

Linguagem de Programação 1

Ciência da Computação

Prof. Fábio José Rodrigues Pinheiro

SETEMBRO DE 2021



**INSTITUTO
FEDERAL**

Catarinense

Campus
Videira

Fundamentos da Linguagem Java

Tipos primitivos

```
1 byte b = 65;  
2 char c = 'A'; // ou c = 65;  
3  
4 int i = 65;  
5 long l = 65L;  
6 short s = 65;  
7  
8 double d = 65.1;  
9 float f = 65.1f; // ou f = (float) 65.1;  
10  
11 boolean b = true; // ou false
```

[Clique aqui para ver a documentação oficial](#)



Lendo informações pelo teclado

```
1 import java.util.Scanner;
2
3 public class LeitorTeclado{
4
5     public static void main(String[] args){
6
7         Scanner teclado = new Scanner(System.in);
8
9         System.out.print("Entre com seu nome: ");
10        String s = teclado.nextLine(); // lendo cadeia de caracteres
11        System.out.println("Nome:   " + s);
12        System.out.print("Informe um número inteiro: ");
13        int i = teclado.nextInt(); // lendo inteiro
14        System.out.print("Informe um número real: ");
15        double r = teclado.nextDouble(); // lendo real
16        System.out.println("inteiro: " + i + ", real: " + r);
17    }
18 }
```



- Se o método `nextLine()` for chamado depois dos métodos `nextInt()`, `nextDouble()`, `nextFloat()`, `nextByte()`, `nextShort()`, `nextLong()` ou `next()`, então ele não irá ler os valores
- Os métodos `nextXXXX()` ignoram o caractere de nova linha (NL) e assim esse caractere é consumido pelo `nextLine()` subsequente
- **Solução:** Adicionar um chamada extra do método `nextLine()`

```
1 int i = teclado.nextInt();  
2 teclado.nextLine(); // chamada extra para consumir NL  
3 String s = teclado.nextLine(); // lendo cadeia de caracteres
```



Classe String

```
1 String s = "IFC Videira";
2
3 // a partir da posição 0 pegue 5 caracteres
4 String sub = s.substring(0,5);
5
6 sub = "Videira";
7 // retorna a posição que a string sub se encontra em s; caso contrário
   retorna -1
8 System.out.println(s.indexOf(sub));
9
10 int tamanho = sub.length(); //obtendo o tamanho
11 char c = sub.charAt(1); // obtendo caractere na posição 1
12
13 String alunos = "Joao:Pedro:Ana";
14
15 // criará vetor de Strings com 3 elementos
16 String[] vetAlunos = alunos.split(":");
17 System.out.println(vetAlunos[0]); // Joao
```



- if-else
- switch case

```
1 String s = "IFC";
2 String s2 = null; // valor nulo
3
4 if (s2.isEmpty()){ // verifica se está vazia
5     System.out.println("Vazia");
6 }
7
8 s2 = "IFC Videira";
9
10 if (s.equals(s2)){ // para comparar Strings
11     System.out.println("Iguais");
12 }else if (s.contains(s2)){
13     System.out.println("s contém s2");
14 }else if (s2.contains(s)){
15     System.out.println("s2 contém s");
16 }
```



- Faça um programa que leia do teclado dois números inteiros e um operador (+, -, \ ou *) e apresente o resultado da operação aritmética.

```
1 Entre com o 1o. operando: 10
2 Entre com o 2o. operando: 20
3 Entre com o operador: +
4
5 Resultado: 10 + 20 = 30
```



Funções Matemáticas

```
1
2 // algumas funções matemáticas
3 double d = Math.sqrt(25); // obtém a raiz quadrada
4 d = Math.pow(4,2); // 4 elevado a 2
5 d = Math.sin(45); // Math.cos(45), Math.tan(45)...
6
7 // obtendo números pseudo-aleatórios de 0 a 9
8 Random r = new Random();
9 int i = r.nextInt(10);
10
11 // formatando a saída
12 // largura de campo de 8 caracteres e precisão de 2 caracteres
13 System.out.printf("%8.2f", d);
```



Usando interface gráfica para interagir com o usuário

```
1 import javax.swing.JOptionPane;
2
3 public class Tela{
4
5     public static void main(String[] args){
6         String s = JOptionPane.showInputDialog("Entre com um numero");
7
8         //convertendo String para int
9         int numero = Integer.parseInt(s);
10
11         JOptionPane.showMessageDialog(null, numero);
12     }
13 }
```



```
1 int[] vet = new int[10]; //vetor de inteiros com 10 posições
2 vet[0] = 5;
3 vet[9] = 4;
4 int[][] mat = new int[2][2]; //vetor com 2 dimensões (matriz)
```



Estruturas de repetição

```
1 int[] vet = new int[10];
2 for (int i = 0; i < vet.length; i++) {
3     vet[i] = i*3;
4 }
5 //foreach
6 for (int x : vet) {
7     System.out.println(x);
8 }
9 int i = vet.length;
10 while(i >=0){
11     i--;
12     System.out.println(vet[i]);
13 }
```



Argumentos de linha de comando

```
1 public class Argumentos{
2
3     public static void main(String[] args){
4
5         System.out.println("Forneceu " + args.length() + "argumentos");
6         System.out.println("Argumentos fornecidos: ");
7
8         for(String argumento: args){
9             System.out.println(argumento);
10        }
11        // o laço abaixo é equivalente ao laço acima feito com foreach
12        for(int i = 0; i < args.length(); i++){
13            System.out.println(args[i]);
14        }
15    }
16 }
```

```
1 javac Argumentos.java
2 java Argumentos Ola mundo
```

