|  |  |
| --- | --- |
| Pedro Henrique Padilha Portella da Cruz | 28/02/2021 |
| Programação para Dispositivos Móveis | Bittencourt |

Resumo Java Básico Aulas 7, 8 e 9



**Operadores Aritméticos:**

+ soma

- subtração

\* produto

/ divisão

% resto da divisão

Ex:

*Public static void main(String[]args) {*

*Int n1 = 3;*

*Int n2 = 5;*

*Flaot m = (n1 + n2) / 2;*

*System.out.println(“a media entre ”+n1+” e ”+n2+”é de ”+m);*

*}*

**Operadores de Incremento e Decremento:**

Os operadores de incremento e decremento server para incrementar ou decrementar 1 valor de variáveis em certas condições, como em iterações de for, foreach, ou validações de dados.

Ex:

*Public static void main(String[]args) {*

*Int n = 8;*

*n--;*

*n++;*

*n++;*

*System.out.println(n);*

*}*

Operadores de Atribuição

Operadores que atribuem diretamente uma operação que envolve um valor nele mesmo.

+= somar e atribuir

-= subtrair e atribuir

\*= multiplicar e atribuir

/= dividir e atribuir

Ex:

a += b => a = a + b

a -= b => a = a - b

a \*= b => a = a \* b

a /= b => a = a / b

Classe Math

Responsável por conter a maioria dos métodos de cálculos matemáticos e constantes aritméticas, importada do pacote java.lang.Math

Math.abs(...);  
Descrição: Retorna o valor absoluto (módulo) do numero passado por parametro.  
Parametro: Pode ser um int, um double, um float ou um long  
Retorno: Mesmo tipo primitivo do valor de entrada (parametro)  
Quote:  
Math.acos(...);  
Descrição: Retorna ao usuário o arco-cosseno do angulo passado por parametro (retorno entre 0 e PI [metade superior de uma circunferencia trigonometrica)  
Parametro: double  
Retorno: doulbe  
Quote:  
Math.asin(...);  
Descrição: Retorna ao arco-seno do angulo passado para o método (retorno entre -PI/2 [3/4 de circunferencia trigonometrica) e PI/2 (1/4 da circunferencia)  
Parametro: double  
Retorno: double  
Quote:  
Math.atan(...);  
Descrição: Retorna o arco-tangente do angulo que o usuario passou (mesmo "range" (intervalo) do Math.asin(...)  
Parametro: double  
Retorno: double  
Quote:  
Math.ceil(...);  
Descrição: Este método retorna o maior numero inteiro (menor que o passado como parametro) - (ATENCAO PARA NUMEROS NEGATIVOS)  
Parametro: double  
Retorno: double  
Quote:  
Math.cos(...);  
Descrição: Retorna o coseno do angulo passado  
Parametro: double  
Retorno: double  
Quote:  
Math.exp(...);  
Descrição: Retona o valor da Constante de Euller "e" elevada ao numero passado  
Parametro: double  
Retorno: double  
Quote:  
Math.floor(...);  
Descrição: Retorna o maior numero inteiro (nao menor que o passado) - (ATENCAO PARA NUMEROS NEGATIVOS)  
Parametro: double  
Retorno: double  
Quote:  
Math.log(...);  
Descrição: Retorna o logaritmo natural do numero passado.  
Parametro: double  
Retorno: double  
Quote:  
Math.max(... , ...);  
Descrição: Retorna o maior entre os numeros passados  
Parametro: pode ser um par de int, de double, de float ou de long (desde que os 2 parametros sejam do mesmo tipo)  
Retorno: depende do tipo de entrada  
Quote:  
Math.min(... , ...);  
Descrição: Retorna o menor entre os numeros passados  
Parametro: pode ser um par de int, de double, de float ou de long (desde que os 2 parametros sejam do mesmo tipo)  
Retorno: depende do tipo de entrada  
Quote:  
Math.pow(... , ...);  
Descrição: Para uma estrutura de potenciacao a^b este metodo retorna o primeiro paramentro como 'a' e o segundo como 'b'  
Parametro: double  
Retorno: double  
Quote:  
Math.random();  
Descrição: um numero aleatório que vai de zero até 1 (0 incluido, 1 nunca será gerado)  
Parametro: nenhum  
Retorno: double  
Quote:  
Math.round(...);  
Descrição: Retorna o long mais próximo do parametro pasado  
Parametro: double  
Retorno: long  
Quote:  
Math.sin(...);  
Descrição: Retorna o seno do parametro  
Parametro: double  
Retorno: double  
Quote:  
Math.tan(...);  
Descrição: Retorna a tangente do angulo  
Parametro: double  
Retorno: double  
Quote:  
Math.sqrt(...);  
Descrição: Retorna a raiz quadrada do numero passado  
Parametro: double  
Retorno: double  
Quote:  
Math.toDregrees(...);  
Descrição: Retorna o angula passado (em radianos) em graus  
Parametro: double  
Retorno: double  
Quote:  
Math.toRadians(...);  
Descrição: Retorna o angula passado (em graus) em radianos  
Parametro: double  
Retorno: double

**Operador ternário**

O operador ternário é um recurso para tomada de decisões com objetivo similar ao do if/else, mas que é codificado em apenas uma linha.

Sintaxe do operador ternário:

(expressão booleana) ? código 1 : código 2;

Ao avaliar a expressão booleana, caso ela seja verdadeira, o código 1, declarado após o ponto de interrogação (?) será executado; do contrário, o programa irá executar o código 2, declarado após os dois pontos (:).

**Operadores Lógicos**

Símbolo && é chamado de E. Este operador retorna true somente se os dois operandos forem true.

**<**operando1**>** **&&** **<**operando2**>**

Se o valor do operando1 for false, então o operador && não verifica o valor do operador2, pois sabe que o resultado já é false.

Símbolo || é chamado de OU. Este operado retorna true caso tenha pelo menos um operando com o valor true.

**<**operando1**>** **||** **<**operando2**>**

Se o valor do operando1 for true, então o operador || não verifica o valor do operando2, pois já sabe que o resultado é true.

Símbolo ! é chamado de negação. Este operador retorna true se o operando tem o valor false, e retorna false se o operando o valor true.

**!** **<**operando**>**

**Estruturas Condicionais**

**double livroLinux;**

**double livroBancosDados;**

**livroLinux = 78.60;**

**livroBancosDados = 56.75;**

**double total = livroLinux + livroBancosDados;**

**System.out.println("O preço total é " + total );**

**if (total < 120.00 ) {**

**System.out.println("O preço está bom!");**

**}**

**else {**

**System.out.println("Livros muito caros!");**

**}**