



Curso de Especialização em Tecnologia Java

Fábio Fernandes da Rocha Vicente

PARTE I

Conceitos da Linguagem de Programação Java

11 Palavras-chave

1. Simples
2. Orientada a objetos
3. Distribuída
4. Robusta
5. Segura
6. Neutra em relação à arquitetura
7. Portável
8. Interpretada
9. Alto desempenho
10. Múltiplas linhas de execução
11. Dinâmica

I. Simple

“Nós quisemos elaborar um sistema que pudesse ser programado facilmente sem muito treinamento esotérico e que facilitasse a prática padrão atual. Assim, embora achássemos que a linguagem C++ não fosse adequada, nós projetamos a linguagem Java o mais parecida possível com o C++ a fim de tornar o sistema mais compreensível. Java omite recursos raramente usados e pouco compreendidos do C++ que, em nossa experiência, trazem mais dificuldades que benefícios.”

2. Orientada a Objetos

Construir softwares de maneira rápida, correta e econômica é sempre o objetivo da programação. “ Objetos ou, mais precisamente, as classes de onde os objetos vêm são essencialmente componentes reutilizáveis de software. Há objetos data, objetos data/hora, objetos áudio, objetos vídeo, objetos automóvel, objetos pessoas etc. Quase qualquer substantivo pode ser razoavelmente representado como um objeto de software em termos dos atributos (por exemplo, nome, cor e tamanho) e comportamentos (por exemplo, calcular, mover e comunicar). [...] Programas orientados a objetos são muitas vezes mais fáceis de entender, corrigir e modificar.”

H. M. Deitel, P. J. Deitel. Java: Como Programar, 10a. Edição. Pearson

Education do Brasil, 2017.

3. Distribuída

A linguagem Java tem uma biblioteca extensa de rotinas para lidar com protocolos TCP/IP, como o HTTP e o FTP. Os aplicativos Java podem abrir e acessar objetos na Internet através de URLs com a mesma facilidade com que se acessa um arquivo no sistema local.

4. Robusta

Java tem a intenção de escrever programas que precisam ser confiáveis de várias formas. O Java coloca muita ênfase na **verificação antecipada** de possíveis problemas, na **verificação dinâmica em tempo de execução** e na **eliminação de situações sujeitas a erros**. A única e grande diferença entre o Java e o C++ está em que o **Java usa um modelo de ponteiros que elimina a possibilidade de sobrescrita de memória e conseqüente destruição de dados**.

5. Segura

A linguagem Java tem a intenção de ser usada em ambientes de rede/distribuídos. Nesta direção, grande ênfase tem sido colocada na segurança. O Java permite a construção de sistemas livres de vírus e adulterações.

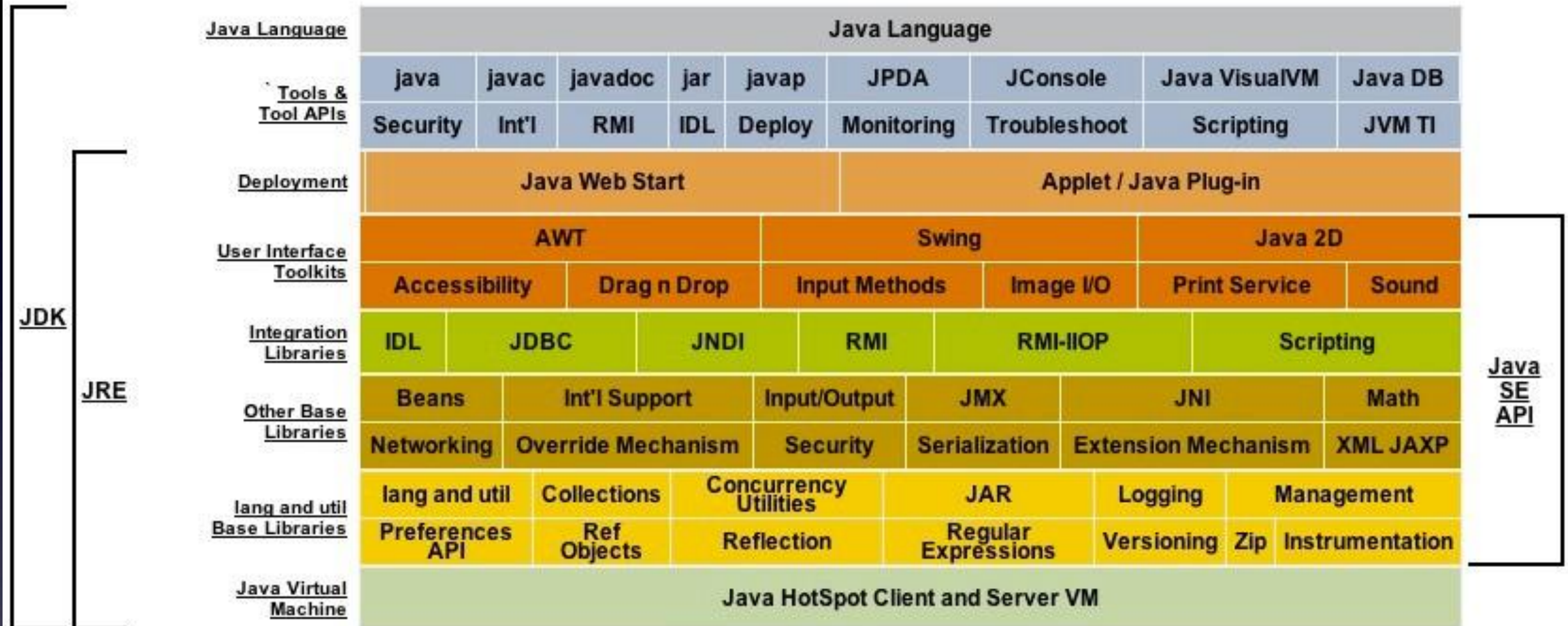
6. Neutra em relação à arquitetura

O compilador gera um formato de arquivo objeto **neutro em relação à arquitetura** - o código compilado é executável em muitos processadores, **dada a presença do sistema Java em tempo de execução.**

O compilador Java faz isso gerando instruções em **bytecodes** que nada têm a ver com uma arquitetura particular de um computador específico.

Em vez disto, elas são projetadas para serem fáceis de interpretar em qualquer máquina e convertidas facilmente para o código nativo da máquina em tempo real.

Description of Java Conceptual Diagram



<http://docs.oracle.com/javase/7/docs/index.html>

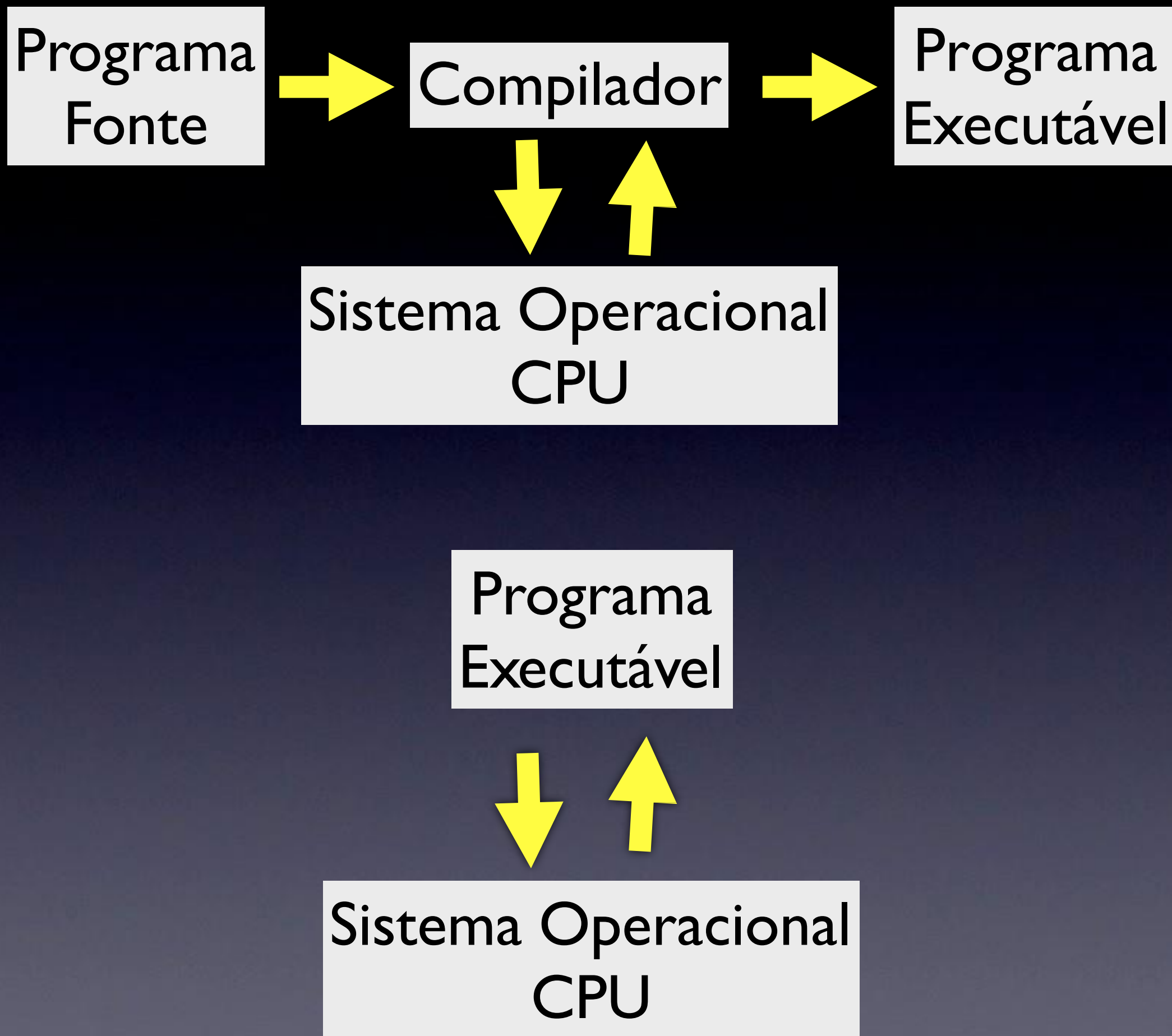
7. Portável

Diferente das linguagens C e C++, não há aspectos “dependentes da implementação” na especificação da linguagem Java. Os tamanhos dos tipos de dados primitivos são especificados, assim como o comportamento da aritmética que os utiliza.

Exemplos: tamanho de inteiros, aritmética de ponto flutuante, etc.

8. Interpretada

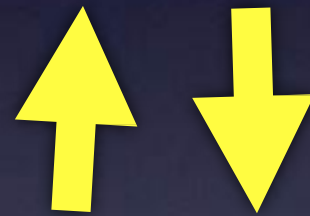
O interpretador Java pode executar bytecodes Java diretamente em qualquer máquina para a qual tenha sido escrito. Como a ligação é um processo mais incremental e leve, o processo de desenvolvimento pode ser muito mais rápido e exploratório.



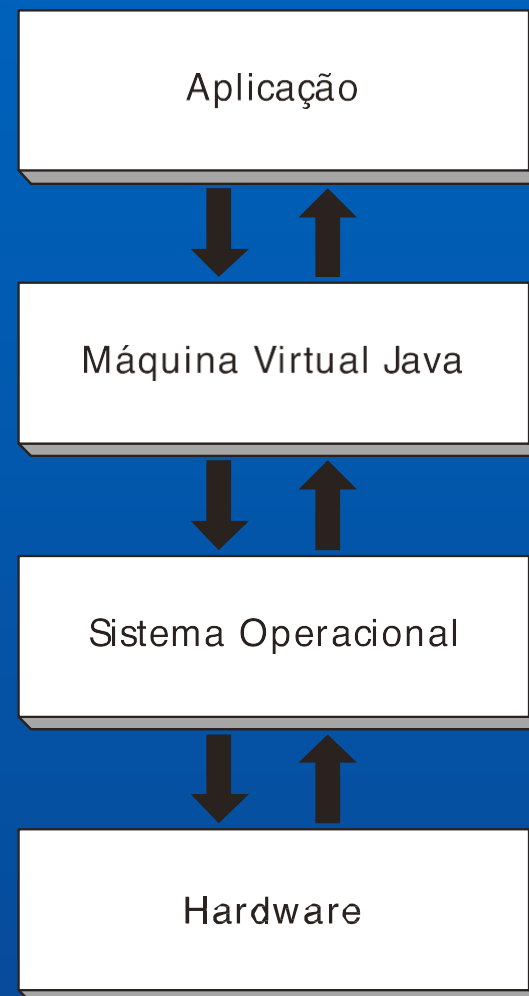
Programa
Fonte



Interpretador



Sistema Operacional
CPU



fonte da figura: <http://www.training.com.br/aso/>

9. Alto Desempenho

Embora o desempenho dos bytecodes interpretados seja geralmente mais do que adequado, há situações em que é necessário um desempenho superior. Os bytecodes podem ser convertidos em tempo real para o código de máquina da CPU particular onde o aplicativo está executando.

10. Múltiplas linhas de execução (threads)

Os benefícios das múltiplas linhas d execução (threads) são melhor capacidade de resposta interativa e melhor comportamento em tempo real de execução

II. Dinâmica

De várias formas Java é uma linguagem mais dinâmica que C ou C++ (...) Na linguagem Java obter informação em tempo de execução é imediato.

Fim da parte I