Tópicos Especiais - Trabalho Prático 1

Aluno: Paulo André Carneiro Fernandes

Descrição do programa

O programa recebe como entrada no mínimo dois e no máximo cinco valores inteiros de 0 a 1000, a partir disso, o algoritmo define o maior e o menor valor entre os números inseridos pelo usuário. Caso algum dos requisitos não sejam cumpridos, o programa exibirá uma mensagem indicando que a entrada é inválida e o motivo da entrada ser inválida, após isso o programa encerra sua execução.

Testes a partir da técnica funcional

- Realizando os testes utilizando os critérios de partição de equivalência e análise do valor limite:

- Testes com tipos de entradas inválidas:

Caso de teste	Entrada	Saída esperada
CT01	a, -1	Entrada inválida
CT02	b	Entrada inválida
CT03		Entrada inválida

- Testando os limites inferiores dos valores 0 e 1000:

Caso de teste	Entrada	Saída esperada
CT04	-2, -1	Entrada inválida
CT05	0, 1	O maior valor é 1 e o menor é 0
CT06	1, 2	O maior valor é 2 e o menor é 1

- Testando os limites superiores dos valores 0 e 1000:

Caso de teste	Entrada	Saída esperada
CT07	1001, 1002	Entrada inválida
CT08	999, 1000	O maior valor é 1000 e o menor é 999
CT09	998, 999	O maior valor é 999 e o menor é 998

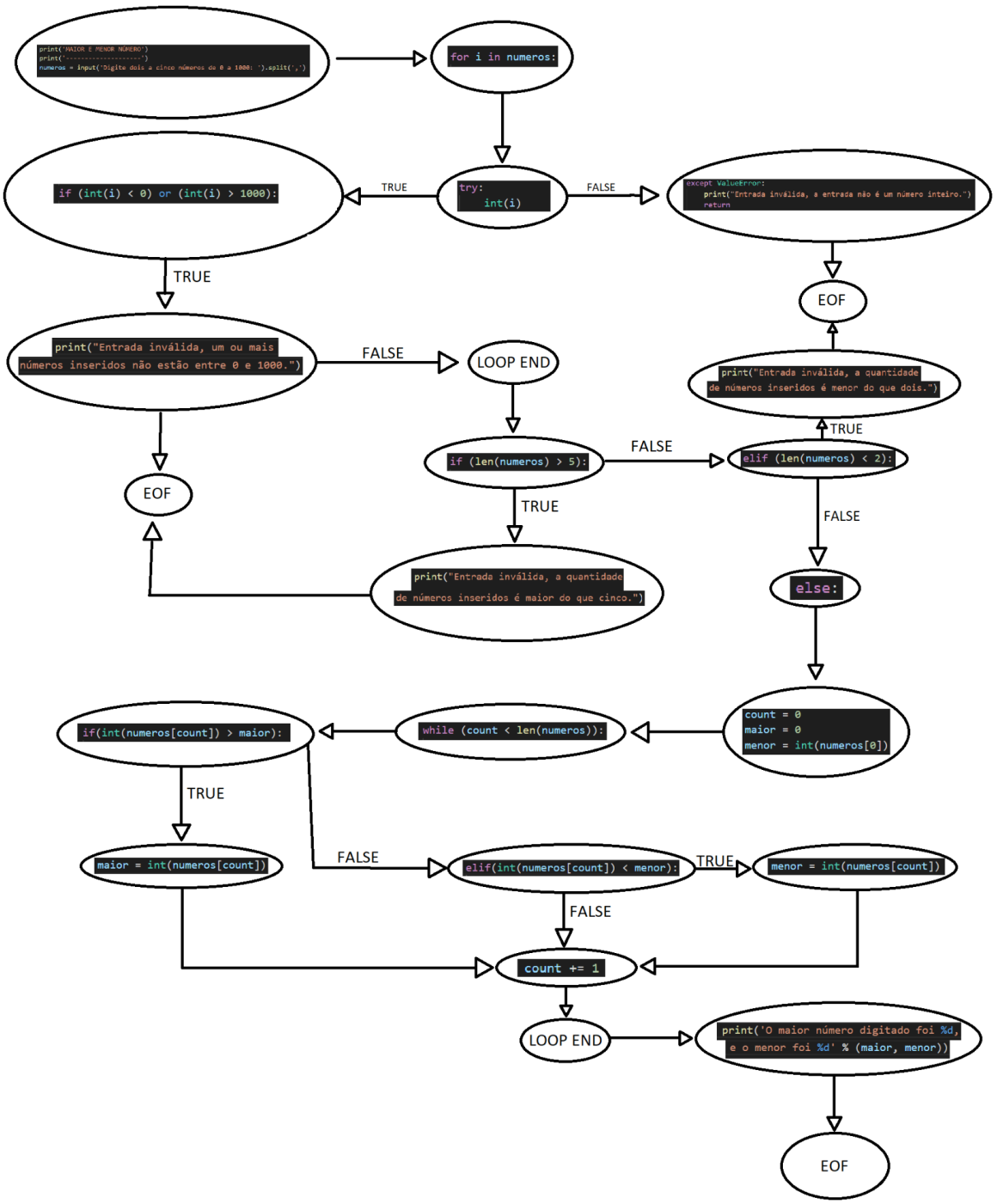
- Testando os limites inferiores da quantidade de valores:

Caso de teste	Entrada	Saída esperada
CT10	0	Entrada inválida
CT11	0, 1	O maior valor é 1 e o menor é 0
CT12	0, 1, 2	O maior valor é 2 e o menor é 0

- Testando os limites superiores da quantidade de valores:

Caso de teste	Entrada	Saída esperada
CT13	0, 1, 2, 3, 4, 5	Entrada inválida
CT14	0, 1, 2, 3, 4	O maior valor é 4 e o menor é 0
CT15	0, 1, 2, 3	O maior valor é 3 e o menor é 0

Grafo do Fluxo de Controle



A imagem também está no classroom e no repositório para melhor visualização.

Teste Unitário

Devido ao fato de o teste unitário ser realizado de forma automática, para fins de simplificação, o escopo do programa apenas nessa etapa será reduzido para o recebimento de três números como parâmetro da função, em vez de receber de dois a cinco valores como foi feito anteriormente, as demais partes do programa serão mantidas.

```
# -*- coding: utf-8 -*-
import py_compile
import unittest
import Maior_e_Menor_Tested

class test_program (unittest.TestCase):

    def test_ct01(self):
        self.assertEqual(Maior_e_Menor_Tested.maior_e_menor('a', 'b',
'c'), "Entrada inválida, a entrada não é um número inteiro.")

    def test_ct02(self):
        self.assertEqual(Maior_e_Menor_Tested.maior_e_menor('a', 1, 2),
"Entrada inválida, a entrada não é um número inteiro.")

    def test_ct03(self):
        self.assertEqual(Maior_e_Menor_Tested.maior_e_menor('a', 1,
'c'), "Entrada inválida, a entrada não é um número inteiro.")

    def test_ct04(self):
        self.assertEqual(Maior_e_Menor_Tested.maior_e_menor(999, 1000,
1001), "Entrada inválida, um ou mais números inseridos não estão entre
0 e 1000.")

    def test_ct05(self):
        self.assertEqual(Maior_e_Menor_Tested.maior_e_menor(-1, 0, 1),
"Entrada inválida, um ou mais números inseridos não estão entre 0 e
1000.")

    def test_ct06(self):
        self.assertEqual(Maior_e_Menor_Tested.maior_e_menor(0, 1, 2),
'O maior número digitado foi 2, e o menor foi 0')

    def test_ct07(self):
        self.assertEqual(Maior_e_Menor_Tested.maior_e_menor(998, 999,
1000), 'O maior número digitado foi 1000, e o menor foi 998')
```

```
    def test_ct08(self):
        self.assertEqual(Maior_e_Menor_Tested.maior_e_menor(462, 943,
738), 'O maior número digitado foi 943, e o menor foi 462')

if __name__ == '__main__':
    unittest.main()
```

```
........
----------------------------------------------------------------------
Ran 8 tests in 0.001s

OK
PS C:\Users\paulo\Downloads>
```