

Programação Orientada a Objeto

Aula 4 – Introdução a Linguagem de Programação Java

Prof. Pedro Baesse
pedro.baesse@ifrn.edu.br

Roteiro

- ▶ História do Java
- ▶ Motivos para usar Java
- ▶ Características da Linguagem



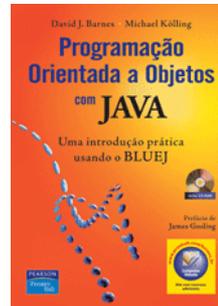
Linguagem de Programação Orientada a Objetos – LPOO



Bibliografia

▶ **Programação Orientada a Objetos com Java: Uma introdução prática usando BlueJ**

- David J. Barnes e Michael Kölling
- Editora Pearson / Prentice Hall



▶ **Java Use a Cabeça**

- Kathy Sierra e Bert Bates
- Editora Alta Books

▶ **Java Como Programar**

- Deitel
- Editora Pearson / Prentice Hall



▶ **Simplesmente Java**

- Michael Wenstrom
- Editora Alta Books

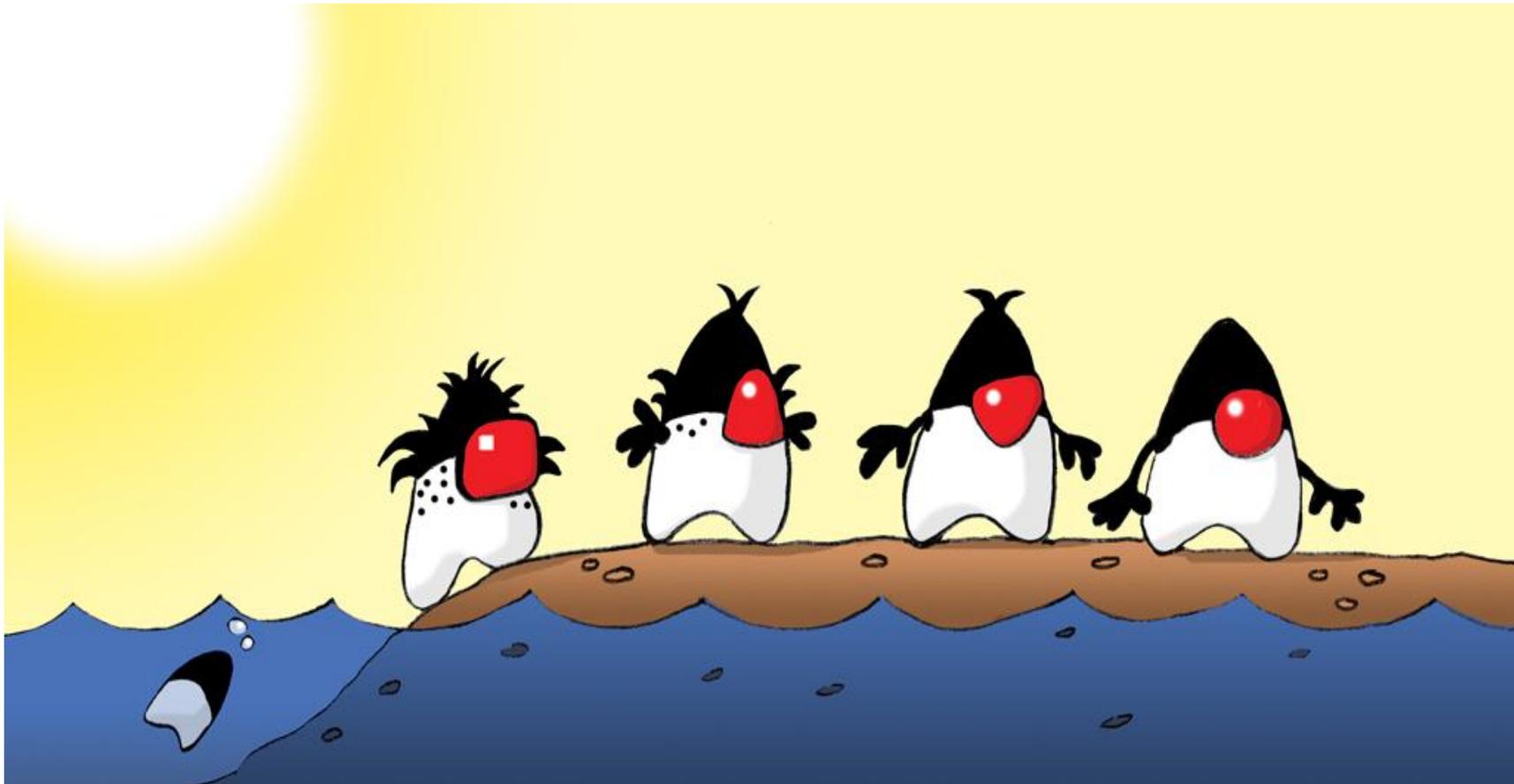


▶ **Big Java**

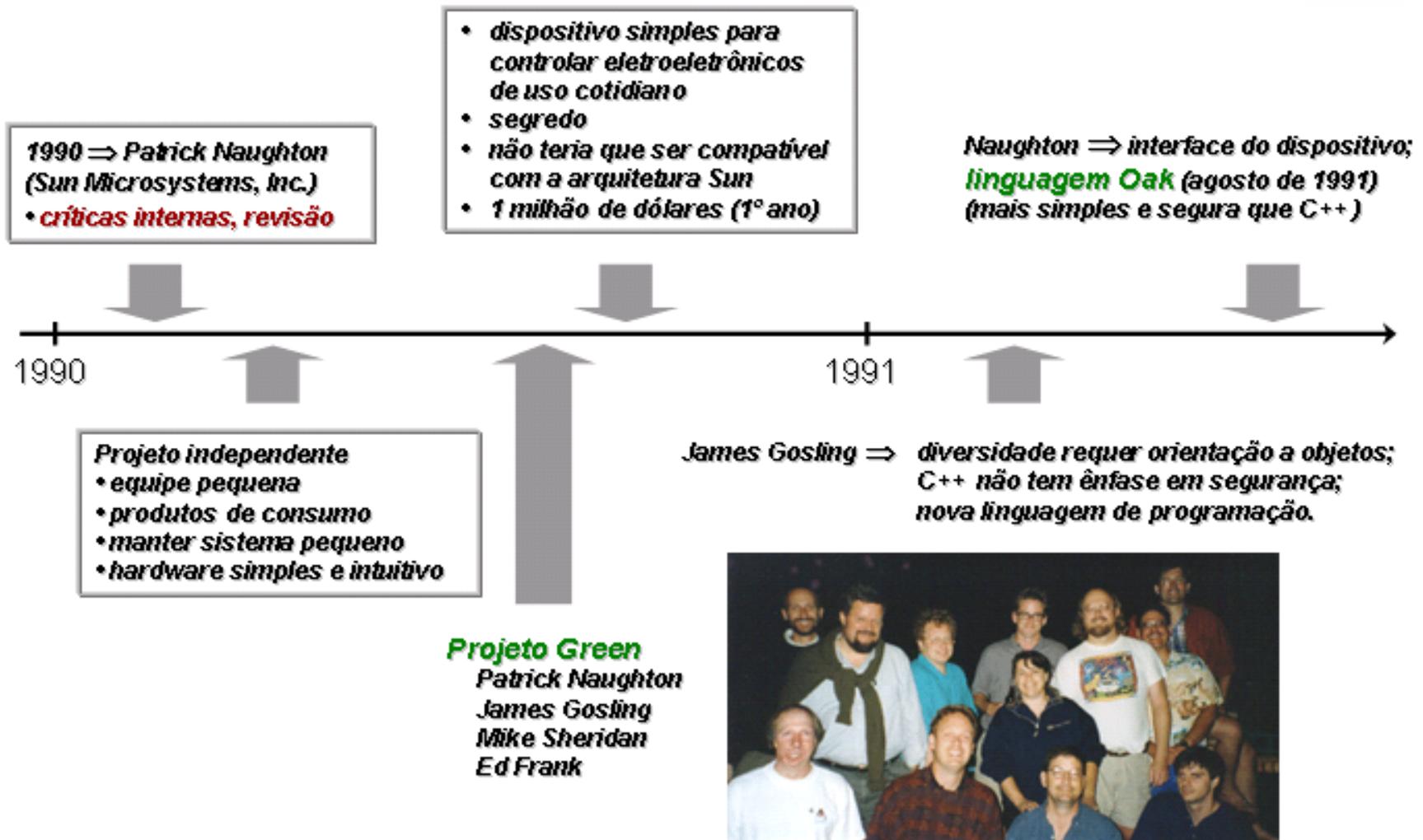
- Cay Horstmann
- Editora Bookman



Evolução



História do Java



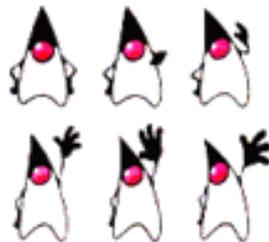
História do Java

- ▶ 1991 – Início do Projeto Green
 - Linguagem Oak criada por James Gosling, Mike Sheridan e Patrick Naughton.
 - Inspirada na UCSD Pascal de Niklaus Wirth, porém, baseada em C++ (Unix)



História do Java

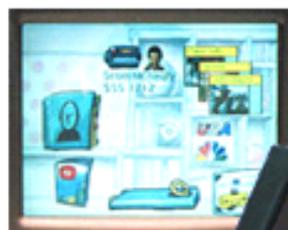
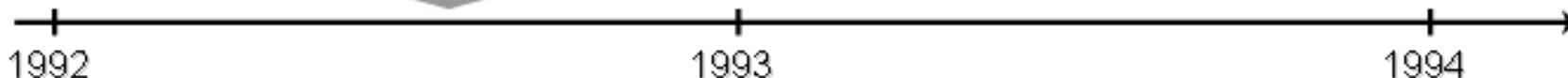
3 de Setembro de 1992,
demonstração do **Star7**
para Scott McNealy
(handheld sem teclado e
sem botões baseado na
plataforma SPARC)



SUN transformou
equipe Green em
FirstPerson, Inc.,
uma subsidiária
gerenciada por
Wayne Rosing.



James Gosling



Star7

LCD colorido de 5" touchscreen
rede sem fio de 900MHz, 200Kbps, PCMCIA,
4MB RAM, codecs de áudio, projeto industrial
versão de Unix usando um megabyte,
flash RAM, drivers

**nova linguagem de programação: pequena,
segura, distribuída, robusta, interpretada, com
coleta automática de lixo, multitarefa, neutra
com relação à arquitetura, com desempenho
elevado, e dinâmica.**

guia e controle remoto de TV
quadro de mensagens pela rede wireless,
agente on-screen

© firstperson, inc.

Oak Language Specification

This document is a preliminary specification of the Oak language. Both the
specification and the language are subject to change. Where a feature has been
added to the specification but not yet implemented in the hardware, the feature should
be considered to work as if done in AWT.

1 Program Structure

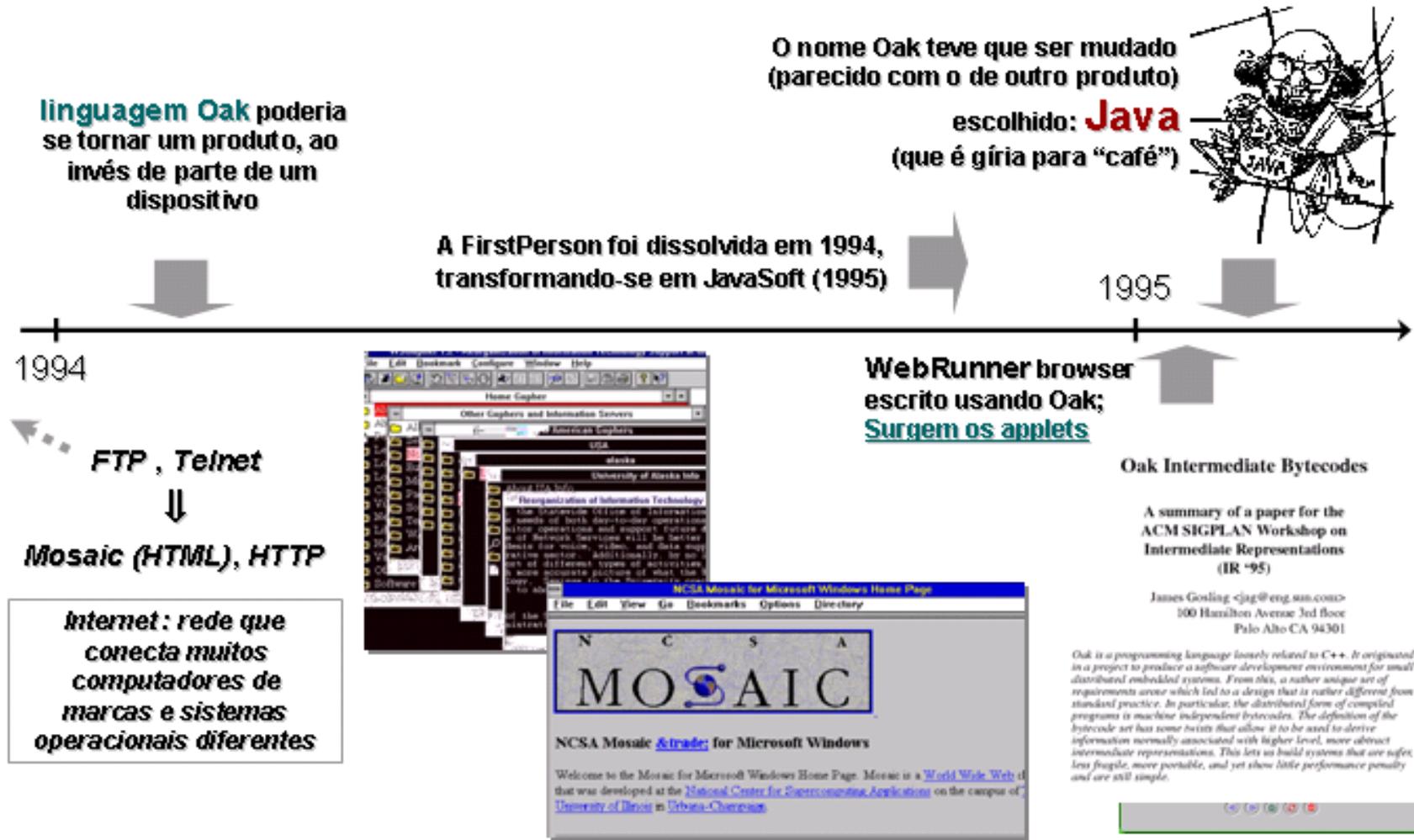
The source code for each program consists of one or more compilation units.
Each compilation unit can contain one or the following (in addition to other source
and comments):

- a package declaration (see "Package" on page 10)
- a class declaration (see "Class" on page 10)
- a module declaration (see "Module" on page 10)

Although each Oak compilation unit can contain multiple classes or modules, it
must contain exactly one package declaration and exactly one "Class" or
"Module" declaration.



História do Java



História do Java

▶ 1994 - HotJava

- Gosling desenvolve o navegador Webrunner depois chamado de HotJava para demonstrar o uso de Java no desenvolvimento de aplicativos para redes
- O compilador Java, originalmente escrito em C, é reescrito em Java



História do Java



Primeira linguagem de programação não associada a um sistema operacional particular e/ou processador.



timeline

1995

- 23 de Maio de 1995 anúncio oficial do Java™
- Seguir o modelo de negócio da Netscape: Produto livre para uso.
- Comercializar parte (licenciamento cód. fonte).
- Acesso ao código fonte do Java liberado.
- Netscape dá suporte para Java (Set95).

Borland, Macromedia, Mitsubishi, Netscape, Oracle, Silicon Graphics, Spyglass, Toshiba...

...Microsoft (dez/1995)

NETSCAPE & SUN ANNOUNCE JAVASCRIPT™
THE OPEN, CROSS-PLATFORM OBJECT
SCRIPTING LANGUAGE FOR ENTERPRISE
NETWORKS & THE INTERNET

28 Industry-Leading Companies Have
Expressed Their Endorsement of ...

December 4, 1995, Mountain View, Calif. - ...



História do Java

- ▶ 1995 – Início da Era Java
 - Lançamento oficial da Linguagem Java, kit JDK 1.0 alpha
 - Netscape com suporte à Java



História do Java

- ▶ 1996 – Lançamento do kit JDK 1.0
 - JVM para MacOS
 - Primeiro JavaOne



História do Java

- ▶ 1997 – Lançamento da versão 1.1
 - Conectividade de bancos de dados, servlets e objetos distribuídos
 - Novo modelo de eventos, internacionalização, modelo de componentes (JavaBeans)
 - JDK 1.1.3 para Linux
 - JavaCard



História do Java

- ▶ 1998 – Lançamento do Java2
 - JDK 1.2 – JFC Swing e segurança
 - Formalização do JCP – Java Community Process
 - API Java 3D



História do Java

- ▶ 1999 – Edições da plataforma: J2SE, J2EE e J2ME
 - Lançamento de JSP – Java Server Pages
 - Apache Jakarta
 - Tecnologia JavaCard adotada como padrão mundial



Edições do Java

- ▶ Java Standard Edition (JavaSE)
 - API's padrão, base da linguagem Java. Aplicações Desktop.
- ▶ Java Micro Edition (JavaME)
 - API's para aplicações em micro-dispositivos (pda, celulares, pagers,...)
- ▶ Java Enterprise Edition (JavaEE)
 - Destinada à aplicações corporativas distribuídas e de arquitetura multicamadas.



História do Java

- ▶ 2000 – J2SE 1.3
 - 400 grupos de usuários no mundo
 - JavaOne com mais de 25.000 desenvolvedores
 - NetBeans Open-Source



História do Java

- ▶ 2000 – J2SE 1.3
 - 400 grupos de usuários no mundo
 - JavaOne com mais de 25.000 desenvolvedores
 - NetBeans Open-Source



História do Java

- ▶ 2001 – IDE Eclipse
 - J2ME para Linux
 - Catálogo J2EE Patterns
 - Lançamento do SDK J2EE 1.3



História do Java

- ▶ 2002 – Lançamento do JDK J2SE 1.4
 - Web Services Developers Pack 1.0
 - Java 3D API 1.3
 - MIDP 2.0
 - API Java Server Faces
 - Mais de 50 milhões de celulares J2ME



História do Java

- ▶ 2003 – java.net
 - 500 grupos de usuários Java no mundo.
 - Anúncio do protótipo do robô de exploração enviado à Marte.
 - J2ME Mobile 3D Graphics API.
 - J2EE 1.4



História do Java

- ▶ 2003 – java.net
 - 500 grupos de usuários Java no mundo.
 - Anúncio do protótipo do robô de exploração enviado à Marte.
 - J2ME Mobile 3D Graphics API.
 - J2EE 1.4



História do Java

- ▶ 2004 – J2SE 1.5 (Tiger)
 - Mais de 250 milhões de telefones móveis.
 - Projeto Looking Glass



Alguns motivos para utilizar Java

- ▶ Em 2003 a linguagem JAVA foi eleita pelo 4º ano consecutivo a linguagem do ano pela revista Info Exame
- ▶ A tecnologia Java está em 100% das empresas Fortune 500
- ▶ 78% dos executivos vêem a linguagem Java como a melhor plataforma para web services (fonte: Giga, Computerworld);
- ▶ O Java está instalada e rodando em quase 500 milhões de computadores desktop pelo mundo;



Alguns motivos para utilizar Java

- ▶ Mais de 300 milhões de smart cards seguros baseados em ambiente Java já foram distribuídos;
- ▶ Quase 1 bilhão de SIM cards para telefonia celular.
- ▶ Quase 74% de profissionais que usam Java como sua principal linguagem de desenvolvimento, ultrapassando os 51% que usam Visual Basic (segundo fonte: IDC).



Características da linguagem Java

▶ Portabilidade

- O código binário das aplicações Java (byte codes) podem ser executados em várias plataformas sem nenhuma alteração

▶ Uso Geral

- Aplicações que vão desde smart-cards até mainframes

▶ Linguagem Distribuída (RMI, EJB)

- Java possui API's para a criação de objetos distribuídos onde partes de uma única aplicação podem estar sendo executada em várias máquinas transparentemente.

▶ Segurança (SandBox – Caixa de Areia)

- Controle de segurança do ambiente de execução das aplicações



Características da linguagem Java

- ▶ **Multithreading**
 - Suporta a programação de aplicações multitarefa compostas por várias linhas de execução (threads)
- ▶ **Linguagem Dinâmica (JCP)**
 - O Java está em constante evolução, suas extensões e sua evolução não são controladas por uma única empresa, ou por seu criador, mas sim pela comunidade
- ▶ **Java e a Internet (Applets, Servlets / JSP)**
 - Total suporte para criação de aplicações em rede especialmente para Web.
- ▶ **Orientação a Objetos**
 - Segue o paradigma de orientação a objetos.



Ambiente de desenvolvimento

- ▶ Ambiente Open Source (Código Aberto)

- ▶ Sistema Operacional

- Linux



- ▶ Servidores Web

- Apache (<http://apache.org>)
- Tomcat (<http://jakarta.apache.org/tomcat>)
- Jboss (www.jboss.org)



- ▶ Ferramentas de Desenvolvimento (IDE)

- Eclipse (www.eclipse.org)
- NetBeans (www.netbeans.org)
- BlueJ (www.bluej.org)



netBeans.ORG



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Ambiente de desenvolvimento

▶ Gerenciadores de Bancos de Dados



▶ MySQL

- www.mysql.com

▶ Postgrees

- www.postgresql.org

PostgreSQL



▶ HSQLDB

- www.hsqldb.org



▶ Firebird

- www.firebirdsql.org



Java é gratuito e código-aberto

A maioria dos produtos *java.sun.com*, assim como os *API's* estão livres para download e uso comercial, incluindo:

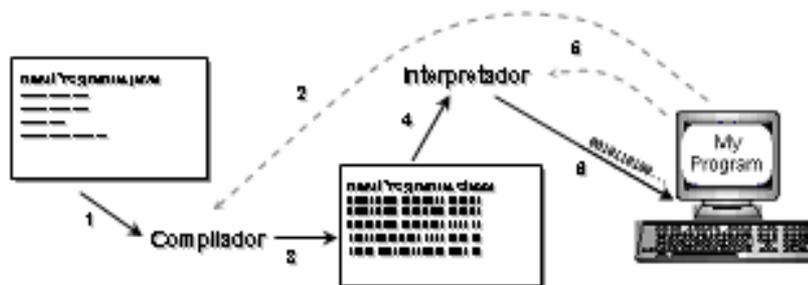
- **Plataforma Java 2, J2SE SDK** (kit padrão de desenvolvimento)
- **Ambiente de Execução Java (JRE)**
- **Plataforma Java 2, Edição Empresa, J2EE SDK**
- **Plataforma Java 2, Micro Edição, (J2ME)**

O J2SE SDK e o JRE podem também ser livremente redistribuídos, contanto que seja distribuído com uma aplicação ou um applet que adicione valor ou com um livro ou revista publicada.

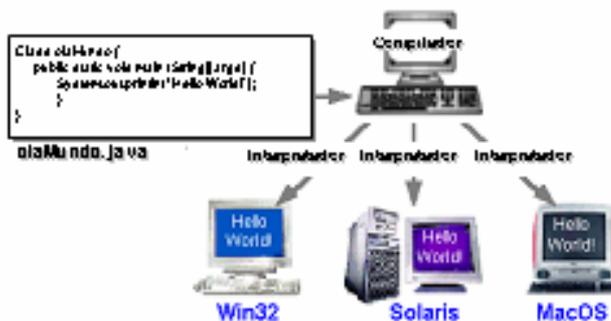
Pré-releases de softwares não estão livres para redistribuição.



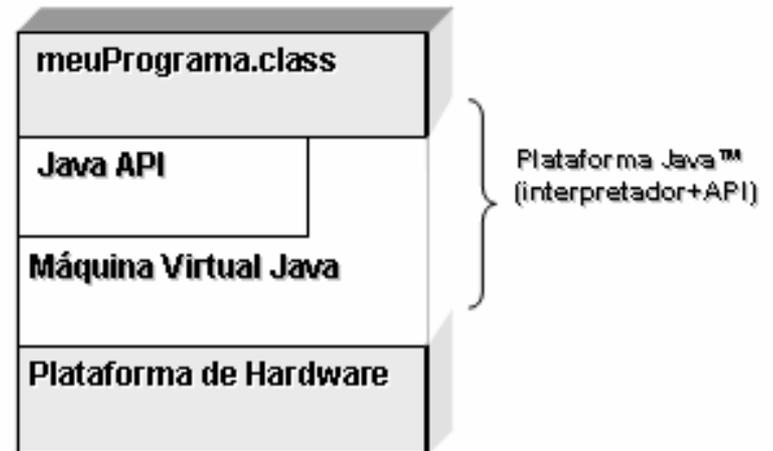
Java: Independência de arquitetura



Etapas para execução de um programa em Java



“Escreva uma vez, rode em qualquer lugar”

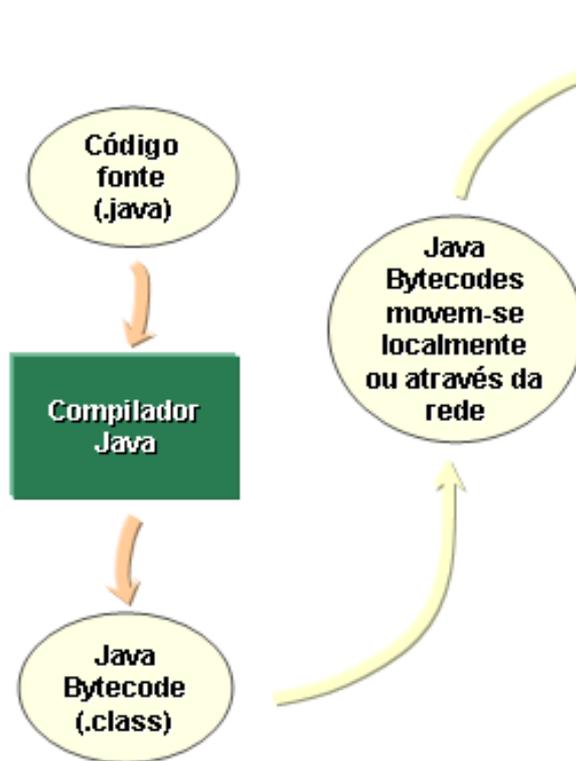


A plataforma Java™

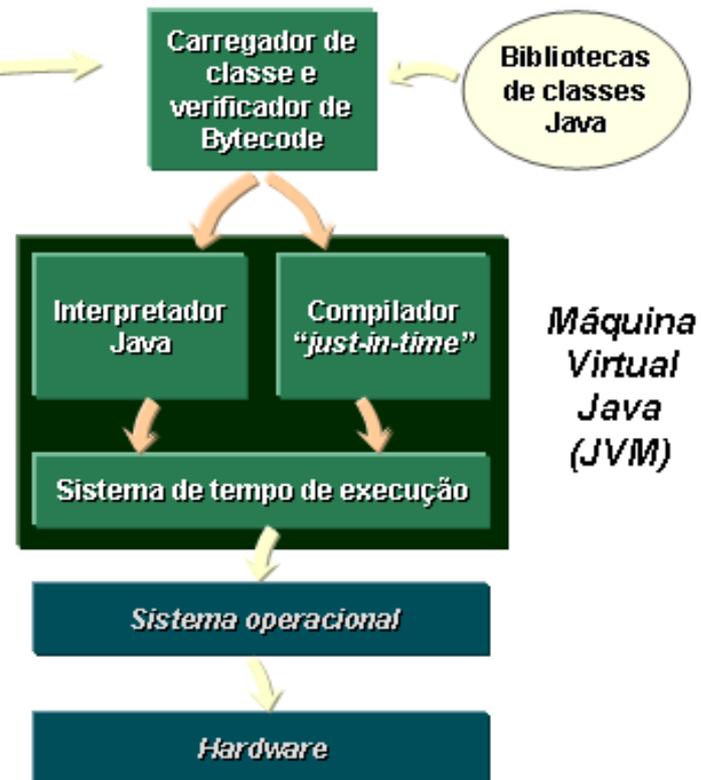


Java: Máquina virtual

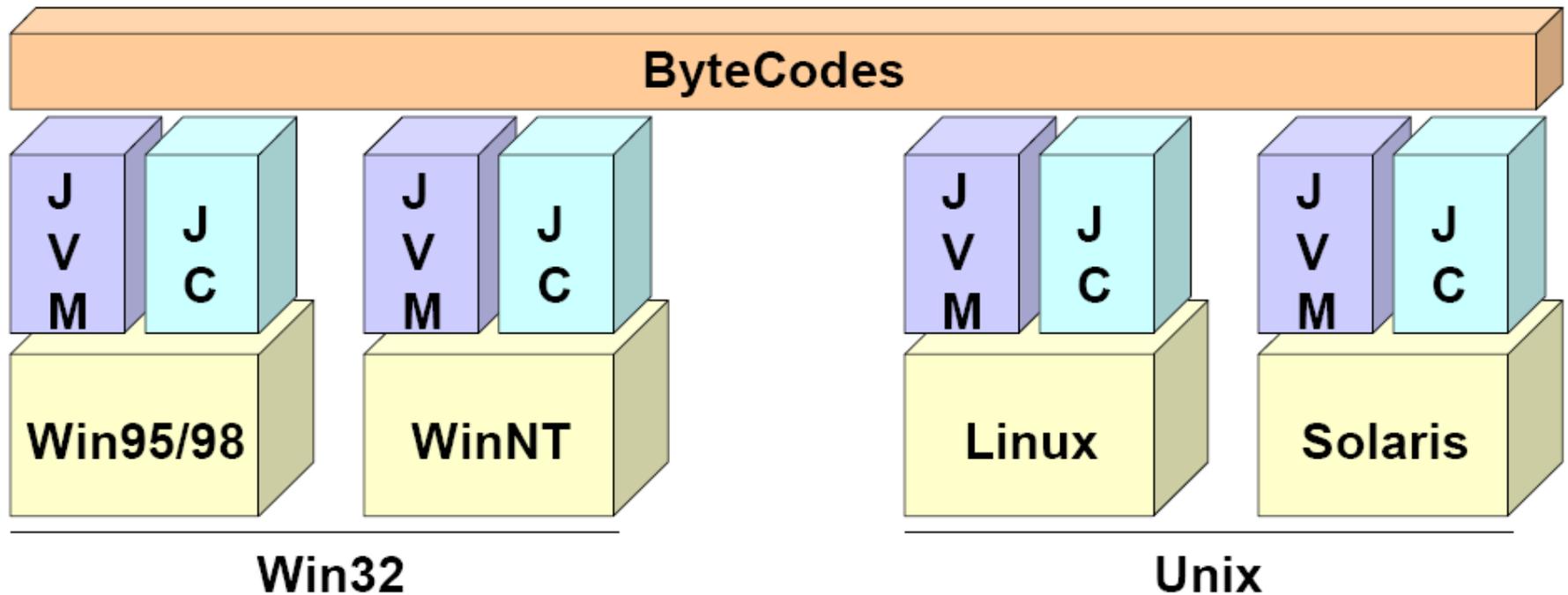
Tempo de Compilação:



Tempo de Execução:



Ambiente Java e os Bytecodes



Ambiente Java e os Bytecodes

- ▶ O código é compilado, gerando um conjunto de instruções chamado de byte-code.
- ▶ O byte-code é aplicado à Máquina Virtual Java (JVM) que se encarrega de interpretar os comandos para o SO onde o programa está rodando.
- ▶ Ou seja, a máquina virtual traduz as instruções do código Java para instruções válidas no SO em que está rodando.
- ▶ Esse byte-code poderá então ser interpretado por qualquer máquina virtual Java (Linux, Windows, Palm OS, Solari, etc).



Ambiente Java e os Bytecodes

- ▶ O resultado gerado pelo ambiente de desenvolvimento Java, embora interpretado inicialmente, torna-se independente de plataforma.
 - Simplifica o projeto de aplicações de rede ou aplicações distribuídas que tenham que operar em ambientes heterogêneos
 - Também permitir a incorporação de vários mecanismos de segurança na aplicação gerada



Ambiente Java e o Garbage collection

- ▶ O Garbage-collection é responsável pela alocação e liberação de memória
- ▶ Quando uma variável é declarada a Java Virtual Machine (JVM) cria um ponteiro para uma área de memória equivalente ao tamanho do tipo de dado utilizado
- ▶ Se a variável é associada a outra região de memória, a JVM coloca o espaço alocado anteriormente em uma pilha de objetos em desuso naquele momento



Ambiente Java e o Garbage collection

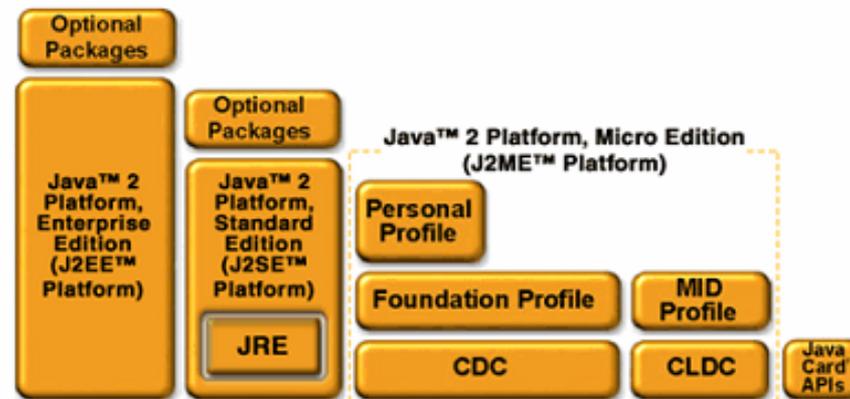
- ▶ Se o computador ficar com pouca memória disponível, a JVM remove os objetos dessa pilha, realocando assim este espaço de memória para outra variável
- ▶ O processo de garbage-collection ocorre automaticamente durante a execução de um programa Java
- ▶ O programador não precisa se preocupar com aritmética de ponteiros (grande dificuldade em linguagens como C e Pascal)



A plataforma Java

Plataforma Java é a combinação de dois componentes: a Máquina Virtual Java (JVM) e a Interface de Programação de Aplicativos Java (Java API).

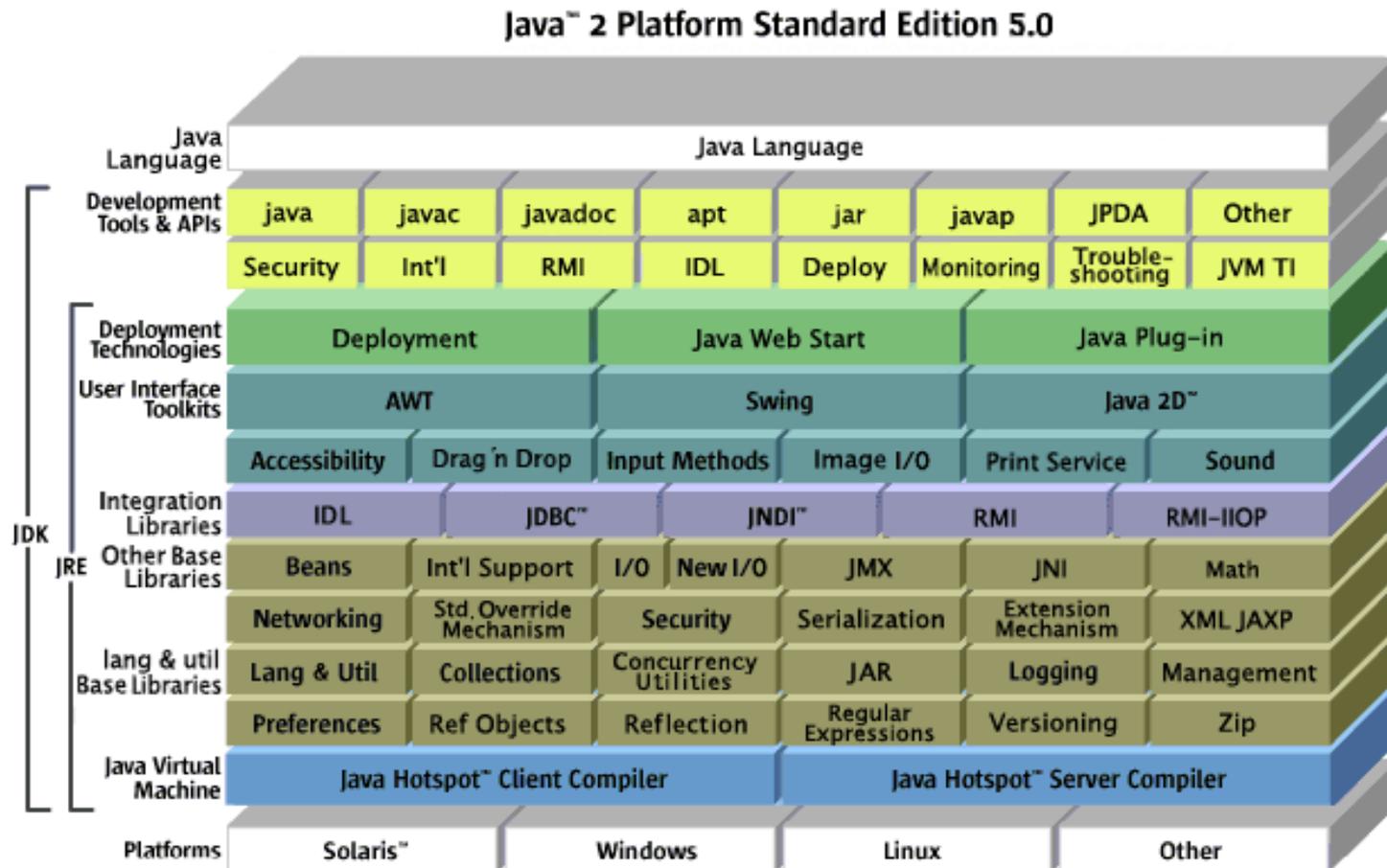
A Java API é uma grande coleção de componentes de software prontos, que fornecem muitas potencialidades úteis, tais como itens de interface gráfica de usuário (GUI), por exemplo. A Java API é agrupada em bibliotecas de classes e interfaces relacionadas, sendo que estas bibliotecas são conhecidas como pacotes.



As edições da Plataforma JAVA



A plataforma Java Standard Edition



Java: Produtos



Java Mobile: Produtos



Java: Produtos

Outros: Enterprise Java Beans, JINI, J2ME, Java Server Faces, etc...



Uma única plataforma – de cartões inteligentes a supercomputadores



Bluej



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

BlueJ

- ▶ **BlueJ** é um ambiente de desenvolvimento **Java**
 - Focado no nível introdutório
 - Criado pela Deakin University, Melbourne, Austrália, e a University of Kent at Canterbury, UK
 - Implementado em **Java**
 - Testado em Solaris, Linux, Macintosh e várias versões do Windows



BlueJ – Objetivos

- ▶ Proporcionar um ambiente fácil de usar para o aprendizado da linguagem Java
- ▶ Ênfase especial para as técnicas de visualização e interação
 - Ambiente altamente interativo
 - Incentiva a experimentação e exploração



BlueJ – Objetivos

- ▶ Proporcionar um ambiente fácil de usar para o aprendizado da linguagem Java
- ▶ Ênfase especial para as técnicas de visualização e interação
 - Ambiente altamente interativo
 - Incentiva a experimentação e exploração



BlueJ – Funcionalidades

- ▶ Ambiente totalmente integrado
- ▶ Mostrar a estrutura de classe gráfica
- ▶ Edição gráfica e textual
- ▶ Built-in editor, compilador, máquina virtual, depurador...



BlueJ – Funcionalidades

- ▶ Interface fácil de usar
- ▶ Criação do objeto interativo
- ▶ Chamadas de objetos interativas
- ▶ Teste interativo
- ▶ Desenvolvimento de aplicações incremental



BlueJ – Requisitos

- ▶ **Kit de Desenvolvimento Java (Java Development Kit – JDK), versão 6 ou superior**
 - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- ▶ **BlueJ**
 - <http://www.bluej.org/download/download.html>



Dúvidas



Referência

- ▶ Material didático do Prof. MSc. Vladimir Camelo

- ▶ Material sobre BlueJ – mabesi.com

