

# Programação Aplicada

Aula 1 – Introdução e Impressão

**Profº Ms Gustavo Molina**

# Agenda

- Introdução ao JAVA
  - Histórico
  - JVM- Java Virtual Machine
    - Características
  - Estudo do Ambiente para o desenvolvimento de programas;
  - Edição, compilação e enlace;
  - Execução e Configuração;



# Introdução ao JAVA [I/II]

## Histórico

- Primeiros passos 1991, primeira versão disponível para *download*: 1995.
- Originalmente desenvolvido para dispositivos eletrônicos inteligentes de consumo popular.
- Depois utilizado para criar páginas da Web com conteúdo dinâmico.

## Agora também utilizado para:

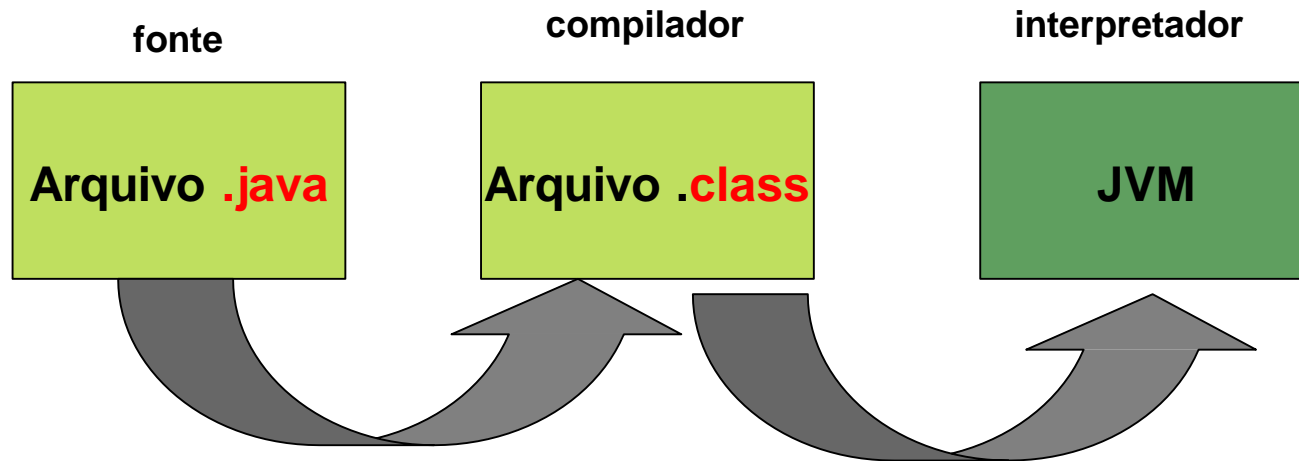
- Desenvolver aplicativos corporativos de larga escala.
- Aprimorar funcionalidades de servidores Web.
- Fornecer aplicativos para dispositivos de consumo popular (telefones celulares, eentre outros.)

# Introdução ao JAVA [II/II]

- Programas Java
  - Consistem em **partes** chamadas **classes**, as quais incluem *métodos* que realizam tarefas e retornam informações ao concluir.
  - Programadores podem criar classes e métodos para construir programas Java.
- O Java oferece bibliotecas de classe
  - Conhecidas como Java **APIs (Application Programming Interfaces)** ou APIs do Java.

# Java Virtual Machine (JVM)

- É uma abstração, provendo, um ambiente multi plataforma para publicação de aplicativos JAVA;
- Recebe chamadas *bytecodes*;
- Chamadas são independentes do SO;
- *Bytecodes* são gerados a partir de um compilador JAVA;



# Exemplo JVM

Teste.java

```
class Teste {  
  
    public static void main (String args[ ]) {  
        System.out.print("Primeira aula");  
        System.out.print("Linguagem Java");  
    }  
}
```

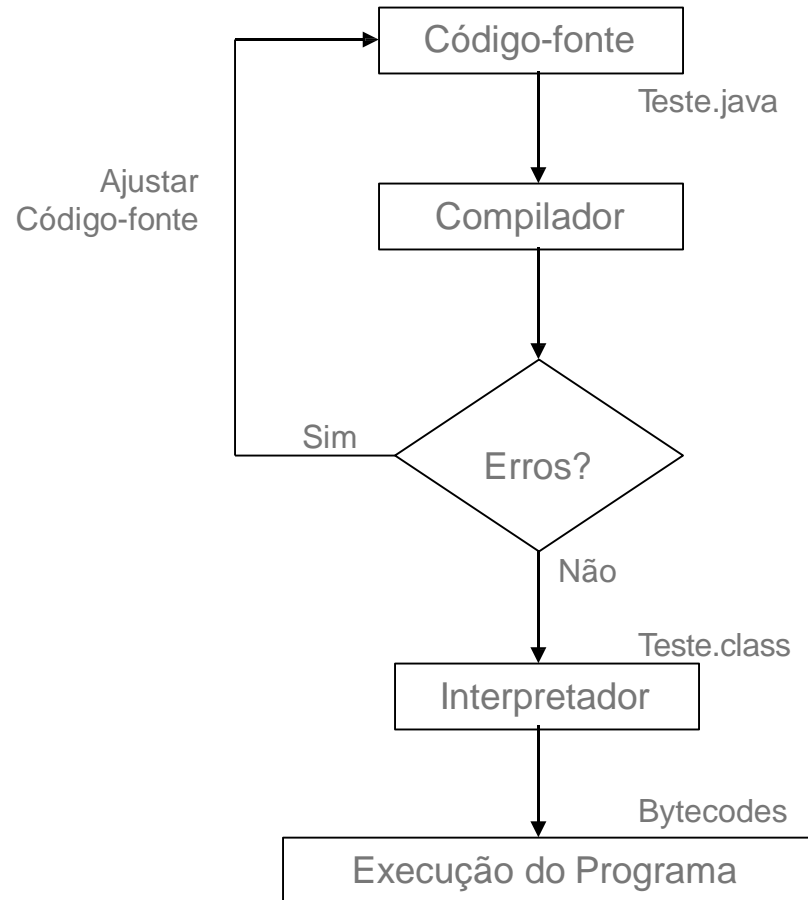
**Compilador JAVA**

Bytecodes

```
Ëp%4???1?  
??      ?? ?  
? ? ? ? ? ? <init> ? ()V ? Code ? LineNumberTable ? main ? ([Ljava/lang/String;)V ?  
          SourceFile ?  
Teste.java? ?      ? ? ? ?
```

**JVM**

# Sequência de criação/execução



Na fase de execução é necessário que haja a Máquina Virtual Java (MVJ)

A MVJ interpreta os *bytecodes* gerados pelo interpretador.

O objetivo da MVJ é permitir que qualquer sistema operacional possa executar uma aplicação Java

# JAVA - Principais Características

- É uma linguagem de desenvolvimento orientada a objetos;
- É Multiplataforma;
- Uma coleção de APIs (classes, componentes, *frameworks*) para o desenvolvimento de aplicações multiplataforma;
- Um ambiente de execução presente em browsers, mainframes, SOs, celulares, cartões inteligentes, eletrodomésticos, entre outros.



# Primeiro programa Java

## Welcome1.java

```
1 // Fig. 2.1: welcome1.java
2 // Programa de impressão de texto.
3
4 public class welcome1
5 {
6     // método main inicia a execução do aplicativo Java
7     public static void main( String args[] )
8     {
9         System.out.println( "Welcome to Java Programming!" );
10
11     } // fim do método main
12
13 } // fim da classe welcome1
```

Welcome to Java Programming!

# Explicação

## Comentários iniciam com: //

- Comentários ignorados durante a execução do programa.
- Documentam e descrevem o código.
- Fornecem legibilidade do código.

## Comentários tradicionais: /\* . . . \*/

```
/* Este é um comentário  
   tradicional. Ele pode se estender  
   por várias linhas */
```

Nota: os números de linha não fazem parte do programa; eles foram adicionados para referência.

# Erros comuns em Java

- O Java diferencia letras maiúsculas de minúsculas **[é “*case sensitive*”]**. Não diferenciar as letras maiúsculas e minúsculas adequadas para um identificador normalmente causa um erro de compilação, onde **a1** e **A1** são diferentes.

# Primeiro Programa em Java com o VS CODE

1) Abrir o VS Code e digite o código abaixo

```
class Teste {  
  
    public static void main (String args[ ]) {  
        System.out.print("Primeira aula");  
        System.out.print("da Linguagem Java");  
    }  
}
```

2) Salve o documento como Teste.java [o mesmo nome da classe obrigatoriamente].

3) Execute o programa.

# Utilizando o println

Agora teste com o println.

```
class Teste {  
  
    public static void main (String args[ ]) {  
        System.out.println("Primeira aula");  
        System.out.println("da Linguagem Java");  
    }  
}
```

# Exemplos de utilização dos caracteres de formatação

```
class Teste {  
  
    public static void main (String args[ ]) {  
        System.out.println("Primeira aula da \n Linguagem Java");  
    }  
}
```

```
class Teste {  
  
    public static void main (String args[ ]) {  
        System.out.println("Primeira aula da \t Linguagem Java");  
    }  
}
```

```
class Teste {  
  
    public static void main (String args[ ]) {  
        System.out.println("Primeira aula da \r Linguagem Java");  
    }  
}
```