

## Variáveis Primitivas e Controle de Fluxo

### Capítulo II

## Algumas Informações

- ✓ Inferência de tipo
  - Em algumas linguagens não é necessário informar o tipo da variável em sua criação. Ex: Python, JavaScript e MatLab.
- ✓ Case Sensitive
  - Diferencia letras maiúsculas de minúsculas
- ✓ Tipagem Forte e Fraca
  - Uma linguagem se denomina de tipagem forte quando não é possível realizar operações com variáveis de tipos diferentes.
  - Tipagem Forte: Java e Python.
  - Tipagem Fraca: PHP e JavaScript.

## Variáveis

No Java existem dois tipos de variáveis, as **Primitivas** e as de **Referência**

- Primitivas: nome associado a um espaço de memória.

```
int valor;           double pi = 3.14159;
```

- Referência: nome associado a um ponteiro que aponta para um espaço de memória. Referenciando um objeto.

```
String palavra = "Oi";      Carro fusca;
```

## Comentários

```
// Comentar é essencial!
```

```
/* Existem diversas formas de comentar um código */
```

```
/**
```

```
 * Nunca esqueça dos comentários.
```

```
 * Eles vão te ajudar muito!
```

```
 */
```

## Algumas convenções Java para variáveis

- ✓ Uma declaração de variável por linha, seguida de comentário;

```
float nota;        // Nota do aluno  
float media;       // Média das notas
```

- ✓ Nome de variáveis: devem começar com letras (minúsculas), \$ ou \_; Se for um nome composto as primeiras letras das palavras subsequentes devem ser maiúsculas;

```
double saldoContaCorrente;
```

- ✓ Nome de constantes: devem ser sempre maiúsculas. Usa-se underline ( \_ ) para separar nomes compostos;

```
final int CREDITOS_MATRICULA = 27;
```

## Tipos Primitivos em Java

Classificação	Tipo	Tamanho	Descrição
Inteiro	<code>byte</code>	8 bits	Utilizado quando o consumo de memória é importante, principalmente em vetores (-127 a 127)
	<code>short</code>	16 bits	Utilizado quando o consumo de memória é importante, principalmente em vetores (-32.768 a 32.767)
	<code>int</code>	32 bits	Bastante utilizado para representar números inteiros (-2147483648 a 2147483647)
	<code>long</code>	64 bits	Utilizando quando o intervalo necessário excede os valores de <code>int</code> ( $-2^{63}$ a $2^{63}-1$ )
Ponto flutuante	<code>float</code>	32 bits	Precisão simples com até 7 dígitos decimais (de acordo com a <b>IEEE 754-2008</b> )
	<code>double</code>	64 bits	Precisão simples com até 15 dígitos decimais (de acordo com a <b>IEEE 754-2008</b> )
Lógico	<code>boolean</code>	indefinido	Utilizado para para flag simples, <code>true</code> e <code>false</code>
Caractere	<code>char</code>	16 bits	Utilizada para armazenar um caractere Unicode, a partir da versão 1.5 (ou 2SE 5.0 o UTF-16 é utilizado)

**$-2^{(\text{numero de bits} - 1)}$  até  $2^{(\text{numero de bits} - 1)} - 1$**

## Classe String

**String** não é um tipo de variável e sim uma classe. Por isso ela é escrita com a primeira letra maiúscula. Esta classe é dotada de vários atributos e métodos que facilitam muito a nossa vida.

[Manipulando métodos da classe String](#)

[Trabalhando String em Java](#)

## Exemplos de promoção e casting ( ou conversion )

```
float numFloat;  
double numDouble;  
// Atribuindo um número de 9 dígitos para float  
numFloat = 1.123456789f;  
// Atribuindo um número de 9 dígitos para double  
numDouble = 1.123456789;  
System.out.println("Float com 9 dígitos: " + numFloat);  
System.out.println("Double com 9 dígitos: " + numDouble);  
  
// Como funciona uma promoção (de inteiro para double)  
numDouble = 10;  
System.out.println("Promoção de um inteiro em Java: " + numDouble);  
  
// Como funciona um casting  
// Podemos usar o underscore para deixar o código mais legível  
numDouble = 10.123_45;  
numFloat = (float) numDouble;  
System.out.println("Casting de um número em Java" + numFloat);
```



## Tabela casting

<b>DE \ PARA</b>	<b><i>byte</i></b>	<b><i>short</i></b>	<b><i>char</i></b>	<b><i>int</i></b>	<b><i>long</i></b>	<b><i>float</i></b>	<b><i>double</i></b>
<b><i>byte</i></b>		Implícito	char	Implícito	Implícito	Implícito	Implícito
<b><i>short</i></b>	byte		char	Implícito	Implícito	Implícito	Implícito
<b><i>char</i></b>	byte	short		Implícito	Implícito	Implícito	Implícito
<b><i>int</i></b>	byte	short	char		Implícito	Implícito	Implícito
<b><i>long</i></b>	byte	short	char	int		Implícito	Implícito
<b><i>float</i></b>	byte	short	char	int	long		Implícito
<b><i>double</i></b>	byte	short	char	int	long	float	

## Conhecendo o *Printf*

Conversor	Descrição
d	Representa decimal
f	Representa ponto flutuante
s	Representa uma string
n	Quebra de linha (Recomendado pela Oracle ao invés de \n)
06	Fixa a saída em 6 caracteres, adicionando zeros se necessário
+	Exibe o sinal (positivo ou negativo)
.3	Exibe 3 casas depois da vírgula
8.3	Exibe 8 casas

## Exemplos

```
double pi = 3.141_592_653;
int decimal = 123456;
double numDouble = 12345.123456;
String texto = "Teste formatação!";

System.out.format("%d %n", decimal);
System.out.format("%f %n", pi);
System.out.format("%s %n", texto);

System.out.printf("%+d %n", decimal);
System.out.printf("%08d %n", decimal);
System.out.printf("%8.3f %n", numDouble);
```

## Entrada de dados

Para entrada de dados via dispositivo padrão (terminal) utilizamos da classe Scanner

Não se esqueça do import!

```
import java.util.Scanner;
```

```
Scanner teclado = new Scanner (System.in);
```

```
System.out.print("Entre com o byte desejado: ");
```

```
byte b = teclado.nextByte();
```

```
System.out.print("Entre com o short desejado: ");
```

```
short s = teclado.nextShort();
```

```
System.out.print("Entre com o int desejado: ");
```

```
int i = teclado.nextInt();
```

```
System.out.print("Entre com o long desejado: ");
```

```
long l = teclado.nextLong();
```

```
System.out.print("Entre com o float desejado: ");
```

```
float f = teclado.nextFloat();
```

```
System.out.print("Entre com o double desejado: ");
```

```
double d = teclado.nextDouble();
```

```
System.out.print("Entre com a String desejada: ");
```

```
String texto = teclado.nextLine();
```

## Entrada de dados

Para ler variáveis do tipo char, nós utilizamos uma String auxiliar e pegamos apenas sua primeira letra e convertemos em char através do método charAt.

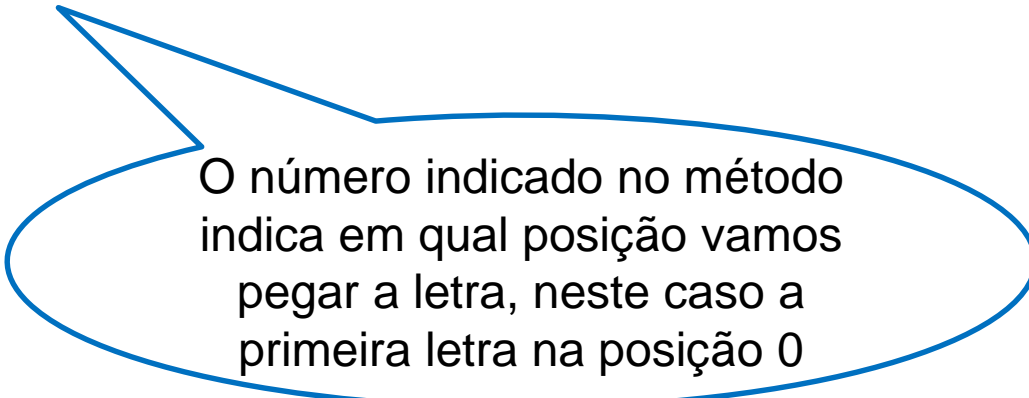
```
Scanner teclado = new Scanner(System.in);
```

```
String leitura;
```

```
char letra;
```

```
leitura = teclado.nextLine();
```

```
letra = leitura.charAt(0);
```



O número indicado no método indica em qual posição vamos pegar a letra, neste caso a primeira letra na posição 0

## Controle de Fluxo

```
if ( num1 == num2 )
    System.out.println("Os numeros sao iguais");
if ( num1 != num2 )
    System.out.println("Os numeros sao diferentes");
if ( num1 > num2 )
    System.out.println("num1 é maior que num2");
if ( num1 < num2 )
    System.out.println("num2 é maior que num1");
if ( num1 >= num2 )
    System.out.println("num1 é maior ou igual à num2");
```

Operador	Descrição
==	Igual à
!=	Diferente de
>	Maior que
>=	Maior ou igual à
<	Menor que
<=	Menor ou igual à

Operador	Descrição
&&	E
	OU

```
if ( (media >= 60) && (freq > 0.75) ) {
    System.out.println("Aluno Aprovado!");
} else if ( (media < 30) || (freq < 0.75) ){
    System.out.println("Aluno Reprovado!");
} else {
    System.out.println(" NP3 !");
}
```

## Operador Ternário

Muito útil quando se deseja atribuir um valor a uma variável dependendo de uma condição

```
// a e b são int e maior é uma String
if ( a > b ) {
    maior = "A é maior que B";
} else {
    maior = "B é maior ou igual a A";
}
```

```
// Operador ternário
maior = ( a > b )? "A é maior que B": "B é maior ou igual a A";
```

## Estruturas de repetição

### While

```
int contador = 0;

System.out.println("Imprimindo valores impares" +
    " de 0 a 30");

while ( contador < 30 ) {
    if ( contador % 2 == 1 ) {
        System.out.println( contador );
    }
    contador ++;
}
```

Operador	Descrição
++	Incremento
+=	Incremento
--	Decremento
-=	Decremento



## Estruturas de repetição

### For

```
System.out.println("Imprimindo valores pares" +  
    " de 0 a 30");  
  
for (int i = 0; i < 30; i++) {  
    if ( i%2 == 0) {  
        System.out.println( i );  
    }  
}
```

## Estruturas de repetição

### Switch-Case

```
Scanner teclado = new Scanner(System.in);  
// Pega o valor digitado e salva na variável  
String aux = teclado.nextLine();  
// Armazena somente a primeira letra da variável aux  
char conceito = aux.charAt(0);
```

```
switch (conceito) {  
    case 'A':  
        System.out.println("Média boa!");  
        break;  
    case 'B':  
        System.out.println("Média Razoável");  
        break;  
    case 'C':  
        System.out.println("Média ruim!");  
    default:  
        System.out.println("Média inválida");  
}
```

Caso nenhuma das opções seja a correta o bloco default é executado

## Operadores

Operador	Descrição
=	Atribuição
!	Complemento (Inverte valor de boolean)
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo
&	Lógica E (bit a bit)
	Lógica OU (bit a bit)
^	Lógica XOR (bit a bit)
~	Inversor (bit a bit)

Aplicável  
somente em  
inteiros



## Mateiral de Apoio

[Variáveis Primitivas e Controle de Fluxo - Caelum](#)

## Exercícios

1. De acordo com a opção do usuário imprima a tabela verdade do operador desejado. ( E ou OU ).
2. Leia 4 valores inteiros A, B, C e D. A seguir, se B for maior do que C e se D for maior do que A, e a soma de C com D for maior que a soma de A e B e se C e D, ambos, forem positivos e se a variável A for par escrever a mensagem "Valores aceitos", senão escrever "Valores nao aceitos". (fonte: **URI 1035**)
3. Crie um programa que calcule o IMC (Índice de Massa Corporal) e mostre o grau de obesidade de acordo com o resultado. Fórmula do imc = ( peso / altura<sup>2</sup> ). Dica: pesquise sobre a *Tabela IMC* para saber os graus de obesidade.
4. Faça uma calculadora que dado 2 números informados pelo usuário realize as seguintes operações (de acordo com a opção do usuário): soma, subtração, multiplicação, divisão e exponencial.

## Exercícios

5. Faça um programa que permita ao usuário entrar com as notas de NP1 e NP2. Após ler estas notas, mostre a média final em um número inteiro.
6. Dentro de uma determinada empresa, alguns funcionários tem seus salários baseados no Salário Mínimo e outros funcionários ganham por hora. Dado que o Salário Mínimo atual é de R\$ 998,00 e o preço da hora de trabalho é de R\$ 9,30, calcule o salário mensal dos seguintes funcionários:
  - Gerente : 3 salários mínimos.
  - Engenheiro : 7 salários mínimos.
  - Serviços terceirizados : 8h/dia
7. Dado um valor em R\$, calcule quantas viagens de ônibus são possíveis. Suponha que o preço seja R\$ 2,75 cada passagem.

**Obrigado!**