

# Introdução a Tipos Primitivos, Declaração de Variáveis, Casting e estrutura de decisão e repetição

- Compreender os tipos primitivos em Java.
- Aprender a declarar variáveis.
- Entender o conceito de casting.
- Praticar a declaração de variáveis, estrutura de decisão, repetição e o uso de casting.



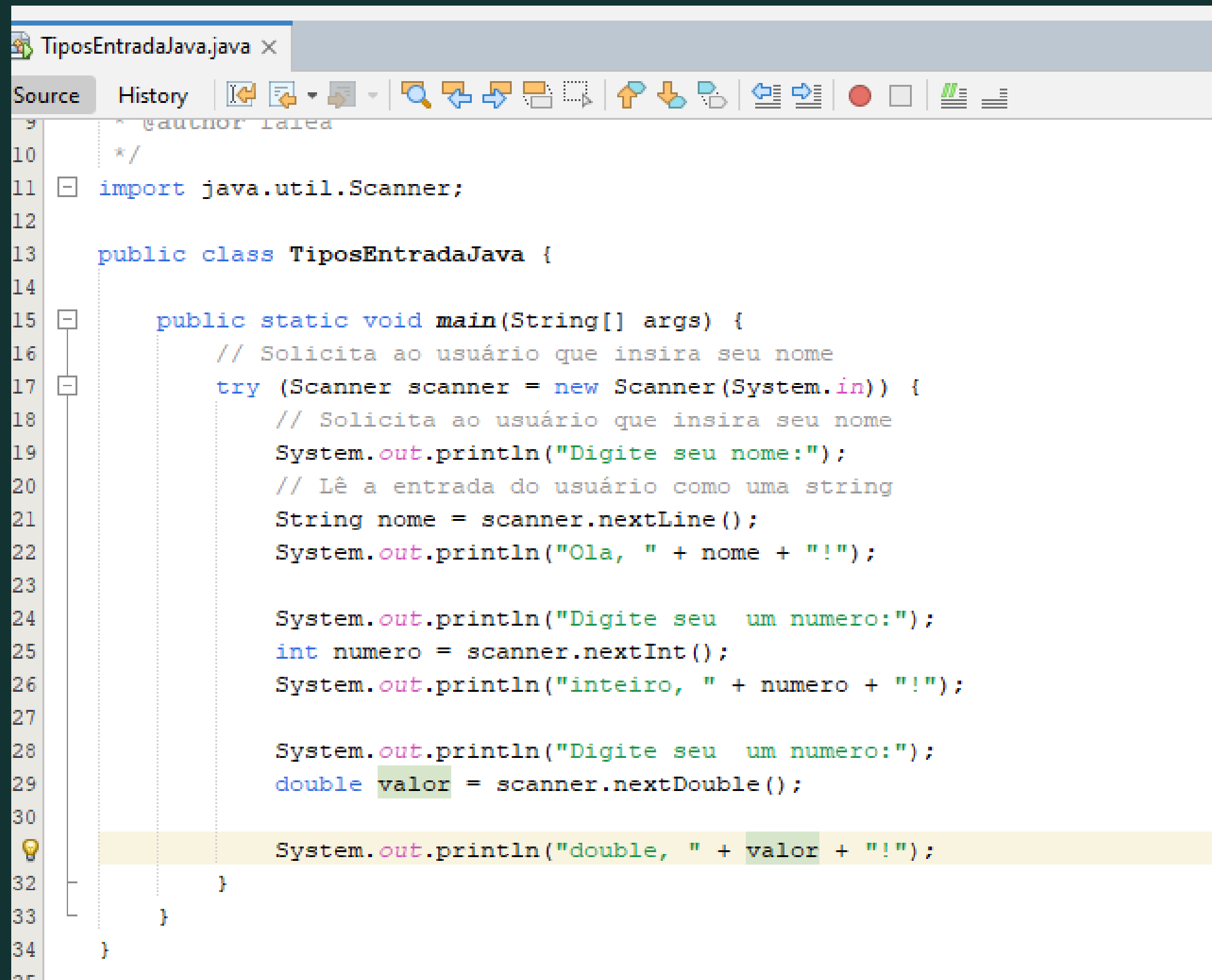
# Scanner o que é ?

O Scanner é uma classe da biblioteca java.util usada para ler entrada de dados do usuário em programas Java. Ele pode capturar diferentes tipos de dados como int, double, String e muito mais.

Tipos de entrada:

Método	Tipo de Entrada	Exemplo de Uso
nextInt()	int	scanner.nextInt();
nextDouble()	double	scanner.nextDouble();
nextFloat()	float	scanner.nextFloat();
nextLong()	long	scanner.nextLong();
nextBoolean()	boolean	scanner.nextBoolean();
next()	String (sem espaços)	scanner.next();
nextLine()	String (linha inteira)	scanner.nextLine();

# Exemplo de Uso



```
9  * @author Ialea
10  */
11  import java.util.Scanner;
12
13  public class TiposEntradaJava {
14
15      public static void main(String[] args) {
16          // Solicita ao usuário que insira seu nome
17          try (Scanner scanner = new Scanner(System.in)) {
18              // Solicita ao usuário que insira seu nome
19              System.out.println("Digite seu nome:");
20              // Lê a entrada do usuário como uma string
21              String nome = scanner.nextLine();
22              System.out.println("Ola, " + nome + "!");
23
24              System.out.println("Digite seu um numero:");
25              int numero = scanner.nextInt();
26              System.out.println("inteiro, " + numero + "!");
27
28              System.out.println("Digite seu um numero:");
29              double valor = scanner.nextDouble();
30
31              System.out.println("double, " + valor + "!");
32          }
33      }
34  }
```

# Tipos Primitivos

Java possui oito tipos de dados primitivos que são classificados em quatro categorias:

## **Numéricos Inteiros:**

- byte: 8 bits, varia de -128 a 127
- short: 16 bits, varia de -32,768 a 32,767
- int: 32 bits, varia de  $-2^{31}$  a  $2^{31} - 1$  ou seja -2.147.483.648 a 2.147.483.647
- long: 64 bits, varia de  $2^{31}$  a  $2^{63}-1$  ou seja -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807.

## **Numéricos de Ponto Flutuante:**

- float: 32 bits, precisão simples
- double: 64 bits, precisão dupla

## **Caractere:**

- char: 16 bits, representa um único caractere Unicode

## **Booleano:**

- boolean: valores true ou false

# Exemplo Declaração de Variáveis

```
*/  
public class Exemplo_Variaveis {  
  
    public static void main(String[] args) {  
        int idade = 25;  
        float altura = 1.75f;  
        boolean isActive = true;  
        char inicial = 'A';  
  
        System.out.println("Idade: " + idade);  
        System.out.println("Altura: " + altura);  
        System.out.println("Ativo: " + isActive);  
        System.out.println("Inicial: " + inicial);  
    }  
}
```

# Casting (Tipo de conversão de dados)

Casting é a conversão de um tipo de dado para outro. Pode ser implícito ou explícito.

Casting Implícito: Ocorre automaticamente quando não há perda de dado.

Casting Explícito: Necessário quando há possibilidade de perda de dados.

```
* @author falea
*/
public class ExemploCasting {

    public static void main(String[] args) {
        int numero = 100;
        long numeroLong = numero; // casting implícito
        System.out.println("Numero long: " + numeroLong);

        double preco = 9.99;
        int precoInt = (int) preco; // casting explícito
        System.out.println("Preco original: " + preco);
        System.out.println("Preco inteiro: " + precoInt);
    }
}
```

# Estrutura de Decisão

```
public class EstruturaDecisao {  
  
    public static void main(String[] args) {  
        int idade = 16;  
  
        if (idade >= 18) {  
            System.out.println("Voce e maior de idade.");  
        } else {  
            System.out.println("Voce e menor de idade.");  
        }  
  
        int nota = 75;  
        if (nota >= 90) {  
            System.out.println("Nota A");  
        } else if (nota >= 80) {  
            System.out.println("Nota B");  
        } else if (nota >= 70) {  
            System.out.println("Nota C");  
        } else {  
            System.out.println("Nota D");  
        }  
  
        int diaDaSemana = 3;  
        switch (diaDaSemana) {  
            case 1 -> System.out.println("Domingo");  
            case 2 -> System.out.println("Segunda-feira");  
            case 3 -> System.out.println("Terca-feira");  
            default -> System.out.println("Dia inválido");  
        }  
    }  
}
```

# Estrutura de Decisão

## Declaração If:

```
if (condição) {  
    // Código a ser executado se a condição for verdadeira  
}
```

```
if (condição) {  
    // Código a ser executado se a condição for verdadeira  
} else {  
    // Código a ser executado se a condição for falsa  
}
```



# Estrutura de Decisão

## Declaração If:

```
if (condição1) {  
    // Código a ser executado se a condição1 for verdadeira  
} else if (condição2) {  
    // Código a ser executado se a condição2 for verdadeira  
} else {  
    // Código a ser executado se todas as condições forem  
    falsas  
}
```

# Estrutura de Decisão: switch

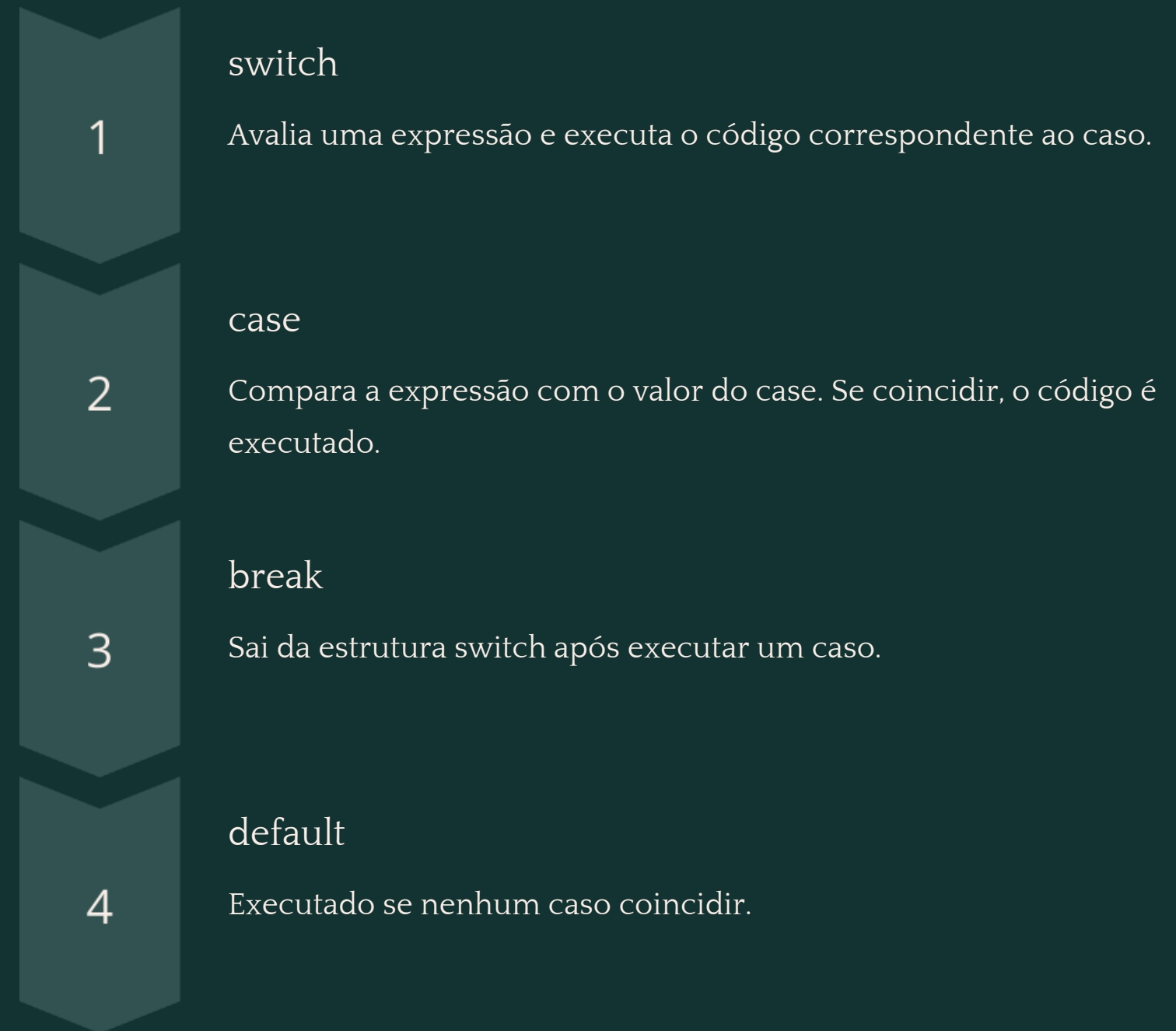
```
switch (expressão) {
```

```
    case valor1:  
        break;
```

```
    case valor2:  
        break;
```

```
    default:
```

```
}
```



# Estrutura de Repetição

```
*/
public class EstruturaRepeticao {

    public static void main(String[] args) {

        for (int i = 0; i < 5; i++) {
            System.out.println("Contagem: " + i);
        }

        int contador = 0;
        while (contador < 5) {
            System.out.println("Contagem: " + contador);
            contador++;
        }

        int conta = 0;
        do {
            System.out.println("Contagem: " + conta);
            conta++;
        } while (conta < 5);

    }
}
```

# Estrutura de Repetição: while Loop

do while (**Enquanto**)

O loop while executa um bloco de código enquanto a condição for verdadeira.

## Declaração while Loop:

```
while (condição) {  
    // Código a ser executado  
    enquanto a condição for  
    verdadeira  
}
```

# Estrutura de Repetição: do while Loop

do while (Faça-enquanto)

Executa o bloco de código pelo menos uma vez.

Verificação da condição ocorre após a execução do código.

## Declaração do- while Loop:

```
do {  
    // Código a ser executado  
} while (condição);
```

# Estrutura de Decisão

## Declaração for Loop:

```
for (inicialização; condição; incremento) {  
    // Código a ser executado  
}
```

# Estrutura de Repetição: For Loop

## For(Para)

O loop for é utilizado quando se sabe previamente quantas vezes o código deve ser executado. Ele possui uma estrutura mais compacta.

```
for (inicialização; condição;  
    incremento) {  
    // Código a ser executado  
}
```

# Estrutura de Decisão

## Declaração for Loop:

```
for (inicialização; condição; incremento) {  
    // Código a ser executado  
}
```



# Referências Bibliográficas

- DUKE. Uma breve história do projeto Green. Disponível em: <https://duke.dev.java.net/green/>.
- JAVA WORLD. So why did they decide to call it Java? Curiosidades sobre o nome da linguagem. JavaWorld, outubro de 1996. Disponível em: <http://www.javaworld.com/javaworld/jw-10-1996/jw-10-javaname.html>.
- SCHWARTZ, Jonathan. Blog do Jonathan Schwartz, ex-CEO da Sun, falando da origem do nome Java. Sun Microsystems. Disponível em: [http://blogs.sun.com/jonathan/entry/better\\_is\\_always\\_different](http://blogs.sun.com/jonathan/entry/better_is_always_different).
- WIKIPÉDIA. A História das versões do Java. Disponível em: [http://en.wikipedia.org/wiki/Java\\_version\\_history](http://en.wikipedia.org/wiki/Java_version_history)
- DUKE. Versão preliminar da especificação do Oak (Oak Language Specification, v0.2, 32 páginas).
- PROJETO JEDI. Página Oficial do JEDI. Disponível em: link desconhecido; <http://jedi.wv.com.br/>.