

Relatório Trabalho Prático de Teste Baseado em Especificação

Validação do método `Main.cal(int month, int year)`

Breno Farias da Silva

Sumário

Introdução	1
Casos de Teste Implementados	1
Detecção de Mutantes e Score de Mutação	1
Mutantes Equivalentes	2
Considerações Finais	2

Introdução

Este relatório descreve o processo de teste aplicado ao método `Main.cal(int month, int year)`, responsável por gerar uma representação textual do calendário de um determinado mês e ano. O objetivo principal foi aplicar testes automatizados até alcançar o maior score de mutação possível, utilizando a ferramenta PITest.

Casos de Teste Implementados

Ao todo, foram implementados **20 casos de teste** com o framework JUnit 5, abrangendo diferentes cenários como anos bissextos, meses com 30 ou 31 dias, alinhamento de datas, validação dos limites de mês, verificação da existência de dias específicos (como o dia 29 em fevereiro), e checagem do cabeçalho do mês.

Esses testes visam garantir que o método respeite as regras do calendário gregoriano e que o alinhamento visual do texto esteja consistente.

Note

Durante a elaboração deste relatório, os métodos de teste relacionados aos limites mínimos e máximos dos parâmetros `months` e `year` foram removidos. Isso ocorreu porque a função `cal`, alvo dos testes, não realiza o tratamento adequado para esses valores extremos. Como resultado, a execução dos testes gerava erros que impediam a compilação do projeto e a geração deste relatório.

Detecção de Mutantes e Score de Mutação

Após a execução da suíte de testes com a ferramenta PIT, obteve-se o seguinte resultado:

- **Mutantes gerados:** 150
- **Mutantes mortos:** 56
- **Mutantes vivos (não detectados pelos testes):** 54
- **Mutantes sem cobertura (nenhum teste passou por eles):** 40

Com esses dados, temos as seguintes métricas:

- **Cobertura de linha:** 64% (29 de 45 linhas)
- **Força dos testes (mutation coverage):** 37%
- **Test strength (proporção de mutantes mortos sobre os cobertos):** 51%
- **Total de testes executados:** 1252 (8,35 testes por mutante em média)

Mutantes Equivalentes

Durante a análise dos resultados fornecidos pelo PIT, observou-se que diversos mutantes vivos não representavam comportamentos semanticamente distintos do original — isto é, produziam a mesma saída mesmo após a mutação. Estes são os chamados **mutantes equivalentes**.

Foram identificados manualmente **11 mutantes equivalentes**, o que justifica parcialmente a limitação do score final.

Considerações Finais

Apesar do esforço de implementação de testes variados e bem distribuídos, o score de mutação alcançado permaneceu abaixo de 50%, indicando a existência de pontos cegos na suíte de testes.

As principais limitações observadas foram:

- A lógica de cálculo do primeiro dia do mês e o alinhamento dos dias na saída impressa são sensíveis a detalhes difíceis de capturar em testes automatizados apenas com `contains` ou `startsWith`.
- Alguns mutantes vivos só poderiam ser pegos com testes que avaliem precisamente o espaçamento e a posição dos caracteres na string retornada.

A principal recomendação para trabalhos futuros é a implementação de verificações baseadas em `regex` ou análises de string linha a linha para validar o posicionamento exato dos dias, aumentando assim a sensibilidade dos testes e a taxa de detecção de mutantes.