

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

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.~~

utilizando gramáticas mais próximas à linguagem humana

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.

Estas observações do
Deitel merecem discussão

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++.

└ principalmente C

Java™



COMO PROGRAMAR

8ª edição

- C++

Uma extensão do C.

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

Fornecer 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

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

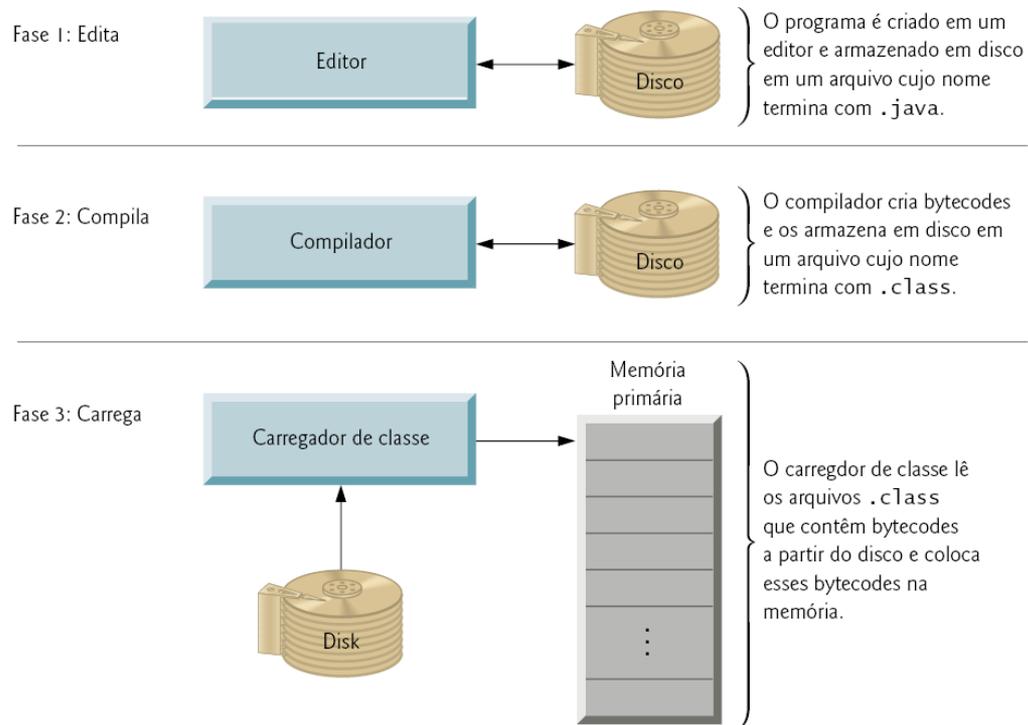


Figura 1.1 | Ambiente típico de desenvolvimento Java. (Parte 1 de 2.)

Java™



COMO PROGRAMAR

8ª edição

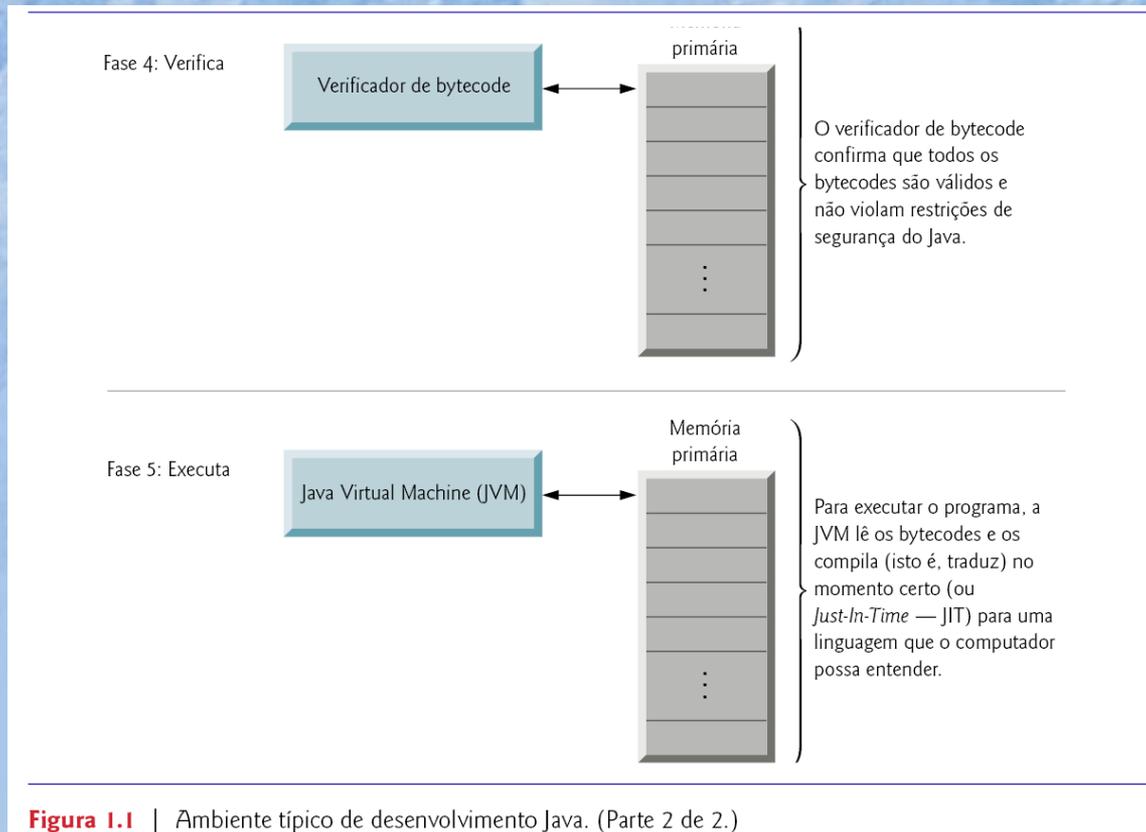


Figura 1.1 | Ambiente típico de desenvolvimento Java. (Parte 2 de 2.)

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**. **gedit, kate**

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

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**.