

Introdução à linguagem de programação Java

Programação Orientada à Objetos I

Prof. MSc. Leanderson André

Universidade da Região de Joinville

25 de fevereiro de 2025

Linguagem de Programação Java

Criada pela Sun na década de 90 e atualmente pertence à Oracle.

Características

- Multiparadigmas
- Multiplataforma
- Tipagem forte e estática
- Compilada

Aplicações

- Domínio geral
- Aplicativos móveis
- Sistemas backend/frontend
- Inteligência Artificial

Histórico de versões

Versão	Data
JDK1.0	23 de janeiro de 1996
JDK 1.1	19 de fevereiro de 1997
J2SE 1.2	8 de dezembro de 1998
J2SE 1.3	8 de maio de 2000
J2SE 1.4	6 de fevereiro de 2002
J2SE 5.0	30 de setembro de 2004
Java SE 6	11 de dezembro de 2006
Java SE 7	28 de julho de 2011
Java SE 8 (LTS)	18 de março de 2014
Java SE 9	21 de setembro de 2017
Java SE 10	20 de março de 2018

Histórico de versões

Versão	Data
Java SE 11 (LTS)	25 de setembro de 2018
Java SE 12	19 de março de 2019
Java SE 13	17 de setembro de 2019
Java SE 14	17 de março de 2020
Java SE 15	15 de setembro de 2020
Java SE 16	16 de março de 2021
Java SE 17 (LTS)	14 de setembro de 2021
Java SE 18	22 de março de 2022
Java SE 19	20 de setembro de 2022
Java SE 20	21 de março de 2023
Java SE 21 (LTS)	19 de setembro de 2023
Java SE 22	19 de março de 2024
Java SE 23	17 de setembro de 2024

Long-Term Support (LTS)

Java SE 21 terá suporte estendido até Setembro de 2031.

Plataforma Java

Java Runtime Environment (JRE)

Fornece as bibliotecas, a Java Virtual Machine e outros componentes para executar miniaplicativos e aplicativos escritos na linguagem de programação Java.

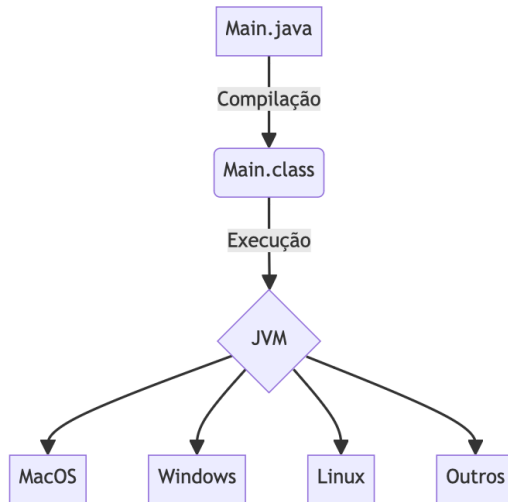
Java Development Kit (JDK)

Contém tudo o que está no JRE, além de ferramentas como compiladores e depuradores necessários para desenvolver miniaplicativos e aplicativos.

Java Virtual Machine (JVM)

Responsável pela independência de hardware e sistema operacional da plataforma Java SE, pelo pequeno tamanho do código compilado (bytecodes) e pela segurança da plataforma..

Plataforma Java



Hello World

Algoritmo 1: Main.java

```
1 /**
2  Comentario de varias linhas
3  */
4  public class Main{
5      // comentario de uma linha
6      public static void main(String args[]){
7          System.out.println("Hello World");
8      }
9  }
```

Observações

- O arquivo deve possuir a extensão **.java**.
- O nome do arquivo deve ser igual da classe pública.

Compilação

Via terminal com o aplicativo javac que está disponível na JDK.

Algoritmo 2: Compilar o arquivo Main.java

```
1 javac Main.java  
2 javac Main.java --release 8
```

Observações

Certificar se o terminal está no mesmo diretório do arquivo java.

Certificar a extensão do arquivo. Deve ser .java.

Caso necessário, utilizar a flag `-release 8` para compilar para a versão 8.

Execução

Via terminal com o aplicativo java que está disponível na JDK ou JRE.

Algoritmo 3: Executar o programa Main

```
1 java Main
```

Observações

Certificar se o terminal está no mesmo diretório do arquivo java compilado.
Certificar a extensão do arquivo compilado. Deve ser .class.

Comandos de terminal

`dir`

Listar os arquivos do diretório corrente.

Algoritmo 4: Executar o programa `dir` no terminal

```
1 dir
```

`cd`

Mudar de diretório.

Algoritmo 5: Alterar de diretório no terminal

```
1 cd c:\meu_diretorio
```

Compilação

O arquivo do código fonte está com extensão diferente de **.java**.

O nome do arquivo está diferente da classe pública.

Faltou o ponto e vírgula no final da instrução.

Faltou as chaves `{}` para delimitar os blocos de instruções.

O tipo de dados `String` está com s minúsculo.

Observações

As mensagens de erro são bem claras e mostram a linha e coluna exatas onde está o problema.

Algoritmo 6: Saída de dados

```
1 // definicao da classe Main foi omitida
2 public static void main(String args[]){
3     // Imprime e no final faz um pulo de linha
4     System.out.println("Hello World");
5     // Imprime sem realizar o pulo de linha
6     System.out.print("Hello World");
7     // Comando de escape para pular linha
8     System.out.print("\n");
9 }
```

Cor	Código de Escape Java
Preto	\u001b[30m
Vermelho	\u001b[31m
Verde	\u001b[32m
Amarelo	\u001b[33m
Azul	\u001b[34m
Magenta	\u001b[35m
Ciano	\u001b[36m
Branco	\u001b[37m
Resetar Cor	\u001b[0m

Tabela: Códigos de Escape para Texto Colorido em Java

Algoritmo 7: Saída de dados

```
1 // codigo escape da cor vermelha
2 System.out.print("\u001B[31m");
3 System.out.print("Texto em Vermelho");
4 // codigo escape para resetar a cor
5 System.out.println("\u001B[0m");
6 System.out.println("Cor original");
```

Tipos de dados

Tipo de dados	Tamanho	Descrição
byte	1 byte	-128 até 127
short	2 bytes	-32,768 até 32,767
int	4 bytes	-2,147,483,648 até 2,147,483,647
long	8 bytes	-9,223,372,036,854,775,808 até 9,223,372,036,854,775,807
float	4 bytes	Suficiente 6 ou 7 dígitos decimais.
double	8 bytes	Suficiente 15 dígitos decimais.
boolean	1 bit	verdadeiro ou falso
char	2 bytes	Um caractere ou valores ASCII

Declaração de variáveis

Algoritmo 8: Declaração de dados

```
1 // <tipo de dados> <nome da variavel> = <valor  
   inicial>;  
2 int inteiro = 10;  
3 // Obrigatorio incluir a letra L no final do  
   numero  
4 long inteiro_longo = 3_000_000_000L;  
5 // Obrigatorio incluir a letra F no final do  
   numero  
6 float decimal = 10.001F;  
7 double decimal_dobrado = 10.001;  
8 // Caracter entre aspas simples  
9 char character = 'A';  
10 boolean decisao = false;
```


Exibir os valores das variáveis

Algoritmo 9: Exibir os valores das variáveis no terminal

```
1 // <tipo de dados> <nome da variavel> = <valor  
   inicial>;  
2 int inteiro = 10;  
3 System.out.println("Inteiro = "+inteiro);  
4 double numero_pi = 3.141592;  
5 System.out.println("Numero PI = "+numero_pi);  
6 // Formatar o numero com 2 casas decimais  
7 System.out.println("Numero PI = "  
8   +String.format("%.2f", numero_pi));
```

Conversão de tipos

Um tipo de dados pequeno pode ser convertido (casting) para um tipo de dados maior. O caminho inverso, é preciso utilizar o operador de casting.

Algoritmo 10: Conversão entre long e int

```
1 long i = 10;
2 int j = 10;
3 // Erro: Conversao de tipos
4 // Um long(8bits) nao cabe em uma variavel
   inteira(4bits)
5 j = i;
6 // Correto
7 // Atencao: perda de valor
8 j = (int) i;
```

O mesmo ocorre entre double e float.

Nomes para as variáveis

Regras

Começar com uma letra, `_` ou `$`.

Preferência para uma letra minúscula e camelCase. Os nomes são case sensitive.

Pode utilizar números no meio do nome.

Não pode utilizar espaços.

Válido

`abelha`, `salarioDoFuncionario`, `jogador12`, `$variavel`, `_altura`, `Nome`

Inválido

`1abelha`, `&nome`, `jogador 12`,

Operações Aritméticas

Operação	Expressão Java
adição	$x = a + b;$
subtração	$x = a - b;$
divisão	$x = a / b;$
multiplicação	$x = a * b;$
resto	$x = a \% b;$

Operadores de atribuição compostos

Operação	Expressão Java	Expressão correspondente
Incremento	<code>x++;</code>	<code>x = x + 1;</code>
Decremento	<code>x--;</code>	<code>x = x - 1;</code>
divisão	<code>x/= y;</code>	<code>x = x / y;</code>
multiplicação	<code>x*= y;</code>	<code>x = x * y;</code>
resto(módulo)	<code>x%= y;</code>	<code>x = x % y;</code>

Operadores bit a bit

Operação	Expressão composta	correspondente
E bit a bit (AND)	$x \& = y;$	$x = x \& y;$
OU bit a bit (bitwise OR)	$x \mid = y$	$x = x \mid y;$
OU exclusivo bit a bit (XOR)	$x \wedge = y;$	$x = x \wedge y;$
Complemento de bits (NOT)		$x = \sim x;$
Deslocamento à esquerda	$x \ll = y;$	$x = x \ll y;$
Deslocamento à direita (com sinal)	$x \gg = y;$	$x = x \gg y;$
Deslocamento à esquerda (sem sinal)	$x \ggg = y;$	$x = x \ggg y;$

Operação	Expressão Java
maior que	$a > b$
maior e igual que	$a \geq b$
menor que	$a < b$;
menor e igual que	$a \leq b$

Operação	Expressão Java
igualdade	<code>a == b</code>
diferença	<code>a != b</code>
negação	<code>!a</code>

Algoritmo 11: Exemplo de utilização das operações

```
1 double x = 9.34;  
2 // Utilize os parenteses para definir  
3 // a ordem de operacoes  
4 double valor = 10 + ( x - 10) * 2;  
5 System.out.println("valor = "+valor);  
6 boolean eh_maior = 10 > 7;  
7 System.out.println("Maior? = "+eh_maior);
```

Algoritmo 12: Entrada de dados do tipo String

```
1 // Importar a classe Scanner
2 import java.util.Scanner;
3 public class Main{
4     public static void main(String args[]){
5         // Crie o leitor de teclado
6         Scanner leitor = new Scanner(System.in);
7         System.out.println("Digite o seu nome:");
8         // Todos os caracteres ate o ENTER sera
9         // capturado e salvo na variavel nome
10        String nome = leitor.nextLine();
11        System.out.println("Seu nome: "+nome);
12    }
13 }
```

Algoritmo 13: Entrada de dados de inteiro e decimal

```
1 // Declaracao da classe Main e metodo main  
  omitidos  
2 Scanner leitor = new Scanner(System.in);  
3 // Capturar um inteiro  
4 int x = leitor.nextInt();  
5 // Capturar um longo  
6 long y = leitor.nextLong();  
7 // Capturar um decimal (float)  
8 float z = leitor.nextFloat();  
9 // Capturar um decimal  
10 double w = leitor.nextDouble();
```

Operação	Expressão
Número PI	Math.PI
Logaritmo natural(base e)	Math.log(x)
Potência	Math.pow(x, y)
Raiz quadrada	Math.sqrt(x)
Raiz cúbica	Math.cbrt(x)
Menor entre x e y	Math.min(x,y)
Menor entre x e y	Math.min(x,y)
Valor absoluto	Math.abs(x)
Número aleatório entre 0.0 e 1.0	Math.random()
Arredondamento	Math.round(x)
Entre outros (cos, pi, tan, sin, etc)	

Área de um círculo

$$A = \pi r^2$$

Algoritmo 14: Implementação da área de um círculo

```
1 double raio = 10;  
2 double area = Math.PI * Math.pow(raio,2);
```

Questão 01

Escreva um programa em java que declare uma variável do tipo inteiro. Atribua o maior valor possível para essa variável. Some mais 1 e apresente o resultado em tela. Observe e analise o resultado.

Questão 02

Escreva um programa em java que receba um inteiro. Verifique se o valor é par ou ímpar. Apresente o resultado em tela.

Questão 03

Escreva um programa em java que receba dois inteiros. Realize a divisão entre os números e apresente o resultado em tela. Observe o que acontece quando se divide um número por zero.

Questão 04

Escreva um programa em java que receba dois textos. Verifique se os dois textos são iguais e apresente o resultado em tela. Observe e analise os resultados.

Questão 05

Escreva um programa em java que declare uma variável do tipo character. Atribui o valor inteiro 88 para a variável apresente o resultado em tela. Observe e analise o resultado.

Questão 06

Implemente e analise o trecho de código abaixo e identifique a diferença entre os operadores.

```
1 int x = 10;  
2 int y = 10;  
3 System.out.println(x++);  
4 System.out.println(++y);  
5 System.out.println("x=" + x);  
6 System.out.println("y=" + y);
```


Questão 07

Implemente e analise o trecho de código abaixo.

```
1 int x = 10;
2 int y = 2;
3 System.out.println("x em binario = "+Integer.
    toBinaryString(x));
4 x = x << y;
5 System.out.println("x = " + x);
6 System.out.println("x em binario = "+Integer.
    toBinaryString(x));
```

Experimente mudar o valor de y para 1 e 3.

Troque o operador << pelo operador >>.

Questão 08

Implemente a função Beale de acordo com a fórmula abaixo: A função possui duas variáveis x e y . As variáveis podem possuir os valores entre $[-4.5, 4.5]$. O resultado da função é zero quando $x = 3$ e $y = 0.5$.

$$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$

As variáveis podem possuir os valores entre $[-4.5, 4.5]$.
O resultado da função é zero quando $x = 3$ e $y = 0.5$.



Prof. MSc. Leanderson André
leandersonandre@univille.br