Modularização – Parte IV

Profa. Dra. Juliana Cristina Braga

Centro de Matemática, Computação e Cognição

Objetivo da Aula

- Entender o que é escopo em programação
- Entender a importância do escopo na programação de módulos
- Entender sobre quantificadores dos módulos

Roteiro da Aula

- Blocos
- Escopo (global e local)
- Escopo em JAVA
- Qualificadores de métodos em JAVA

Blocos

- O que são Blocos?
  - Um bloco em Portugol ou em JAVA é definido por
  - {
  - }

Blocos - Exemplos

Cada Cor
Indica um Bloco
Na Figura ao Lado

```
programa
{
    funcao inicio()
    {
      real num1, num2
      escreva("Entre com o valor do primeiro número\n")
      leia(num1)
      escreva("Entre com o valor do segundo número\n")
      leia(num2)
      escreva("A soma dos valores é ", soma(num1, num2))
    }

     funcao real soma(real a, real b){
          real s
          s = a+b
          retorne s
     }
}
```

Bloco 1

Bloco 2

Bloco 3

Blocos - Exemplos

Cada Cor
Indica um Bloco
Na Figura ao Lado

```
funcao real pesoIdeal(real altura, caracter sexo){

    real p=0
    se (sexo == 'f'){
        p = ((62.1*altura) - 44.7)
    }
    senao {
        p = ((27.7*altura) - 58)
    }
    retorne p

}
```

Bloco 1

Bloco 2

Bloco 3

Blocos

- Porque é importante saber a limitação de um bloco?
- Porque é o bloco que define o escopo das entidades (ex: variáveis, arquivos, módulos) que estão nele contidas
- Ou seja, um bloco define o ESCOPO das entidades

Escopo Global

- Escopo global – são entidades que podem ser utilizadas nos módulos em que foram declaradas e também nos módulos internos aos módulos que foram declarados

Exemplo Escopo Global

```
programa
{
        inteiro a = 10
        funcao inicio()
        {
          inteiro b = 20
          escreva(a+b)
        }
}
```

Qual o escopo da variável a?

Módulos – Escopo Local

- Escopo local – entidade somente pode ser usada no módulo em que foi declarado. Entidades locais a um módulo não têm significado fora desse módulo

Exemplo Escopo Local

```
programa
{
        inteiro a = 10
        funcao inicio()
        {
            inteiro b = 20
            escreva(a+b)
        }
}
```

Qual o escopo da variável b?

```
programa
{
        inteiro a = 10
        funcao inicio()
        {
            inteiro b = 20
            escreva(a+b)
        }
        funcao moduloA()
        {
            escreva(a)
        }
}
```

Esse código está correto?

Posso usar a variável a dentro do moduloA?

```
programa
{
	inteiro a = 10
	funcao inicio()
	{
	  inteiro b = 20
	  escreva(a+b)
	}
	funcao moduloA()
	{
	  escreva(a)
	  escreva(b)
	}
}
```

Esse código está correto?

Posso usar a variável **b** dentro do moduloA?

```
programa
{
	inteiro a = 10
	funcao inicio()
	{
	  inteiro b = 20
	  escreva(a+b)
	}
	funcao moduloA()
	{
	  escreva(a)
	  escreva(b)
	}
}
```

Uma variável **LOCAL** só existe dentro do bloco em que foi declarada. Depois disso ela é “destruída” da memória.

Uma variável **GLOBAL** existe enquanto o programa estiver na memória.

```
programa
{
	funcao inicio() {
		inteiro a = 20
		escreva(a)
		moduloA()
	}
	funcao moduloA(){
		inteiro a = 10
		escreva("\n")
		escreva(a)
	}

}
```

Esse código está correto?
Posso ter dois nomes iguais para Variáveis de escopo distinto?
Qual a saída desse programa?

Modularização – Parte IV

As regras de escopo
Valem para qualquer bloco e não
Somente para módulos!

```
programa
{
      inteiro a
      funcao inicio() {
        para (inteiro i=0; i<3;i++){
            escreva(i)
        }
      }

}
```

Variável local ao bloco para

Modularização – Parte IV

```
programa
{
    funcao inicio() {

        para (inteiro i=0;i<3;i++){
            escreva(i)
        }
        escreva(i)
    }

}
```

Esse código está correto?
Qual a saída?

```
programa
{
	funcao inicio() {
	  inteiro i
	  para (i=0;i<3;i++){
	     escreva(i)
	  }
	  escreva(i)
	}

}
```

Esse código está correto?
Qual a saída?

```
funcao inicio()
{
  real a
  caracter s
  escreva("entre com sua altura\n")
  leia(a)
  escreva("entre com seu sexo (f ou m)\n")
  leia(s)
  escreva("Seu peso ideal é ", pesoIdeal(a,s))

}
funcao real pesoIdeal(real altura, caracter sexo){
     real p=0
     se (sexo == 'f'){
          p = ((62.1*altura) - 44.7)
     }
     senao {
          p = ((27.7*altura) - 58)
     }
     retorne p
}
```

Os parâmetros dos mód
Possuem escopo LOCAL!!

Escopo

Evitem utilizar variáveis globais!

- Ocupam mais espaço de memória, pois ocupam a memória enquanto o programa estiver sendo executado
- Sempre que surgir a dúvida:
  - Essa variável deve ser global ou local?
    - R: será local somente se você necessitar dela o tempo todo!
  - Dê preferência pelas variáveis locais!

EM JAVA

Modularização – Parte IV

```
package aula3;
public class Main {
    static int c=0;
    public static void main(String[] args) {
        int a=5;
        System.out.println(a);
        exemplo(a);
        System.out.println(c);

    }
    public static void exemplo(int b){
        int a=3;
        System.out.println(a);
        System.out.println(b);
        c = a*b;
        System.out.println(c);
    }

}
```

Exemplo de declaração
De variável global
Em JAVA

Colocar a palavra reservada **static**

Escopo - Qualificadores dos métodos

- Qualificadores dos métodos em JAVA
- Também está relacionado com o escopo
- **public**: método acessível em outros programas
- **private**: método acessível apenas no programa em que foi definido
- **protected**: método acessível na classe, subclasses

Orientação a objetos
Foge do objetivo da disciplina

Coisas para não esquecer

- Um bloco agrupa zero ou mais instruções.
- Um bloco é delimitado pelos CHAVES { e }.
- Uma entidade que é declarada dentro de um método é chamada de entidade local.
- A entidade local existe somente dentro do método.
- O método fica na memória durante a sua execução, depois disso ele é “destruído” da memória e suas variáveis também.

Coisas para não esquecer

- Uma varíável criada fora de qualquer método e dentro da classe principal é chamada de variável global.
- A variável global existe dentro de qualquer método criada dentro do programa.
- A variável global existe durante toda execução do programa
- Uma variável local existe apenas enquanto o bloco que a contém está a ser executado.
- Uma variável local pode ser declarada em qualquer ponto do bloco a que pertence, e não apenas no início, mas sempre antes de ser usada.
- EVITE CRIAR VARIÁVEIS GLOBAIS