



INSTITUTO FEDERAL
Rio Grande do Sul
Campus Farroupilha

Paradigmas de Linguagens de Programação

Aula 13
Escopo

Professor Cléber Macieski – Tecnólogo em ADS
cleber.macieski@farroupilha.ifrs.edu.br

Introdução

Assuntos:

Escopos, blocos e constantes.

Objetivos:

Seguir conhecendo conceitos básicos das linguagens de programação.



Escopo

- O escopo de uma variável é a faixa de sentenças nas quais ela é visível.
- Uma variável é **visível** se ela pode ser referenciada ou atribuída.
- Variável local:
 - Definida e visível no bloco em questão.
- Variável não local:
 - Visível no bloco mas não definida nele.
 - Variáveis globais são uma categoria a parte das não locais.
- O escopo pode ser estático ou dinâmico.



Escopo Estático

- Método de vincular **nomes** a variáveis não locais.
- Introduzido por ALGOL 60.
- Sabe-se o escopo antes de executar o programa.
- Para ligar um nome de referência para uma variável, você (ou o compilador) deve encontrar a sua declaração.



Categorias de Linguagens Quanto a Escopo Estático

- **Unidade de Programa** ([Program Unit](#) – PU): Uma parte constituinte de uma programa maior, e em algum sentido, autocontido.
- Duas categorias:

Linguagens que Permitem Subprogramas Aninhados

Linguagens que Não Permitem Subprogramas Aninhados

```
4 int main()  
5 {  
6     int x = 4;  
7     void funcao_aninhada()  
8     {  
9         printf("%d\n", x);  
10    }  
11  
12    funcao_aninhada();  
13    return 0;  
14 }  
15
```

```
5 public static void main (String[] args) {  
6  
7     public void metodoAninhadoNaoPermitido() {  
8         System.out.println("Isso é inválido");  
9     }  
10 }
```

Escopo Estático: Determinação de Atributos de Variáveis

- Linguagem de escopo estático com subprogramas.
- Busca dos atributos ocorre buscando-se a declaração.
- Primeiro no subprograma.
- Depois no **pai estático**, sucessivamente.
- O conjunto de pais são chamados de **ancestrais estáticos**.

```
>> function externa(){  
    function interna1(){  
        var x = 7;  
        interna2();  
    }  
  
    function interna2(){  
        console.log(x);  
        var y = x;  
    }  
  
    var x = 3;  
    interna1();  
}
```

Blocos

- Conceito introduzido no Algol 60.
- Definem um novo escopo estático.
- Permitem a seção de código ter suas próprias variáveis locais.
- Redecaração interna é inválida em algumas linguagens.

```
12 public static void main(String[] args) {  
13     int i = 0;  
14     {  
15         int i = 0;  
16         System.out.println(i);  
17     }
```

```
4 int main()  
5 {  
6     int x = 4;  
7     int y = 11;  
8     {  
9         int x = 5;  
10        printf("%d\n", x);  
11  
12        y = 13;  
13        printf("%d\n", y);  
14    }  
15    printf("%d\n", x);  
16  
17    printf("%d\n", y);  
18  
19    return 0;
```



Escopo Global

- C e C++ têm declarações e definições.
- C, C++, PHP e Python permitem uma estrutura de programa que é uma sequência de definição de funções, nas quais as definições de variáveis podem aparecer **fora das funções**.
- Variáveis externas a funções costumam ser visíveis por todo o arquivo.

```
4   int global = 10;
5
6   int main()
7   {
8       int x = 4;
9       int y = 11;
10      {
11          int x = 5;
12          printf("%d\n", x);
13
14          printf("Global= %d\n", global);
15
16          y = 13;
17          printf("%d\n", y);
18      }
```


Escopo Dinâmico

- Utiliza **ancestralidade dinâmica**.
- Os atributos da variável dependendo de quem chamou a função em *runtime*.
- Não é possível garantir olhando para parte do código estático.
- Possível de ser utilizado em Perl e Common Lisp.
- Causa diversos problemas como:
 - Menor ocultação de informação.
 - Impossibilidade de verificar tipos.
 - Aumenta a dificuldade do entendimento.
 - Performance.
- Vantagem:
 - Passagem de parâmetros desnecessária.

```
>> function externa(){  
    function interna1(){  
        var x = 7;  
    }  
  
    function interna2(){  
        console.log(x);  
        var y = x;  
    }  
  
    var x = 3;  
    interna1();  
}
```



Constantes Nomeadas

- Uma constante nomeada é uma variável que está vinculada a um valor apenas uma vez.
- Vantagens:
 - Legibilidade .
 - Confiabilidade.
- Usadas para parametrizar programas.
- Vinculação ocorre antes do tempo de execução.

```
4  int main()
5  {
6      float const PI = 3.14;
7      int const VERDADEIRO = 1;
8      int const FALSO = 0;
9
10     int x = 0;
11
12     if ( x == VERDADEIRO )
13     {
14         printf("%f\n", PI);
15     }
16
17     PI = 1.5;
18
19
20     return 0;
21 }
```

others

Code::Blocks X Search results X Cccc X Build log X Build message:

	Line	Message
		=== Build: Debug in Testes (compiler: GNU GCC Co
		In function 'main':
Fontes\wor...	17	error: assignment of read-only variable 'PI'

No próximo encontro

Tipos de Dados.



INSTITUTO FEDERAL
Rio Grande do Sul
Campus Farroupilha



Concluindo

Dúvidas, a disposição.



INSTITUTO FEDERAL
Rio Grande do Sul
Campus Farroupilha



Referências e Materiais Interessantes

SEBESTA, R. W. **Conceitos de Linguagens de Programação**. 5ª ed. Porto Alegre: Bookman, 2003.



INSTITUTO FEDERAL
Rio Grande do Sul
Campus Farroupilha

