

# Parte 02

## Desenvolvimento de Programação Orientada a Objetos

Java – Conceitos e  
Definições

Prof. Pedro Neto

Aracaju – Sergipe - 2011

# Conteúdo

## 2 O que é Java

- i. Java
- ii. Máquina Virtual
- iii. Java lento? Hotspot e JIT
- iv. Versões do Java e a confusão do Java2
- v. JVM? JRE? JDK? O que devo baixar?
- vi. Onde usar e os objetivos do Java
- vii. Especificação versus implementação
- viii. Preparando o Ambiente de Trabalho
- ix. Seu primeiro programa
- x. O que aconteceu?
- xi. Exercícios: Modificando o Programa
- xii. O que pode dar errado?
- xiii. Exercícios adicionais

# Java – Conceitos e Definições

## Java

Maiores problemas de desenvolvimento na década de 1990:

- Uso de Ponteiros?
- Gerenciamento de memória?
- Organização?
- Falta de bibliotecas?
- Ter de reescrever parte do código ao mudar de sistema operacional?
- Custo financeiro de usar a tecnologia?



A linguagem Java veio a resolver estes problemas, comuns a outras linguagens de programação na época.

# Java – Conceitos e Definições

## Java

A linguagem, a princípio, tinha o objetivo de ser executada em pequenos dispositivos como TVs, video-cassetes, liquidificadores, e eletro-eltrônicos em geral. Porém para o lançamento, a linguagem já teve seu foco redirecionado para o desenvolvimento de clientes web para serem executados nos browsers, os chamados *applets*.



Porém, apesar de ter sido criado com um propósito e lançado com outro, o Java ganhou destaque e espaço no desenvolvimento de aplicações *server-side*.

# Java – Conceitos e Definições

## Java – Breve História

- 1991 - A **Sun** criou o Green Team para desenvolver inovações tecnológicas.  
Esse time foi liderado por James Gosling, considerado o pai do Java.
- 1992 – Apresentação da proposta inicial.  
Interpretador para pequenos dispositivos (VM)(reuso do software)  
Plataforma Oak  
Uso para interação em TV a cabo  
Fracasso da idéia a princípio  
Mudança de foco: Internet
- 1995 - Lançamento da plataforma Java  
Desenvolvimento da tecnologia **applet**.

Crescimento contínuo até os dias atuais, com foco no desenvolvimento de aplicações para internet (arquitetura web), principalmente no lado servidor. Grande comunidade de desenvolvedores e empresas que desenvolvem tecnologias e soluções com a plataforma Java (IBM, Oracle, etc.).

# Java – Conceitos e Definições

## Java – Breve História

O Java foi desenvolvido e mantido pela Sun (<http://www.sun.com>) através de um comitê (<http://www.jcp.org>). Com a compra da Sun pela Oracle em 2009, atualmente é esta empresa que mantém a tecnologia. (<http://www.oracle.com/technetwork/java/index.html>)

The Oracle logo, featuring the word "ORACLE" in a red sans-serif font with a registered trademark symbol (®) to the right, all contained within a white rectangular box with a thin grey border.

ORACLE®

# Java – Conceitos e Definições

## Java – Características

- Orientação a objetos - Baseado no modelo da Simula67;
- Portabilidade - Independência de plataforma;
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;
- Segurança - Pode executar programas via rede com restrições de execução;
- Sintaxe similar a C/C++.
- Facilidades de Internacionalização - Suporta nativamente caracteres Unicode;
- Simplicidade na especificação, tanto da linguagem como do "ambiente" de execução (JVM);
- É distribuída com um vasto conjunto de bibliotecas (ou APIs);
- Possui facilidades para criação de programas distribuídos e multitarefa (múltiplas linhas de execução num mesmo programa);
- Desalocação de memória automática por processo de coletor de lixo;
- Carga Dinâmica de Código - Programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização.

# Java – Conceitos e Definições

## Java – Máquina Virtual

Linguagem de programação tradicional (C, C++, Pascal, etc.)

Uma compilação  
para cada S.O.

Código Fonte

Ex.: Arquivos .pas,  
.c, .cpp, etc.



**Compilação**

Código Binário  
do S.O.

Ex.: Arquivos .exe do  
Windows

# Java – Conceitos e Definições

## Java – Máquina Virtual

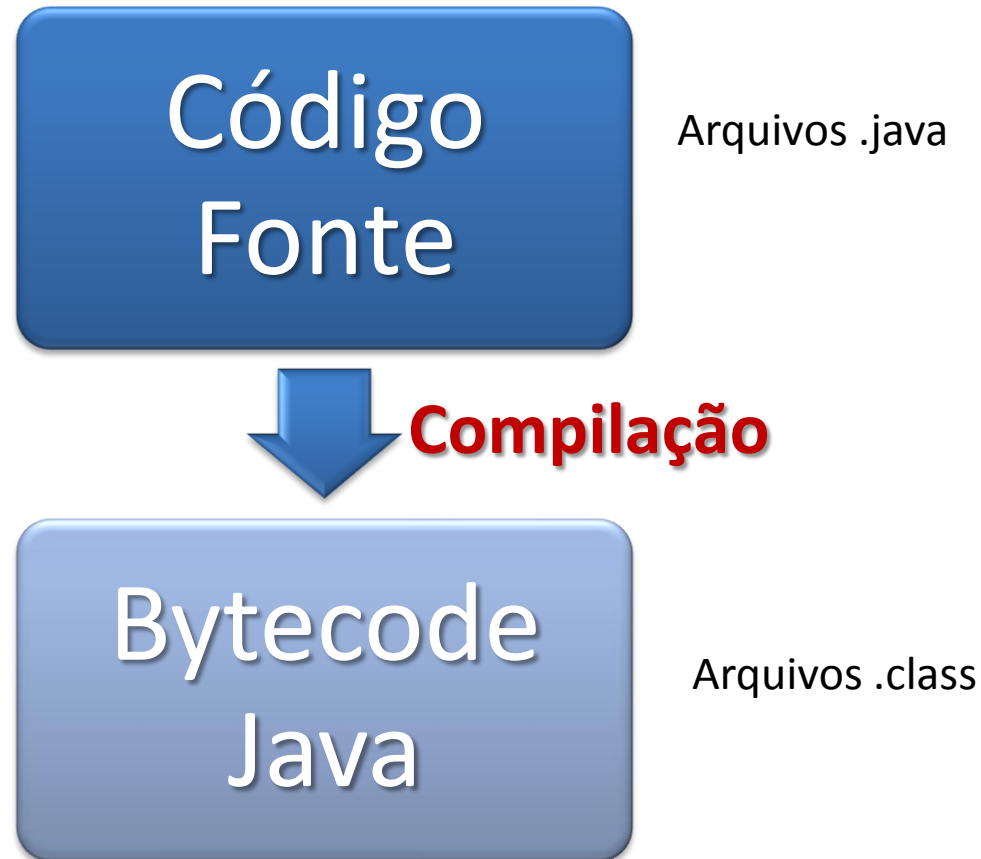
Linguagem de programação tradicional (C, C++, Pascal, etc.)



# Java – Conceitos e Definições

## Java – Máquina Virtual

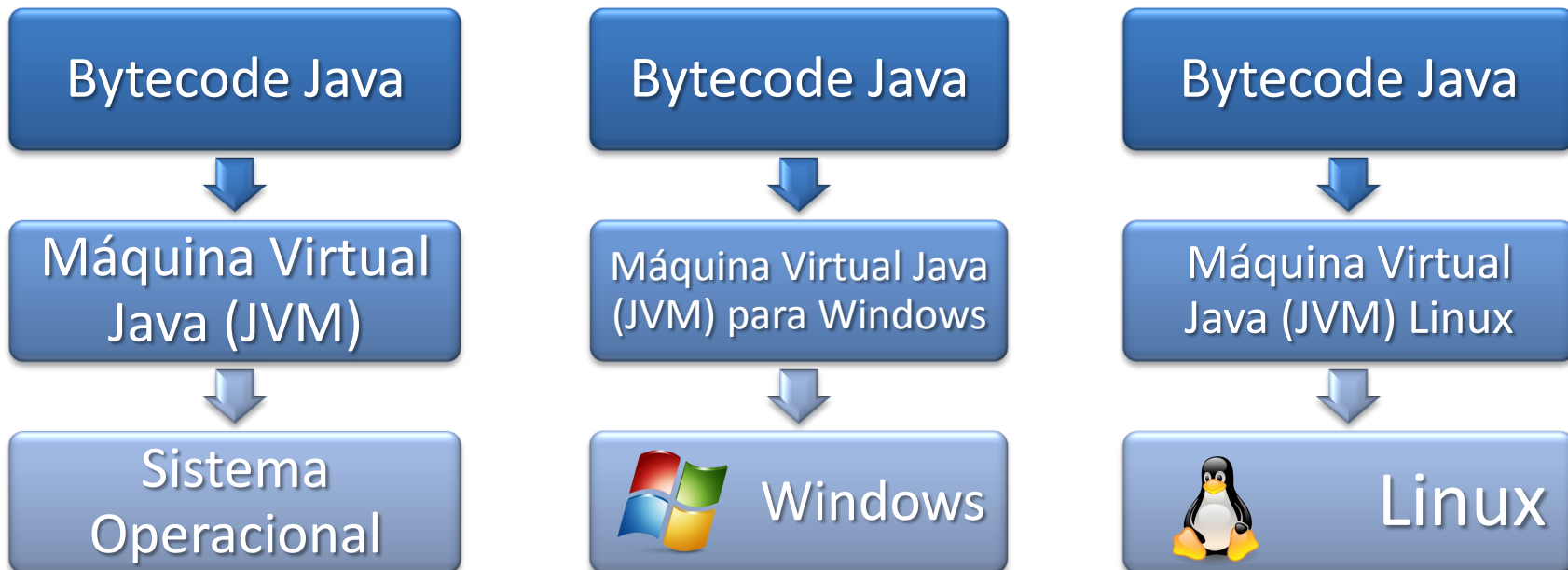
Linguagem de programação Java



# Java – Conceitos e Definições

## Java – Máquina Virtual

Linguagem de programação Java



## Java – Conceitos e Definições

### Java – Máquina Virtual

A Máquina Virtual Java (**JVM – Java Virtual Machine**) é um conceito bem mais amplo que o de um interpretador. Como o próprio nome diz, uma máquina virtual é como um “computador de mentira”: tem tudo que um computador tem. Em outras palavras, ela é responsável por gerenciar memória, threads, a pilha de execução, etc.

O desenvolvedor ganha independência de sistema operacional. Ou, melhor ainda, independência de plataforma em geral: não é preciso se preocupar em qual sistema operacional sua aplicação está rodando, nem em que tipo de máquina, configurações, etc.

A aplicação roda sem nenhum envolvimento com o sistema operacional!  
Sempre conversando apenas com a **JVM**.

# Java – Conceitos e Definições

## Java – Máquina Virtual

Essa característica é interessante: como tudo passa pela JVM, ela pode tirar métricas, decidir onde é melhor alocar a memória, entre outros. Uma JVM isola totalmente a aplicação do sistema operacional. Se uma JVM termina abruptamente, só as aplicações que estavam rodando nela irão terminar: isso não afetará outras JVMs que estejam rodando no mesmo computador, nem afetará o sistema operacional.

Essa camada de isolamento também é interessante quando pensamos em um servidor que não pode se sujeitar a rodar código que possa interferir na boa execução de outras aplicações. Essa camada, a máquina virtual, não entende código java, ela entende um código de máquina específico. Esse código de máquina é gerado por um compilador java, e é conhecido por “**bytecode**”, pois existem menos de 256 códigos de operação dessa linguagem, e cada “**opcode**” gasta um byte. O compilador Java gera esse **bytecode** que, diferente das linguagens sem máquina virtual, vai servir para diferentes sistemas operacionais, já que ele vai ser “traduzido” pela JVM.

## Java – Java Lento? Hotspot e JIT

Hotspot é a tecnologia que a JVM utiliza para detectar pontos quentes da sua aplicação: código que é executado muito, provavelmente dentro de um ou mais loops. Quando a JVM julgar necessário, ela vai **compilar** estes códigos para instruções realmente nativas da plataforma, tendo em vista que isso vai provavelmente melhorar a performance da sua aplicação. Esse compilador é o JIT: Just inTime Compiler, o compilador que aparece “bem na hora” que você precisa.

Você pode pensar então: porque a JVM não compila tudo antes de executar a aplicação? É que teoricamente compilar dinamicamente, a medida do necessário, pode gerar uma performance melhor. O motivo é simples: imagine um .exe gerado pelo VisualBasic, pelo gcc ou pelo Delphi; ele é estático. Ele já foi otimizado baseado em heurísticas, o compilador pode ter tomado uma decisão não tão boa.

Já a JVM, por estar compilando dinamicamente durante a execução, pode perceber que um determinado código não está com performance adequada e otimizar mais um pouco aquele trecho, ou ainda mudar a estratégia de otimização. É por esse motivo que as JVMs mais recentes (como a do Java 6) em alguns casos chegam a ganhar de códigos C compilados com o GCC 3.x.

## Java – Versões do Java e a confusão do Java 2

Java 1.0 e 1.1 são as versões muito antigas do Java, mas já traziam bibliotecas importantes como o JDBC e o java.io. Com o Java 1.2 houve um aumento grande no tamanho da API, e foi nesse momento em que trocaram a nomenclatura de Java para Java2, com o objetivo de diminuir a confusão que havia entre Java e Javascript. Mas lembre-se, não há versão “Java 2.0”, o 2 foi incorporado ao nome, tornando-se Java2 1.2.

Depois vieram o Java2 1.3 e 1.4, e o Java 1.5 passou a se chamar Java 5, tanto por uma questão de marketing e porque mudanças significativas na linguagem foram incluídas. É nesse momento que o “2” do nome Java desaparece. Repare que para fins de desenvolvimento, o Java 5 ainda é referido como Java 1.5. Até a versão 1.4, existia um terceiro número (1.3.1, 1.4.1, 1.4.2, etc), indicando bug fixes e melhorias. A partir do Java 5 existem apenas updates: Java 5 update 7, por exemplo.

## Java – Versões do Java e a confusão do Java 2

Hoje a última versão disponível do Java é a 7.0, lançada em 2011. Existe compatibilidade para trás em todas as versões do Java. Um class gerado pelo javac da versão 1.2 precisa necessariamente rodar na JVM 7.0. Por isso, recomendamos sempre usar a última versão do Java para usufruir das tecnologias mais modernas mas sem correr o risco de quebrar aplicações antigas.

## Java – JVM? JRE? JDK? O que devo baixar?

O que gostaríamos de baixar no site da Oracle?

**JVM** = apenas a virtual machine, esse download não existe, ela sempre vem acompanhada.

**JRE** = *Java Runtime Environment*, ambiente de execução Java, formado pela JVM e bibliotecas, tudo que você precisa para executar uma aplicação Java. Mas nós precisamos de mais.

**JDK** = *Java Development Kit*: Nós desenvolvedores, faremos o download do JDK do Java SE (Standard Edition). Ele é formado pela JRE somado a ferramentas, como o compilador.

Tanto o JRE e o JDK podem ser baixados do site

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Lá você encontra instruções de instalação.

## Java – Onde usar e os objetivos do Java

Ao desenvolver em Java, você poderá achar que tecnologia a tem menor produtividade quando comparada com a linguagem que você está acostumado. É preciso ficar claro que a premissa do Java não é a de criar sistemas pequenos, onde temos um ou dois desenvolvedores, mais rapidamente que linguagens como *php*, *perl*, e outras.

O foco da plataforma é outro: aplicações de médio a grande porte, onde o time de desenvolvedores tem várias pessoas e sempre pode vir a mudar e crescer. Não tenha dúvidas que criar a primeira versão de uma aplicação usando Java, mesmo utilizando IDEs e ferramentas poderosas, será mais trabalhoso que muitas linguagens script ou de alta produtividade. Porém, com uma linguagem orientada a objetos e madura como o Java, será extremamente mais fácil e rápido fazer alterações no sistema, desde que você siga as boas práticas e recomendações sobre design orientado a objetos.

## Java – Onde usar e os objetivos do Java

O Java possui uma quantidade enorme de bibliotecas gratuitas para realizar os mais diversos trabalhos (tais como relatórios, gráficos, sistemas de busca, geração de código de barra, manipulação de XML, tocadores de vídeo, manipuladores de texto, persistência transparente, impressão, etc.) e isso é um ponto fortíssimo para adoção da plataforma: você pode criar uma aplicação sofisticada, usando diversos recursos, sem precisar comprar um componente específico, que costuma ser caro. O ecossistema do Java é enorme.

Cada linguagem tem seu espaço e seu melhor uso. O uso do Java é interessante em aplicações que virão a crescer, em que a legibilidade do código é importante, onde temos muita conectividade e se há muitas plataformas (ambientes e sistemas operacionais) heterogêneas (Linux, Unix, OSX e Windows misturados). Você pode ver isso pela quantidade enorme de ofertas de emprego procurando desenvolvedores Java para trabalhar com sistemas web e aplicações de integração no servidor.

## Java – Preparando o Ambiente de Trabalho

Para estudarmos o Java utilizaremos a última versão da JDK e a IDE NetBeans. Ambos estão disponíveis respectivamente para download nos links abaixo, onde também são encontradas as instruções de instalação:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>  
[www.netbeans.org/](http://www.netbeans.org/)



Primeiro deve ser instalado o JDK e posteriormente o Netbeans.

# Java – Conceitos e Definições

## Java – Seu primeiro programa

Nosso primeiro programa em Java será no tradicional estilo “Hello World”:

```
class MeuPrograma {  
    public static void main(String[] args) {  
        System.out.println("Minha primeira aplicação Java!");  
    }  
}
```

# Java – Conceitos e Definições

## Java – O que aconteceu?

O miolo do programa é o que será executado quando chamamos a máquina virtual. Por enquanto, todas as linhas anteriores, onde há a declaração de uma classe e a de um método, não importam para nós nesse momento. Mas devemos saber que toda aplicação Java começa por um ponto de entrada, e este ponto de entrada é o método **main**.

Ainda não sabemos o que é método, mas veremos adiante. Até lá, não se preocupe com essas declarações. Sempre que um exercício for feito, o código que nos importa sempre estará nesse miolo. No caso do nosso código, a linha do **System.out.println** faz com que o conteúdo entre aspas seja colocado na tela.

## Java – Exercícios: Modificando o Programa

- 1) Altere seu programa para imprimir uma mensagem diferente.
- 2) Altere seu programa para imprimir duas linhas de texto usando duas linhas de código System.out.
- 3) Sabendo que os caracteres \n representam uma quebra de linhas, imprima duas linhas de texto usando uma única linha de código System.out.

## Java – O que pode dar errado?

Vamos mostrar alguns erros mais comuns cometidos no início da programação em Java:

- Esquecer o ponto e vírgula “;” no final das linhas
- Palavras chaves (class, static, etc.) em maiúsculo
- Esquecer de abrir/fechar {}
- Esquecer de colocar `static` ou o argumento `String[] args` no método main
- Não colocar o método main como `public`

## Java – Conceitos e Definições

### Java – Exercícios Adicionais

Um arquivo fonte Java deve sempre ter a extensão `.java`, ou o compilador o rejeitará. Além disso, existem algumas outras regras na hora de dar o nome de um arquivo Java. Experimente gravar o código deste capítulo com `OutroNome.java` ou algo similar.

Compile e verifique o nome do arquivo gerado. Qual o resultado?

## Dados de Contato



79 9949 4098



pedro@pyxistec.com.br



psneto@emsergipe.com



pedro.pyxistec@gmail.com



<http://www.facebook.com/pedro.neto.se>



pedropyxis



<http://lattes.cnpq.br/4891420246888248>



<http://pedroneto.yolasite.com/>