

Variáveis Primitivas e Controle de Fluxo

Capítulo II

Variáveis

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

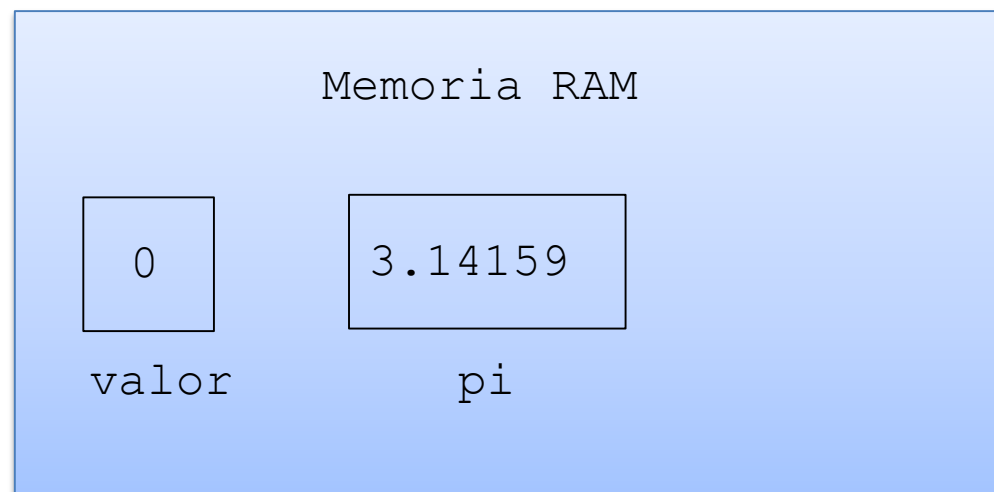
- Primitivas: nome associado a um espaço de memória.

```
int valor;
```

<tipo> <nome>

```
double pi = 3.14159;
```

<tipo> <nome> <valor inicial>



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 IEEE 754-2008)
	<code>double</code>	64 bits	Precisão simples com até 15 dígitos decimais (de acordo com a IEEE 754-2008)
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 $2^{(\text{numero de bits} - 1)} - 1$

Variáveis

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

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

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

Algumas convenções Java para variáveis (Boas práticas)

- ✓ 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;
- ✓ Utilize o padrão: "camel case";

```
double saldoContaCorrente;
```



- ✓ Nome de constantes: devem ser sempre maiúsculas. Usa-se underline (_) para separar nomes compostos;
- ```
final int CREDITOS_MATRICULA = 27;
```

Algumas convenções Java para variáveis (Boas práticas)

- ✓ Quando inicializar uma variável sempre indique o tipo do número, caso ela não seja do tipo inteiro (int);
- ✓ Para float:

```
float a = 10f;
float b = 10.2f;
```

- ✓ Para double:

```
double a = 10.0;
double b = 10.2;
```

Exemplos de promoção e casting ( ou conversion )

✓ Vamos analisar o seguinte trecho de código:

✓ **Exemplo 01:**

```
int a = 5;
int b = 2;
double resultado;

resultado = a / b;
```

```
System.out.println("Resultado: " + resultado);
```

POR QUE?

O resultado esperado seria: "Resultado: 2.5"

Porém na saída obtemos o valor: "Resultado: 2.0"

Resultado: 2.0

BUILD SUCCESSFUL (total time: 0 seconds)



Isso acontece pois as variáveis do tipo "a" e "b" são ambas do tipo inteiro, assim o compilador identifica que o resultado esperado também deve ser do tipo inteiro.

✓ Como resolvemos este problema?

✓ **Solução exemplo 01:**

```
int a = 5;
int b = 2;
double resultado;
```

Utilizamos o "casting".  
Transformamos o resultado agora em double!

```
resultado = (double) a / b;
```

```
System.out.println("Resultado: " + resultado);
```

```
Resultado: 2.5
```

```
BUILD SUCCESSFUL (total time: 1 second)
```

Exemplos de promoção e casting ( ou conversion )

- ✓ Vamos analisar o seguinte trecho de código:
- ✓ **Exemplo 02:**

```
double a = 5.5;
```

```
int b;
```

```
b = a;
```

Erro!  
Não é possível  
converter de double  
para int

```
System.out.println("Resultado: " + b);
```

```
b = a;
```

1 error

BUILD FAILED (total time: 1 second)

Isso acontece pois uma variável do tipo **double** não pode ser atribuída a uma variável do tipo **int**.

- ✓ Como resolvemos este problema?
- ✓ **Solução exemplo 02:**

```
double a = 5.5;
int b;
```

Utilizamos o  
"casting"  
novamente!

```
b = (int) a;
```

```
System.out.println("Resultado: " + b);
```

Resultado: 5

BUILD SUCCESSFUL

Porém, perceba que houve  
uma perda de informação ao  
fazermos o casting.

```
)
```

## 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;
```

Sempre que terminar de utilizar o objeto "sc", é necessário fechar este recurso.

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

```
System.out.println("Entre com um valor de byte: ");
```

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

```
System.out.println("Entre com um valor de short: ");
```

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

```
System.out.println("Entre com um valor de int: ");
```

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

```
System.out.println("Entre com um valor de long: ");
```

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

```
System.out.println("Entre com um valor de float: ");
```

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

```
System.out.println("Entre com um valor de double: ");
```

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

```
System.out.println("Entre com uma cadeia de String: ");
```

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

```
System.out.println("Entre com uma cadeia de String sem espaço: ");
```

```
String textoSemEspaco = sc.next();
```

```
sc.close();
```

## Entrada de dados

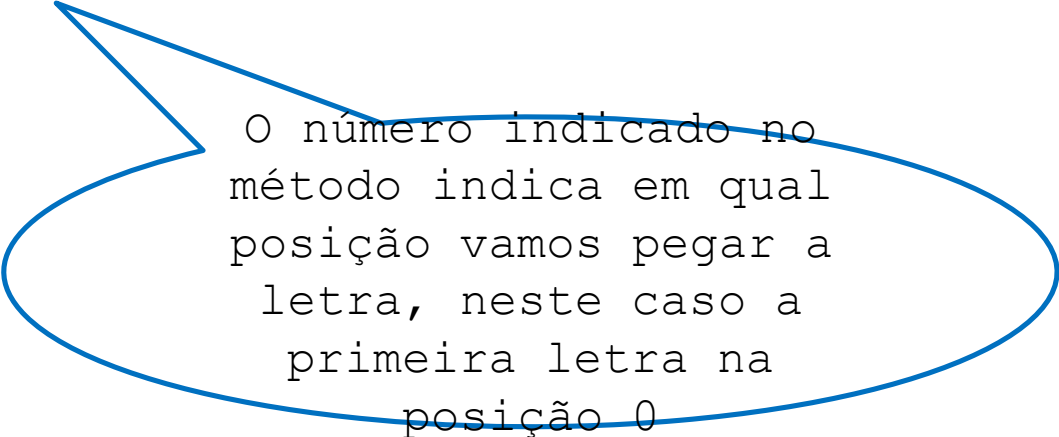
Para ler variáveis do tipo char, nós utilizamos uma String auxiliar e pegamos apenas sua primeira letra e convertemos em char através do método charAt.

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

```
String leitura = sc.nextLine();
```

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

```
sc.close();
```



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-ELSE

```
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 |
|----------|-----------|
| &&       | E         |
|          | OU        |

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

```
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 !");
}
```

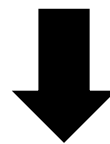
## Controle de Fluxo

Expressões condicionais ternárias (Opcional ao if-else)

Exemplo:

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

IF-ELSE  
tradicional



```
System.out.println((media >= 60 && freq > 0.75 ? "Aprovado" : "NP3"));
```

Condição

verdadeiro

falso

Como se lê: "a condição é verdadeira (?), se sim "Aprovado", caso contrário (:) "NP3".

## Estruturas de repetição

### While

```
int contador = 0;

System.out.println("Imprimindo valores ímpares" +
 " 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 sc = new Scanner(System.in);
int op; //opção

op = sc.nextInt();
switch(op) {
 case 1:
 System.out.println("Opção 01");
 break;

 case 2:
 System.out.println("Opção 02");
 break;

 case 3:
 System.out.println("Opção 03");
 break;

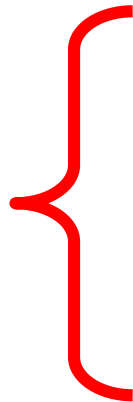
 default:
 System.out.println("Opção inválida");
 break;
}
sc.close();
```

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

## Operadores

| Operador | Descrição                                      |
|----------|------------------------------------------------|
| =        | Atribuição                                     |
| !        | Complemento (Inverte valor de <b>boolean</b> ) |
| +        | 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. 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> ).

| IMC               | Classificação              |
|-------------------|----------------------------|
| abaixo de 18,5    | abaixo do peso             |
| entre 18,6 e 24,9 | Peso ideal (parabéns)      |
| entre 25,0 e 29,9 | Levemente acima do peso    |
| entre 30,0 e 34,9 | Obesidade grau I           |
| entre 35,0 e 39,9 | Obesidade grau II (severa) |
| acima de 40       | Obesidade III (mórbida)    |



## Exercícios

3. 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.
4. Escreva um programa que repita a leitura de uma senha até que ela seja válida. Para cada leitura de senha incorreta informada, escrever a mensagem "Senha Invalida". Quando a senha for informada corretamente deve ser impressa a mensagem "Acesso Permitido" e o algoritmo encerrado. Considere que a senha correta é o valor 2002.

**Obrigado!**