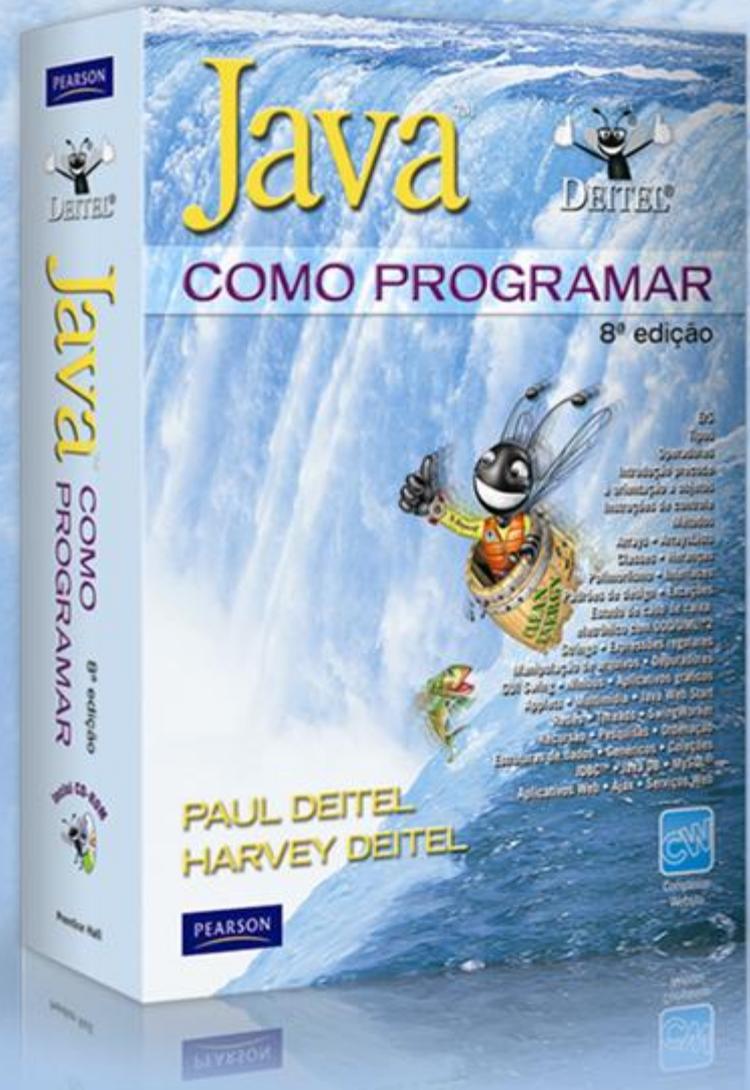


Capítulo 1 Introdução aos computadores, à Internet e à World Wide Web4

Java™ Como Programar, 8/E



Java™



COMO PROGRAMAR

8ª edição

OBJETIVOS

Neste capítulo, você aprenderá:

- Os conceitos básicos de rede, hardware e software de computador.
- Os conceitos básicos de tecnologia de objeto, como classes, objetos, atributos, comportamentos, encapsulamento e herança.
- Os diferentes tipos de linguagens de programação e quais linguagens são mais amplamente utilizadas.
- Um ambiente de desenvolvimento de programa Java típico.
- O papel do Java no desenvolvimento de aplicativos cliente/servidor distribuídas para a Internet e a Web.
- A história da UML — a linguagem de design orientado a objetos padrão no setor.
- A história da Internet e da World Wide Web até a Web 2.0.
- A fazer test-drive de aplicativos Java.
- Algumas das principais tecnologias de software mais recentes.

Java™



COMO PROGRAMAR

8ª edição

- I.1** Introdução
- I.2** Computadores: hardware e software
- I.3** Organização do computador
- I.4** Primeiros sistemas operacionais
- I.5** Computação pessoal distribuída e computação cliente/servidor
- I.6** A Internet e a World Wide Web
- I.7** Linguagens de máquina, linguagens assembly e linguagens de alto nível
- I.8** História do C e do C++
- I.9** História do Java
- I.10** Bibliotecas de classe do Java
- I.11** Fortran, Cobol, Pascal e Ada
- I.12** Basic, Visual Basic, Visual C++, C# e .NET
- I.13** Ambiente típico de desenvolvimento Java
- I.14** Notas sobre o Java e este livro
- I.15** Testando um aplicativo Java

Java™



COMO PROGRAMAR

8ª edição

- I.16** Estudo de caso de engenharia de software: introdução à tecnologia de objetos e à UML
- I.17** Web 2.0
- I.18** Tecnologias de software
- I.19** Conclusão
- I.20** Recursos da Web

Java™



COMO PROGRAMAR

8ª edição

1.1 Introdução

- O núcleo do livro enfatiza como alcançar clareza no programa por meio de técnicas comprovadas de programação orientada a objetos.
- **abordagem live-code** — recursos do Java são apresentados no contexto de programas Java funcionais completos.
- Programas de exemplo para download
www.deitel.com/books/jhttp8/
www.prenhall.com/deitel
- Consulte a Seção “Antes de você começar” depois do *Prefácio*

Java™



COMO PROGRAMAR

8ª edição

- Você aprenderá a escrever instruções para comandar o computador e executar as ações.
- O software (isto é, as instruções que você escreve) controla o hardware (isto é, os computadores).
- O Java é uma das mais populares linguagens para desenvolvimento de software atualmente.

Java™



COMO PROGRAMAR

8ª edição

- O uso de computadores está aumentando em quase todos os campos de trabalho.
- Os custos da computação estão caindo drasticamente.
- Mais de um bilhão de computadores de uso geral estão em utilização em todo o mundo.

Java™



COMO PROGRAMAR

8ª edição

- Ao longo dos anos, muitos programadores aprenderam programação estruturada.
- Você aprenderá a programação estruturada e **programação orientada a objetos** — a metodologia de programação-chave utilizada pelos programadores de hoje em dia.
- Você criará e trabalhará com muitos objetos de software.

Sua estrutura interna costuma ser construída, utilizando técnicas de programação estruturada.

- A lógica de manipular objetos é, ocasionalmente, expressa com programação estruturada.

Java™



COMO PROGRAMAR

8ª edição

- O Java tornou-se a linguagem preferida para implementar aplicativos baseados na Internet e softwares para dispositivos que se comunicam em uma rede.
- Atualmente há bilhões de celulares e dispositivos portáteis compatíveis com Java!
- O Java é a linguagem preferida para atender às necessidades de programação de muitas organizações.

Java™



COMO PROGRAMAR

8ª edição

- Este livro é baseado na **Java Standard Edition (Java SE) 6**, Atualização 11
- **Java Enterprise Edition (Java EE)**
projetado para desenvolver aplicativos em rede distribuídos e em grande escala e também aplicativos baseados na Web.
- **Java Micro Edition (Java ME)**
projetado para o desenvolvimento de aplicativos de pequenos dispositivos com limitações de memória, como telefones celulares, pagers e PDAs.

Java™



COMO PROGRAMAR

8ª edição

- E-mail do autor
deitel@deitel.com.
- *Deitel® Buzz Online*
www.deitel.com/newsletter/subscribe.html
- Java Resource Centers (e outros relacionados)
www.deitel.com/ResourceCenters.html.

Java™



COMO PROGRAMAR

8ª edição

1.2 Computadores: Hardware e Software

- **Computador** — Um dispositivo que pode realizar cálculos e tomar decisões lógicas fenomenalmente mais rápido do que os seres humanos.
- Os computadores pessoais de hoje em dia podem realizar bilhões de cálculos em um segundo.
- **Supercomputadores** já realizam *milhares de trilhões (quatrilhões) de instruções por segundo!*
- Os computadores processam **dados** *sob o controle de conjuntos de instruções chamados programas de computador.*
- Esses programas orientam o computador por meio de conjuntos ordenados de ações especificadas por pessoas chamadas **programadores de computador.**

Java™



COMO PROGRAMAR

8ª edição

- Um computador consiste em vários dispositivos conhecidos como **hardware**, por exemplo, teclado, monitor, mouse, discos, memória, unidades de DVD e de CD-ROM e unidades de processamento).
- Os programas que executam em um computador são referidos como **software**.

Java™



COMO PROGRAMAR

8ª edição

1.3 Organização do computador

- **Unidades lógicas** ou seções de um computador.
- **Unidade de entrada.** Essa seção de “recebimento” obtém informações (dados e programas de computador) de **dispositivos de entrada** e as coloca à disposição de outras unidades para serem processadas.
- **Dispositivos de entrada.** Teclado, mouse, microfone, scanner, unidades de disco, unidades de CD, unidades de DVD, dispositivos USB etc.

Java™



COMO PROGRAMAR

8ª edição

- **Unidade de saída.**

Essa seção de “entrega” pega as informações que o computador processou e as coloca em vários **dispositivos de saída** a fim de torná-las disponíveis para serem utilizadas fora do computador.

- **Dispositivos de saída**

Monitor, impressora, alto-falantes etc.

Java™



COMO PROGRAMAR

8ª edição

- **Unidade de memória.**

Seção de “armazenamento” de acesso rápido e relativamente baixa capacidade retém as informações que foram inseridas por meio da unidade de entrada, tornando-as imediatamente disponíveis para processamento quando necessário.

Retém informações processadas até que elas possam ser colocadas em dispositivos de saída pela unidade de saída.

As informações na unidade de memória são **voláteis** — em geral, são perdidas quando o computador é desligado.

Costuma-se chamá-la de **memória** ou *memória primária*.

Java™



COMO PROGRAMAR

8ª edição

- **Unidade de aritmética e lógica (*arithmetic and logic unit* – ALU).**

A seção “produção” realiza cálculos, como adição, subtração, multiplicação e divisão.

Contém os mecanismos que permitem que o computador tome decisões.

Nos sistemas atuais, a ALU normalmente é implementada como parte da próxima unidade lógica, a CPU.

Java™



COMO PROGRAMAR

8ª edição

- **Unidade central de processamento (*central processing unit* – CPU).**

A seção “administrativa” coordena e supervisa a operação das outras seções.

Diz à unidade de entrada quando as informações devem ser lidas e transferidas para a unidade de memória.

Avisa à ALU quando as informações da unidade de memória devem ser utilizadas em cálculos e diz a unidade de saída quando enviar as informações da unidade de memória para certos dispositivos de saída.

Muitos computadores de hoje têm múltiplas CPUs — esses computadores são chamados de **multiprocessadores**.

Um **processador de múltiplos núcleos (ou multi-core)** implementa o multiprocessamento em um único chip de circuito integrado.

Java™



COMO PROGRAMAR

8ª edição

- **Unidade de armazenamento secundária.**

Seção de “armazenamento” de longo prazo e alta capacidade.

Programas ou dados que não são utilizados ativamente pelas outras unidades, em geral, são colocados em dispositivos de armazenamento secundário (por exemplo, unidades de disco) até que sejam necessários, possivelmente horas, dias, meses ou até mesmo anos mais tarde.

Dizemos que as informações nos dispositivos de armazenamento secundário são **persistentes** — são preservadas mesmo quando a energia elétrica é desligada.

Exemplos de dispositivos de armazenamento secundário: CDs, DVDs e cartões de memória

Java™



COMO PROGRAMAR

8ª edição

1.4 Primeiros sistemas operacionais

- Os primeiros computadores podiam realizar apenas um **trabalho** *ou tarefa por vez*.
Processamento em lotes de um único usuário.

O computador executava um único programa ao processar dados em grupos ou **lotes**.

- **Sistemas operacionais**

Desenvolvidos para tornar o uso dos computadores mais conveniente.

Os primeiros sistemas operacionais facilitavam e aceleravam a transição entre trabalhos, aumentando a quantidade de trabalho, ou **throughput**, que os computadores poderiam processar em um dado tempo.

- **Multiprogramação**

Operação simultânea de muitos trabalhos que competem para compartilhar os recursos do computador.

Java™



COMO PROGRAMAR

8ª edição

1.5 Computação pessoal distribuída e computação cliente/servidor

- Em 1977, a Apple Computer popularizou a **computação pessoal**.
- Em 1981, a IBM introduziu o IBM Personal Computer.
Isso legitimou rapidamente a computação pessoal em negócios, indústria e organizações governamentais.
- Os primeiros computadores pessoais podiam ser interconectados em redes de computadores, às vezes por linhas telefônicas e às vezes em **redes locais** (*local area networks* – **LANs**) dentro de uma organização.

Isso levou à **computação distribuída** — a computação de uma organização é distribuída em redes para os sites em que o trabalho da organização é realizado.

Java™



COMO PROGRAMAR

8ª edição

- Os computadores pessoais agora são tão poderosos quanto as máquinas de milhões de dólares de apenas algumas décadas atrás.
- **Servidores** armazenam dados que podem ser utilizados por computadores clientes distribuídos por toda a rede; daí o termo **computação cliente/servidor**.
- O Java é amplamente utilizado para escrever softwares de rede de computador e aplicativos distribuídos para cliente/servidor.

Java™



COMO PROGRAMAR

8ª edição

1.6 A Internet e a World Wide Web

- **A Internet**

Rede global de computadores.

Tem suas raízes na década de 1960, quando o Departamento da Defesa dos EUA financiou o projeto.

Agora acessível por bilhões de computadores e dispositivos controlados por computador no mundo inteiro.

- **World Wide Web**

Permite que usuários de computador localizem e visualizem documentos baseados em multimídia sobre quase qualquer assunto pela Internet.

Java™



COMO PROGRAMAR

8ª edição

1.7 Linguagens de máquina, linguagens assembly e linguagens de alto nível

- Os programadores escrevem instruções em várias linguagens de programação, algumas diretamente compreensíveis por computadores e, outras, requerendo passos intermediários de **tradução**.
- Três tipos gerais de linguagem:
 - Linguagens de máquina
 - Linguagens assembly
 - Linguagens de alto nível

Java™



COMO PROGRAMAR

8ª edição

- Qualquer computador pode entender diretamente apenas sua própria **linguagem de máquina**.

Essa é a “linguagem natural” do computador, definida pelo projeto de hardware.

Em geral, as linguagens de máquina consistem em strings de números (em última instância reduzidas a 1s e 0s) que instruem os computadores a realizar suas operações mais elementares uma de cada vez.

Dependentes de máquina — uma linguagem de máquina específica que só pode ser utilizada em um tipo de computador.

Java™



COMO PROGRAMAR

8ª edição

- Abreviações em inglês que representam operações elementares formavam a base das **linguagens assembly**.
- **Programas tradutores** chamados **assemblers** convertem programas de linguagem assembly em linguagem de máquina.

Java™



COMO PROGRAMAR

8ª edição

- **Linguagens de alto nível**

Instruções simples realizam tarefas substanciais.

Compiladores convertem programas de linguagem assembly em linguagem de máquina.

Permitem aos programadores escrever instruções parecidas com o inglês cotidiano e contêm notações matemáticas comumente utilizadas.

- C, C++, linguagens .NET da Microsoft (por exemplo, Visual Basic, Visual C++ e C#) estão entre as linguagens de programação de alto nível mais amplamente utilizadas; o Java é, de longe, a linguagem mais amplamente utilizada.

Java™



COMO PROGRAMAR

8ª edição

- O processo de compilação de um programa de linguagem de alto nível em linguagem de máquina pode consumir uma quantidade considerável de tempo do computador.
- **Interpretadores** executam linguagem de alto nível diretamente, embora sejam mais lentos que programas compilados.
- O Java utiliza uma combinação inteligente de compilação e interpretação para executar programas.

Java™



COMO PROGRAMAR

8ª edição

1.8 História do C e do C++

- O Java evolui do C++, que evolui do C, que evoluiu do BCPL e do B.
- C

Originalmente implementada em 1972.

Criada por Dennis Ritchie, na Bell Laboratories, a partir de uma evolução da linguagem B.

Inicialmente tornou-se amplamente conhecida como a linguagem de desenvolvimento do sistema operacional UNIX.

Hoje, a maior parte do código para sistemas operacionais de uso geral é escrita em C ou C++.

Java™



COMO PROGRAMAR

8ª edição

- C++

Uma extensão do C.

Desenvolvida por Bjarne Stroustrup no início da década de 1980 na Bell Laboratories.

Fornece capacidades de *programação orientada a objetos*.

Linguagem híbrida — é possível programar tanto no estilo C como no estilo orientado a objetos, ou em ambos.

Java™



COMO PROGRAMAR

8ª edição

1.9 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.

- 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.

Java™



COMO PROGRAMAR

8ª edição

1.10 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**.

Java™



COMO PROGRAMAR

8ª edição



Observação de engenharia de software I.1

Utilize uma abordagem de bloco de construção para criar seus programas. Evite reinventar a roda — utilize partes existentes onde quer que possível. Essa reutilização de software é um benefício fundamental da programação orientada a objetos.

Java™



COMO PROGRAMAR

8ª edição

- 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.

Java™



COMO PROGRAMAR

8ª edição

- 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).

Java™



COMO PROGRAMAR

8ª edição



Observação de engenharia de software 1.2

Ao programar em Java, você geralmente utilizará os seguintes blocos de construção: classes e métodos de bibliotecas de classe, classes e métodos que você mesmo cria e classes e métodos que outros criam e tornam disponíveis para você.

Java™



COMO PROGRAMAR

8ª edição



Dica de desempenho 1.1

Usar as classes e métodos da Java API em vez de escrever suas próprias versões pode melhorar o desempenho de programa, porque eles são cuidadosamente escritos para executar de modo eficiente. Essa técnica também diminui o tempo de desenvolvimento de programa.

Java™



COMO PROGRAMAR

8ª edição



Dica de portabilidade 1.1

Utilizar as classes e métodos da Java API em vez de escrever suas próprias versões melhora a portabilidade de programa, porque esses são incluídos em cada implementação Java.

Java™



COMO PROGRAMAR

8ª edição

1.11 Fortran, COBOL, Pascal e Ada

- **Fortran** (*FORmula TRANslator*)

Desenvolvida pela IBM Corporation em meados da década de 1950

Utilizada para aplicativos científicos de engenharia que requerem computações matemáticas complexas.

Ainda é amplamente utilizada, especialmente em aplicativos de engenharia.

- **COBOL (COmmon Business Oriented Language)**

Desenvolvida no fim da década de 1950 por um grupo de fabricantes de computador e usuários de computadores do governo dos EUA e da indústria.

Utilizada para aplicativos comerciais que exigem manipulação precisa e eficiente de grandes quantidades de dados.

Grande parte do software utilizado em grandes empresas varejistas ainda é programada em COBOL.

Java™



COMO PROGRAMAR

8ª edição

- Pesquisas na década de 1960 resultaram na evolução da **programação estruturada**.

Uma abordagem disciplinada para escrever programas mais claros, mais fáceis de testar e depurar e mais fáceis de modificar do que os grandes programas produzidos com a técnica anterior.

- Pascal

Desenvolvida pelo Professor Niklaus Wirth em 1971.

Projetado para ensinar a programação estruturada em ambientes acadêmicos.

Java™



COMO PROGRAMAR

8ª edição

- **Programação Ada**

Desenvolvida sob o patrocínio do Departamento da Defesa dos EUA durante a década de 1970 e início da de 1980.

O DOD queria que uma única linguagem atendesse à maioria de suas necessidades.

Seu nome foi dado em homenagem a Lady Ada Lovelace, filha do poeta Lord Byron.

Lady Lovelace tem o mérito de ter escrito o primeiro programa de computador do mundo no começo do século 19.

Suporta **multitarefa** — que permite que os programadores especifiquem quantas atividades em um programa devem ocorrer em paralelo.

O Java, por meio de uma técnica chamada *multithreading*, também permite que os programadores escrevam programas com atividades paralelas.

Java™



COMO PROGRAMAR

8ª edição

1.12 BASIC, Visual Basic, Visual C++, C# e .NET

- **BASIC** (Beginner's All-Purpose Symbolic Instruction Code)
Desenvolvida no Dartmouth College em meados da década de 1960 como um meio de escrever programas simples.
Utilizada para familiarizar os iniciantes com as técnicas de programação.
- **Visual Basic** da Microsoft
Introduzida no início da década de 1990 para simplificar o desenvolvimento de aplicativos para a Microsoft Windows.
- As mais recentes ferramentas de desenvolvimento da Microsoft
Estratégia de escopo corporativo para integrar a Internet e a Web em aplicativos de computador.
Implementada na **plataforma .NET**
As três principais linguagens de programação: Visual Basic (baseada no BASIC original), **Visual C++** (baseada no C++) e **C#** (baseada no C++ e no Java, e desenvolvida especificamente para a plataforma .NET).

Java™



COMO PROGRAMAR

8ª edição

1.13 Ambiente de desenvolvimento Java típico

- Desenvolvimento de programa e ciclo de execução em Java (ilustrado na Figura 1.1).

Java™



COMO PROGRAMAR

8ª edição

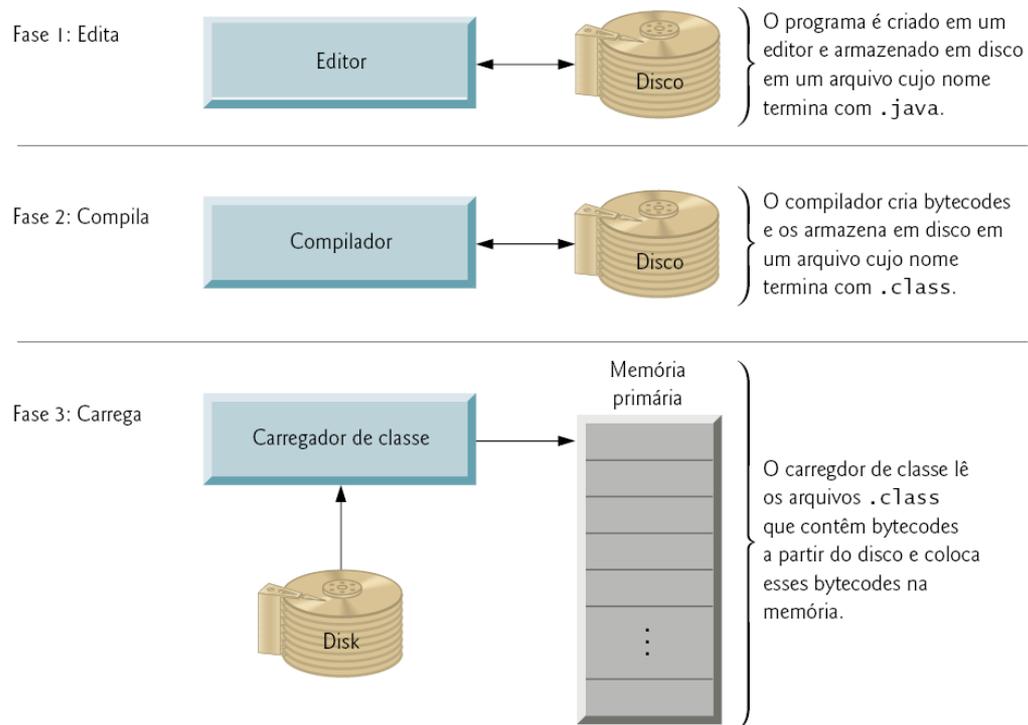


Figura 1.1 | Ambiente típico de desenvolvimento Java. (Parte 1 de 2.)

Java™



COMO PROGRAMAR

8ª edição

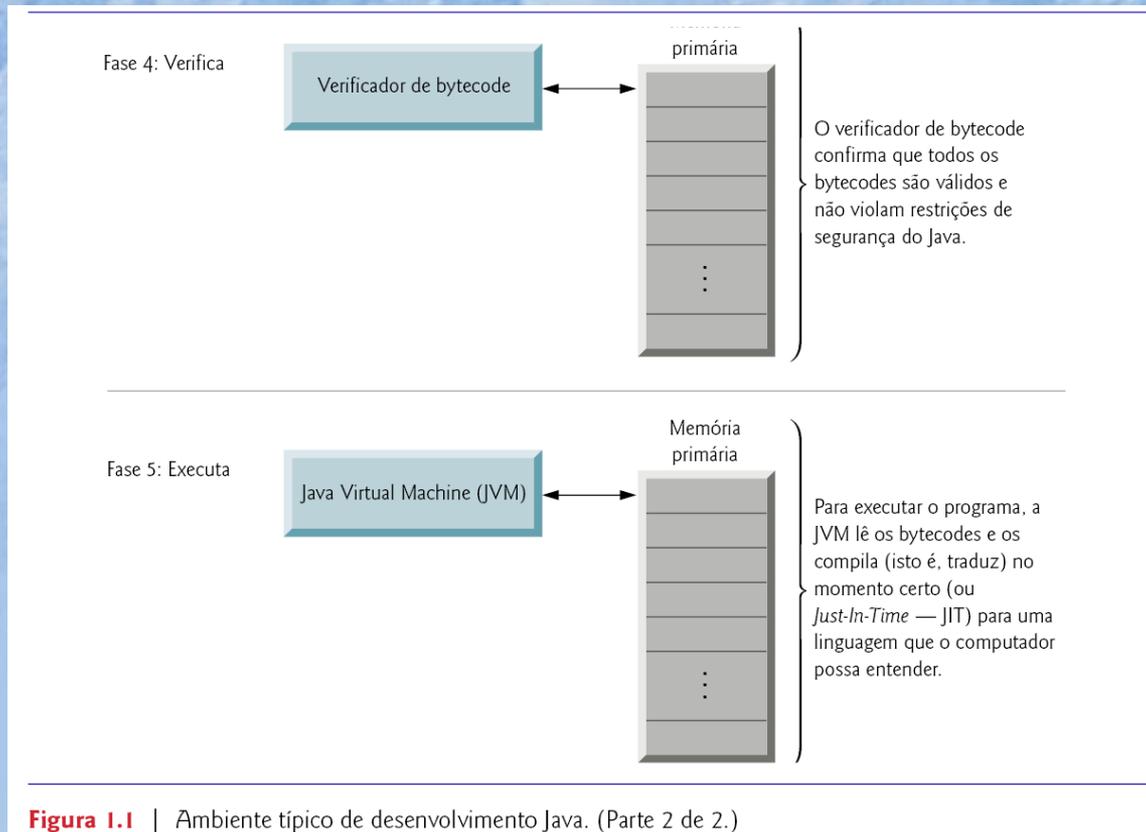


Figura 1.1 | Ambiente típico de desenvolvimento Java. (Parte 2 de 2.)

Java™



COMO PROGRAMAR

8ª edição

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

Java™



COMO PROGRAMAR

8ª edição

- Discutimos essas fases no contexto do J2SE Development Kit (JDK) versão 5.0 da Sun Microsystems, Inc.
- Faça o download do JDK mais recente e sua documentação a partir de java.sun.com/javase/downloads/.
- *Siga cuidadosamente as instruções de instalação do JDK fornecidas na Seção “Antes de você começar” deste livro a fim de certificar-se de que você configura seu computador apropriadamente para compilar e executar programas Java.*
- Sun’s New to Java Center em:
java.sun.com/new2java/

Java™



COMO PROGRAMAR

8ª edição

- 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 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).

Java™



COMO PROGRAMAR

8ª edição

- **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)

Java™



COMO PROGRAMAR

8ª edição

- 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.

Java™



COMO PROGRAMAR

8ª edição

- 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.

Java™



COMO PROGRAMAR

8ª edição

- 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
```

Java™



COMO PROGRAMAR

8ª edição

- 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.

Java™



COMO PROGRAMAR

8ª edição

- 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).

Java™



COMO PROGRAMAR

8ª edição

- 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.

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.

Java™



COMO PROGRAMAR

8ª edição



Erro comum de programação 1.1

*Erros como divisão por zero ocorrem durante a execução de um programa, por isso são chamados de **erros de runtime** ou **erros de tempo de execução**. **Erros de tempo de execução fatais** fazem com que os programas sejam imediatamente encerrados sem terem realizado seu trabalho com sucesso. **Erros de tempo de execução não fatais** permitem que os programas executem até sua conclusão, produzindo frequentemente resultados incorretos.*

Java™



COMO PROGRAMAR

8ª edição

1.14 Notas sobre o Java e Java Como Programar, Oitava Edição

- Uma versão baseada na Web da documentação da Java API
java.sun.com/javase/6/docs/api/index.html
- Baixe essa documentação para seu computador
java.sun.com/javase/downloads/
- Detalhes técnicos adicionais sobre muitos aspectos do desenvolvimento Java.
java.sun.com/reference/docs/index.html

Java™



COMO PROGRAMAR

8ª edição



Boa prática de programação 1.1

Escreva seus programas Java de maneira simples e direta. Isso é às vezes chamado de KIS (“Keep It Simple”, ou seja, “mantenha a coisa simples”). Não “estenda” a linguagem tentando usos estranhos.

Java™



COMO PROGRAMAR

8ª edição



Dica de portabilidade 1.2

Embora seja mais fácil escrever programas portáveis em Java do que na maioria das outras linguagens de programação, diferenças entre compiladores, JVMs e computadores podem tornar a portabilidade difícil de alcançar. Simplesmente escrever programas em Java não garante portabilidade.

Java™



COMO PROGRAMAR

8ª edição



Dica de prevenção de erro 1.1

Sempre teste os programas Java em todos os sistemas em que você quiser executá-los, para assegurar que funcionarão corretamente para o público-alvo.

Java™



COMO PROGRAMAR

8ª edição



Boa prática de programação 1.2

Leia a documentação da versão do Java que você está utilizando. Consulte-a frequentemente para certificar-se de que você está ciente da rica coleção de recursos Java e de que está utilizando esses recursos corretamente.

Java™



COMO PROGRAMAR

8ª edição

1.15 Testando um aplicativo Java

- Verificando sua configuração. Leia a Seção “Antes de você começar” do livro para confirmar se você configurou o Java corretamente no seu computador e se copiou os exemplos do livro para sua unidade de disco rígido.

Java™



COMO PROGRAMAR

8ª edição

Usando o comando `cd` para mudar de diretório

Localização dos arquivos do aplicativo de ATM

```
Administrator: Command Prompt
C:\> cd C:\examples\ch01\ATM
C:\examples\ch01\ATM>
```

Figura 1.2 | Abrindo um **Prompt de Comando** e mudando de diretório.

Java™



COMO PROGRAMAR

8ª edição

```
Administrator: Command Prompt
C:\>cd C:\examples\ch01\ATM
C:\examples\ch01\ATM>java ATMCaseStudy
```

Figura 1.3 | Utilizando o comando java para executar o aplicativo ATM.

Java™



COMO PROGRAMAR

8ª edição

Mensagens de boas-vindas do aplicativo de ATM Prompt para inserir o número da conta

```
Administrator: Command Prompt - java ATMCaseStudy
C:\examples\ch01\ATM>java ATMCaseStudy
Welcome!
Please enter your account number: 12345
```

Figura 1.4 | Solicitando o número de uma conta ao usuário.

Java™



COMO PROGRAMAR

8ª edição

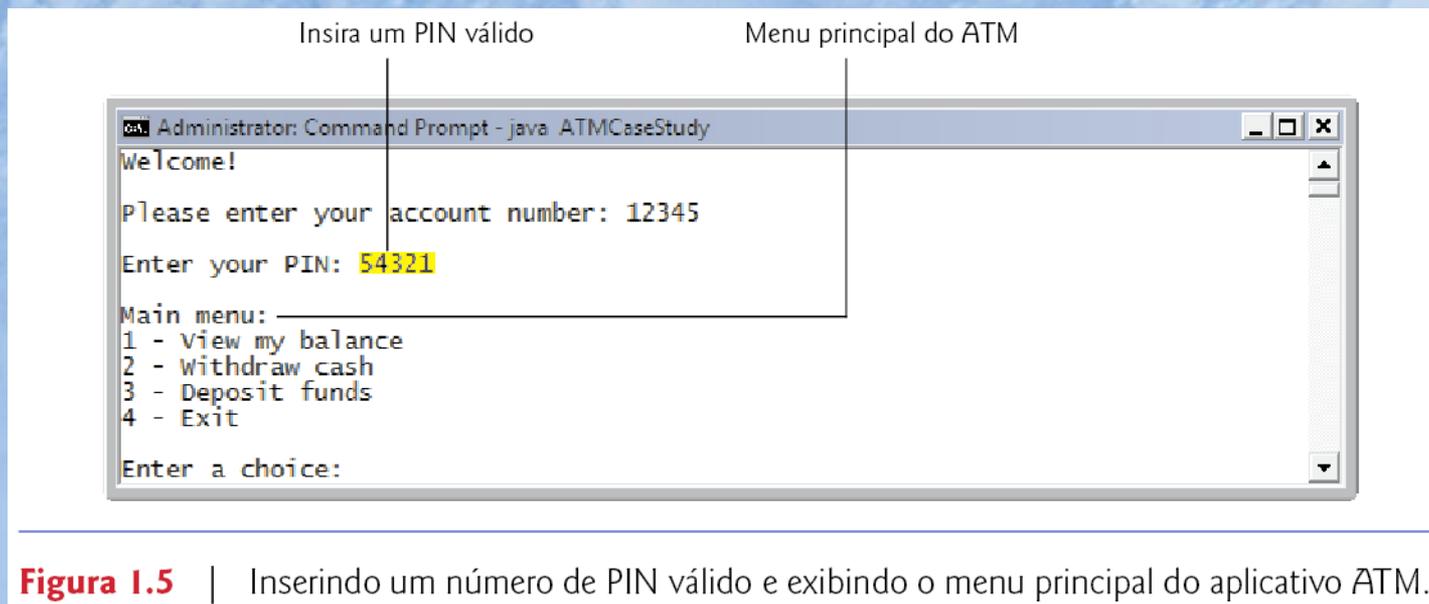


Figura 1.5 | Inserindo um número de PIN válido e exibindo o menu principal do aplicativo ATM.

Java™



COMO PROGRAMAR

8ª edição

Informações sobre o saldo da conta

```
Administrator: Command Prompt - java ATMCaseStudy
Enter a choice: 1
Balance Information:
- Available balance: $1,000.00
- Total balance: $1,200.00
Main menu:
1 - View my balance
2 - Withdraw cash
3 - Deposit funds
4 - Exit
Enter a choice:
```

Figura 1.6 | O aplicativo ATM exibindo as informações de saldo da conta do usuário.

Java™



COMO PROGRAMAR

8ª edição

Menu de saque do ATM

```
Administrator: Command Prompt - java ATMCaseStudy
Enter a choice: 2
Withdrawal Menu:
1 - $20
2 - $40
3 - $60
4 - $100
5 - $200
6 - Cancel transaction
Choose a withdrawal amount: 4
Please take your cash now.
Main menu:
1 - View my balance
2 - Withdraw cash
3 - Deposit funds
4 - Exit
Enter a choice:
```

Figura 1.7 | Retirando dinheiro da conta e retornando ao menu principal.

Java™



COMO PROGRAMAR

8ª edição

Confirmando as informações de atualização
do saldo da conta após transação de saque

```
Administrator: Command Prompt - java ATMCaseStudy
Enter a choice: 1
Balance Information:
- Available balance: $900.00
- Total balance: $1,100.00
Main menu:
1 - View my balance
2 - Withdraw cash
3 - Deposit funds
4 - Exit
Enter a choice:
```

Figura 1.8 | Verificando o novo saldo.

Java™



COMO PROGRAMAR

8ª edição

Mensagem de despedida do ATM

Prompt de número da conta para o próximo usuário

```
Administrator: Command Prompt - java ATMCaseStudy
Enter a choice: 4
Exiting the system...
Thank you! Goodbye! _____
welcome!
Please enter your account number: _____
```

Figura 1.9 | Concluindo uma sessão de transação de um ATM.

Java™



COMO PROGRAMAR

8ª edição

Nome do aplicativo	Localização do capítulo	Comandos a serem executados
Tic-Tac-Toe (jogo da velha)	Capítulo 8	<code>cd C:\examples\ch01\Tic-Tac-Toe java TicTacToeTest</code>
Jogo de adivinhação	Capítulo 14	<code>cd C:\examples\ch01\GuessGame java GuessGame</code>
Animador de logotipo	Capítulo 24	<code>cd C:\examples\ch01\LogoAnimator java LogoAnimator</code>
Bola que rebate	Capítulo 26	<code>cd C:\examples\ch01\BouncingBall java BouncingBall</code>

Figura 1.10 | Alguns dos aplicativos sugeridos para um test-drive neste livro.

Java™



COMO PROGRAMAR

8ª edição

1.16 Estudo de caso de engenharia de software: introdução à tecnologia de objetos e à UML

- **Unified Modeling Language™ (UML™)**

Uma linguagem gráfica que permite que as pessoas que projetam sistemas de software utilizem uma notação padrão da indústria para representá-las.

- Os Capítulos 12–13 apresentam um projeto orientado a objetos e a implementação do software de um caixa eletrônico (*automated teller machine* — ATM) simples.

Java™



COMO PROGRAMAR

8ª edição

1.16 Estudo de caso de engenharia de software (Cont.)

- **Objetos**

Componentes reutilizáveis de software que modelam itens do mundo real.

Os humanos pensam em termos de objetos. Pessoas, animais, plantas, carros etc.

Têm **atributos** (por exemplo, tamanho, forma, cor e peso)

Exibem **comportamentos**; por exemplo, uma bola rola, pula, enche-se e murcha; um bebê chora, dorme, engatinha, anda e pisca; um carro acelera, freia e volta; uma toalha absorve água.

Java™



COMO PROGRAMAR

8ª edição

- **Projeto orientado a objetos** (*object-oriented design* – OOD).

Modela software em termos semelhantes àqueles que as pessoas utilizam para descrever objetos do mundo real.

Relacionamentos de **classe**.

Relacionamentos de herança.

Modela comunicação entre objetos (via mensagens).

Encapsula atributos e **operações** (comportamentos) em objetos.

Ocultamento de informações

Os objetos podem saber comunicar-se entre si por meio de **interfaces**, bem-definidas, mas normalmente não são autorizados a saber como outros objetos são implementados.

Java™



COMO PROGRAMAR

8ª edição

- **Linguagens orientadas a objetos.**

A programação nessa linguagem, chamada **programação orientada a objetos (object-oriented programming – OOP)**.

Permite implementar um projeto orientado a objetos como um sistema funcional.

O Java é orientado a objetos.

- **Foco na criação de classes.**

Cada classe contém campos e o conjunto de métodos que manipulam os campos e fornecem serviços aos **clientes** (isto é, outras classes que utilizam a classe).

Os programadores utilizam classes existentes como blocos de construção para construir novas classes.

- **As classes estão para os objetos assim como as plantas arquitetônicas estão para as casas.**

- **Associações** — relacionamentos entre classes.

- Os pacotes também facilitam a **reutilização**.

Java™



COMO PROGRAMAR

8ª edição

- **Análise e design orientado a objetos (*object-oriented analysis and design* – OOAD)**
Analisar os requisitos do seu projeto (isto é, determinar *o que o sistema deve fazer*) e desenvolver um **design** que os cumpra (i.e., *decidir como o sistema deve fazê-lo*).
- Unified Modeling Language (UML) — Uma linguagem gráfica simples para comunicar os resultados de *qualquer processo OOAD* tornou-se amplamente utilizada.
- A UML é o esquema de representação gráfica mais amplamente utilizado para modelar sistemas orientados a objetos.

Java™



COMO PROGRAMAR

8ª edição

1.17 Web 2.0

- A Web explodiu na metade da década de 1990.
- O estouro da bolha “ponto com” trouxe tempos difíceis no início dos anos 2000.
- O ressurgimento começou em 2004 com a **Web 2.0**.
- O Google é amplamente considerado a empresa símbolo da Web 2.0.
- Os **serviços Web** viabilizaram a Web 2.0.

Java™



COMO PROGRAMAR

8ª edição

- **Ajax**

O Ajax é uma das principais tecnologias da Web 2.0.

Ajuda aplicativos baseados na Internet a ter um desempenho comparável ao dos aplicativos desktop.

- **Blogs**

Websites são como diários on-line.

Blogosfera — A coleção de todos os blogs e a comunidade que os escreve.

- **RSS feeds**

Permitem que sites enviem informações aos assinantes.

Comumente utilizados para transmitir os blogs.

Java™



COMO PROGRAMAR

8ª edição

- **Web 3.0**

Outro nome para a última geração da Web, também chamada **Web semântica**.

A Web 1.0 era quase puramente baseada em HTML.

A Web 2.0 cada vez mais utiliza a XML, especialmente em tecnologias como RSS feeds e serviços Web.

A Web 3.0 utilizará intensamente a XML, criando “uma teia de significados”.

Java™



COMO PROGRAMAR

8ª edição

1.18 Tecnologias de software

- O **Agile Software Development** é um grupo de metodologias que tenta implementar softwares rapidamente, utilizando menos recursos do que as metodologias anteriores.
- A **refatoração** envolve a reformulação do código para torná-lo mais claro e mais fácil de manter e, ao mesmo tempo, preservar sua funcionalidade.
- **Padrões de design** são arquiteturas testadas para construir softwares orientados a objetos flexíveis e que podem ser mantidos.
- **Programação de jogos** — O negócio de jogos de computador é maior do que o negócio de filmes inéditos. Cursos universitários e mesmo disciplinas agora dedicam-se às técnicas sofisticadas utilizadas na programação de jogos.

Java™



COMO PROGRAMAR

8ª edição

- **Software de código-fonte aberto**

Um estilo de desenvolvimento de software em que pessoas e empresas contribuem com seus esforços para desenvolver, manter e aprimorar os softwares em troca do direito de utilizar esses softwares para objetivos próprios, tipicamente gratuitamente.

A Sun tornou livre e aberta sua implementação do Java Development Kit.

Eclipse Foundation (o Eclipse IDE é popular para desenvolvimento de software Java).

Mozilla Foundation (criadores do navegador da Web Firefox)

Apache Software Foundation (criadores do servidor Web Apache).

SourceForge (que fornece as ferramentas para gerenciar projetos de código-fonte aberto e atualmente tem mais de 100.000 projetos de código-fonte aberto em desenvolvimento).

Java™



COMO PROGRAMAR

8ª edição

- **Linux**

Um sistema operacional de código-fonte aberto.

Um dos maiores sucessos do movimento do código-fonte aberto.

- **MySQL**

Um sistema de gerenciamento de bancos de dados de código-fonte aberto.

- **PHP**

A mais popular linguagem para criação de scripts de servidor com código-fonte aberto para desenvolver aplicativos baseados na Internet.

- **LAMP**

Um acrônimo para o conjunto de tecnologias de código-fonte aberto que muitos desenvolvedores utilizavam para construir aplicativos Web — ele significa Linux, Apache, MySQL e PHP (ou Perl ou Python — duas outras linguagens utilizadas para objetivos semelhantes).

Java™



COMO PROGRAMAR

8ª edição

- **Ruby on Rails**

Combina a linguagem de criação de scripts Ruby com a estrutura de aplicativo Web do Rails desenvolvida pela empresa 37Signals.

Muitos desenvolvedores Ruby on Rails informaram ganhos de produtividade significativos em relação ao uso de outras linguagens ao desenvolver aplicativos Web que utilizam intensamente banco de dados.

Java™



COMO PROGRAMAR

8ª edição

1.18 Tecnologias de software (Cont.)

- **Software as a Service (SAAS)**

Software que roda em servidores via Internet.

Quando esse servidor é atualizado, todos os clientes no mundo inteiro veem as novas capacidades; nenhuma instalação local é necessária.

Você acessa o serviço via um navegador.

Você pode executar os mesmos aplicativos em diferentes tipos de computadores a partir de qualquer lugar no mundo.