



INSTITUTO FEDERAL

Catarinense

Campus Camboriú

AULA 6

(Subalgoritmos - parte I)

Professora: Lidiane Visintin

lidiane.visintin@ifc.edu.br

Professor: Rafael de Moura Speroni

rafael.speroni@ifc.edu.br

O que são subalgoritmos

- Sempre é possível dividir problemas grandes em problemas menores, e de menor complexidade de resolução.
- É um algoritmo que resolve parte de um problema.
- Nesse caso, o algoritmo completo é dividido num algoritmo principal e em diversos subalgoritmo (tantos quantos forem necessários).
- Geralmente está subordinado a um outro algoritmo.

Vantagens de subalgoritmos

- Diminui a complexidade do problema.
- Permite focalizar a atenção em um problema pequeno de cada vez, o que ao final produzirá uma melhor compreensão do todo.
- Reusabilidade

Definição de subalgoritmos

- Cabeçalho, onde estão definidos o *nome e o tipo do subalgoritmo*, bem como os seus *parâmetros*;
- Corpo do subalgoritmo, onde se encontram as instruções, que serão executadas cada vez que ele é chamado.

Tipos de Subalgoritmos (Conceitos):

- As **funções** retornam um, e somente um valor ao algoritmo chamador;
- Os **procedimentos**, que retornam vários valores, ou nenhum, ao algoritmo chamador.

Forma geral de um procedimento

def NOME(PARÂMETROS):
 COMANDOS

```
def soma(a,b):  
    print(a+b)
```

Forma geral de uma função

```
def NOME( PARÂMETROS ):
    COMANDOS
    return ●
```

```
def soma(a,b):
    return a+b
```

Chamada de um subalgoritmo

```
def soma(a,b):  
    print(a+b)
```

```
soma(2,9)
```

```
soma(7,8)
```

```
soma(10,15)
```

11

15

25

Chamada de um subalgoritmo

```
def soma(a,b):  
    return a+b  
  
print(soma(2,9))  
  
print(soma(7,8))  
  
print(soma(10,15))
```

11
15
25

Chamada de um subalgoritmo

```
def e_impar(x):  
    return x % 2 == 1  
  
def par_ou_impar(x):  
    if e_impar(x):  
        return "Impar"  
    else:  
        return "Par"  
  
print(par_ou_impar(4))  
print(par_ou_impar(5))
```

```
Par  
Impar
```

Exercícios para praticar

1. Escreva uma função que receba dois números e retorne o maior.
Maximo(5, 6) ☐ retorno 6
Maximo(2, 1) ☐ retorno 2
2. Escreva uma função que receba um número e retorne **TRUE** se o número for par.
epar(6) ☐ retorno True
epar(1) ☐ retorno False
3. Escreva uma função que receba o lado de um quadrado e retorne a sua área ($A = \text{lado}^2$).
Area_quadrado(4) ☐ retorno 16
Area_quadrado(9) ☐ retorno 81
4. Escreva uma função que receba a base e a altura de um triângulo e retorne a sua área ($A = (\text{base} \times \text{altura})/2$).
Area_triangulo(6, 9) ☐ retorno 27
Area_triangulo(5, 8) ☐ retorno 20

Próximos Assuntos

- Variáveis Globais e Locais
- Passagem de parâmetros
- Recursividade

Referências



Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de programação: a construção de algoritmos e estruturas de dados**. 3. ed. Pearson Prentice Hall. 2005

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 27. ed.. Érica. 2014

Referências Complementares

MENEZES, Nilo Ney de Coutinho. **Introdução a programação com Python**. 3ª Ed. Novatec. 2019.

CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. 2. ed. Elsevier, Campus,. 2002

Referências na Internet

<https://docs.python.org/3/>

<https://www.w3schools.com/python/default.asp>

<https://panda.ime.usp.br/pensepy/static/pensepy/05-Funcoes/funcoes.html>