Sistemas de Informações

Júlio Geovani Oliveira Guimarães

Atividade 2 – Testes de Mutação

Ufs (SC) - 02/09/2024

Repostório: https://github.com/valdas-v1/A-Simple-Python-Calculator/tree/main

Tecnologias utilizadas

Sistema Operacional: Windowns

IDE: Visual Studio Code

Ferramentas: Venv(Ambiente Virtual), pytest, pytest-cov e mutmut.

1. **Configurações feitas no ambiente:**

- **Em primeiro, um clone no repositório:**

  Código: "git clone https://github.com/valdas-v1/A-Simple-Python-Calculator.git "

- **Em segundo, a instalação do venv:**

  Código: "python -m venv venv"
  "python3 -m venv meu-diretório\venv"

- **Em terceiro, a Ativação do venv (Ambiente Virtual):**

  Código: "venv/Scripts/activate"

- **Em quarto, a instalação dos pacotes já disponíveis no repositório:**

Código: “pip3 install -r requirements.txt”

Obs: Adição dos pacotes pytest-cov e mutmut ao requirements.txt.

2. Execução dos testes:

- **Execução do arquivo de teste utilizando o pytest:**

Código: “pytest -vv meu-diretório\tests\test_calculator.py”

Métodos executados: test_sum(), test_subtract(), test_multiply(), test_division(), test_square_root(), teste_invalide_value(), test_memory.

Resultado: Todos os testes foram realizados e todos passaram corretamente.

```
A-Simple-Python-Calculator\tests\test_calculator.py::test_sum PASSED                [ 14%]
A-Simple-Python-Calculator\tests\test_calculator.py::test_subtract PASSED           [ 28%]
A-Simple-Python-Calculator\tests\test_calculator.py::test_multiply PASSED           [ 42%]
A-Simple-Python-Calculator\tests\test_calculator.py::test_division PASSED           [ 57%]
A-Simple-Python-Calculator\tests\test_calculator.py::test_square_root PASSED        [ 71%]
A-Simple-Python-Calculator\tests\test_calculator.py::test_invalid_value PASSED      [ 85%]
A-Simple-Python-Calculator\tests\test_calculator.py::test_memory PASSED             [100%]

====================================== 7 passed in 0.06s =======================================
```

- **Execução do teste de cobertura:**

  Código: “pytest -vv meu-diretório\tests\test_calculator.py --cov=calculator”

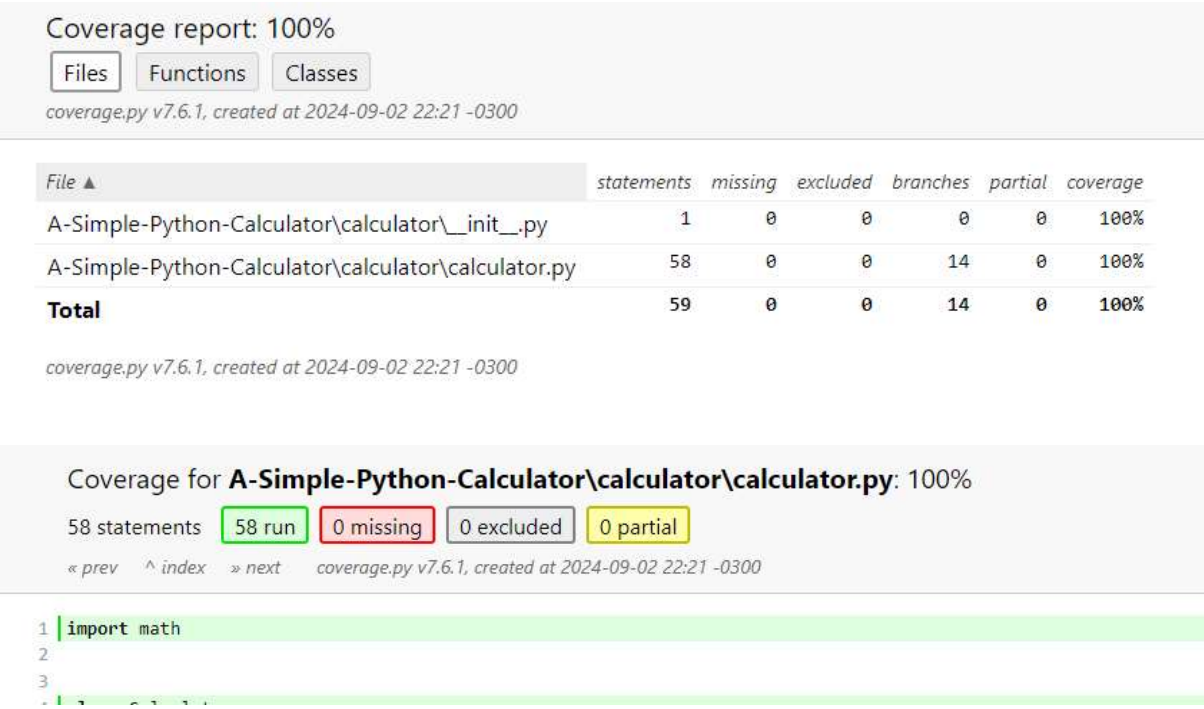
```
---------- coverage: platform win32, python 3.12.5-final-0 -----------
Name                                                    Stmts   Miss  Cover
---------------------------------------------------------------------------
A-Simple-Python-Calculator\calculator\__init__.py           1      0   100%
A-Simple-Python-Calculator\calculator\calculator.py        58      0   100%
---------------------------------------------------------------------------
TOTAL                                                      59      0   100%


======================================== 7 passed in 0.08s ========================================
```

Resultado: Todos os testes foram realizados e recebi o retorno que 100% do código está coberto de testes.

- **Gerando html com as informações do teste de cobertura:**

  Código: "pytest -vv meu-diretório\tests\test_calculator.py --cov=calculator --cov-branch --cov-report html"

Informações extraídas com a visualização na página web:



Coverage report: 100%

Files | Functions | Classes

coverage.py v7.6.1, created at 2024-09-02 22:21 -0300

File ▲	statements	missing	excluded	branches	partial	coverage
A-Simple-Python-Calculator\calculator__init__.py	1	0	0	0	0	100%
A-Simple-Python-Calculator\calculator\calculator.py	58	0	0	14	0	100%
Total	59	0	0	14	0	100%

coverage.py v7.6.1, created at 2024-09-02 22:21 -0300

Coverage for **A-Simple-Python-Calculator\calculator\calculator.py**: 100%

58 statements | 58 run | 0 missing | 0 excluded | 0 partial

« prev ^ index » next coverage.py v7.6.1, created at 2024-09-02 22:21 -0300

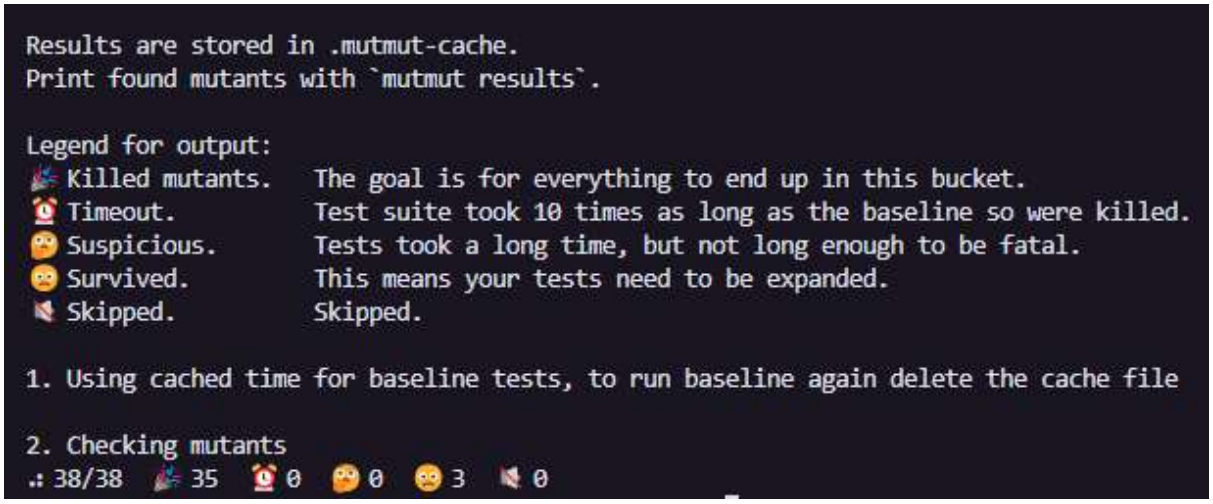
```
1 import math
2
3
```

- Relatório de cobertura em 100%, ou seja, todas as linhas de código foram devidamente testadas.

- **Obs: 58 linhas de código foram executadas, 0 ausentes, 0 excluídas e 0 parciais. Ou seja, êxito em 100% em relação aos testes de cobertura!**

- **Execução do teste de mutação:**

  Código: “mutmut run --paths-to-mutate=meu-diretório\calculator\calculator.py --tests-dir=meu-diretório\tests”



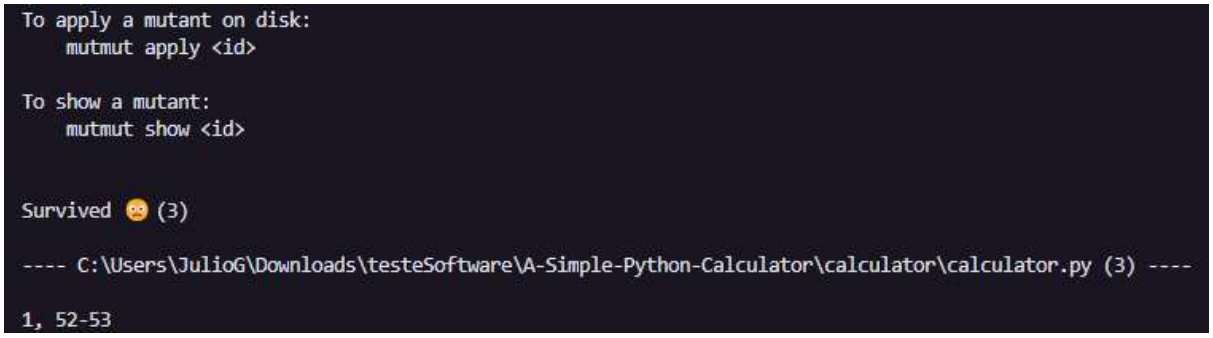
Resultados:

- **38 mutants foram inseridos no código.**
- **35 foram mortos.**
- **Nenhum teve o tempo expirado.**
- **Nenhum suspeito.**
- **3 sobreviveram.**
- **Nenhum foi ignorado.**

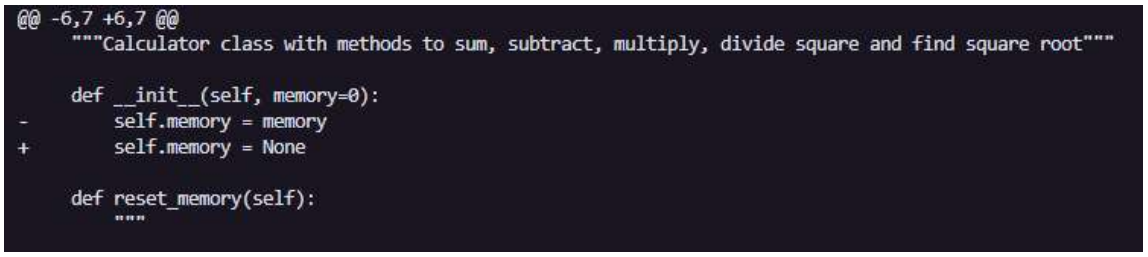
Conclusão: De forma geral, os casos de teste se comportaram bem em relação ao teste de mutação, eliminaram praticamente todos os 38 mutants e apenas 3 sobreviveram.

- **Analisando mutants que sobreviveram:**

- Imprimindo os mutantes: Código: “mutmut results”



- Checando mais informações sobre o mutant de acordo com seu id: Código: “mutmut show 1”



Resultado: No caso deste teste de mutação, foi passado para o self.memory o valor None em vez do 0 inicialmente. Como se trata de uma calculadora com métodos para execução de operações aritméticas entre números, valores nulos podem causar erros de execução. No código não existe uma tratativa na classe Calculator que trata None da mesma forma que trata 0 ou outros valores iniciais.

Forma de melhoria:

Garantir que a classe `Calculator` trate `None` da mesma forma

que trata outros valores iniciais.

Código que o teste foi realizado:

```
def __init__(self, memory=0):
    self.memory = 0

def reset_memory(self):
    """
    Resets the memory back to zero
    """

    self.memory = 0
```

Código com a alteração

```
def __init__(self, memory=0):
    self.memory = memory

def reset_memory(self):
    """
    Resets the memory back to zero
    """

    self.memory = 0
```

Alteração de melhoria: Fazer com que self.memory receba o valor 0, do memory já iniciado no init.

Resultado gerado com a alteração:

```
🎉 Killed mutants.   The goal is for everything to end up in this bucket.
⏰ Timeout.          Test suite took 10 times as long as the baseline so were killed.
🤔 Suspicious.       Tests took a long time, but not long enough to be fatal.
🙁 Survived.         This means your tests need to be expanded.
🔇 Skipped.          Skipped.

1. Using cached time for baseline tests, to run baseline again delete the cache file

2. Checking mutants
.: 37/37  🎉 35  ⏰ 0  🤔 0  🙁 2  🔇 0
```

Resultado: Um dos 3 mutants foi eliminado, gerando assim um melhor resultado em relação aos testes de mutação.