

Conceitos de Programação  
Entrada e Saída de Dados  
Variáveis  
Conversão de Strings

Programação de Computadores – Aula 01  
Prof. Sergio Bonato

Adaptado do material do Prof. Fúlvio Prevot com autorização.

# Aviso

- Os conceitos iniciais apresentados nesta aula serão aprofundados ao longo da disciplina de Lógica de Programação.
- Portanto, não se preocupe se formos um pouco rápido em alguns pontos.
- O que será ensinado em cada disciplina:
  - Lógica de Programação: pensar programaticamente.
  - Programação de Computadores: programar em Java.

# Conceitos e Fundamentos de Programação

- Programação é toda atividade envolvida na criação, testes, desenvolvimento, manutenção e implantação de programas e sistemas para computadores.
- Algoritmo é uma sequência de passos ou etapas ordenadas de maneira racional e lógica, com a finalidade de resolver um problema.
- Programa é uma sequência de instruções ou comandos, ordenados de maneira racional e lógica, de modo que o computador possa automaticamente executar tarefas ou resolver um problema.
- Sistema (Computacional) é um conjunto de programas que devem trabalhar de maneira integrada e harmônica, de modo a realizar um determinado conjunto de tarefas ou resolver determinado problema.

# Conceitos e Fundamentos de Programação

- Linguagem de programação é um conjunto de símbolos, comandos e funções, usada para escrever um programa de computador.
- O Programador, após analisar o problema e construir o algoritmo, usa uma linguagem de programação, para escrever o programa, de forma que este execute automaticamente a ideia da solução expressa no algoritmo.
- Há várias maneiras de se construir e representar um algoritmo; duas técnicas bastante usuais são:
  - Fluxograma ou diagrama de blocos
  - Português estruturado (ou “Portugol”).
  - Obs: estas representações de algoritmos serão vistas em detalhes na disciplina de Raciocínio Lógico.

# Representação de Algoritmos

- No “Portugol”, escreve-se a solução passo a passo, usando-se um português bem simples.
- Exemplo: Algoritmo para soma de dois números.

**inicio**

ler a;

ler b;

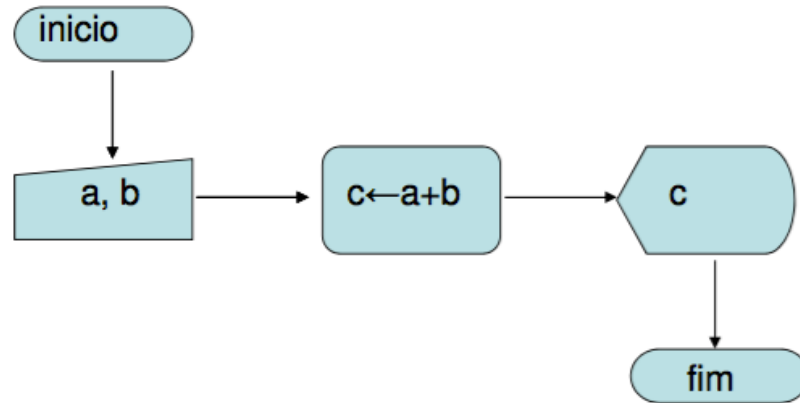
$c \leftarrow a + b$ ;

escrever c;

**fim**

# Representação de Algoritmos

- Num diagrama de blocos (ou fluxograma), o algoritmo é representado, usando-se figuras geométricas.
- Exemplo: Algoritmo para somar dois números.



# Programa de Computador

- Uma vez construído o algoritmo, ou seja, uma vez que as etapas e os detalhes da solução estão definidos, pode-se escrever um programa de computador, a fim de se ter uma forma automática de se testar e usar a solução do problema.
- Normalmente, um programa é constituído de três fases:
  - Entrada de dados
  - Processamento
  - Saída de dados
- Após escrever o programa, usando uma determinada linguagem de programação, o programador usa o compilador.

# Compilador e Variáveis

- Compilador é o programa que faz a tradução do programa, escrito na linguagem de programação escolhida pelo programador (por exemplo, Java), para a linguagem de máquina (ou para uma linguagem intermediária).
- 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 ao Java



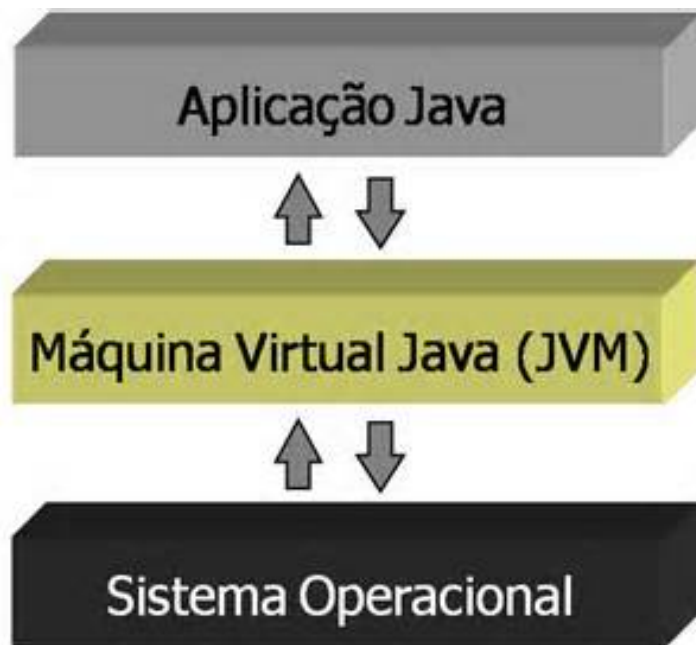
- Linguagem de programação criada por James Gosling, da Sun, em 1995
- Baseada na linguagem C
- Orientada a objetos
- Teve como princípio ser um C++ melhorado



# Introdução ao Java



- O princípio do Java é “escreva o código uma vez, rode em várias máquinas”.
- Por isso é baseada em máquinas virtuais.
- Atualmente está na versão 8.



# 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";  
ePar = true;
```

# Introdução à Linguagem Java



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

# Introdução à Linguagem Java



- Classe `JOptionPane` (da 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 retornados a uma String após o usuário clicar o botão Ok.
    - Ex: `String z = JOptionPane.showInputDialog("Digite seu nome");`
  - método `showMessageDialog(null, x)`: mostra uma caixa de diálogo que exibe o texto contido na variável `x`.
    - Ex: `JOptionPane.showMessageDialog(null, "Resultado = " + y);`

# 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. Ex:

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

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

# 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");` isto retorna o inteiro 123
- Para converter para `double`, use:
  - `Double.parseDouble("0.75");` isto retorna o double 0.75

# Conversão de Strings

- Para converter para boolean, use:
  - `Boolean.parseBool("true");` isto retorna true – qualquer outra string, exceto "TRUE", "tRUe" e outras combinações de "true", retorna false.
- Para converter para char, use:
  - `"a".charAt(0);` isto 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).