

PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON - ARA0066

Semana Aula: 5

ESCOPO

Tema

2 . NOMES, VINCULAÇÕES E ESCOPO

Objetivos

Entender o conceito de Escopo de uma linguagem e sua importância na manipulação de variáveis

Tópicos

2 .2 ESCOPO

Procedimentos de Ensino-Aprendizagem

Explicar ao aluno o conceito de escopo e sua importância, apresentar o conceito de blocos e revisar as estruturas de decisão e estruturas de repetição utilizando a linguagem Python para exemplificar. Como sugestão, segue o roteiro abaixo:

- Situação-problema:

Muitos programadores utilizam o mesmo nome de variável em várias partes do programa cujos valores são diferentes, por exemplo, é muito comum utilizar a variável chamada `i` para denotar a posição de um vetor. Como é possível que elas tenham valores diferentes mesmo tendo o mesmo nome?

- Metodologia:

Aula expositiva inicia com a apresentação do conceito de escopo de uma variável, em seguida o professor deve demonstrar o conceito a partir de um programa simples em Python. Como exemplo ele pode utilizar o seguinte trecho:

```
def func():
```

```
    x = 1
```

```
    print(x)
```

```
x = 10  
func()  
print(x)
```

cujos valores impressos são 1 e 10. Nesse momento o professor deve explicar, a partir do escopo, os valores que foram impressos.

Após a apresentação do conceito de escopo, o professor deve seguir a aula apresentando as estruturas de decisão e de repetição na linguagem Python.

- Atividade verificadora de aprendizagem:

O aluno deve acessar a lista de exercícios do site Python Brasil [2] e deve implementar os exercícios 3, 5 e 7. Também deve acessar a lista de exercícios sobre estruturas de repetição [3] e implementar os exercícios 1 e 9.

Esta atividade computará 1,0 ponto para a AV1. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pela curretude 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 5 (Nomes e vinculações de escopos), páginas 215 a 226, Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582604694/>

[2] Lista de exercícios de estruturas de decisão do site Python Brasil disponível em: <https://wiki.python.org.br/EstruturaDeDecisao>

[3] Lista de exercícios de estruturas de repetição do site Python Brasil disponível em <https://wiki.python.org.br/EstruturaDeRepeticao>

Aprenda +

[4] Resolver os seguintes exercícios propostos na plataforma URI ONLINE JUDGE:

- URI Online Judge | 1035 - Teste de Seleção 1

(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1035>)

- URI Online Judge | 1036 - Fórmula de Bhaskara

(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1036>)

- URI Online Judge | 1037 - Intervalo

(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1037>)
- URI Online Judge | 1038 - Lanche
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1038>)
- URI Online Judge | 1041 - Coordenadas de um Ponto
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1041>)
- URI Online Judge | 1043 - Triângulo
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1043>)
- URI Online Judge | 1059 - Números Pares
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1059>)
- URI Online Judge | 1060 - Números Positivos
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1060>)
- URI Online Judge | 1064 - Positivos e Média
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1064>)
- URI Online Judge | 1067 - Números Ímpares
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1067>)
- URI Online Judge | 1071 - Soma de Impares Consecutivos I
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1071>)
- URI Online Judge | 1072 - Intervalo 2
(<https://www.urionlinejudge.com.br/judge/pt/problems/view/1072>)

Atividade Autônoma Aura:

Questão 1: (QUADRIX - COBRA TECNOLOGIA, 2014)(Adaptada) Dado o programa em Python para troca de valores:

```
A = input("Digite o valor de A: ")
```

```
B = input("Digite o valor de B: ")
```

```
X = A
```

```
A = B
```

```
B = X
```

```
print("O valor de A é: ", A)
```

```
print("O valor de B é: ", B)
```

Considerando o algoritmo acima, as variáveis:

I. X, A e B foram criadas com escopo global na área de dados da memória.

II. X, A e B foram criadas com escopo local na área de dados da memória.

III. A e B poderiam, refazendo o algoritmo, ser qualificadas com escopo local e a variável X com escopo global.

IV. A e B poderiam, refazendo o algoritmo, ser qualificadas com escopo global e a variável X com escopo local.

Está correto o que consta somente em:

- a) III
- b) II
- c) I e III
- d) I e IV
- e) II e IV

Questão 2: (ENADE, 2011) Escopo dinâmico: para as linguagens com escopo dinâmico, a vinculação das variáveis ao escopo é realizada em tempo de execução. (...) Se uma variável é local ao bloco, então o uso da dada variável no bloco será sempre vinculado àquela local. Contudo, se a variável for não local, a sua vinculação depende da ordem de execução, a última vinculada na execução. A consequência disso é que, em um mesmo bloco de comandos, um identificador pode ter significados diferentes, e o programador precisa ter a ideia precisa de qual variável está sendo usada.

de MELO, A. C. V.; da SILVA, F. S. C. **Princípios de Linguagens de Programação**. São Paulo: Edgard Blücher, 2003. p.65.

Suponha que uma linguagem de programação tenha sido projetada com vinculação e verificação estáticas para tipos de variáveis, além de passagem de parâmetros por valor. Também é exigido pela especificação da linguagem que programas sejam compilados integralmente e que não é permitido compilar bibliotecas separadamente. Durante uma revisão da especificação da linguagem, alguém propôs que seja adicionado um mecanismo para suporte a variáveis com escopo dinâmico.

A respeito da proposta de modificação da linguagem, analise as seguintes afirmações.

I. As variáveis com escopo dinâmico podem ser tratadas como se fossem parâmetros para os subprogramas que as utilizam, sem que o programador tenha que especificá-las ou declarar seu tipo (o compilador fará isso). Assim, eliminase a necessidade de polimorfismo e é possível verificar tipos em tempo de compilação.

II. Como diferentes subprogramas podem declarar variáveis com o mesmo nome mas com tipos diferentes, se as variáveis com escopo dinâmico não forem declaradas no escopo onde são referenciadas, será necessário que a linguagem suporte polimorfismo de tipos.

III. Se as variáveis dinâmicas forem declaradas tanto nos escopos onde são criadas como nos subprogramas em que são referenciadas, marcadas como tendo escopo dinâmico, será

possível identificar todos os erros de tipo em tempo de compilação.

É correto apenas o que se afirma em:

- a) I.
- b) II.
- c) I e III.
- d) II e III.
- e) I, II e III.