

Revisão: Variáveis, Operadores Aritméticos e  
Expressões Aritméticas  
Strings: tamanho, partes, comparação

Programação Orientada a Objetos – Aula 02  
Professores: Bonato e Machion

Adaptado do material do Prof. Fúlvio Prevot com autorização.

## Variáveis

- Para armazenar dados na memória do computador são usadas as **variáveis**.
- Uma variável corresponde a uma porção bem delimitada e bem definida da memória do computador, na qual serão armazenados dados de um determinado tipo.

## Tipos de Dados

- De acordo com o tipo de dado ou informação armazenada, uma variável pode usar mais ou menos quantidade de memória do computador.
- Dados podem ser:
  - **De tipo primitivo:** são os tipos mais simples, tais como números inteiros, números reais, lógico (verdadeiro ou falso) e caracter.
  - **De tipo “composto” (ou abstrato, “de objeto”):** resultante da combinação de tipos primitivos ou de outros tipos abstratos (exemplo: sequência de caracteres).
- Tipos primitivos: números inteiros, números reais, caracteres, variáveis lógicas (verdadeiro ou falso)
- Tipos objeto: cadeias de caracteres (strings), tipos definidos pelo usuário, compostos por tipos primitivos e tipos objetos. Ex: tipo Aluno, composto por nome (cadeia de caracteres) e idade (inteiro).

## Introdução à Linguagem Java



- Alguns tipos de Dados
  - **int:** número inteiro
  - **double:** número real
  - **char:** caractere
  - **boolean:** valor lógico (true ou false)
  - **String:** sequência de caracteres

## Introdução à Linguagem Java



- Declarando uma classe

```
public class ContaCorrente
{
    . . . .
}
```

## Introdução à Linguagem Java



- Declarando o método main

```
public class ContaCorrente
{
    public static void main(String[] args)
    {
        . . . .
    }
}
```

## Introdução à Linguagem Java



- O método main
  - precisa ser escrito exatamente como no slide anterior
  - é necessário para que o programa escrito em uma classe seja executado
  - mas não precisa existir em todas as classes de um programa, como veremos ao longo do curso
  - as classes podem possuir outros métodos, como também veremos ao longo do curso

## Introdução à Linguagem Java



- **Comentários** são observações e explicações feitas pelo programador no corpo da classe.
- **São desconsiderados pelo compilador. Há dois tipos:**
  - Comentário de uma linha, indicado por um `//`
    - Exemplo: `//este método calcula o saldo`
  - Comentário de várias linhas, que começa com um `/*` e termina com um `*/`
    - Exemplo: `/*esta classe contém os dados de conta corrente do cliente, possuindo métodos para sacar e depositar valores e para calcular o saldo */`

## Introdução à Linguagem Java



- **Declarando variáveis locais:** basta escrever, dentro do bloco, o tipo e o nome da variável.

```
public class ContaCorrente
{
    public static void main(String[] args)
    {
        int x;
        double y;
        String nome;
    }
}
```

## Introdução à Linguagem Java



- **Atribuição:** é feita usando o sinal de igual =; coloca-se ao lado esquerdo do sinal um valor, outra variável ou uma expressão aritmética. Ex:

```
int x = 10;
y = 45.3;
char c = 'c';
String nome = "Vitória";
z = x * 3;
y = y / x;
ePar = true;
```

## Introdução à Linguagem Java



- **Operadores aritméticos:** são usados para construir expressões aritméticas.
  - adição: **+**
  - subtração: **-**
  - multiplicação: **\***
  - divisão: **/**
  - módulo (resto da divisão de inteiros): **%**
  - Exemplo: `d = 5 * (c * 10 - a / ( x + 14 ) );`

## Introdução à Linguagem Java



- **Operadores aritméticos especiais:**
  - `x += 5;` equivale a `x = x + 5`
  - `v -= d;` equivale a `v = v - d;`
  - `f *= 10;` equivale a `f = f * 10;`
  - `z /= t;` equivale a `z = z / t;`
  - `u %= 2;` equivale a `u = u % 2;`
  - `c++;` equivale a `c = c + 1;`
  - `b--;` equivale a `b = b - 1;`

## Introdução à Linguagem Java



- **Classe Math:** contém vários métodos de funções matemáticas, como por exemplo:
  - `Math.sqrt(x)` : retorna o valor da raiz quadrada do número colocado entre parênteses.
  - `Math.random()` : retorna um número aleatório real entre 0 e 1.
  - `Math.abs(x)` : retorna o valor absoluto do valor colocado entre parênteses.
  - `Math.pow(x, y)` : retorna o valor do primeiro parâmetro elevado ao segundo parâmetro.

## Introdução à Linguagem Java



- Classe `JOptionPane` (package `javax.swing`): contém métodos que apresentam diálogos para entrada ou saída de dados.
  - método `showInputDialog(x)` : mostra uma caixa de diálogo que apresenta a String `x` e permite ao usuário digitar dados que serão retornados a uma String após o usuário clicar o botão Ok ou teclar <Enter>.
    - Ex: `String nome = JOptionPane.showInputDialog("Digite seu nome");`
  - método `showMessageDialog(null, x)` : mostra uma caixa de diálogo que exibe o texto contido na String `x`.
    - Ex: `JOptionPane.showMessageDialog(null, "Resultado = " + y);`

Para usar esses métodos é preciso escrever no início do programa:

```
import javax.swing.JOptionPane;
```

## Introdução à Linguagem Java



- Método **println(x)**: exibe, na linha de comando, a String que estiver como parâmetro dentro dos parênteses e pula uma linha. Exemplos:

```
System.out.println("Erro do sistema");
```

```
System.out.println("Soma = " + (x + y) + " reais.");
```

## String



- Para criar uma string, usamos:  

```
String s = "esta é uma string";
```
- Usamos a palavra string com letra minúscula para indicar uma cadeia de caracteres; mas o tipo String em Java começa sempre com letra maiúscula.



## String



- **Concatenação de Strings:** é a união de duas ou mais Strings. Todos os tipos de dados concatenados a Strings são convertidos para Strings.

```
int g = 10;  
String x = "Bom";  
x += " Dia"; //o conteúdo de x agora é "Bom Dia"  
x += ", " + g; //agora x vale "Bom Dia, 10"
```

## O tamanho da String



- Para saber o tamanho de uma string, basta usar o método `length()`
  - `"12345678".length()` retorna 8 (que é um int).
  - `String s = ""; s.length()` retorna 0
  - `String s1; s1.length()` dá erro (`NullPointerException`); você não pode chamar métodos de strings não inicializadas

## Um pedaço da String



- Para pegar um pedaço de uma string use o método `substring(inicio, fim)`.
- Início é um inteiro que marca o começo do pedaço que queremos; não se esqueça que a primeira posição de uma string é 0;
- Fim é um inteiro que marca o fim do pedaço que queremos + 1; se colocarmos 2, o método vai retornar até o caractere que está na posição 1; se colocarmos o tamanho da string, vai retornar até o último caractere.
  - `"012345".substring(0,6)` retorna `"012345"`
  - `"012345".substring(0,2)` retorna `"01"`
  - `"012345".substring(3,4)` retorna `"3"`
  - `"012345".substring(3,3)` retorna `""`
  - `"012345".substring(0,7)` dá erro (`StringOutOfBoundsException`), pois esta string começa em 0 e termina em 5, não em 6.

## Conversão de Strings

- Como o `JOptionPane.showInputDialog("")` sempre retorna um valor do tipo `String`, é necessário fazer conversões para armazenar este valor em uma variável correspondente.
- Para converter para `int`, use:
  - `Integer.parseInt("123");` retorna o inteiro 123
- Para converter para `double`, use:
  - `Double.parseDouble("0.75");` retorna o double 0.75

# Conversão de Strings

- Para converter para boolean, use:
  - `Boolean.parseBool("true");` retorna true – qualquer outra string, exceto "TRUE", "tRUe" e outras combinações de "true", retorna false.
- Para converter para char, use:
  - `"a".charAt(0);` retorna o primeiro caractere da string; neste caso, 'a'.
- **Obs:** caso o valor digitado não possa ser convertido, como querer que "Dez" vire 10, ocorrerá um erro que irá encerrar o programa (depois veremos como evitar isso).