



Java Fundamentos

RECODE
pro



Módulo 03 – Java Fundamentos Aula 01



A plataforma e a Linguagem Java

A Plataforma Java é o nome dado ao ambiente computacional, ou plataforma, criada pela empresa estadunidense Sun Microsystems, e vendida para a Oracle depois de alguns anos. A plataforma permite desenvolver programas utilizando a linguagem de programação Java. Uma grande vantagem da plataforma é a de não estar presa a um único sistema operacional ou hardware, pois seus programas rodam através de uma máquina virtual que pode ser emulada em qualquer sistema usando versões diferentes e adaptadas da JVM (Java Virtual Machine) para o sistema que vai receber a aplicação.

Módulo 03 – Java Fundamentos Aula 01

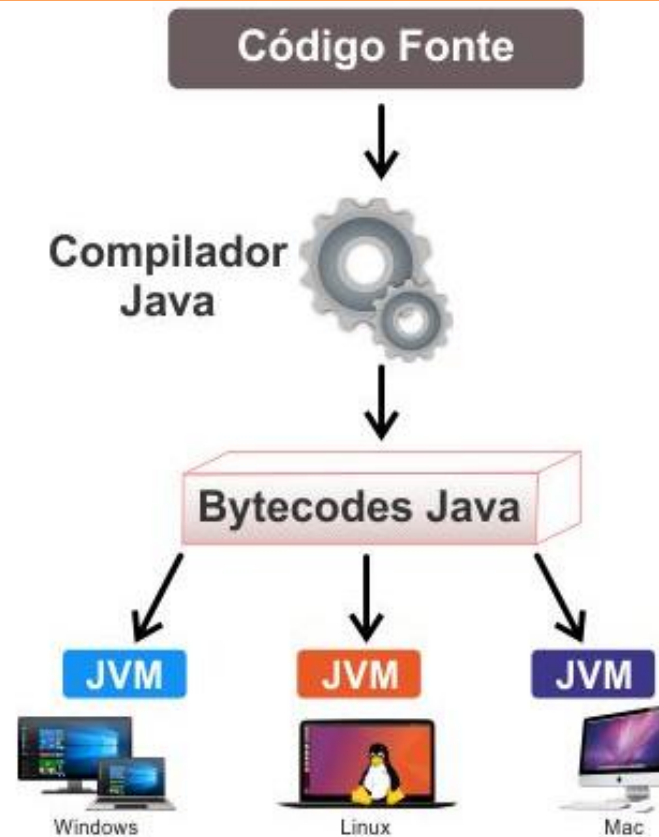


JVM - Java Virtual Machine

JVM é um aplicativo de software que simula um computador, mas oculta o sistema operacional e o hardware subjacentes dos programas que interagem com ela. Se a mesma máquina virtual é implementada em muitas plataformas de computador, os aplicativos escritos para ela podem ser utilizados em todas essas plataformas. A JVM é uma das máquinas virtuais mais utilizadas. O .NET da Microsoft utiliza uma arquitetura de máquina virtual semelhante



Módulo 03 – Java Fundamentos Aula 01



Na linguagem de programação Java, todos os códigos-fonte são primeiramente escritos em um arquivo de texto com a extensão .java. Esses arquivos-fonte são compilados nos arquivos .class pelo compilador javac. O arquivo .class não contém código que é nativo para seu processador; ele em seu lugar contém bytecodes – a linguagem de máquina da Java Virtual Machine.

Módulo 03 – Java Fundamentos Aula 01



JDK - Kit de Desenvolvimento Java

O JDK é Kit que provê ferramentas para o desenvolvimento de programas Java. Ou seja, ele contém um compilador, um depurador e o próprio JRE para você executar os seus programas.



Módulo 03 – Java Fundamentos Aula 01



Versões do Java

- Java SE (Java Platform, Standard Edition). É a base da plataforma. Inclui o ambiente de execução e as bibliotecas comuns.
- Java EE (Java Platform, Enterprise Edition). A edição voltada para o desenvolvimento de aplicações corporativas e para internet.
- Java ME (Java Platform, Micro Edition). A edição para o desenvolvimento de aplicações para dispositivos móveis e embarcados.



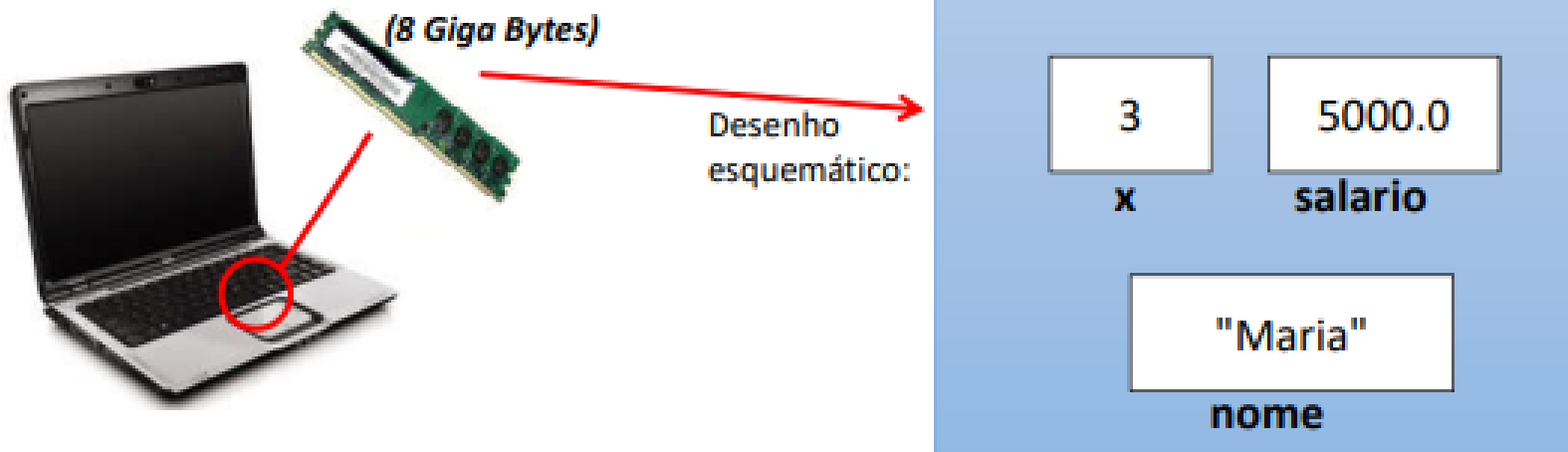
Módulo 03 – Java Fundamentos Aula 01



Variáveis

Definição informal:

Em programação, uma variável é uma porção de memória (RAM) utilizada para armazenar dados durante a execução dos programas.



Módulo 03 – Java Fundamentos Aula 01



Tipos de dados no Java

Tipo	Descrição	Tamanho
byte	Valor inteiro entre -128 e 127 (inclusivo)	1 byte
short	Valor inteiro entre -32.768 e 32.767 (inclusivo)	2 byte
int	Valor inteiro entre -2.147.483.648 e 2.147.483.647 (inclusivo)	4 byte
long	Valor inteiro entre -9.223.372.036.854.775.808 e 9.223.372.036.854.775.807 (inclusivo)	8 byte
float	Valor componto flutuante entre 1,40129846432481707E10j45 e 3,40282346638528860 E 1038 (positivo ou negativo)	4 byte
double	Valor componto flutuante entre 4,94065645841246544E10j324 e 1,79769313486231570E10308 (positivo ou negativo)	8 byte
boolean	true ou false	1 bit
char	Um único caractere Unicode de 16 bits. Valor inteiro e positivo entre 0 (ou '\u0000') e 65.535 (ou '\uffff')	2 bytes
String	Cadeia de caracteres (palavras ou textos)	

Módulo 03 – Java Fundamentos Aula 01



Declaração de variáveis em JAVA

Sintaxe:

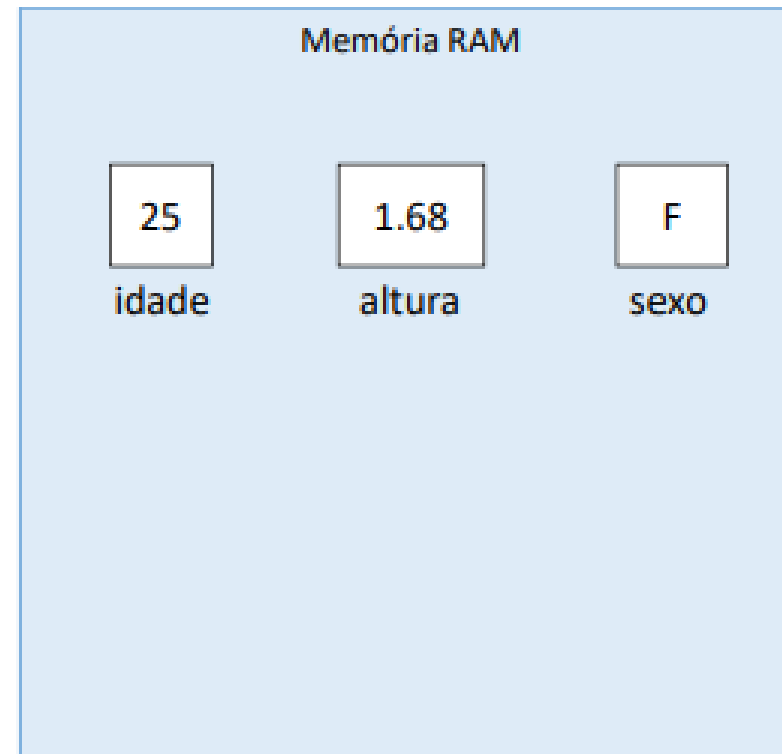
`<tipo> <nome> = <valor inicial>;`
(opcional)

Exemplos:

```
int idade = 25;  
double altura = 1.68;  
char sexo = 'F';
```

Uma variável possui:

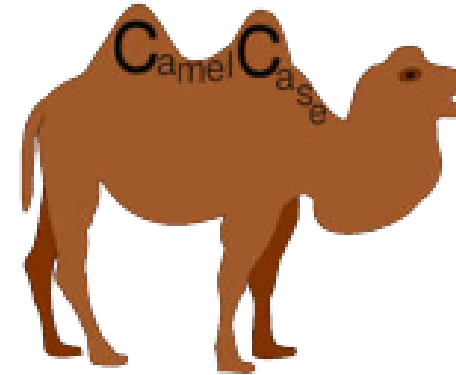
- Nome (ou identificador)
- Tipo
- Valor
- Endereço





Nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não pode ter espaço em branco
- Não usar acentos ou til
- Sugestão: use o padrão "camel case"



Errado:

```
int 5minutos;  
int salário;  
int salário do funcionario;
```

Correto:

```
int _5minutos;  
int salario;  
int salarioDoFuncionario;
```

Módulo 03 – Java Fundamentos Aula 01



Inicialização

Toda variável deve ser inicializada antes de ser utilizada pela primeira vez. Se isso não for realizado, ocorrerá um erro de compilação. A inicialização é realizada através do operador de atribuição =. Esse operador guarda um valor em uma variável.

```
// Declarações
int numero ;
double valor ;
// Inicialização
numero = 20;
// Uso Correto
System.out.println(numero);
// Erro de compilação
System.out.println (valor);
```

Módulo 03 – Java Fundamentos Aula 01



Operadores em Java

- Aritmético (+, -, *, /, %)
- Atribuição (=, +=, -=, *=, /=, %=)
- Relacional (==, !=, <, <=, >, >=)
- Lógico (&&, ||)

Módulo 03 – Java Fundamentos Aula 01



Operadores Aritméticos

ADIÇÃO

```
System.out.println(24 + 6);
```

Ou

```
Int a = 24;
```

```
Int b = 6;
```

```
System.out.println(a + b);
```

SUBTRAÇÃO

```
System.out.println(20 - 15);
```

Ou

```
Int a = 20;
```

```
Int b = 15;
```

```
System.out.println(a - b);
```

MULTIPLICAÇÃO

```
System.out.println(3 * 5);
```

Ou

```
Int a = 3;
```

```
Int b = 5;
```

```
System.out.println(a * b);
```

MULTIPLICAÇÃO

```
System.out.println(40 / 4);
```

Ou

```
Int a = 40;
```

```
Int b = 4;
```

```
System.out.println(a / b);
```

Módulo 03 – Java Fundamentos Aula 01



Operadores de Atribuição

- Simples =
- Incremental +=
- Decremental -=
- Multiplicativa *=
- Divisória /=
- Modular %=

```
int valor = 1; // valor = 1
valor += 2; // valor = 3
valor -= 1; // valor = 2
valor *= 6; // valor = 12
valor /= 3; // valor = 4
valor %= 3; // valor = 1
```

As instruções ao lado poderiam ser escritas de outra forma:

```
int valor = 1; // valor = 1
valor = valor + 2; // valor = 3
valor = valor - 1; // valor = 2
valor = valor * 6; // valor = 12
valor = valor / 3; // valor = 4
valor = valor % 3; // valor = 1
```

Módulo 03 – Java Fundamentos Aula 01



Operadores Relacional

- Igualdade ==
- Diferença !=
- Menor <
- Menor ou igual <=
- Maior >
- Maior ou igual >=

```
int valor = 2;
```

```
boolean t = false ;
```

```
t = ( valor == 2); // t = true
```

```
t = ( valor != 2); // t = false
```

```
t = ( valor < 2); // t = false
```

```
t = ( valor <= 2); // t = true
```

```
t = ( valor > 1); // t = true
```

```
t = ( valor >= 1); // t = true
```

Módulo 03 – Java Fundamentos Aula 01



Operadores Lógicos

A linguagem Java permite verificar duas ou mais condições através de operadores lógicos. Os operadores lógicos devolvem valores do tipo primitivo boolean. Os operadores lógicos são:

- “E” lógico &&
- “OU” lógico ||

```
int valor = 30;
```

```
boolean teste = false ;
```

```
teste = valor < 40 && valor > 20; // teste = true
```

```
teste = valor < 40 && valor > 30; // teste = false
```

```
teste = valor > 30 || valor > 20; // teste = true
```

```
teste = valor > 30 || valor < 20; // teste = false
```

```
teste = valor < 50 && valor == 30; // teste = true
```


Módulo 03 – Java Fundamentos Aula 01



Eclipse é uma IDE para desenvolvimento Java, porém suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e Kotlin. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software

[https://pt.wikipedia.org/wiki/Eclipse_\(software\)](https://pt.wikipedia.org/wiki/Eclipse_(software))

Módulo 03 – Java Fundamentos Aula 01



Saída de dados em Java

Para escrever na tela um texto qualquer

Sem quebra de linha ao final:

```
System.out.print("Bom dia!");
```

Com quebra de linha ao final:

```
System.out.println("Bom dia!");
```

Para escrever o conteúdo de uma variável de algum tipo básico

Suponha uma variável tipo int declarada e iniciada: `int y = 32;`

```
System.out.println(y);
```

Módulo 03 – Java Fundamentos Aula 01



Para escrever o conteúdo de uma variável com ponto flutuante

Suponha uma variável tipo double declarada e iniciada: `double x = 10.35784;`

```
System.out.println(x);
```

```
System.out.printf("%.2f%n", x);
```

`%n` = quebra de linha
(independente de plataforma)

```
System.out.printf("%.4f%n", x);
```

Para considerar o separador de decimais como ponto, ANTES da declaração do Scanner, faça:
`Locale.setDefault(Locale.US);`

Módulo 03 – Java Fundamentos Aula 01



Para concatenar vários elementos em um mesmo comando de escrita

Regra geral para print e println:

elemento1 + elemento2 + elemento3 + ... + elementoN

```
System.out.println("RESULTADO = " + x + " METROS");
```

%f = ponto flutuante
%n = quebra de linha

Regra geral para printf:

"TEXT01 %f TEXT02 %f TEXT03", variavel1, variavel2

```
System.out.printf("RESULTADO = %.2f metros%n", x);
```

Módulo 03 – Java Fundamentos Aula 01



Para concatenar vários elementos em um mesmo comando de escrita

Regra geral para printf:

"TEXT01 %f TEXT02 %f TEXT03", variavel1, variavel2

String nome = "Maria";

int idade = 31;

double renda = 4000.0;

System.out.printf("%s tem %d anos e ganha R\$ %.2f reais%n", nome, idade, renda);

%f = ponto flutuante
%d = inteiro
%s = texto
%n = quebra de linha

Módulo 03 – Java Fundamentos Aula 01



Processamento de dados

Comando de atribuição.

Sintaxe:

`<variável> = <expressão>;`

Lê-se "recebe"

REGRA:

- 1) A expressão é calculada
- 2) O resultado da expressão é armazenado na variável

Módulo 03 – Java Fundamentos Aula 01



Exemplo 1

```
int x, y;  
x = 8;  
y = 4 * x;
```

```
System.out.println(x);  
System.out.println(y);
```

Exemplo 2

```
int x;  
double y;  
x = 8;  
y = 4 * x;
```

```
System.out.println(x);  
System.out.println(y);
```

```
double b, B, h, area;
```

```
b = 6.0;  
B = 8.0;  
h = 5.0;
```

```
area = (b + B) / 2.0 * h;
```

```
System.out.println(area);
```

Boa prática:

Sempre indique o tipo do número, se a expressão for de ponto flutuante (não inteira).

Para **double** use:

.0

Para **float** use:

f

```
float b, B, h, area;
```

```
b = 6f;  
B = 8f;  
h = 5f;
```

```
area = (b + B) / 2f * h;
```

```
System.out.println(area);
```

Boa prática:

Sempre indique o tipo do número, se a expressão for de ponto flutuante (não inteira).

Para **double** use:

.0

Para **float** use:

f

Módulo 03 – Java Fundamentos Aula 01



Casting

É a conversão explícita de um tipo para outro. É necessário quando o compilador não é capaz de “adivinhar” que o resultado de uma expressão deve ser de outro tipo.

Exemplo 1

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

```
System.out.println(resultado);
```

Exemplo 2

```
int a, b;  
double resultado;  
a = 5;  
b = 2;  
resultado = (double) a / b;
```

```
System.out.println(resultado);
```

Exemplo 3

```
double a;  
int b;  
a = 5.0;  
b = a;
```

```
System.out.println(b)
```


Módulo 03 – Java Fundamentos Aula 01



Vamos praticar

- 1 - O custo de um carro novo ao consumidor é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos (aplicados ao custo de fábrica). Supondo que o percentual do distribuidor seja de 28% e os impostos de 45%, escrever um código para processar e mostrar o custo final ao consumidor.
- 2 – Desenvolva um código que seja atribuído a uma variável o nome de um piloto, uma distância percorrida em km e o tempo que o piloto levou para percorrê-la (em horas). O programa deve calcular a velocidade média em km/h, e exibir a seguinte frase: A velocidade média de XX foi YY km/h. onde XX é o nome do piloto, e YY é sua velocidade média.
- 3 - Considerando que para um consórcio, sabe-se o número total de prestações, a quantidade de prestações pagas e o valor atual da prestação, escreva um algoritmo que determine o total pago pelo consorciado e o saldo devedor.

RECODE



www.recode.org.br

RECODE
pro

recodepro.org.br

Institucional



[/rederecode](#)



[/recoderede](#)