



INTRODUÇÃO AO JAVA

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HABNER FABRÍCIO BOESING
habner.boesing@unoesc.edu.br



OBJETIVOS

- História do Java
- Ferramentas para desenvolvimento
- Tipos de dados
- Declaração de variáveis
- Convenção de nomenclatura
- Expressões com operadores aritméticos, relacionais e lógicos
- Estruturas Condicionais

JAVA - HISTÓRIA

- Concebido no ano de 1995 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems.
- O objetivo era criar uma linguagem orientada à objetos com compatibilidade com os mais diversos tipos de equipamentos do dia-a-dia, até mesmo com eletrodomésticos.
- Em 2008 o Java foi adquirido pela empresa Oracle Corporation.
- Diferente das linguagens de programação modernas, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é interpretado por uma Máquina Virtual Java (JVM).



Java™

ORACLE



Apoiadores:



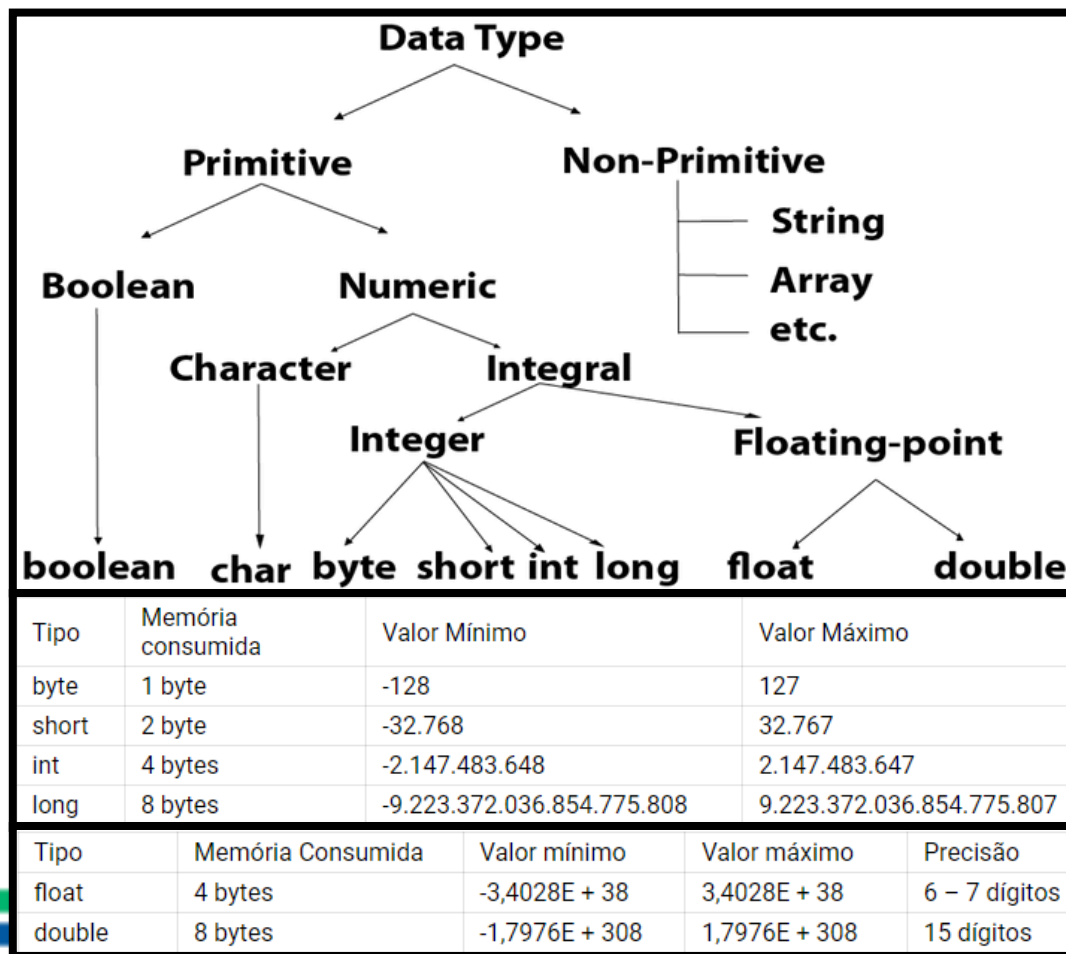
JAVA - CARACTERÍSTICAS

- Linguagem de programação de alto nível
- Orientada a objetos
- Multiplataforma
- Tipagem estática e forte (na declaração da variável e necessário especificar seu tipo e a mesma não pode mudar seu tipo durante sua execução)

```
public class Teste {  
  
    public static void main(String[] args) {  
  
        int idade;  
        idade = 15;  
        System.out.println(idade);  
  
    }  
}
```

TIPOS DE DADOS PRIMITIVOS E NÃO-PRIMITIVOS

- Dentre os tipos de dados disponíveis no Java, temos:



JAVA – AMBIENTE DE DESENVOLVIMENTO

- Para que possam ser executados os algoritmos utilizando a linguagem de programação Java, é necessário instalar o Java Development Kit (JDK) para possibilitar a execução da JVM.
- Link download: <https://www.oracle.com/java/technologies/downloads/>
- Além disso, precisamos de um ambiente de desenvolvimento (IDE) no computador.
- Existem diversas opções disponíveis, como por exemplo:
 - NetBeans: <https://netbeans.apache.org/download/index.html>
 - Eclipse: <https://www.eclipse.org/downloads/>
 - Eclipse (STS): <https://spring.io/tools>
 - IntelliJ IDEA: <https://www.jetbrains.com/pt-br/idea/download/>



CONVENÇÃO DE NOMENCLATURA - JAVA

- Uma convenção de nomenclatura é uma regra a seguir quando você decide como nomear seus identificadores (por exemplo, classe, pacote, variável, método, etc.).
- Diferentes programadores Java podem ter diferentes estilos e abordagens para a maneira como programam. Usando as convenções de nomenclatura padrão do Java, eles tornam seu código mais fácil de ler para eles próprios e para outros programadores. A legibilidade do código Java é importante porque significa que menos tempo é gasto tentando descobrir o que o código faz, deixando mais tempo para corrigi-lo ou modificá-lo.



REGRAS DE NOMENCLATURA

■ Pacote

- Deve ser uma letra minúscula, como java, lang.
- Se o nome contiver várias palavras, ele deverá ser separado por pontos (.) Como java.util, java.lang.

```
package com.javatpoint; //pacote
class Employee {
    //fragmento de código
}
```


REGRAS DE NOMENCLATURA - JAVA

■ Classe

- Deve começar com a letra maiúscula.
- Deve ser um substantivo como Cor, Interruptor, Sistema, Cliente, Aluno, etc.
- Use palavras apropriadas, em vez de siglas.

```
public class Employee {  
    //fragmento de código  
}
```

■ Interface

- Deve começar com a letra maiúscula.
- Deve ter um nome que descreve uma operação que uma classe pode fazer como: Runnable, Comparable, Enumerable.
- Use palavras apropriadas, em vez de siglas.

```
interface Printable {  
    //fragmento de código  
}
```

REGRAS DE NOMENCLATURA

■ Método

- Deve começar com letra minúscula.
- Deve ser um verbo como `acender()`, `desligar()`, `trocar()`.
- Se o nome contiver várias palavras, inicie-o com uma letra minúscula seguida por uma letra maiúscula, como `testarIntegridade()`.

```
class Employee {  
    //método  
    void desenhar() {  
        //fragmento de código  
    }  
}
```

REGRAS DE NOMENCLATURA

■ Variável

- Ele deve começar com uma letra minúscula, como id, nome.
- Não deve começar com caracteres especiais como & (e comercial), \$ (dólar), _ (sublinhado).
- Se o nome contiver várias palavras, inicie-o com a letra minúscula seguida por uma letra maiúscula como primeiroNome, anoFabricacao.
- Evite usar variáveis de um caractere, como x, y, z.

```
class Employee {  
    //variável  
    int id;  
    //fragmento de código  
}
```

REGRAS DE NOMENCLATURA

■ Constante

- Deve estar em letras maiúsculas, como VERMELHO, AMARELO.
- Se o nome contiver várias palavras, ele deverá ser separado por um sublinhado (_), como MAX_PRIORITY.
- Pode conter números, mas não como a primeira letra.

```
class Employee {  
    //constante&  
    final static int MIN_AGE = 18 ;  
    //fragmento de código  
}
```

PRIMEIROS ALGORITMOS – COMANDOS BÁSICOS

- Exibir valor na tela:

```
System.out.println("Olá Mundo");
```

- Atribuição de valores à variáveis:

```
int num1, num2, soma;  
num1 = 10;  
num2 = 5;  
soma = num1 + num2;
```

- Exibição de valores concatenados (MÉTODO I):

```
System.out.println("O valor da soma entre: "+num1+" e "+num2+" é "+soma);
```

- Armazenar valores digitados pelo usuário:

```
Scanner scan = new Scanner(System.in);  
String nome;  
System.out.println("Digite seu nome: ");  
nome = scan.next();  
System.out.println(nome);
```

Apoiadores:



TIPOS DE OPERADORES

- Os operadores são importantes para realizar as operações entre as diferentes variáveis, criando cálculos ou expressões que resultem em verdadeiro ou falso.
- Operadores aritméticos: +, -, *, /, %, ++, --
- Operadores de comparação: >, <, ==, >=, <=, !=
- Operadores lógicos: &&, ||
- Operadores de atribuição: =, +=, -=

OPERADORES BÁSICOS (+,-,*,/)

■ Exemplo:

```
float num1, num2, soma, sub, mult, div;  
Scanner scan = new Scanner(System.in);  
//solicitar números ao usuário  
System.out.print("Digite o primeiro número: ");  
num1 = scan.nextFloat();  
System.out.print("Digite o segundo número: ");  
num2 = scan.nextFloat();  
//operações  
soma = num1 + num2;  
sub = num1 - num2;  
mult = num1 * num2;  
div = num1 / num2;  
//exibição das mensagens  
System.out.println("A valor da soma é " + soma);  
System.out.println("A valor da subtração é " + sub);  
System.out.println("A valor da multiplicação é " + mult);  
System.out.println("A valor da divisão é " + div);
```

ESTRUTURA CONDICIONAL COMPOSTA (IF-ELSE)

Exemplo:

```
float numero;  
Scanner ler = new Scanner(System.in);  
System.out.print("Digite um número diferente de 0: ");  
numero = ler.nextFloat();  
//estrutura condicional composta  
if (numero > 0) {  
    System.out.println("Este é um número positivo");  
} else {  
    System.out.println("Este é um número negativo");  
}
```

Apoiadores:



ESTRUTURA CONDICIONAL ENCADEADA HOMOGÊNEA (IF-ELSE IF-ELSE)

Exemplo:

```
float numero;  
Scanner ler = new Scanner(System.in);  
System.out.println("Digite um número diferente de 0: ");  
numero = ler.nextFloat();  
//estrutura condicional encadeada homogênea  
if (numero > 0) {  
    System.out.println("Este é um número positivo");  
} else if (numero < 0) {  
    System.out.println("Este é um número negativo");  
} else {  
    System.out.println("Este é um número neutro");  
}
```

OPERADOR LÓGICO (E)

Exemplo:

```
String aluno;
float media, frequencia;
Scanner ler = new Scanner(System.in);
System.out.print("Qual é o nome do aluno? ");
aluno = ler.next();
System.out.print("Qual é a média final do aluno? ");
media = ler.nextFloat();
System.out.print("Qual é a frequência do aluno? ");
frequencia = ler.nextFloat();
if (media >= 7 && frequencia >= 75) {
    System.out.println("O aluno " + aluno + " está aprovado.");
} else {
    System.out.println("O aluno " + aluno + " está reprovado.");
}
```

OPERADOR LÓGICO (OU)

■ Exemplo:

```
String aluno, curso;
Scanner ler = new Scanner(System.in);
System.out.print("Qual é o nome do aluno? ");
aluno = ler.next();
System.out.print("Qual é o curso que o aluno está cursando? ");
curso = ler.next();
if ((curso.equals("Eng. de Produção")) || (curso.equals("Eng. Elétrica"))) {
    System.out.println("O aluno " + aluno + " está aprovado.");
} else {
    System.out.println("O aluno " + aluno + " está reprovado.");
}
```

LISTA DE EXERCÍCIOS JAVA

- 1) Escreva um algoritmo que pergunte quanto o funcionário ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês.
- 2) 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 algoritmo para ler o custo de fábrica de um carro, calcular e escrever o custo final ao consumidor.
- 3) Crie um algoritmo que imprima o gênero de uma pessoa utilizando como valores de entrada M ou F.
- 4) Leia o nome e 3 notas de um aluno. Se obtiver média maior que 7 exibir a mensagem de aprovado, se ficar entre 5 e 7, exibir a mensagem em exame, e se for abaixo disso, exibir reprovado. Lembrando que terá que exibir o nome do aluno também.