

Algoritmos e Programação I

Módulo 5: documentação e testes de programa

Prof. Dr. Said Sadique Adi



Programas em Python

- Um **programa** em Python é um arquivo com extensão .py que contém todas as instruções e estruturas de programação necessárias para a sua execução.
- Um programa em Python pode ser executado no terminal utilizando o seguinte comando

```
python3 <nome_do_programa>.py
```

Documentação de programas

- Documentar um programa consiste em inserir textos explicativos dentro dele, que não irão interferir em sua execução, de modo a:
 - facilitar a compreensão do programa pelo próprio autor do programa e por terceiros;
 - facilitar a manutenção do programa;
 - facilitar a busca e eliminação de erros no programa.

Documentação de programas

- Aos textos explicativos inseridos dentro de um programa damos o nome de **comentários**.
- Em Python, comentários podem ser inseridos de duas formas distintas:
 - Através do uso do caractere # (comentário de uma linha).
 - Tudo que vier após o # será ignorado pelo interpretador.
 - Através do uso de três aspas simples ou duplas (comentário de várias linhas);
 - Tudo que vier entre dois trios de aspas simples ou duplas será ignorado pelo interpretador.

Exemplo de documentação

"Programa que determina se um inteiro positivo é primo ou não, terminando sua execução quando um 0 for digitado"

```
numero = int(input("Digite um número positivo: "))
while(numero != 0):
    if numero % 2 == 0: # verifica se o número é par
        print("O número", numero, "não é primo.")
    else:
        contador = 0;
        for i in range(1, numero + 1, 1): # laço para contar quantos divisores tem o número
            if numero % i == 0:
                contador = contador + 1
        if contador == 2:
            print("O número", numero, "é primo.")
```

Ainda sobre documentação

- As funções providas pelo Python possuem uma documentação própria (**docstrings**) que podem ser acessadas via o comando:

`help(<nome_da_função_Python>).`

- Para funções definidas pelo próprio programador, um docstring pode ser criado adicionando um comentário logo após a linha de definição da função entre aspas simples.

Teste de programas

- Testar um programa corresponde à tarefa de verificar se o programa está correto, ou seja, verificar se ele não possui erros; esses erros podem ser:
 - Erros de sintaxe: usualmente indicam que você escreveu algo fora da regra da linguagem (ex. não colocar : ao final de uma linha que se iniciou com **def**);
 - Erros de execução: ocorrem durante a execução do programa e indicam que você cometeu um erro de lógica no seu programa (usar uma variável ou função que não foi definida);
 - Erros de semântica: o programa é executado do começo ao fim, mas não produz o resultado desejado.

Exemplos dos tipos de erros

```

numero = int(input("Digite um número positivo: "))
while(numero != 0):
    if numero % 2 == 0 ← Erro de sintaxe
        print("O número", numero, "não é primo.")
    else:
        contador = 0;
        for i in range(1, numero + 1, 1):
            if numero % i == 1: ← Erro de semântica
                contador = contadores + 1 ← Erro de execução
        if contador == 2:
            print("O número", numero, "é primo.")
        else:
            print("O número", numero, "não é primo.")
numero = int(input("Digite um número positivo: "))

```


Depuradores

- Ferramentas que auxiliam o programador na busca por erros no seu código.
- Para a linguagem Python, existe um depurador padrão incluso, denominado pdb.
- Algumas IDEs, como a PyCharm, fornecem depuradores próprios.
- O **pytest** (<https://docs.pytest.org/>) é um arcabouço para execução de testes automatizados em Python.