

# PROGRAMAÇÃO ORIENTADA A OBJETOS

Professora Lyrane Bezerra

# Estrutura de um programa Java

Todo código Java reside dentro de uma **classe**.

Um programa simples começa com a declaração da classe principal e um método especial, o main, que é o ponto de entrada do seu programa.

```
// Declaração da classe
public class NomeDaClasse {

    // Método principal (ponto de entrada do programa)
    public static void main(String[] args) {
        // Seu código vai aqui
    }
}
```

# Estrutura de um programa Java

- `public class NomeDaClasse`: Define uma classe chamada `NomeDaClasse`. O nome do arquivo Java deve ser exatamente o mesmo do nome da classe (`NomeDaClasse.java`).
- `public static void main(String[] args)`: Este é o método principal. `public` significa que ele pode ser acessado de qualquer lugar, `static` significa que você não precisa criar um objeto da classe para executá-lo, `void` indica que ele não retorna nenhum valor, e `String[] args` é um array de strings que pode receber argumentos da linha de comando.

## Comando de Saída (Imprimindo na tela)

Para exibir informações no console, usamos a classe `System.out`.

Existem dois comandos principais:

- `System.out.println()`: Imprime a mensagem e pula para a próxima linha.
- `System.out.print()`: Imprime a mensagem, mas não pula para a próxima linha.

```
public class SaidaDeDados {  
    public static void main(String[] args) {  
        // Imprime "Olá, Mundo!" e pula a linha  
        System.out.println("Olá, Mundo!");  
  
        // Imprime duas mensagens na mesma linha  
        System.out.print("Esta é a primeira parte. ");  
        System.out.print("Esta é a segunda parte.");  
    }  
}
```



# Comandos de Entrada (Lendo dados do usuário)

Para ler dados digitados pelo usuário, a maneira mais comum e simples é usar a classe **Scanner**.

Para usá-la, você precisa:

- 1.Importar a classe Scanner do pacote java.util.
- 2.Criar uma instância de Scanner, passando System.in (o fluxo de entrada padrão) como argumento.
- 3.Usar os métodos da classe Scanner para ler diferentes tipos de dados.

# Exemplo

// 1. Importa a classe Scanner

```
import java.util.Scanner;
```

```
public class EntradaDeDados {
```

```
    public static void main(String[] args) {
```

// 2. Cria um objeto Scanner para ler a entrada  
do teclado

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

// Saída para o usuário

```
    System.out.print("Digite seu nome: ");
```

// 3. Lê uma linha de texto (uma String)

```
    String nome = scanner.nextLine();
```

// Saída para o usuário

```
    System.out.print("Digite sua idade: ");
```

// 3. Lê um número inteiro (um int)

```
    int idade = scanner.nextInt();
```

// Imprime os dados lidos

```
    System.out.println("Olá, " + nome + "! Você  
tem " + idade + " anos.");
```

// É uma boa prática fechar o objeto  
Scanner quando não for mais necessário

```
    scanner.close();
```

```
    } }
```

## Observações:

No exemplo anterior, o programa solicita o nome e a idade do usuário, armazena essas informações em variáveis (nome e idade) e, em seguida, as exibe na tela.

O uso de `scanner.nextLine()` para strings e `scanner.nextInt()` para inteiros demonstra como ler diferentes tipos de dados.

# Tipos de dados em Java

- Em Java, os tipos de dados são divididos em duas categorias principais:
  - **tipos primitivos**
  - **tipos de referência.**
- Essa distinção é crucial para entender como as variáveis são armazenadas e manipuladas na memória.



# Tipos de dados

## Tipos Primitivos

São os tipos de dados mais básicos e fundamentais. Eles armazenam o valor diretamente na memória. Java tem 8 tipos primitivos, que são:

### Tipos para Números Inteiros:

Usados para armazenar números inteiros (sem casas decimais). Eles variam em tamanho e faixa de valores, o que permite otimizar o uso de memória.

- byte: O menor tipo de dado inteiro, com 8 bits. Faixa de -128 a 127.
- short: Tem 16 bits. Faixa de -32.768 a 32.767.
- int: O tipo inteiro mais comum, com 32 bits. Faixa de aproximadamente  $-2^{31}$  a  $2^{31}$ . É a escolha padrão para a maioria dos casos.
- long: O maior tipo inteiro, com 64 bits. Faixa de aproximadamente  $-2^{63}$  a  $2^{63}$ . É usado para valores muito grandes.

### Tipos para Números Decimais (Ponto Flutuante)

Usados para armazenar números com casas decimais.

- float: Tem 32 bits e é usado para números de ponto flutuante de precisão simples. Geralmente, é usado quando a precisão não é crítica e a economia de memória é importante.
- double: O tipo padrão para números decimais, com 64 bits e precisão dupla. É a escolha recomendada para a maioria das situações.

# Tipos de dados

## Tipos Primitivos

São os tipos de dados mais básicos e fundamentais. Eles armazenam o valor diretamente na memória. Java tem 8 tipos primitivos, que são:

### Tipos para Caracteres e Lógicos

Usados para representar valores lógicos e caracteres individuais.

- char: Armazena um único caractere Unicode de 16 bits. O valor deve ser colocado entre aspas simples, como 'A'.
- boolean: Armazena um valor lógico, que pode ser apenas true (verdadeiro) ou false (falso).

# Tipos de Referência

Diferentemente dos tipos primitivos, que armazenam o valor, os tipos de referência armazenam um endereço de memória (uma referência) para o objeto que contém os dados.

Eles são criados a partir de classes. Os principais exemplos são:

- **Classes:** Como a classe String para sequências de caracteres, Scanner para entrada de dados, e todas as classes que você cria.
- **Interfaces:** Definem um contrato de métodos que uma classe deve seguir.
- **Arrays:** Coleções ordenadas de elementos do mesmo tipo.

A principal diferença é que os tipos primitivos são passados por valor (uma cópia do valor é usada), enquanto os tipos de referência são passados por referência (uma cópia do endereço de memória é usada).

```
public class TiposDeDadosExemplo {  
    public static void main(String[] args) {  
        // Tipos Primitivos  
  
        byte idade = 30;  
  
        int populacao = 150000;  
  
        long distancia = 9876543210L; // O  
        'L' indica que é um long  
  
        double salario = 2500.50;  
  
        float peso = 68.5f; // O 'f' indica que é  
        um float  
  
        char primeiraLetra = 'A';  
  
        boolean estaAtivo = true;
```

## Exemplo:

```
        // Tipo de Referência (String é uma classe)  
        String nome = "Maria";  
  
        System.out.println("Idade: " + idade);  
        System.out.println("População: " + populacao);  
        System.out.println("Salário: " + salario);  
        System.out.println("Nome: " + nome);  
        System.out.println("Está ativo? " + estaAtivo);  
    }  
}
```