

R

Java  
Fundamentos e  
Orientação a  
Objetos

**Java**

# Java - História

Java é uma linguagem de programação, orientada a objetos, desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems.

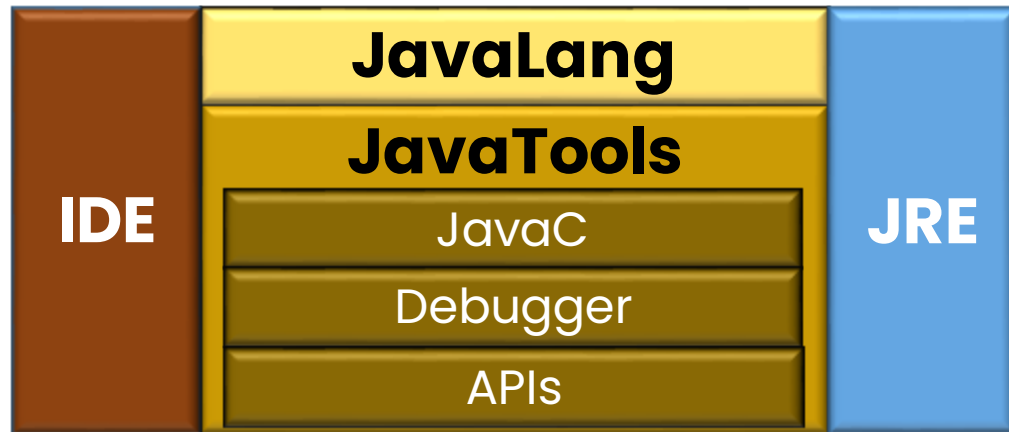
<https://youtu.be/sTX0UEpIF54>

Vídeo do Gustavo Guanabara (2015)

# Java - Funcionamento

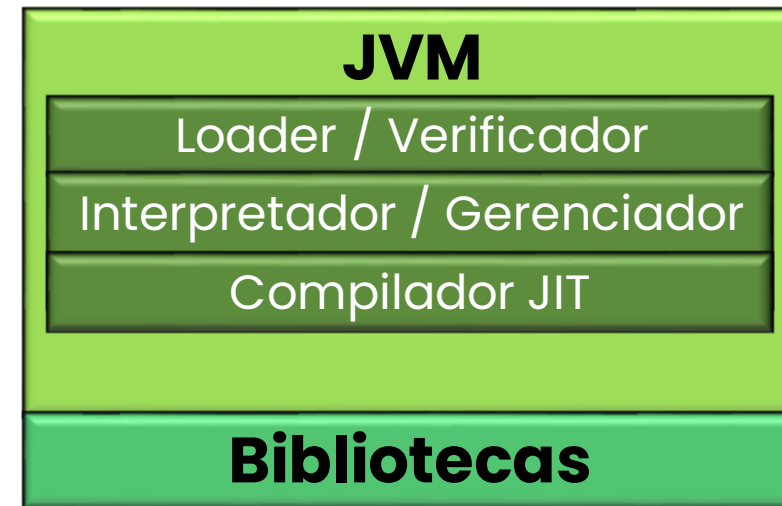
## JDK

### Java Development Kit



## JRE

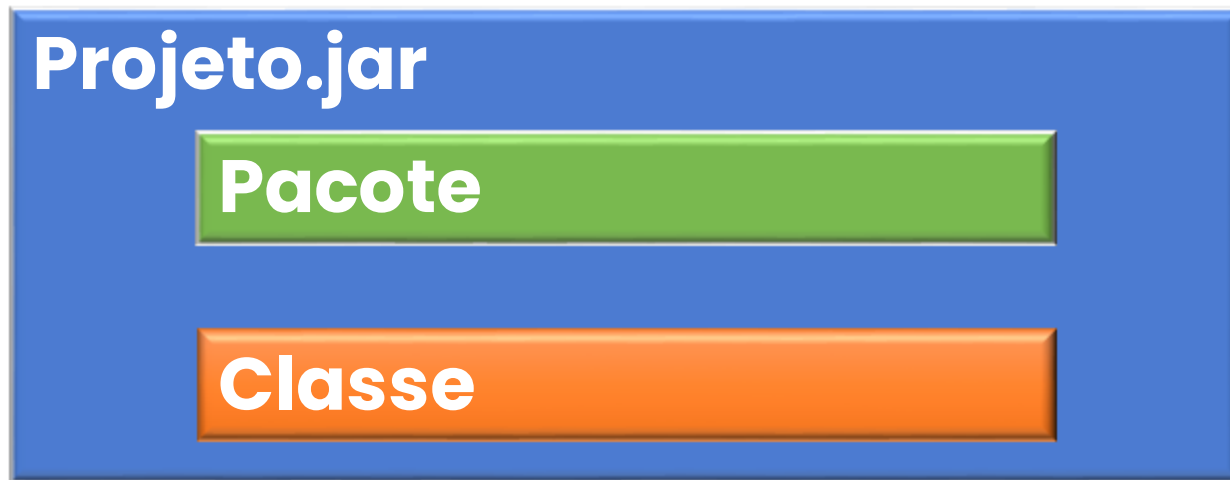
### Java Runtime Enviroment



# Java - IDE



# Java – Estrutura do Código Java



```
package primeiroprograma  
public class PrimeiroPrograma {  
    public static void main (String[] args) {  
    }  
}
```

← Pacote  
← Classe  
↑ Método

# Java – Estrutura do Código Java

package primeiroprograma ← Pacote

public class PrimeiroPrograma {  
← Classe

public static void main (String[] args) {  
← Método

código do método;

}



Pertence a Classe PrimeiroPrograma

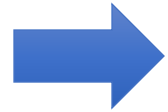
## Java – Camel Case

**m**eu**p**acote



primeiroprograma

**M**inha**C**lasse



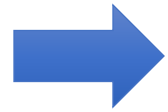
PrimeiroPrograma

**m**eu**M**etodo



lancarNotaAluno

**m**eu**A**tributo



nomeAluno

**m**inha**V**ariavel



mediaPrimeiroBimestre



Java  
Fundamentos e  
Orientação a  
Objetos

# Tipos de Dados

# Java – Tipos de Dados

Família	Tipo Primitivo	Classe Invólucro	Tamanho	Exemplo
Lógico	boolean	Boolean	1 bit	true
Literais	char	Character	1 byte	'A'
	-	String	1 byte/cada	"JAVA"
Inteiros	byte	Byte	1 byte	127
	short	Short	2 bytes	32 767
	int	Integer	4 bytes	2 147 483
	long	Long	8 bytes	$2^{63}$
Reais	float	Float	4 bytes	$3.4e^{+38}$
	double	Double	8 bytes	$1.8e^{+308}$

Java  
Fundamentos e  
Orientação a  
Objetos

# Saída de Dados

# Java – Saída de Dados

```
System.out.print ("Seja bem-vindo!");
```

```
System.out.print ("Sua idade é " + 19);
```

```
System.out.println ("Seja bem-vindo!");
```

```
System.out.println ("Iniciamos o curso);
```



**Pula linha ao exibir a informação**

# Java – Operadores para Formatação

Código	Tipo	Elemento Armazenado
%c	char	Um Único Caractere
%s	char	Uma Cadeia de Caracteres
%d ou %i	int	Um Inteiro
%f	flout e double	Um Número em Ponto Flutuante

Código	Ação
\n	Leva o cursor para a próxima linha
\t	Executa uma tabulação
\b	Executa um retrocesso
\f	Leva o cursor para a próxima página
\a	Emite um sinal sonoro (beep)
\"	Exibe o caractere "
\\	Exibe o caractere \
%%	Exibe o caractere %

## Java – Saída de Dados (Formatação)

```
Float nota = 8.5f;
```

```
System.out.printf ("Sua nota %.2f", nota);
```

```
System.out.format ("Sua nota %.2f", nota);
```

Java  
Fundamentos e  
Orientação a  
Objetos

# **Entrada de Dados Via Teclado**

# Java – Entrada de Dados Via Teclado

A classe Scanner, pois tem justamente a finalidade de facilitar a entrada de dados no modo Console.

O objeto *System.in* é o que faz a leitura do que se escreve no teclado. Veja abaixo como são invocados alguns dos métodos principais que correspondem com a assinatura que retorna um valor do tipo que foi invocado. Ou seja, para cada um dos primitivos existe uma chamada do método para retornar o valor especificado na entrada de dados, sempre seguindo o formato `nextTipoDado()`.

**Import java.util.Scanner**  **Classe**

**Scanner teclado = new Scanner(System.in)**  **Novo Objeto**

Int idade = **teclado.nextInt()**;

Float salario = **teclado.nextFloat()**;

String nome = **teclado.nextLine()**;

**teclado.close()**;



# Java – Entrada de Dados Via Teclado

Método	Descrição
close()	Fecha o escaneamento de leitura.
findInLine()	Encontra a próxima ocorrência de um padrão ignorando máscaras ou strings ignorando delimitadores.
hasNext()	Retorna um valor booleano verdadeiro (true) se o objeto Scanner tem mais dados de entrada.
hasNextXYZ()	Retorna um valor booleano como verdadeiro (true) se a próxima entrada a qual XYZ pode ser interceptada como Boolean, Byte, Short, Int, Long, Float ou Double.
match()	Retorna o resultado da pesquisa do último objeto Scanner atual.
next()	Procura e retorna a próxima informação do objeto Scanner que satisfazer uma condição.
nextBigDecimal(), nextBigInteger()	Varre a próxima entrada como BigDecimal ou BigInteger.
nextXYZ()	Varre a próxima entrada a qual XYZ pode ser interceptado como boolean, byte, short, int, long, float ou double.
nextLine()	Mostra a linha atual do objeto Scanner e avança para a próxima linha.
radix()	Retorna o índice atual do objeto Scanner.
remove()	Essa operação não é suportada pela implementação de um Iterator.
skip()	Salta para a próxima pesquisa de um padrão especificado ignorando delimitadores.
string()	Retorna uma string que é uma representação do objeto Scanner.

# Java – Entrada de Dados Via Teclado

## Métodos para Entrada de Dados

Método	Descrição
nextBoolean()	Usada para variável do tipo Boolean
nextByte()	Usada para variável do tipo Byte
nextDouble()	Usada para variável do tipo Double
nextFloat()	Usada para variável do tipo Float
nextInt()	Usada para variável do tipo Int
nextLine()	Usada para variável do tipo String
nextLong()	Usada para variável do tipo Long
nextShort()	Usada para variável do tipo Short

# Java – Entrada de Dados Via Teclado

## Exercício:

**1) Fazer um programa para que o usuário possa digitar um nome, uma idade e um salário, na sequência exibir na tela as informações digitadas.**

Java  
Fundamentos e  
Orientação a  
Objetos

# Conversão de Tipos de Dados

# Java – Conversão de Int to Char

```
public class IntToCharExemplo {  
    public static void main (String args []) {  
        int a = '6';  
        char c = (char) a;  
        System.out.println (c);  
    }  
}
```

O uso do (char) é chamado de **typecasting**.

Caso o número não esteja entre aspas simples, será exibido um código ACSII.

```
public class IntToCharExemplo {  
    public static void main (String args []) {  
        int radix = 10; //para obter o valor decimal, obrigatório o uso.  
        int value_int = 8;  
        char value_char = Character.forDigit(value_int, radix);  
        System.out.println(value_char);  
    }  
}
```

# Java – Conversão de Char to Int

```
public class CharToIntExemplo {  
    public static void main (String args []) {  
        char a = '1';  
        int b = a;  
        System.out.println (b);  
    }  
}
```

Será exibido um código ACSII.

```
public class CharToIntExemplo {  
    public static void main (String args []) {  
        char gen = '1';  
        int numgen = Character.getNumericValue(gen);  
        System.out.println("Número Gênero --> " + numgen);  
    }  
}
```

Valor exibido de forma correta.

# Java – Conversão de Int to String

## Integer.toString ()

O método Integer.toString () converte Int em String. O toString() é o método estático da classe Integer.

Conversão simples de Int em String usando o método Integer.toString ().

```
Int i = 10;  
String s = Integer.toString (i); // Retornará "10"
```

# Java – Conversão de String to Int

## Integer.parseInt ()

Podemos converter String em um int.

```
public class StringToIntExample1 {  
    public static void main (String args []) {  
        String s = "200";  
        int i = Integer.parseInt (s);  
        System.out.println (i);  
    }  
}
```



# Java – Conversão de Float to String

## Float.toString ()

O método Float.toString () converte Float em String. O toString() é o método estático da classe Float.

Conversão simples de Float em String usando o método Float.toString ().

```
float i = 15.8f;  
String s = Float.toString (i); // Retornará "15.8"
```

# Java – Conversão de String to Float

## Float.parseFloat ()

```
public class StringToFloatExemplo {  
    public static void main (String args []) {  
        String s = "28,1";  
        float d = Float.parseFloat (s);  
        System.out.println (d);  
    }  
}
```

# Java – Conversão de Boolean to String

## Boolean.toString ()

O método Boolean.toString () converte o booleano em String. O toString () é o método estático da classe Boolean.

```
public class BooleanToStringExemplo {  
    public static void main (String args []) {  
        boolean b1 = true;  
        boolean b2 = false;  
        String s1 = Boolean.toString (b1);  
        String s2 = Boolean.toString (b2);  
        System.out.println (s1);  
        System.out.println (s2);  
    }  
}
```

Java  
Fundamentos e  
Orientação a  
Objetos

# Operadores

# Java – Operadores Aritméticos

Aritmético	Operação	Prioridade
+	Adição	3
-	Subtração	3
%	Resto da divisão	2
*	Multiplicação	1
/	Divisão	1

# Java – Operadores de Atribuição

Operadores	Função	Operação	Operação
+=	Somar e Atribuir	a += b	a = a + b
-=	Subtrair e Atribuir	a -= b	a = a - b
%=	Resto da divisão e Atribuir	a %= b	a = a % b
*=	Multiplicar e Atribuir	a *= b	a = a * b
/=	Dividir e Atribuir	a /= b	a = a / b

# Java – Operadores de Atribuição

valor int = 1; **// valor = 1**

valor + = 2; **// valor = 3**

valor - = 1; **// valor = 2**

valor \* = 6; **// valor = 12**

valor / = 3; **// valor = 4**

valor% = 3; **// valor = 1**

# Java – Operadores Unários

Aritmético	Operação
++	Incremento
--	Decremento

valor int = 1; **// valor = 1**

valor ++ **// valor = 3**

valor -- **// valor = 1**



# Java – Operadores Relacionais

Uma das formas de se utilizar estes operadores é no uso da estrutura de decisão.

Símbolo	Função
<code>!=</code>	Diferente
<code>&gt;</code>	Maior
<code>&lt;</code>	Menor
<code>&lt;=</code>	Menor igual
<code>&gt;=</code>	Maior igual
<code>==</code>	Igual
<code>%</code>	Resto da divisão entre números inteiros
<code>x = (a &gt; b) ? a : b</code>	Ternário

# Java – Operadores Lógicos

Uma das formas de se utilizar estes operadores é no uso da estrutura de decisão.

Símbolo	Função
&&	And (e)
	Or (ou)
!	Not (não)
^	XOU (OU Exclusivo)

# Operadores Lógicos

Tabela verdade - AND		
Condição 1	Condição 2	Resultado
FALSO	FALSO	FALSO
VERDADEIRO	FALSO	FALSO
FALSO	VERDADEIRO	FALSO
VERDADEIRO	VERDADEIRO	VERDADEIRO

# Operadores Lógicos

Tabela verdade - OR		
Condição 1	Condição 2	Resultado
FALSO	FALSO	FALSO
VERDADEIRO	FALSO	VERDADEIRO
FALSO	VERDADEIRO	VERDADEIRO
VERDADEIRO	VERDADEIRO	VERDADEIRO

# Operadores Lógicos

Tabela Verdade NOT	
Condição	Resultado
Verdadeiro	Falso
Falso	Verdadeiro

Tabela Verdade XOU		
Condição 1	Condição 2	Resultado
Verdadeiro	Verdadeiro	Falso
Verdadeiro	Falso	Verdadeiro
Falso	Verdadeiro	Verdadeiro
Falso	Falso	Falso

# Java – Classe Math

Para se trabalhar com os operadores abaixo é preciso fazer uso da Classe Math.

Operadores	Função	Código	Valor
<b>PI</b>	Constante	Math.PI	3.1415....
<b>pow</b>	Exponenciação	Math.pow (5, 2)	25
<b>sqrt</b>	Raiz Quadrada	Math.sqrt (25)	5



f i t @rederecode | y @recoderede

<https://recode.org.br>