

# Linguagem de Programação Orientada a Objetos I

Introdução a Linguagem Java  
Prof. Tales Bitelo Viegas

<https://facebook.com/ProfessorTalesViegas>

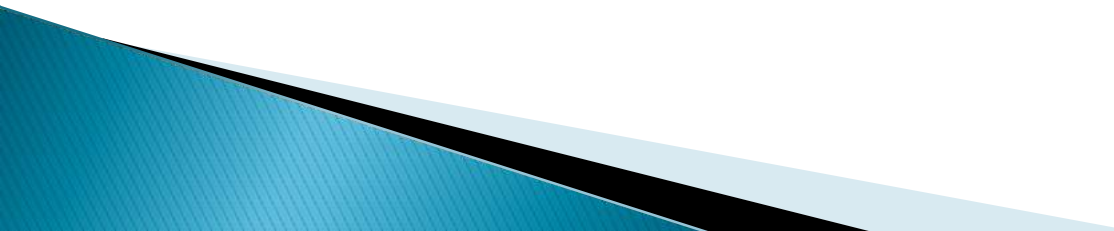
# Introdução à Linguagem Java

- ▶ Desenvolvimento
  - Sun Microsystems
  - Inspiração em C++
- ▶ Primeira implementação: 1995

# Vantagens

- ▶ Algumas vantagens da tecnologia Java
  - Linguagem OO (orientada a objeto)
  - Reuso de código
  - Portabilidade
  - Uso em diversas aplicações
  - Desenvolvimento rápido de código
  - Redução do tempo de projeto
  - Redução do custo do projeto

# Principais Objetivos

- ▶ Tornar possível a troca de programas executáveis Java entre computadores na Internet. O receptor deste programa não deveria realizar modificação alguma
  - ▶ Exemplo: programa Java compilado em uma SPARC/Solaris poderia ser enviado pela Internet para um Pentium/Windows que o executaria sem modificação alguma
- 

# Como executar uma linguagem ?

- ▶ Quais são as formas para executar uma linguagem em uma determinada máquina ?
  - Tradução
  - Interpretação
  - Máquina Virtual

# Tradução

- ▶ Método de trocar todas as instruções de L1 por uma sequência equivalente de instruções em L0
- ▶ Resultando em um programa consistente completamente em L0

# Interpretação

- ▶ Escrever um programa em L0 que examina cada instrução de L1 (programa+dados) e executa a sequência equivalente de instruções diretamente
- ▶ Esta técnica não requer primeiramente a geração de um novo programa em L0

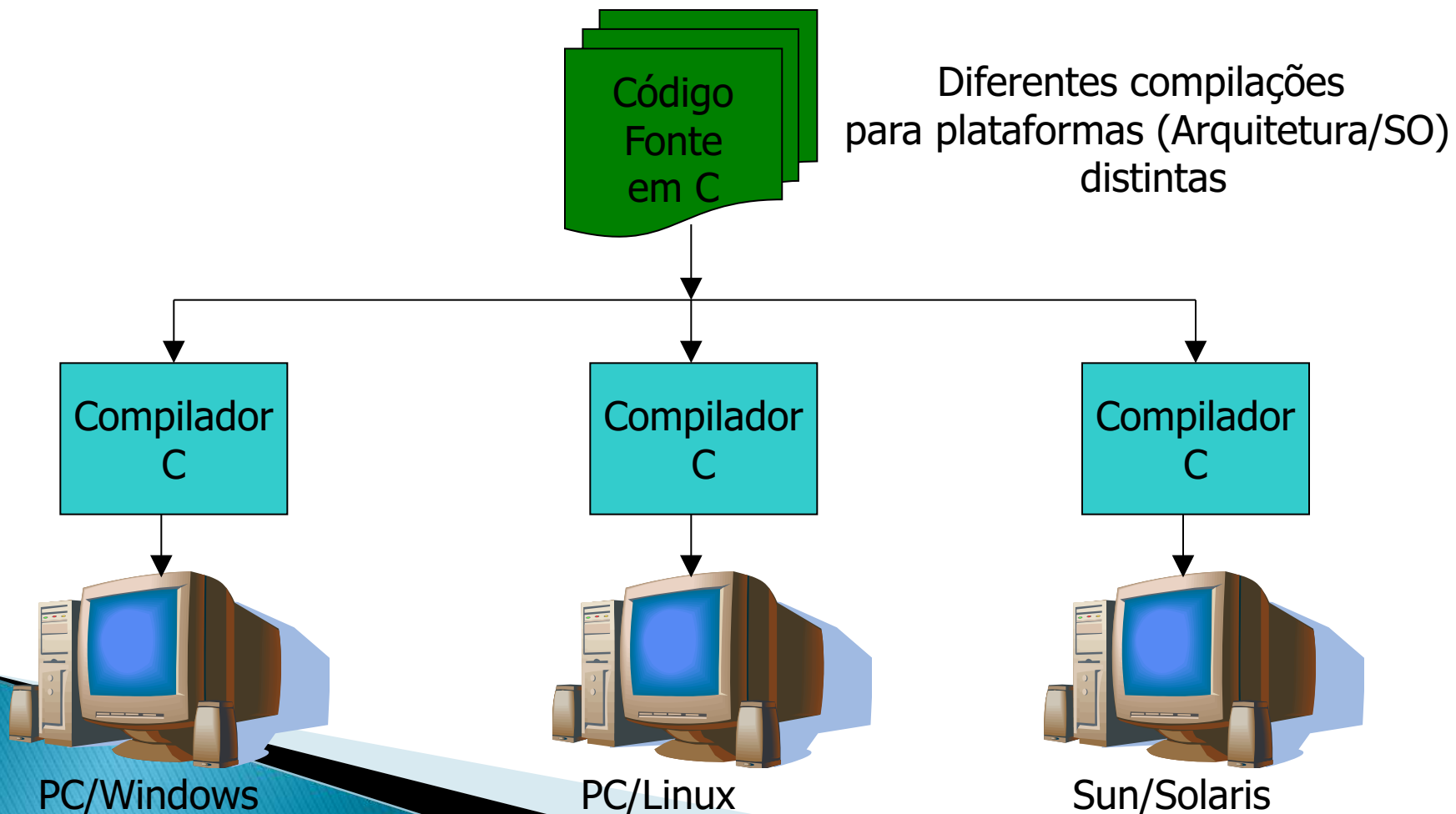
# Máquinas Virtuais

- ▶ Por outro lado, ao invés de pensar em tradução ou interpretação é imaginar a existência de uma máquina hipotética (computador) ou máquina virtual, onde a linguagem de máquina é L1
- ▶ Podemos chamar esta máquina virtual de M1, assim podemos construir (escrever) programas em L1 e possuir o computador que executa diretamente esta linguagem



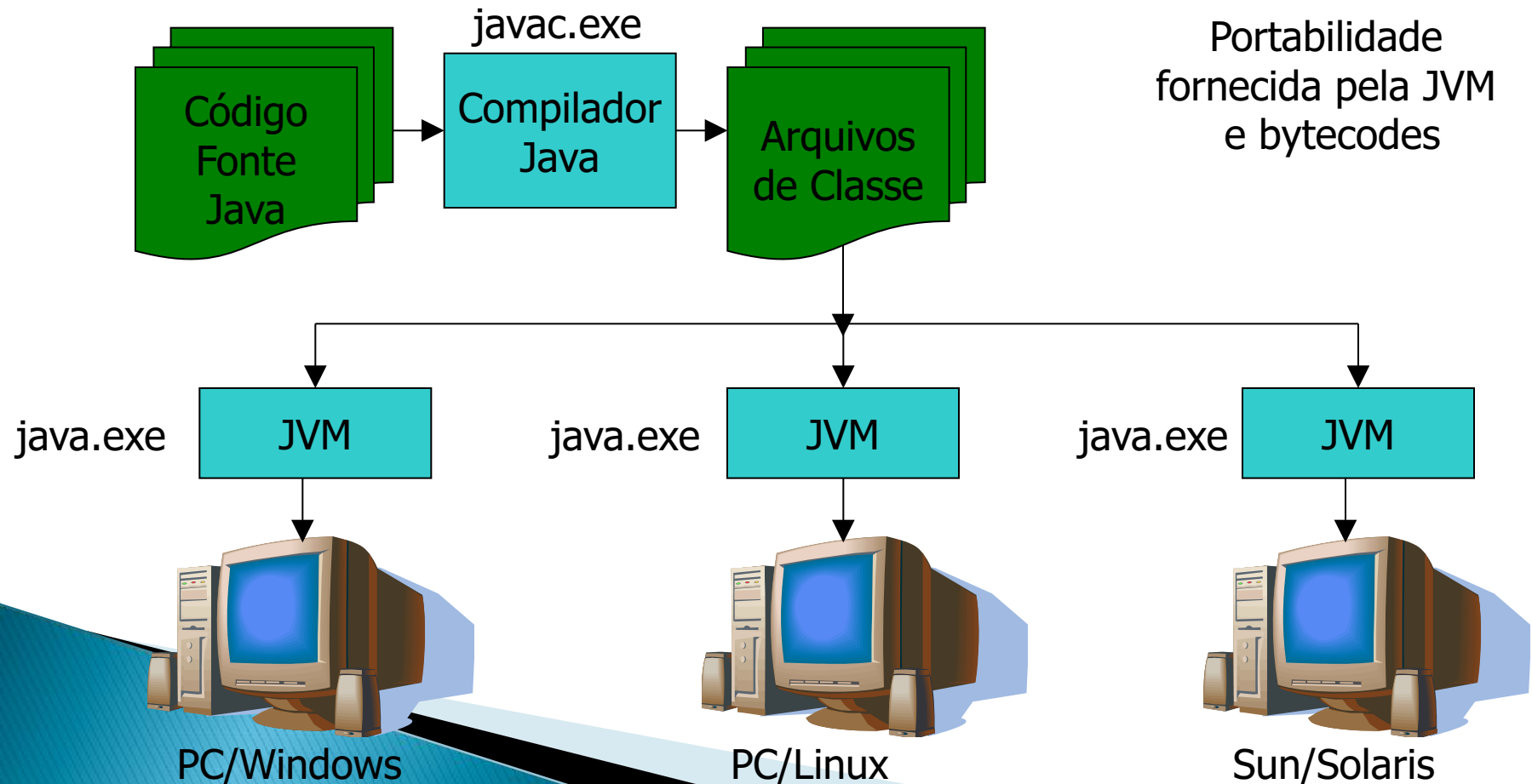
# Distribuição de Programas

## ► Desenvolvimento com Tradução (exemplo C)

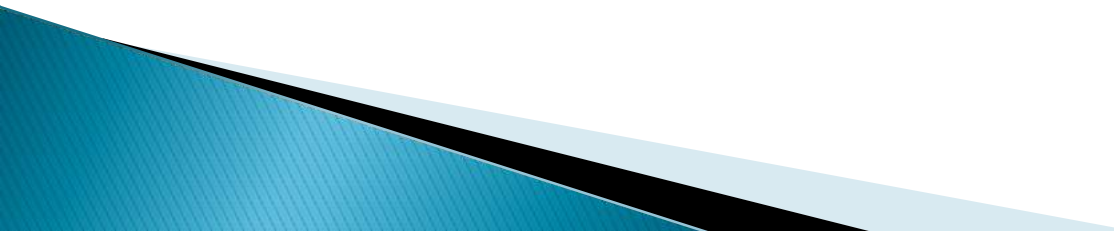


# Distribuição de Programas

## ► Desenvolvimento Java



# JVM – Java Virtual Machine

- ▶ MVJ – Máquina Virtual Java
  - ▶ Tornar os programas binários portáveis através de diferentes máquinas
  - ▶ Necessário um compilador que compila para a JVM e um interpretador para executar os programas Java binários
- 

# Características

- ▶ Simples e orientada a objetos
  - Inspiração em C++ (porém mais simples)
  - Extensa biblioteca de classes
  - Conceitos OO (herança, encapsulamento, polimorfismo, ...)
- ▶ Robusta e segura
  - Verificações durante a compilação e em tempo de execução
  - Não existem ponteiros
  - *Garbage collector*

# Características

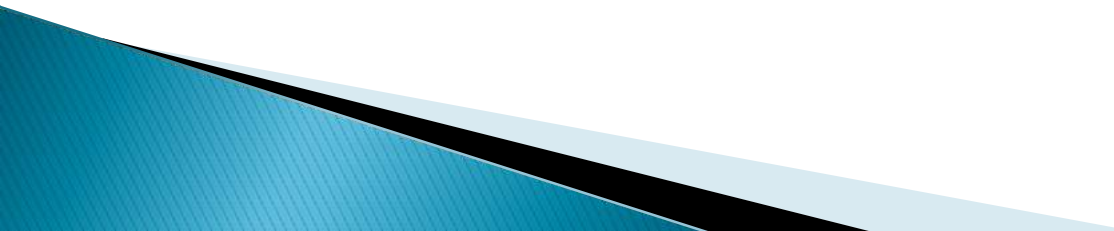
- ▶ Arquitetura neutra e portátil
  - Os *bytecodes* (gerados pelos compilador Java) formam um código intermediário que pode ser transportado e executado em diversas plataformas
  - Através da Máquina Virtual Java (JVM ou MVJ)

# Características

- ▶ Interpretada, *multithread* e dinâmica
  - Os bytecodes são portáteis (podem ser executados em qualquer plataforma que possua a JVM)
  - Java suporta *multithreading* no nível da linguagem
  - A linguagem Java é dinâmica: as classes somente são ligadas à aplicação quando necessário

# Características

## ▶ Desempenho

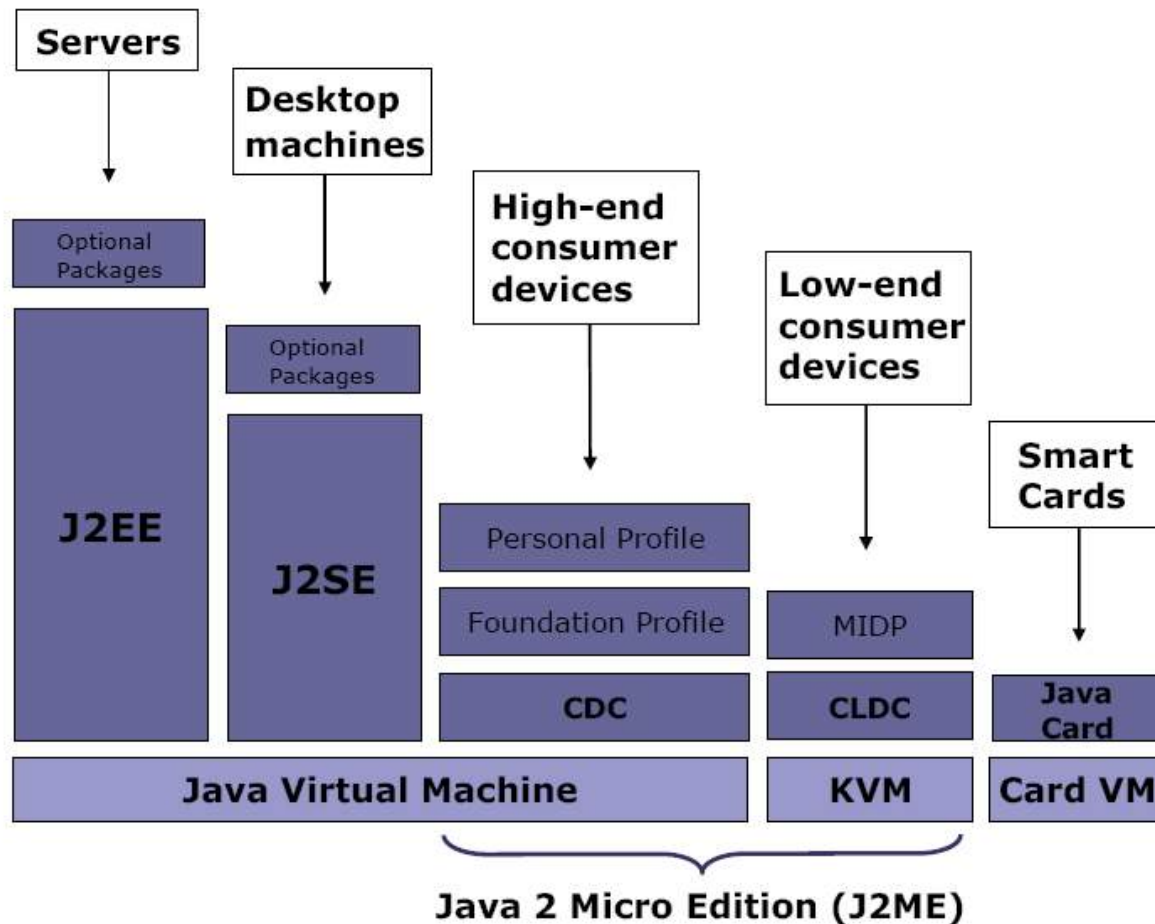
- Inferior ao de linguagens compiladas para linguagem de máquina
  - Existem alternativas para melhorar o desempenho
  - Entretanto, a diferença não é tão grande assim
- 

# Plataformas

- ▶ Java EE (Enterprise Edition)
  - Aplicações de grande porte
  - Geralmente client-server, Web
- ▶ Java SE (Standard Edition)
  - Desenvolvimento de aplicativos
- ▶ Java for Mobile Devices (Java ME)
  - Para dispositivos móveis (celulares/tablets)



# Plataformas e Máquinas Virtuais



## Plataformas

- J2EE
- J2SE
- J2ME
- JavaCard

## Máquinas Virtuais

- JVM
- CVM
- KVM
- Card VM
- HotSpot (JIT Compiler)

# Tecnologias

- ▶ Java EE – Java Enterprise Edition
  - Conjunto de tecnologias para desenvolvimento de aplicações servidor e distribuídas
- ▶ Java SE – Java Standard Edition
  - Antigamente era a única edição do Java (Java Platform), contém as principais classes da plataforma Java
- ▶ Java ME – Java Micro Edition
  - Plataforma mais jovem
  - Java para pequenos dispositivos (Pagers, celulares, etc...)
  - Vantagem: mesma linguagem e praticamente mesmas APIs no desenvolvimento de aplicações
- ▶ Java Card
  - Desenvolvimento de aplicações para cartões

# Tecnologias

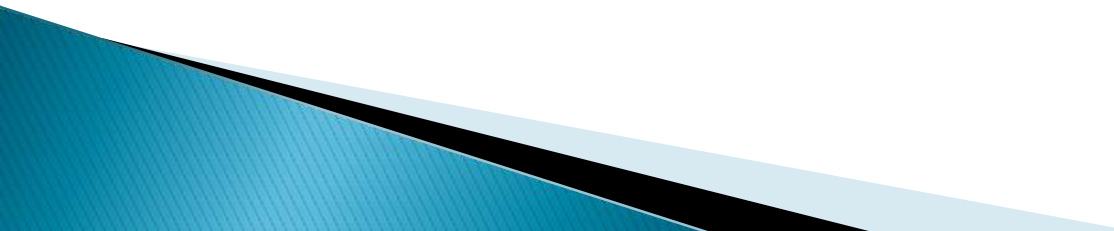
## ▶ JVM

- Java Virtual Machine

## ▶ API

- Application Programming Interfaces
- São bibliotecas de código pré-compilado (classes), que podem ser reutilizadas no desenvolvimento de aplicativos Java.

## ▶ JRE

- Java Runtime Environment
  - Contém as bibliotecas, máquina virtual e demais componentes necessários para executar applets e aplicações escritas em Java
  - Voltado para usuários
- 

# Tecnologias

## ▶ JDBC

- Java Database Connectivity
- Solução para acesso multiplataforma à banco de dados.

## ▶ RMI

- Remote Method Invocation

## ▶ JSP

- JavaServer Pages
- Criação de páginas web dinâmicas

## ▶ Applets

- Programa escrito em Java e que pode ser executado a partir de uma página HTML

# The Java SE Development Kit

- ▶ JDK: Kit de desenvolvimento
- ▶ Conjunto de ferramentas de desenvolvimento de aplicações
  - compilador
  - ambiente de execução de programas: máquina virtual, bibliotecas de classes e outros arquivos
  - classes de demonstração
  - depurador
  - documentação de classes
  - ...

# JSE – Java Standard Edition

- ▶ Algumas ferramentas
  - javac: compilador Java
  - java: interpretador Java
  - javadoc: gera documentação HTML para arquivos java, desde que estes tenham comentários no formato adequado
  - appletviewer: viabiliza a visualização de Applets através da emulação de um browser
  - jar: permite o agrupamento de diversos arquivos .class em um único arquivo .jar

# Onde aprender mais ?

- ▶ Site da Oracle para Java:  
<http://www.oracle.com/technetwork/java>
  - ▶ JavaSE –  
<http://www.oracle.com/technetwork/java/javase>
  - ▶ JavaEE –  
<http://www.oracle.com/technetwork/java/javaee>
  - ▶ J2ME –  
<http://www.oracle.com/technetwork/java/javame>
- 