

Aula 1 – Introdução ao Java

Prof. Me. Rodrigo Brito Battilana

Roteiro

- Introdução
- A história do JAVA
- Bibliotecas JAVA
- Ambiente de desenvolvimento JAVA
- Referências

Introdução

- **Java**: uma das linguagens de programação mais utilizadas no mundo.
- Instruções fazem com que os computadores realizem tarefas.
- O software (isto é, as instruções que você escreve) controla o hardware (isto é, os computadores).

Introdução

- **Programação orientada a objetos** — atualmente a metodologia-chave de programação.
- Ele também é amplamente utilizado para implementar aplicativos e softwares baseados na internet para dispositivos que se comunicam por uma rede.

Introdução

- De acordo com um estudo realizado pela Gartner, os dispositivos móveis continuarão a ultrapassar os PCs como os dispositivos de computação principais dos usuários.
- Isso está criando oportunidades profissionais significativas para pessoas que programam aplicativos móveis, muitos dos quais são programados em Java.

Hardware e Software

- Os computadores processam dados sob o controle de conjuntos de instruções chamados **programas de computador**.
- Esses programas de software orientam o computador por meio de ações ordenadas especificadas por pessoas chamadas **programadores de computador**.
- **Programação orientada a objetos**: melhora a produtividade do programador, reduzindo os custos de desenvolvimento de softwares.

Hardware e Software

► Organização do computador

- Os computadores podem ser visualizados como divididos em várias unidades lógicas ou seções lógicas:
 - ✓ Unidade de entrada
 - ✓ Unidade de saída
 - ✓ Unidade de memória
 - ✓ Unidade de aritmética e lógica (ALU)
 - ✓ Unidade de processamento central (CPU)
 - ✓ Unidade de armazenamento secundária

Linguagens de máquina, assembly e de alto nível

- As linguagens de programação podem ser divididas em três tipos gerais:
- **1. Linguagens de máquina:** strings de números que instruem os computadores a realizar suas operações mais elementares uma de cada vez.
- **2. Linguagens assembly:** abreviações em inglês para representar operações elementares formam a base de linguagens assembly.
- **3. Linguagens de alto nível:** instruções únicas podem ser escritas para realizar tarefas substanciais.

Linguagens de máquina, assembly e de alto nível

- **Assemblers:** *Programas tradutores* desenvolvidos para converter os primeiros programas de linguagem assembly em linguagem de máquina a velocidades de computador.
- **Compiladores:** programas tradutores desenvolvidos para converter os programas de linguagem de alto nível em linguagem de máquina.

A história do Java

- Os microprocessadores têm um impacto profundo em dispositivos inteligentes eletrônicos de consumo popular.

- 1991

Reconhecendo isso, a Sun Microsystems financiou um projeto de pesquisa corporativo interno, que resultou em uma linguagem baseada em C++ chamada Java.

Criada por James Gosling.

A história do Java

- 1993
- A Web explodiu em popularidade.
- A Sun viu o potencial de utilizar o Java para adicionar conteúdo dinâmico a páginas da Web.
- O Java chamou a atenção da comunidade de negócios por causa do enorme interesse pela Web.

A História do Java

- Linguagem de programação orientada a objetos.
- “Escreva uma vez, execute em qualquer lugar”.
- Utilizado para desenvolver aplicativos corporativos de grande porte, aprimorar a funcionalidade de servidores da web, fornecer aplicativos para dispositivos voltados ao consumo popular (por exemplo, telefones celulares, smartphones, televisão, *set-up boxes* etc.) e para muitos outros propósitos.
- Linguagem-chave para desenvolvimento de aplicativos Android adequados a smartphones e tablets.

A História do Java

- Você pode criar cada classe e método de que precisa para formar seus programas Java. Porém, a maioria dos programadores Java tira proveito das ricas coleções de classes existentes e métodos nas **bibliotecas de classe Java**, também conhecidas como **Java APIs (*application programming interfaces*)**.

Bibliotecas de classe do Java

- Os programas Java consistem em partes chamadas **classes**.
- As classes incluem partes chamadas **métodos** que realizam tarefas e retornam informações quando as tarefas são concluídas.

- **Java, bibliotecas de classe**

Ricas coleções de classes

Também conhecidas como **Java APIs (Application Programming Interfaces)**

- Há dois aspectos para aprender o “mundo” do Java.

A linguagem Java em si.

As classes nas extensas bibliotecas de classe Java.

- Download da documentação da Java API

java.sun.com/javase/downloads/

Role até a seção **Additional Resources** e clique no botão **Download** à direita de **Java SE 6 Documentation**.

Bibliotecas de classe do Java

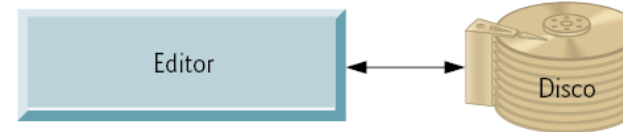
- Dicas de programação.
 - **Observações de engenharia de software** — explicam conceitos que afetam e aprimoram a arquitetura total e a qualidade de sistemas de software.
 - **Boas práticas de programação** ajudam a escrever programas que são mais claros, mais compreensíveis, mais fáceis de manter e mais fáceis de testar e depurar — isto é, remover erros de programação).
 - **Erros de programação comuns** — discutem problemas a que se deve prestar atenção e evitar.

Bibliotecas de classe do Java

- Dicas de programação (cont.):
 - **Dicas de desempenho** — técnicas para escrever programas que executam mais rapidamente e utilizam menos memória).
 - **Dicas de portabilidade** — técnicas para ajudá-lo a escrever programas que podem executar, com pouca ou nenhuma modificação, em diferentes computadores
 - **Dicas para prevenção de erros** — técnicas para remover bugs dos programas.
 - **Observações sobre a aparência e comportamento** — técnicas para ajudá-lo a projetar a “aparência” e o “comportamento” das interfaces com o usuário dos seus aplicativos em termos da aparência e facilidade de uso).

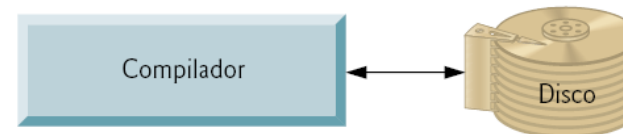
Ambiente de desenvolvimento JAVA

Fase 1: Edita



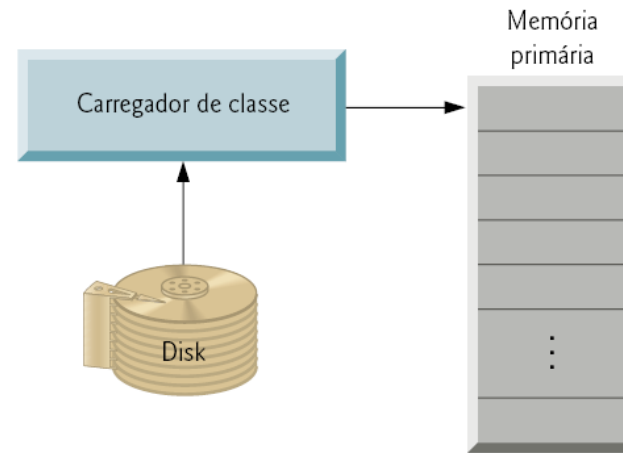
O programa é criado em um editor e armazenado em disco em um arquivo cujo nome termina com `.java`.

Fase 2: Compila



O compilador cria bytecodes e os armazena em disco em um arquivo cujo nome termina com `.class`.

Fase 3: Carrega



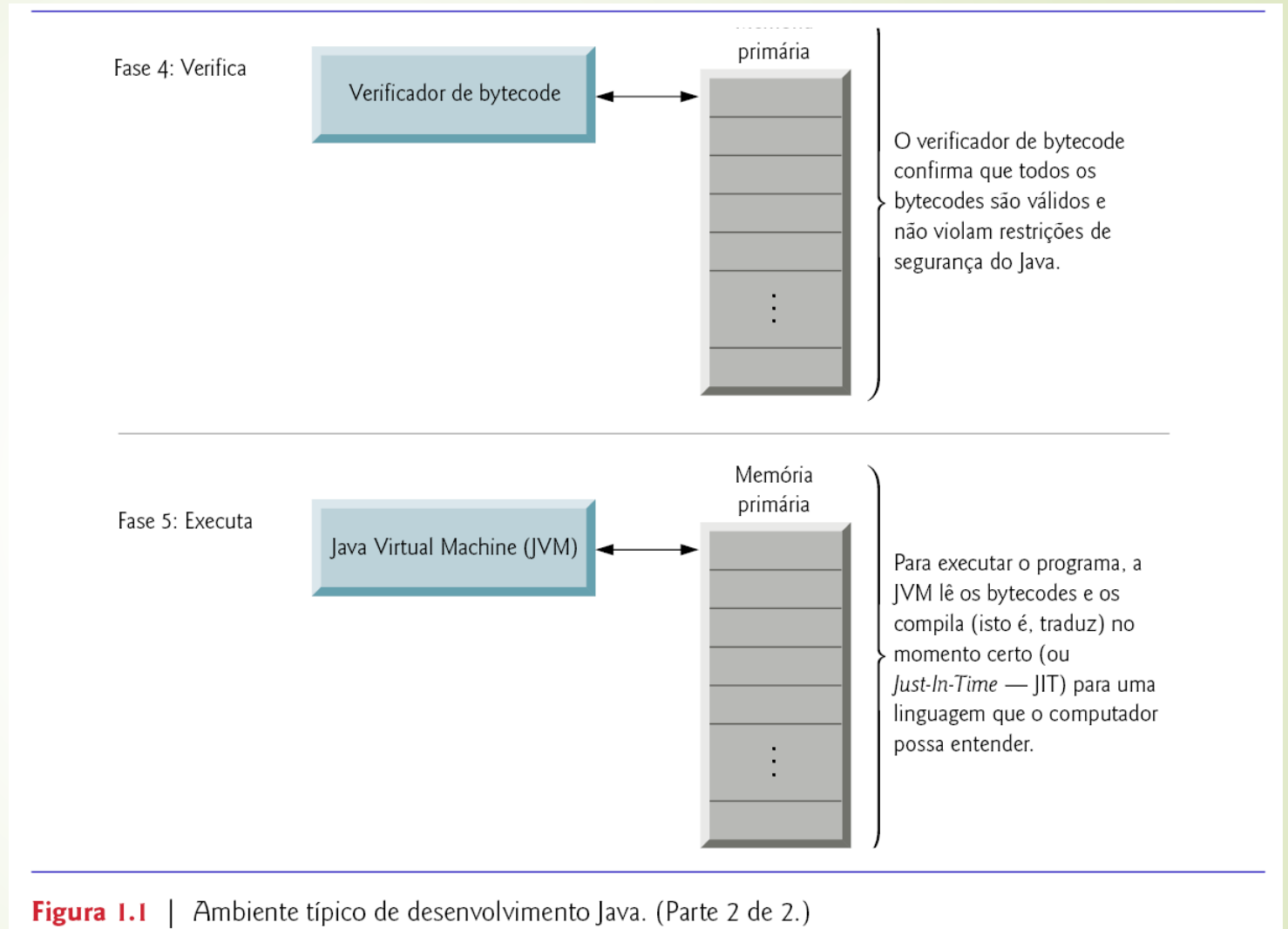
O carregador de classe lê os arquivos `.class` que contêm bytecodes a partir do disco e coloca esses bytecodes na memória.

➡ FONTE: DEITEL 2010

Figura 1.1 | Ambiente típico de desenvolvimento Java. (Parte 1 de 2.)

Ambiente de desenvolvimento JAVA

➤ FONTE: DEITEL 2010



Fases de desenvolvimento JAVA

- ▀ Programas Java normalmente passam por cinco fases:
 - ▀ 1) editar
 - ▀ 2) compilar
 - ▀ 3) carregar
 - ▀ 4) verificar
 - ▀ 5) executar

Fases de desenvolvimento JAVA

- A Fase 1 consiste em editar um arquivo com um **programa editor** (em geral, conhecido simplesmente como **editor**).
 - Escrever um programa Java (**código-fonte**), usando o editor
 - Fazer quaisquer correções necessárias.
 - Salvar o programa.
 - Um nome de arquivo que termina com a extensão **.java** indica que o arquivo contém o código-fonte Java.
 - Editores Linux: vi e emacs.
 - Editores Windows: Notepad, EditPlus (www.editplus.com), TextPad (www.textpad.com) e jEdit (www.jedit.org).

Fases de desenvolvimento JAVA

- **Ambientes de desenvolvimento integrado (IDEs)**

- IDEs fornecem ferramentas que suportam o processo de desenvolvimento de software, incluindo editores para escrever e editar programas e depuradores para localizar **erros de lógica** — erros que fazem os programas executar incorretamente.

- IDEs populares

- Eclipse (www.eclipse.org)
 - NetBeans (www.netbeans.org)
 - JBuilder (www.codegear.com)
 - JBuilder (www.codegear.com)
 - BlueJ (www.blueJ.org)
 - jGRASP (www.jgrasp.org)

Fases de desenvolvimento JAVA

➤ Fase 2

➤ Use o comando **javac** (o **compilador Java**) para **compilar** um programa. Por exemplo, para compilar um programa chamado `Welcome.java`, você digitaria `javac Welcome.java`

➤ Se o programa compilar, o compilador produz um arquivo **.class** chamado `Welcome.class` que contém a versão compilada do programa.

Fases de desenvolvimento JAVA

23

- O compilador Java converte o código-fonte Java em **bytecodes** que representam as tarefas a ser executadas.
- Os bytecodes são executados pela **Java Virtual Machine (JVM)** — uma parte do JDK e a base da plataforma Java.
- **Máquina virtual (VM)** — um aplicativo de software que simula um computador.
 - Oculta o sistema operacional e o hardware subjacentes dos programas que interagem com ela.
- Se a mesma VM for implementada nas várias plataformas de computador, os aplicativos que ela executa podem ser utilizados em todas essas plataformas.

Fases de desenvolvimento JAVA

- Os bytecodes são independentes de plataforma.
 - Eles não dependem de uma plataforma de hardware específica.
- Bytecodes são **portáveis**.
 - Os mesmos bytecodes podem executar em qualquer plataforma contendo uma JVM que entende a versão do Java em que os bytecodes foram compilados.
- A JVM é invocada pelo comando **java**. Por exemplo, para executar um aplicativo Java chamado Welcome, você digitaria o comando.
 - `java Welcome`

Fases de desenvolvimento JAVA

■ Fase 3

- A JVM armazena o programa na memória para executá-lo.
- Isso é conhecido como **carregamento**.
- O **carregador de classe** pega os arquivos .class que contêm os bytecodes do programa e transfere-os para a memória primária.
- Também carrega qualquer arquivo .class fornecido pelo Java que seu programa utiliza.
- Os arquivos .class podem ser carregados a partir de um disco em seu sistema ou por intermédio de uma rede.

Fases de desenvolvimento JAVA

■ Fase 4

- À medida que as classes são carregadas, o **verificador de bytecode** examina seus bytecodes.
- Assegura que eles são válidos e não violam as restrições de segurança do Java.
- O Java impõe uma forte segurança para certificar-se de que os programas Java que chegam pela rede não danificam os arquivos ou o sistema (como vírus e vermes de computador).

Fases de desenvolvimento JAVA

▀ Fase 5

▀ A JVM executa os bytecodes do programa.

▀ Em geral, as JVMs atuais executam uma combinação de interpretação e **compilação just-in-time (JIT)**.

▀ Analisa os bytecodes à medida que eles são interpretados, procurando **hot spots** — partes dos bytecodes que executam com frequência.

▀ Um **compiler just-in-time (JIT)** (o compilador **Java HotSpot**) traduz os bytecodes para a linguagem de máquina do computador subjacente.

Fases de desenvolvimento JAVA

► Fase 5

- Quando a JVM encontra novamente essas partes compiladas, o código de linguagem de máquina mais rápido é executado.
- Os programas Java, na realidade, passam por duas fases de compilação:
- Uma fase em que código-fonte é traduzido em bytecodes (para a portabilidade entre JVMs de diferentes plataformas de computador).
- Uma segunda em que, durante a execução, os bytecodes são traduzidos em linguagem de máquina para o computador real em que o programa é executado.

Referências

- ▶ DEITEL; PAUL, DEITEL; HARVEI, **Java: Como Programar** 8ª edição, Editora: Pearson, São Paulo, 2010.