



Universidade Federal do ABC

Modularização ou Subrotinas

Parte I



Objetivos da Aula

- Entender os benefícios no uso dos módulos (ou subrotinas)
- Entender conceitos de módulos
- Entender como utilizar módulos (ou subrotinas) já existentes no **PortugolStudio**

Refletindo ...

Até o momento temos escrito nossos programas no PortugolStudio de maneira “sequencial” dentro da função início

```
1 programa
2 {
3     funcao inicio()
4     {
5
6         // DIGITE AQUI O SEU PROGRAMA
7
8     }
9 }
```

Refletindo ...

Mas também além dos comandos da **linguagem** (**leia**, **escreva**, **se**, **enquanto**) podemos **CHAMAR** (utilizar) rotinas (funções) prontas na linguagem

```
inclua biblioteca Matematica -->mat

funcao inicio() {
    real base, expoente, resultado
    escreva("Digite a base: ")
    leia(base)
    escreva("Digite o expoente: ")
    leia(expoente)

    // chamando a função que calcula a potencia
    // a função já existe na biblioteca Matematica
    resultado = mat.potencia(base, expoente)

    escreva(base, " elevado a ", expoente, " = ", resultado)
}
```

Refletindo.

- Note que, se não tivéssemos a função potencia pronta (na biblioteca Matematica) e precisássemos calcular a potência (x elevado a y) teríamos que construir esse código (programar).
- Mas como já existe (e nós conhecemos seu nome) nós apenas a chamamos.

```
resultado = mat.potencia(base, expoente)|
```

```
escreva(base, " elevado a ", expoente, " = ", resultado)
```

potencia é um exemplo de uma função (módulo ou subrotina) já existente no Portugol Studio

Módulos - Conceituando

- Um **módulo ou subrotina**, é um trecho de programa com finalidade específica, construído sob um nome.

potencia é uma rotina que tem por finalidade “retornar” a potenciação entre dois números (x, y) : **x elevado a y**. “Alguém” deu o nome de potencia para essa rotina.

```
resultado = mat.potencia(base, expoente)
```

```
escreva(base, " elevado a ", expoente, " = ", resultado)
```

Módulos - conceituando

Existem outras funções “prontas” no Portugol Studio. Por exemplo:

- **real raiz(real radicando, real indice):** Realiza a radiciação (extrai a raíz) de um número por um determinado índice
- **inteiro sorteia(inteiro minimo, inteiro maximo):** Sorteia um número aleatório entre os valores mínimo e máximo especificados

Módulos - conceituando

Veremos mais adiante que nós também podemos construir nossas próprias funções (módulos ou subrotinas). Isso chama-se **MODULARIZAÇÃO!**

- A modularização de um programa é a **divisão desse programa em subprogramas**

Módulos – Parte 1

programa

{

funcao inicio()

{

//codigo da funcao principal

}

funcao moduloA() {

//código do módulo A

}

funcao moduloB() {

//código do módulo B

}

}

EXEMPLO DE USO DA FUNÇÃO RAIZ

Módulos – Formas de Utilização

PortugolStudio

```
1 programa
2 {
3     inclui biblioteca Matematica -->mat
4
5     funcao inicio()
6     {
7         real num, resultado
8         escreva("Digite um numero ")
9         leia(num)
10
11         resultado = mat.raiz(num, 2.0)
12
13         escreva("Raiz Quadrada de ", num, " = ", resultado)
14     }
15 }
16
```

Biblioteca

// calcula a raiz quadrada (2.0) de num

Módulos – Formas de Utilização

PortugolStudio

```
inclua biblioteca Util --> util
funcao inicio()
{
    inteiro vet[10], i

    // preenche o vetor
    para(i = 0; i < 10; i++){
        vet [i] = util.sorteia(1,100)
    }
```

Biblioteca

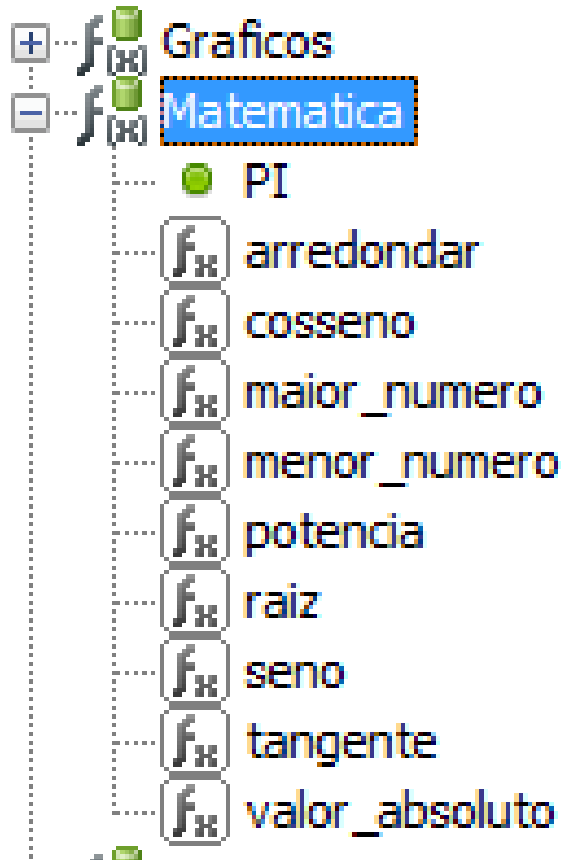
// SORTEIA um número entre 1 e 100

VANTAGENS DE USO

Módulos – Vantagens

- Neste ponto podemos entender **uma das vantagens** da utilização do módulo: **reuso!**
 - Imagine a economia de código que se faz ao chamar o a função **RAIZ** sem necessitar implementá-la?
 - O mesmo vale para a função **SORTEIA**.
 - E não é nem necessário CONHECER o código. Basta saber o **NOME da função, o que ela faz e sua sintaxe (parâmetros, etc)**

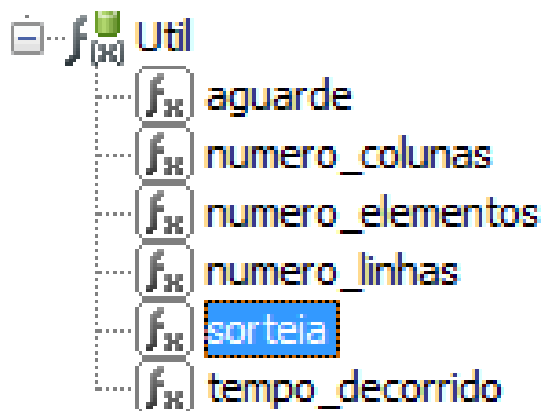
Módulos – Bibliotecas no Portugol Studio



Para conhecer as funções (módulos) existentes nós precisamos consultar a documentação **(AJUDA)**

Módulos – Bibliotecas no Portugol Studio

Na documentação cada função é descrita e seu uso é explicado



funcao **inteiro** sorteia(**inteiro** minimo, **inteiro** maximo)

Descrição: Sorteia um número aleatório entre os valores mínimo e máximo especificados

Parâmetros:

- **minimo**: o menor número que pode ser sorteado
- **maximo**: o maior número que pode ser sorteado

Retorna: O número sorteado

Módulos – Mais Vantagens

- Reaproveitamento de código: diminuição do código, diminuição do retrabalho (se tivéssemos que implementar de novo).
- Organização
- Legibilidade
- Flexibilidade

Essas vantagens ficarão mais claras quando começarmos a construir nossas próprias funções

COMPONENTES DE UM MÓDULO

Módulos - Componentes

- **(1) Identificador** (nome da função)
- **(2) Parâmetros** (o que a função recebe para processar)
- **(3) retorno** (o que a função retorna)

(2) PARÂMETROS

```
resultado = mat.potencia(base, expoente)
```

(3) RETORNO

(1) IDENTIFICADOR

PARÂMETROS

- São os valores ou variáveis **enviadas ao módulo**, para que ele possa realizar o seu processamento
- Podemos passar **nenhum** ou **vários** parâmetros para o módulo
 - Tudo depende da finalidade do módulo (e de como ele foi construído)

(2) PARÂMETROS

```
resultado = mat.potencia(base, expoente)
```

Módulos – Componentes - Parâmetros

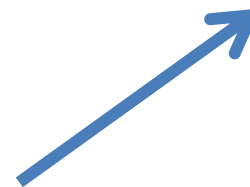
```
1 programa
2 {
3     inclui biblioteca Matematica -->mat
4
5     funcao inicio()
6     {
7         real num, resultado
8         escreva("Digite um numero ")
9         leia(num)
10
11         resultado = mat.raiz(num, 2.0)
12
13         escreva("Raiz Quadrada de ", num, " = ", resultado)
14     }
15 }
16
```

2 parâmetros

Módulos – Componentes - Parâmetros

```
1 programa
2 {
3     inclui biblioteca Matematica -->mat
4
5     funcao inicio()
6     {
7         real base, expoente, resultado
8         escreva("Digite a base: ")
9         leia(base)
10        escreva("Digite o expoente: ")
11        leia(expoente)
12
13        resultado = mat.potencia(base, expoente)
14
15        escreva(base, " elevado a ", expoente, " = ", resultado)
16    }
17 }
```

2 parâmetros



Módulos – Componentes - Parâmetros

```
inclua biblioteca Matematica -->mat
```

```
funcao inicio()
```

```
{
```

```
    real base, expoente, resultado
```

```
    escreva("Digite a base: ")
```

```
    leia(base)
```

```
    escreva("Digite o expoente: ")
```

```
    leia(expoente)
```

```
    resultado = mat.potencia(base, expoente)
```

```
    limpa()
```



0 parâmetros

```
    escreva(base, " elevado a ", expoente, " = ", resultado)
```

```
}
```

RETORNO

- O Retorno de um módulo é o valor (ou valores) que o módulo retorna para o local em que ele foi chamado

```
resultado = mat.potencia(base, expoente)
```

(3) RETORNO

Módulos – Componentes - Retorno



Uma
variável
recebe o
valor de
retorno do
módulo

```
1 programa
2 {
3     inclui biblioteca Matematica -->mat
4
5     funcao inicio()
6     {
7         real base, expoente, resultado
8         escreva("Digite a base: ")
9         leia(base)
10        escreva("Digite o expoente: ")
11        leia(expoente)
12
13        resultado = mat.potencia(base, expoente)
14
15        escreva(base, " elevado a ", expoente, " = ", resultado)
16    }
17 }
```

INFORMAÇÕES ADICIONAIS

- O Retorno de um módulo é o que define a classificação módulo: **PROCEDIMENTO** ou **FUNÇÃO**
- Módulos que não retornam valores são denominados **PROCEDIMENTOS**
- Módulos que retornam valores são denominados **FUNÇÕES**

**NESSE CURSO VAMOS CHAMAR TUDO DE
FUNÇÃO!**

Módulos – Como utilizar?

PortugolStudio

- No Portugol Studio, consulte a Ajuda

Aprender

Escolha sua forma de aprender:

:: Sair Programando (Ctrl + N)

:: Conhecer a Linguagem (F1)

:: Conhecer as Bibliotecas (Shift + F1)

:: Atalhos do teclado (F11)

:: Explorar os Exemplos (Alt+E)

Portugol: exemplos de funções pré-definidas

- **real arredondar(real numero, inteiro casas):** Arredonda um número real para o número de casas decimais informado.
- **real potencia(real base, real expoente):** Realiza uma exponenciação através da multiplicação da base por ela mesma tantas vezes quanto indicar o expoente
- **real raiz(real radicando, real indice):** Realiza a radiciação (extrai a raíz) de um número por um determinado índice
- **real valor_absoluto(real numero):** Calcula o valor absoluto do **número** informado
- **inteiro sorteia(inteiro minimo, inteiro maximo):** Sorteia um número aleatório entre os valores mínimo e máximo especificados

Portugol: Funções para manipulação de cadeias de caracteres (strings)

- cadeia **caracteres_maiusculos**(cadeia cad) :
Transforma os caracteres de uma cadeia em caracteres maiúsculos
- cadeia **caracteres_minusculos**(cadeia cad) :
Transforma os caracteres de uma cadeia em caracteres minúsculos
- cadeia **substituir**(cadeia cad, cadeia texto_pesquisa, cadeia texto_substituto): Pesquisa por um determinado texto em uma cadeia e substitui todas as ocorrências por um texto alternativo

MATERIAL ADICIONAL

MÓDULOS M JAVA



Módulos - Nomenclatura

- Em Portugal Studio
 - Os módulos são chamados de **funções**
- Em Java
 - Os módulos são chamados de **MÉTODOS**
- **Usaremos como sinônimos: chamaremos tudo de FUNÇÕES!**

Módulos – Formas de Utilização



```
class ExemploModulosJava {  
  
    public static void main(String args[]) {  
  
        System.out.println("Metodo abs(-30): "  
        System.out.println("Metodo pow(2,3): "  
        System.out.println("Metodo sqrt(16): "  
  
    }  
}
```

abs – retorna o valor absoluto do número
pow – retorna a potência do número
sqrt – retorna a raiz quadrada do número

No java temos a biblioteca Math

```
+ Math.abs(-30) );  
+ Math.pow(2,3) );  
+ Math.sqrt(16) );
```



No JAVA indica-se também o nome da biblioteca que contém o módulo (função)



Módulos – Componentes - Retorno

```
class ExemploModulosJava {  
  
    public static void main(String args[]) {  
  
        System.out.println("Metodo abs(-30): "      + Math.abs(-30) );  
        System.out.println("Metodo pow(2,3): "      + Math.pow(2,3) );  
        System.out.println("Metodo sqrt(16): "      + Math.sqrt(16) );  
  
    }  
}
```

Só é possível imprimir o valor do módulo, potência e quadrado, porque os módulos *abs*, *pow* e *sqrt* respectivamente retornam valores

Todos os valores retornados por *abs*, *pow* e *sqrt* poderiam ter sido atribuídos a uma variável



Módulos – Componentes – Resumindo...

INDEPENDENTE DA LINGUAGEM:

PortugolStudio

OU



Math.sqrt(9)

mat.raiz(9, 2)

Resolvem o mesmo problema
O identificador é mnemônico
Retornam os valores 3
Possuem um parâmetro

limpa()

Não retorna valor
Não possui parâmetro

TODOS SÃO
MÓDULOS!

Módulos – Componentes – Resumindo...

INDEPENDENTE DA LINGUAGEM:

PortugolStudio

OU



Math.sqrt(9)

mat.raiz(9,2)

São FUNÇÕES

Limpa()

É PROCEDIMENTO

TODOS SÃO
MÓDULOS!

Módulos – Como utilizar?

No Java consulte a API: <http://download.oracle.com/javase/6/docs/api/>



[java.awt.image](#)
[java.awt.image.renderable](#)
[java.awt.print](#)
[java.beans](#)
[java.beans.beancontext](#)
[java.io](#)
[java.lang](#)
[java.lang.annotation](#)
[java.lang.instrument](#)
[java.lang.management](#)

[InheritableThreadLocal](#)
[Integer](#)
[Long](#)
[Math](#)
[Number](#)
[Object](#)
[Package](#)
[Process](#)
[ProcessBuilder](#)
[Runtime](#)
[RuntimePermission](#)
[SecurityManager](#)
[Short](#)
[StackTraceElement](#)
[StrictMath](#)
[String](#)
[StringBuffer](#)
[StringBuilder](#)
[System](#)
[Thread](#)
[ThreadGroup](#)
[ThreadLocal](#)
[Throwable](#)
[Void](#)

Method Summary

| | | |
|---------------|--|--|
| static double | abs (double a) | Returns the absolute value of a double value. |
| static float | abs (float a) | Returns the absolute value of a float value. |
| static int | abs (int a) | Returns the absolute value of an int value. |
| static long | abs (long a) | Returns the absolute value of a long value. |
| static double | acos (double a) | Returns the arc cosine of a value; the returned angle is in the range 0.0 through π . |
| static double | asin (double a) | Returns the arc sine of a value; the returned angle is in the range $-\pi/2$ through $\pi/2$. |
| static double | atan (double a) | Returns the arc tangent of a value; the returned angle is in the range $-\pi/2$ through $\pi/2$. |
| static double | atan2 (double y, double x) | Returns the angle θ from the conversion of rectangular coordinates (x, y) to polar coordinates (r, θ). |
| static double | cbrt (double a) | Returns the cube root of a double value. |
| static double | ceil (double a) | Returns the smallest (closest to negative infinity) double value that is greater than or equal to the argument and is equal to a mathematical integer. |
| static double | copySign (double magnitude, double sign) | Returns the first floating-point argument with the sign of the second floating-point argument. |

Módulos – Como utilizar?



**** Assinatura do módulo

```
static double pow(double a, double b)
```

Returns the value of the first argument raised to the power of the second argument.

Valor de retorno Nome

- Número de parâmetros
- Tipo de cada parâmetro

```
System.out.println("O resultado é: "+Math.pow(base, expoente));
```

Módulos – Como utilizar?



```
Math.pow(2, 0.5);
```

correto

```
double pot = Math.pow(2, 0.5);
```

correto

```
Math.pow(0.5);
```

incorreto

```
Math.pow("2", "2.5");
```

incorreto

Explique porque está correto ou incorreto.

Módulos – Como utilizar?



- Alguns métodos em JAVA podem ser usados de várias maneiras. Quem “*manda*” é a *assinatura* dele! Veja o método `abs`:

| | |
|----------------------------|--|
| <code>static double</code> | <code><u>abs</u>(double a)</code> Returns the absolute value of a double value. |
| <code>static float</code> | <code><u>abs</u>(float a)</code> Returns the absolute value of a float value. |
| <code>static int</code> | <code><u>abs</u>(int a)</code> Returns the absolute value of an int value. |
| <code>static long</code> | <code><u>abs</u>(long a)</code> Returns the absolute value of a long value. |

Esse é um dos exemplos de polimorfismo em Programação Orientada a Objetos

COISAS PARA NÃO ESQUECER

Coisas para não esquecer

- Módulos são *subprogramas* que podemos chamar dentro do programa principal ou dentro de outros módulos
- Um módulo DEVE possuir: um identificador ou nome de chamada
- Um módulo PODE possuir: parâmetros e/ou valor de retorno
- Um módulo PODE ou não retornar valores

Coisas para não esquecer

- Um módulo que Não possui valor de retorno é chamado de PROCEDIMENTO
- Um módulo que possui valor de retorno é chamado de FUNÇÃO

O PRÓXIMO PASSO É APRENDER A CRIAR OS NOSSO PRÓPRIOS MÓDULOS (FUNÇÕES)