jr.

→ Telefone: +55 83 9908 - 4649

→ Email: antonio.roberto@academico.ifpb.edu.br

Aryelson Gonçalves Messias.

→ Telefone: +55 83 9986 - 3202

→ Email: goncalves.aryelson@academico.ifpb.edu.br

Marcelo Ribeiro da Silva.

→ Telefone: +55 83 8859 - 2904

→ Email: ribeiro.silva@academico.ifpb.edu.br

Todos os integrantes testaram as funcionalidades juntos.