

PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON - ARA0066

Semana Aula: 10

FUNDAMENTOS DOS SUBPROGRAMAS

Tema

5 . SUBPROGRAMAS

Objetivos

Empregar modularização de programas para decompor problemas complexos em partes menores favorecendo a solução completa.

Tópicos

5 .1 FUNDAMENTOS DOS SUBPROGRAMAS

Procedimentos de Ensino-Aprendizagem

O professor deve iniciar a aula apresentando situações nas quais seja possível estabelecer relação entre o tópico desta aula e o objetivo. Como sugestão, segue o roteiro abaixo:

Situação-problema:

Suponha que um programador é contratado para desenvolver um sistema ERP (Enterprise Resource Planning) que é um software integrado para gestão empresarial formado por diversos módulos como: Contabilidade, Recursos Humanos, Suprimentos, Patrimônio, Vendas entre outros. Desenvolver um sistema desse porte em um único arquivo que contém todo o programa seria extremamente complexo tanto para sua manutenção como para o próprio entendimento haja vista a grande quantidade de código sequencial que o mesmo teria. Como poderíamos resolver esse problema dando mais organização para o esse programa?

Metodologia:

O professor deve iniciar a aula apresentando ao aluno os fundamentos de subprogramas, demonstrando a importância da segmentação de problemas grandes em pequenos problemas logicamente encadeados. O professor deve apresentar a sintaxe da construção

de funções na linguagem de programação Python além de apresentar os métodos de passagem de parâmetros. Deve-se apresentar um exemplo de programa recursivo, como o cálculo de fatorial. O professor também deverá implementar exemplos de subprogramas utilizando a linguagem de programação Python.

Atividade verificadora de aprendizagem:

Faça um programa com uma função chamada somaImposto. A função possui dois parâmetros formais: taxaImposto, que é a quantia de imposto sobre vendas expressa em porcentagem e custo, que é o custo de um item antes do imposto. A função "altera" o valor de custo para incluir o imposto sobre vendas.

Faça um programa que converta da notação de 24 horas para a notação de 12 horas. Por exemplo, o programa deve converter 14:25 em 2:25 P.M. A entrada é dada em dois inteiros. Deve haver pelo menos duas funções: uma para fazer a conversão e uma para a saída. Registre a informação A.M./P.M. como um valor "A" para A.M. e "P" para P.M. Assim, a função para efetuar as conversões terá um parâmetro formal para registrar se é A.M. ou P.M. Inclua um loop que permita que o usuário repita esse cálculo para novos valores de entrada todas as vezes que desejar.

Esta atividade computará 1,0 ponto para a AV2. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pelo curreteude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

Recursos Didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

Leitura Específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 9 (Subprogramas), páginas 364 a 393, Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582604694/>

[2] Documentação do Python "Princípios da Programação Funcional em Python". Disponível em <https://wiki.python.org.br/PrincipiosFuncionais>

[3] Listas de exercícios do Python Brasil sobre Funções, disponível em: <https://wiki.python.org.br/ExerciciosFuncoes>

Aprenda +

[4] Vídeo "Recursão". Disponível em <https://www.youtube.com/watch?v=KEEK7Me-ms> (Ative a legenda e a tradução automática para o português)

-

[5] Vídeo "Como usar funções no Python (Tutorial do Python # 3)". Disponível em <https://www.youtube.com/watch?v=NSbOtYZIQI0> (Ative a legenda e a tradução automática para o português)

Atividade Autônoma Aura:

Questão 1: (IF Sul Riograndense, 2019) (adaptada) Observe a função print do código escrito em Python a seguir:

```
def foo(n):  
    if n > 1:  
        return n * foo(n-1)  
    print(foo(4))
```

Qual o resultado impresso por essa função?

- a) 4
- b) 16
- c) 24
- d) 20
- e) 48

Questão 2: (AOCP - MJSP, 2020) Assinale a alternativa que apresenta o código Python que implementa corretamente uma função para o cálculo da área de um retângulo, bem como o código que imprime o seu resultado.

a)

```
def area_retangulo(comp, larg){  
    return comp * larg}  
print(' A área é ' +  
    area_retangulo (2,4))
```

b)

```
func area_retangulo(comp, larg):return comp * larg  
print(' A área é ' ||  
    area_retangulo (2,4))
```

c)

```
def area_retangulo(comp, larg)(  
    return comp * larg);  
print(' A área é {}'.format (area_retangulo (2,4)))
```

d)

```
func area_retangulo(comp, larg):return comp * larg;println(' A área é ' +
```

```
area_retangulo (2,4))
```

```
e) def area_retangulo(comp, larg):  
    return comp * larg  
    print(' A área é {}'.format  
    (area_retangulo (2,4)))
```