

Curso de Java

Variáveis, Tipos e Expressões

Prof. Fernando Vanini

IC – Unicamp

- Variáveis e memória
- Rigidez de tipos
- Tipos simples
- Tipos inteiros em Java
- Tipo booleano
- Declaração de variável
- Comando de atribuição
- Expressões
- Operadores

- Uma *variável* é uma área de memória, associada a um *nome*, que pode armazenar valores de um determinado *tipo*.
- Um *tipo de dado* define um conjunto de *valores* e um conjunto de *operações*.
- Uma variável de um certo tipo T pode conter, num certo instante, um valor pertencente ao tipo T.

- Java é uma linguagem com rigidez de tipos (em inglês *strongly typed*).
 - um valor pertencente a um determinado tipo só pode ser usado como argumento em operações que prevêem operações desse tipo.

- Os tipos simples ou tipos primitivos em Java são:
 - o tipo booleano
 - tipos numéricos
 - inteiros
 - ponto flutuante
 - decimais

- Java oferece diversos tipos de inteiros. Cada um deles é definido por:
 - uma gama de valores
 - uma representação interna
- O programador tem a opção de decidir qual o tipo inteiro a ser usado em função das necessidades de sua aplicação.

Tipo	Tamanho (bits)	Intervalo de Valores
byte	8	-128 a 127 ($-(2^7)$ a 2^7-1)
short	16	-32768 a 32767 ($-(2^{15})$ a $2^{15}-1$)
int	32	$-(2^{31})$ a $2^{31}-1$
long	64	$-(2^{63})$ a $2^{63}-1$

- O tipo booleano, em Java chamado de **boolean**, e em geral é utilizado para operações lógicas.
- Os valores possíveis para uma variável do tipo **boolean** são **true** e **false**.

- Em Java a declaração de uma variável tem a seguinte estrutura:

```
<tipo> nome;
```

OU

```
<tipo> nome = valor inicial;
```

- Exemplos

```
String n;
```

```
int x = 10; boolean b = false;
```

- O comando de atribuição é usado para alterar o valor de uma variável.
- Sua forma:

<variável> = <expressão>;

- Exemplos:

x = z + 10;

z = z + 1;

b = true;

Uma declaração da forma

int x = 10;

é na verdade a combinação de uma declaração de variável com um comando de atribuição.

- Uma expressão é uma combinação de operandos e operadores.
- Expressões em Java são semelhantes às expressões usadas em outras linguagens como VB, C ou Pascal.

- No caso de operações encadeadas, como em $a+b*c$ o cálculo da expressão é feito de acordo com a precedência entre os operadores.
- Parênteses podem ser utilizados para alterar a ordem de cálculo das operações.
- Exemplos:

$$(a+b) * c$$

e

$$a + (b * c)$$

- No caso de encadeamento de operações com a mesma precedência, a linguagem define o modo de associatividade (à esquerda ou à direita). Os operadores aritméticos em Java têm associatividade à esquerda.
- Exemplo:
 $6/2/3$ é calculado como $(6/2)/3$ pois neste caso a associatividade é à esquerda. Em Java, apenas os operadores de atribuição ($=$ $*=$ $/=$ $\%=$ $+=$ $-=$ $<<=$ $>>=$ $\&=$ $\^=$ $\|=$) e o operador ternário ($? :$), têm associatividade à direita.

- Principais operadores, em ordem decrescente de precedência

Categoria	Operadores	Associatividade
Unário	+ - !	esquerda
Multiplicativo	* / %	esquerda
Aditivo	+ -	esquerda
Relacional	< = > >= <= == is	esquerda
Igualdade	== !=	esquerda
'and' (bool)	&&	esquerda
'ou' (bool)		esquerda

```
public class Opr
{
    public static void main(String[] args) {
        boolean t = true;
        boolean f = false;
        System.out.println("t:" + t + " f:" + f);
        System.out.println("t || f:" + t || f );
    }
}
```

```
/* Operadores relacionais */
public class Opr2 {
    public static void main(string[] args){
        boolean gt,lt,eq;
        int a = 100; int b = 333;
        gt = a > b;
        lt = a < b;
        eq = a == b;
        System.out.println("a > b:"+gt);
        System.out.println("a < b:"+lt);
        System.out.println("a == b:"+eq);
    }
}
```



```
/* Operadores aritméticos */  
class Opr3 {  
    public static void main(String[] args) {  
        int a = 127, b = 16, c = 4;  
        int d = a/b/c;    /* (a/b)/c */  
        int m = a+b*c;    /* a+(b*c) */  
        System.out.println("a/b/c:"+d);  
        System.out.println("a+b*c:"+m);  
    }  
}
```

- O operador de atribuição pode ser combinado com outros operadores aritméticos.
- $a \text{ op} = \text{exp};$ é equivalente a $a = a \text{ op} (\text{exp});$
- Exemplos:
 - $a += b;$ equivalente a $a = a + b;$
 - $a -= 2;$ equivalente a $a = a - 2;$
 - $a *= 1+1;$ equivalente a $a = a * (1+1);$
 - $a \% = b*c+d;$ equivalente a $a = a \% (b*c+d);$

- Numa expressão, os operadores `++` e `--` podem ser usados antes ou depois da variável:
 - se usado após a variável, o incremento será feito depois de usar o valor da variável na expressão
 - se usado antes da variável, o incremento será feito antes do uso do valor na expressão
- Exemplos:
 - `a = a/i++;` equivalente a `a = a/i; i = i+1;`
 - `b = --k*2;` equivalente a `k=k-1; b = k*2;`