



# Java™

## Aula 1 - JAVA

Roberto Silva Araújo Assis

# Introdução à Linguagem Java

- Surgiu em 1991
- Cientistas da Sun: liderados por Patrick Naughton e James Gosling
- Linguagem pequena para ser utilizada em eletrodomésticos em geral
- O projeto se chamava “Green”.
- basearam a linguagem em C++ e ela foi chamada originalmente de “Oak”.
- Oak é um tipo de árvore na qual Gosling gostava de apreciar.
- mas o nome já existia, então, trocaram o nome para Java.

# Introdução à Linguagem Java

- 1993 e 1994 procuraram uma empresa para comprar a tecnologia, mas não encontraram ninguém.
- Patrick Naughton, uma das pessoas responsáveis pelo marketing, conseguiu acumular 300.000 milhas aéreas de bônus tentando vender a tecnologia Java.
- Com o crescimento da World Wide Web (www), para mostrar o poder do java construíram o browser HotJava.
- HotJava reconhecia applets, e por isso tinha uma máquina virtual.
- 1995 Netscape incorporou (versão 2.0) reconheceu o applet.
- Posterior todas as novas versão dos outros browsers também reconheciam applet.

# Características do Java

## Parecida com C, C++:

- A sintaxe da linguagem Java é muito semelhante a da linguagem C ou de C++. Logo, o programador que já conhece a linguagem C achará a transição para Java simples e fácil.
- A sintaxe de Java fornece uma versão mais limpa do que C++. Em Java não existe aritmética de ponteiros, estruturas, uniões e etc.

# Características do Java

## Orientada a Objetos:

- Java é uma linguagem voltada para a programação orientada a objetos e, por isso, todo o código está contido dentro de classes.
- Java suporta herança simples, mas não herança múltipla. A ausência de herança múltipla pode ser compensada pelo uso de herança e interfaces, onde uma classe herda o comportamento de sua superclasse além de oferecer uma implementação para uma ou mais interfaces.

# Características do Java

## Compilada e Interpretada:

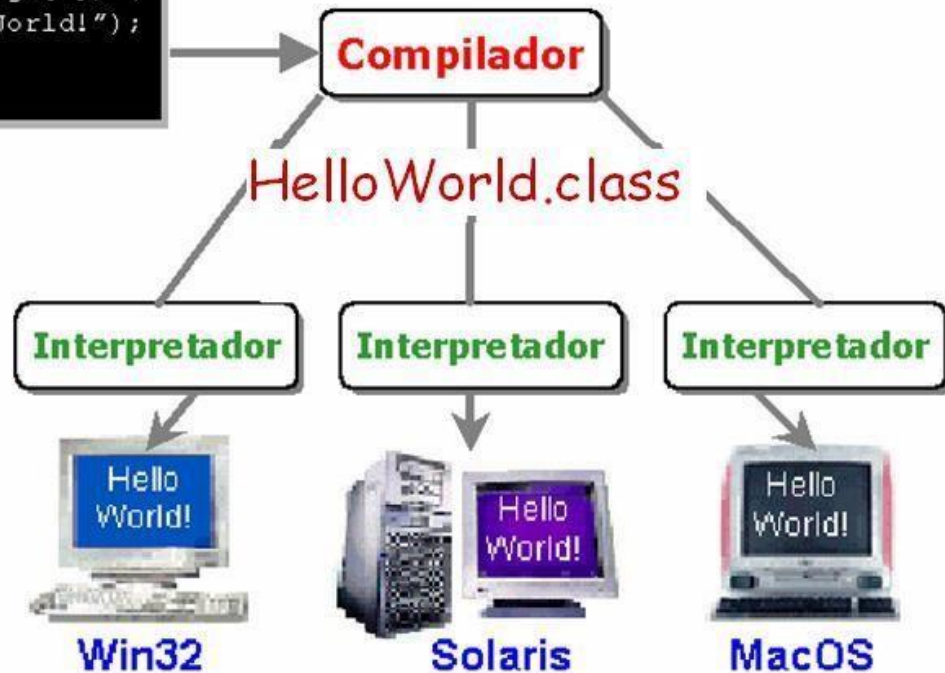
- Todo programa Java é compilado e interpretado. Um programa em Java é compilado para um código composto por instruções chamadas de “bytecode”. O “bytecode” é um código de uma máquina virtual, chamada Máquina Virtual Java (Java Virtual Machine - JVM), idealizada pelos criadores da linguagem. Os bytecodes são independentes de plataforma e são interpretados pela JVM para serem executados no computador.

# Características do Java

## Programa Java

```
public class HelloWorld {  
    public static void main (String args[ ]) {  
        System.out.println ("Hello, World!");  
    }  
}
```

HelloWorld.java



# Características do Java

## Segura:

- A presença de coleta automática de lixo (Garbage Collection) evita erros comuns que os programadores cometem quando são obrigados a gerenciar diretamente a memória (C, C++, Pascal).
- A eliminação do uso de ponteiros, em favor do uso de vetores, objetos e outras estruturas substitutivas.
- O programador é proibido de obter acesso à memória que não pertence ao seu programa, além de não ter chances de cometer erros comuns tais como uso indevido de aritmética de ponteiros.
- A presença de mecanismos de tratamento de exceções torna as aplicações mais robustas



# Características do Java

## Suporta concorrência:

- Java suporta Multithreading que é a capacidade de um programa fazer mais de uma coisa ao mesmo tempo como, por exemplo, imprimir enquanto lê um fax. A linguagem permite a criação, de maneira fácil, de vários “threads” de execução, que eliminam o período de inatividade do processador executando concorrentemente ou tomando vantagem da execução paralela real em sistemas multiprocessadores.

# O Primeiro Programa Java

## Passo a Passo

```
1. /* Meu programa Java */  
2. public class HelloWorld  
3. {  
4.     public static void main (String args[ ])  
5.     {  
6.         System.out.println ("Hello, World!");  
7.     } // da rotina main  
8. } // da class
```

**Linha 2:** public class HelloWorld

Esta linha utiliza a palavra reservada class para declarar que uma nova classe será definida aqui. HelloWorld é o nome usado para identificar a classe. Toda a definição da classe, inclusive todo o código e os dados, estarão entre a chave de abertura "{" e a chave final "}" que se encontram nas linhas 5 e 8 deste exemplo.

# O Primeiro Programa Java

## Passo a Passo

```
1. /* Meu programa Java */
2. public class HelloWorld
3. {
4.     public static void main (String args[ ])
5.     {
6.         System.out.println ("Hello, World!");
7.     } // da rotina main
8. } // da class
```

**Linha 4:** public static void main (String args[ ])

A linha 4 contém a declaração do método **main**. O método main é simplesmente um ponto de partida para o interpretador Java. É por onde será iniciada a execução. O método main deverá sempre ser declarado na forma acima.;

# O Primeiro Programa Java

## Passo a Passo

```
1. /* Meu programa Java */
2. public class HelloWorld
3. {
4.     public static void main (String args[ ])
5.     {
6.         System.out.println ("Hello, World!");
7.     } // da rotina main
8. } // da class
```

**Linha 6:** `System.out.println ("Hello, World!");`

Esta linha executa o método **println** do objeto **out**. Este objeto é uma instância da classe **OutputStream** e foi declarado como variável de classe (static) na classe **System**. Este método imprime na tela uma mensagem texto, no caso, **"Hello, World!"**.

# Que nome dar ao arquivo?

- Os arquivos que contém o código-fonte Java devem sempre ter a terminação “.java”.
- Geralmente, em Java, coloca-se uma classe dentro de cada arquivo. O arquivo conterá o mesmo nome da classe.
- Cuidado, o compilador Java diferencia letras maiúsculas de minúsculas (case sensitive).
- Se você utilizar o modificador public para a classe, por exemplo, `public class HelloWorld`, o arquivo deve possuir o mesmo nome que a classe. Caso não utilize public, o arquivo pode ter outro nome.

# Compilando o código

Para compilar o código acima você deve digitar:

```
C:\> javac HelloWorld.java
```

O compilador é chamado pelo comando `javac` seguido do nome do arquivo sempre com a terminação `“.java”`. Ao ser compilado, se o código não possuir nenhum erro, será gerado um arquivo chamado `HelloWorld.class` composto por `bytecodes`. Esse programa é independente de plataforma e, por isso, o seu programa `HelloWorld (.class)` pode ser executado em qualquer sistema operacional que possua a JVM instalada.

# Executando o programa

Após a compilação do programa, em que foi gerado o arquivo HelloWorld.class, você pode executá-lo digitando o seguinte comando:

```
C:\>java HelloWorld
```

No código acima estamos chamando o interpretador Java (java.exe) para carregar a classe HelloWorld e executá-la. No comando de execução não se deve digitar a extensão do arquivo (.class).

# Exercício

## Exercício 1 – Pede Argumento

Faça um programa Java leia um argumento passado para a Máquina Virtual Java. O programa deve ser formado por uma única classe e contém apenas o método main ( ). A classe deve se chamar PedeArgumento e você passará o seu nome como argumento. Por exemplo, se o seu nome é Nicolas, você deve executar o programa através do seguinte comando:

```
C:/> java PedeArgumento SeuNome
```



# Dica:

Todo o parâmetro passado para a MVJ é recebido pela array de String do método main(). Para acessar um argumento de uma array, forneça a posição da array que você deseja acessar. Por exemplo, como nós fornecemos apenas uma String ela estará na primeira posição da array que é zero: `String nome = args [0];`

Se você digitasse duas strings (por exemplo, o seu nome e sobrenome, seria necessário acessar as duas primeiras posições da array: 0 e 1).

**Qual a diferença entre  
System.out.println  
e**

**System.out.print**

Faça um programa e exemplifique os dois métodos.

# Estruturas Fundamentais de Programação em Java:

## **Espaço em Branco:**

Java é uma linguagem de formato livre. Não é necessário endentar para que ela funcione corretamente. Por exemplo, o nosso programa HelloWorld poderia ser escrito da seguinte maneira:

```
public class HelloWorld{public static void main (String a[ ] ) {System.out.println (“Oi, mundo!”);}}
```

Devemos, porém, tomar cuidado para que haja pelo menos um espaço, Tab ou uma nova linha entre palavras que não estejam separadas por um operador ou separador.

# Estruturas Fundamentais de Programação em Java:

## Comentários em Java:

Os comentários em código-fonte Java podem ser de 3 tipos:

- `//` (duas barras) e terminam no final da linha.
- O comentário de várias linhas é usado para comentários mais longos que uma linha e é iniciado com `/*` (barra-asterisco) e finalizado com `*/`
- Existe ainda o comentário de documentação que começa com `/**` (barra-asterisco-asterisco) e finaliza com `*/` (asterisco-barra). O uso da ferramenta Javadoc para gerar documentação HTML.

# Palavras-chave reservadas:

abstract	boolean	break	byte	byvalue
case	cast	catch	char	class
const	continue	default	do	double
else	extends	false	final	finally
float	for	future	generic	goto
if	implements	import	inner	instanceof
int	interface	long	native	new
null	operator	outer	package	private
protected	public	rest	return	short
static	super	switch	synchronized	this
throw	throws	transient	true	try
var	void	volatile	while	