

O que é JAVA?

O conceito.

Java é uma linguagem de programação orientada a objetos desenvolvida pela Sun Microsystems. Modelada depois de C++, a linguagem Java foi projetada para ser pequena, simples e portátil a todas as plataformas e sistemas operacionais, tanto o código fonte como os binários. Esta portabilidade é obtida pelo fato da linguagem ser interpretada, ou seja, o compilador gera um código independente de máquina chamado byte-code. No momento da execução este byte-code é interpretado por uma máquina virtual instalada na máquina.

Para portar Java para uma arquitetura hardware específica, basta instalar a máquina virtual (interpretador). Além de ser integrada à Internet, Java também é uma excelente linguagem para desenvolvimento de aplicações em geral. Dá suporte ao desenvolvimento de software em larga escala.

2.0 O Básico

2.1. Variáveis e tipos de dados

Variáveis são alocações de memória nas quais podemos guardar dados. Elas têm um nome, tipo e valor. Toda vez que necessite usar de uma variável você precisa declará-la e só então poderá atribuir valores a mesma.

2.1.1. Declarando variáveis

As declarações de variáveis consistem de um tipo e um nome de variável: como segue o exemplo:

```
int idade;  
String nome;  
boolean existe;
```

Os nomes de variáveis podem começar com uma letra, um sublinhado (_), ou um cifrão (\$). Elas não podem começar com um número. Depois do primeiro caracter pode-se colocar qualquer letra ou número.

2.1.2. Tipos de variáveis

Toda variável deve possuir um tipo. Os tipos que uma variável pode assumir uma das três “coisas” a seguir:

- Uma das oito primitivas básicas de tipos de dados
- O nome de uma classe ou interface
- Um Array

Veremos mais sobre o uso de arrays e classes mais a frente.

Os oito tipos de dados básicos são: inteiros, números de ponto-flutuante, caracteres e booleanos (verdadeiro ou falso).

Tipos Inteiros:

Tipo	Tamanho	Alcance
byte	8 bits	-128 até 127
short	16 bits	-32.768 até 32.767
int	32 bits	-2.147.483.648 até 2.147.483.647
long	64 bits	-9223372036854775808 até 9223372036854775807

Existem dois tipos de números de ponto-flutuante: float (32 bits, precisão simples) e double (64 bits, precisão dupla).

2.1.3. Atribuições a variáveis

Após declarada uma variável a atribuição é feita simplesmente usando o operador '=':

```
idade = 18;  
existe = true;
```

2.1.4 Constantes

Para declarar uma constante, use a palavra chave final antes da declaração da variável e inclua um valor inicial para esta variável. Exemplo:

```
final float pi=4.141592;  
final boolean debug=false;  
final int maxsize = 40000;
```

2.2. Comentários

Java possui três tipos de comentário, o /* e */ como no C e C++. Tudo que estiver entre os dois delimitadores são ignorados:

```
/*  
Este comentário ficará visível somente no código o compilador ignorará  
completamente este trecho entre os delimitadores  
*/
```

Duas barras (//) também podem ser usadas para se comentar uma linha:

```
int idade; // este comando declara a variável idade
```

E finalmente os comentários podem começar também com `/**` e terminar com `*/`. Este comentário é especial e é usado pelo javadoc e para gerar uma documentação API do código. Para aprender mais sobre o javadoc acesse a home page (<http://www.javasoft.com>).

2.3. Caracteres especiais

Caracter	Significado
<code>\n</code>	Nova Linha
<code>\t</code>	Tab
<code>\b</code>	Backspace
<code>\r</code>	Retorno do Carro
<code>\f</code>	“Formfeed” (avança página na impressora)
<code>\\</code>	Barra invertida
<code>\'</code>	Apóstrofe
<code>\"</code>	Aspas
<code>\ddd</code>	Octal
<code>\xdd</code>	Hexadecimal

2.4. Expressões e operadores

2.4.1. Operadores Aritméticos

Operador	Significado	Exemplo
+	soma	3 + 4
-	subtração	5 - 7
*	multiplicação	5 * 5
/	divisão	14 / 7
%	modulo	20 % 7

Exemplo Aritmético:

```

class ArithmeticTest {

    public static void main ( Strings args[] ) {
        short x = 6;
        int y = 4;
        float a = 12.5f;
        float b = 7f;
        System.out.println ( "x é " + x + ", y é " + y );
        System.out.println ( "x + y = " + (x + y) );
        System.out.println ( "x - y = " + (x - y) );
        System.out.println ( "x / y = " + (x / y) );
        System.out.println ( "x % y = " + ( x % y ) );
        System.out.println ( "a é " + a + ", b é " + b );
        System.out.println ( " a / b = " + ( a / b ) );
    }
}
    
```

A saída do programa acima é :

```
x é 6, y é 4  
x + y = 10  
x - y = 2  
x / y = 1  
x % y = 2  
a é 12.5, b é 7  
a / b = 1.78571
```

2.4.2. Mais sobre atribuições

Variáveis podem atribuídas em forma de expressões como:

```
int x, y, z;  
x = y = z = 0;
```

No exemplo as três variáveis recebem o valor 0;

Operadores de Atribuição:

Expressão	Significado
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$

2.4.3. Incrementos e decrementos

Como no C e no C++ o Java também possui incrementadores e decrementadores :

```
y = x++;
y = --x;
```

As duas expressões dão resultados diferentes, pois existe uma diferença entre prefixo e sufixo. Quando se usa os operadores (`x++` ou `x--`), `y` recebe o valor de `x` antes de `x` ser incrementado, e usando o prefixo (`++x` ou `--x`) acontece o contrário, `y` recebe o valor incrementado de `x`.

2.5. Comparações

Java possui várias expressões para testar igualdade e magnitude. Todas as expressões retornam um valor booleano (`true` ou `false`).

2.5.1. Operadores de comparação

Operador	Significado	Exemplo
<code>==</code>	Igual	<code>x == 3</code>
<code>!=</code>	Diferente (Não igual)	<code>x != 3</code>
<code><</code>	Menor que	<code>x < 3</code>
<code>></code>	Maior que	<code>x > 3</code>
<code><=</code>	Menor ou igual	<code>x <= 3</code>
<code>>=</code>	Maior ou igual	<code>x >= 3</code>

2.5.2. Operadores lógicos

Operador	Significado
&&	Operação lógica E (AND)
	Operação lógica OU (OR)
!	Negação lógica
&	Comparação bit-a-bit E (AND)
	Comparação bit-a-bit OU (OR)
^	Comparação bit-a-bit OU-Exclusivo (XOR)
<<	Deslocamento a esquerda
>>	Deslocamento a direita
>>>	Deslocamento a direita com preenchimento de zeros
-	Complemento bit-a-bit
x <<= y	Atribuição com deslocamento a esquerda (x = x << y)
x >>= y	Atribuição com deslocamento a direita (x = x >> y)
x >>>= y	Atribuição com deslocamento a direita e com preenchimento de zeros (x = x >>> y)
x &= y	atribuição AND (x = x & y)
x = y	atribuição OR (x = x y)
x ^= y	atribuição XOR (x = x ^ y)