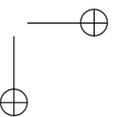
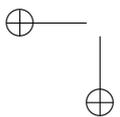
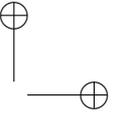
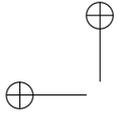
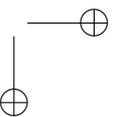
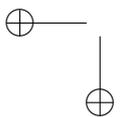
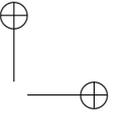
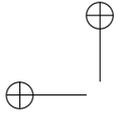




# Introdução à Programação Orientada a Objetos usando Java

Rafael Santos





© 2013, Elsevier Editora Ltda.

Todos os direitos reservados e protegidos pela Lei nº 9.610, de 19/02/1998. Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

*Revisão: Hugo de Lima Correa*  
*Editoração Eletrônica: Rafael Santos*

Elsevier Editora Ltda.  
Conhecimento sem Fronteiras  
Rua Sete de Setembro, 111 – 16º andar  
20050-006 – Centro – Rio de Janeiro – RJ – Brasil

Rua Quintana, 753 – 8º andar  
04569-011 – Brooklin – São Paulo – SP

Serviço de Atendimento ao Cliente  
0800-0265340  
sac@elsevier.com.br

ISBN 978-85-352-7433-2  
ISBN e-book: 978-85-352-7434-9

**Nota:** Muito zelo e técnica foram empregados na edição desta obra. No entanto, podem ocorrer erros de digitação, impressão ou dúvida conceitual. Em qualquer das hipóteses, solicitamos a comunicação ao nosso Serviço de Atendimento ao Cliente, para que possamos esclarecer ou encaminhar a questão.

Nem a editora nem o autor assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, originados do uso desta publicação.

CIP-BRASIL. CATALOGAÇÃO-NA-FONTE  
SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ

---

S238i  
2. ed.

Santos, Rafael  
Introdução à programação orientada a objetos usando Java /  
Rafael Santos. - 2. ed. - Rio de Janeiro : Elsevier, 2013.  
336 p. : il. ; 24 cm.

Inclui bibliografia  
ISBN 978-85-352-7433-2

1. Java (Linguagem de programação de computador). 2.  
Programação orientada a objetos (Computação). I. Título.

13-02857

CDD: 005.133  
CDU: 004.43

---

# Prefácio para a segunda edição

## Sobre este livro

Este livro tem como finalidade possibilitar ao estudante o aprendizado dos conceitos básicos de programação orientada a objetos usando a linguagem Java, e adquirir conhecimentos sobre a linguagem que possibilitem estudos avançados sobre a mesma.

O livro parte do pressuposto de que o estudante conhece computadores como ferramentas de desenvolvimento de software, tem noções elementares de algoritmos e sabe o que é uma linguagem de programação, mas sem necessariamente possuir conhecimentos profundos de uma. O livro também pode ser usado por estudantes que já conheçam alguma linguagem de programação mas estejam interessados em aprender Java.

Ao escrever este livro, minha motivação principal foi apresentar os conceitos de programação orientada a objetos com clareza e simplicidade, mostrando exemplos e sugerindo problemas práticos e ilustrativos de conceitos e de algoritmos em geral. Muitos livros de programação orientada a objetos (abreviadamente, POO), especialmente usando a linguagem Java, concentram-se em aspectos visuais, deixando de lado a teoria e explicação dos detalhes que são necessárias para melhor compreensão das técnicas e conceitos. Na maioria dos casos, demonstrações das capacidades da linguagem são apresentadas desde cedo, sem as explicações necessárias para compreensão do que acontece nos programas-exemplo. Em contraste, este livro concentra-se nas técnicas de programação orientada a objetos, explicando a teoria e apresentando a prática de cada tópico. Espero que com isto os estudantes estejam mais confortáveis com os conceitos de classes e objetos quando forem utilizá-los em aplicações mais complexas.

Para reforçar e ilustrar os conceitos apresentados em cada capítulo, **162** listagens contendo classes, programas e trechos de código foram incluídas. O código nestas listagens é comentado detalhadamente, tanto para esclarecer os pontos mais importantes quanto para acostumar o leitor com a ideia de ler comentários em paralelo com código e também incentivá-lo a comentar código.

Cada capítulo conta com uma lista de exercícios, disponível no site de suporte, em <http://www.elsevier.com.br/rafaelsantos>. No site também podem ser obtidos material complementar a este livro, como erratas, listas de exercícios, material adicional, *links* úteis e informações para professores que adotarem o livro como texto em suas aulas.

## Sobre a organização deste livro

Cada capítulo deste livro cobre um tema específico. Os capítulos e um resumo de seu conteúdo são apresentados a seguir:

- **Capítulo 1: Introdução à Programação Orientada a Objetos** apresenta conceitos básicos de programação orientada a objetos, esclarecendo as diferenças e relações entre modelos, classes, objetos e instâncias. Neste capítulo, os conceitos apresentados são puramente teóricos, mas **essenciais** para a compreensão dos princípios básicos de orientação a objetos.
- **Capítulo 2: Criando Classes em Java** mostra como classes em Java são declaradas, e apresenta os tipos de dados e os mecanismos de criação de campos e métodos nas classes.
- **Capítulo 3: Criando Aplicações em Java** mostra como podemos criar aplicações ou programas em Java (classes que podem ser executadas pela máquina virtual Java), e como podemos usar classes criadas anteriormente nestes programas.
- **Capítulo 4: Construtores e Sobrecarga** apresenta *construtores*, métodos especiais que serão chamados quando objetos das classes que os contêm forem criados. O capítulo também descreve como fazer a sobrecarga de métodos, que permite que vários métodos com o mesmo nome mas diferentes funções possam ser declarados e usados.
- **Capítulo 5: Campos e Métodos Estáticos** mostra como podemos criar métodos e campos estáticos em aplicações ou outras classes em Java. Métodos e campos estáticos não precisam ter suas classes instanciadas para ser usados.
- **Capítulo 6: Estruturas de Decisão e Controle - Condicionais** apresenta as estruturas de execução de trechos do código baseadas em condições, juntamente com uma introdução aos operadores lógicos de Java.
- **Capítulo 7: Estruturas de Decisão e Controle - Repetição** apresenta as estruturas de repetição de Java baseadas em condições, juntamente com noções sobre recursão.
- **Capítulo 8: Reutilização de Classes** mostra dois mecanismos de Programação Orientada a Objetos que permitem o uso de uma classe já existente para criação de uma classe derivada, mais especializada.
- **Capítulo 9: Classes Abstratas e Interfaces** apresenta os conceitos de *classes abstratas* e *interfaces*, que também servem como mecanismos de herança de Java com características especiais.
- **Capítulo 10: Arrays em Java** apresenta os mecanismos de criação e uso de *arrays* (vetores de dados) unidimensionais e multidimensionais em Java, assim como técnicas de Programação Orientada a Objetos para a criação de classes que implementam arrays de forma robusta.



- **Capítulo 11: Classes para Manipulação de Strings** apresenta e demonstra o uso de duas classes para processamento de cadeias de caracteres: **String** e **StringBuilder**.
- **Capítulo 12: Coleções de Objetos** apresenta algumas classes que implementam estruturas de dados simples em Java e suas aplicações.

Além destes capítulos o livro contém o **Apêndice A: A classe Keyboard**, que apresenta a classe **Keyboard**, que pode ser usada para facilitar a leitura de valores de tipos nativos e instâncias da classe **String** a partir do teclado, adicionando flexibilidade aos programas que não usam interfaces gráficas.

## Convenções usadas neste livro

Para destacar alguns conceitos apresentados neste livro, fontes diferentes serão usadas. Nomes de classes, métodos, pacotes, comandos etc. serão mostrados em fonte **proporcional**. Conceitos apresentados pela primeira vez ou que mereçam destaque e palavras que não sejam parte do português corrente serão mostrados em fonte *itálica*.

Adicionalmente, as listagens incluídas no livro serão *estilizadas*, isto é, palavras-chave e partes do código como comentários serão mostradas em tonalidades ou estilos de texto diferentes para facilitar a leitura. As linhas dos programas também serão numeradas para fácil referência.

## Por que Java?

Existem diversas linguagens de programação orientadas a objeto, cada uma com diferentes características e apelos de mercado, educacionais ou acadêmicos. Nesta seção algumas das razões da escolha da linguagem Java serão apresentadas.

**Java é obrigatoriamente orientada a objetos.** Em Java, **todas** as variáveis e métodos devem estar organizados em classes, forçando o uso de orientação a objetos até mesmo em tarefas simples. Dessa forma, o estudante de programação orientada a objetos que esteja usando Java estará usando mais as técnicas de POO.

**Java é simples.** A estrutura de programas e classes em Java segue a organização de linguagens tradicionais como C e C++, mas sem elementos que tornam programas e programação mais complexos. Após o aprendizado dos conceitos básicos de programação orientada a objetos, o estudante da linguagem pode começar a criar aplicativos úteis de forma mais rápida.

A simplicidade se reflete também na maneira com que arquivos contendo programas em Java são compilados e executados: se as recomendações básicas forem seguidas, o compilador se encarregará de compilar todas as classes necessárias em uma aplicação automaticamente, sem necessidade de arquivos adicionais de configuração e inclusão de bibliotecas.

**Java é portátil.** O código-fonte de um programa ou classe em Java pode ser compilado em qualquer computador, usando qualquer sistema operacional, contanto que este tenha uma máquina virtual Java adequada – a máquina virtual é o ambiente de execução das aplicações em Java. As classes criadas podem ser copiadas e executadas em qualquer computador nas mesmas condições, aumentando a utilidade da linguagem através da independência de plataformas, contanto que versões compatíveis da máquina virtual sejam usadas.

**Java é flexível.** Conhecimentos sobre a linguagem podem ser usados para desenvolver aplicações que são executadas em terminais, com ou sem interfaces gráficas, aplicações que são executadas em um servidor com navegadores como cliente, e aplicações embarcadas em *smartphones* e outros dispositivos móveis. O conhecimento básico sobre a linguagem habilita o programador a aprender os conceitos específicos das diversas plataformas mais rapidamente.

**Java é robusta.** Administração de memória (alocação e liberação) e o uso de ponteiros, duas das fontes de erros e *bugs* mais frequentes em programas em C e C++, são administrados internamente na linguagem, de forma transparente para o programador. De maneira geral, programas em Java têm restrições no acesso à memória que resultam em maior segurança para os programas sem diminuir a utilidade dos mesmos.

Java também tem um poderoso mecanismo de exceções que permite melhor tratamento de erros em tempo de execução dos programas.

**Java tem bibliotecas prontas para diversas aplicações.** As bibliotecas de classes da máquina virtual Java contêm várias classes que implementam diversos mecanismos de entrada e saída, acesso à Internet, manipulação de strings (cadeias de caracteres) em alto nível, poderosas estruturas de dados, utilitários diversos e um conjunto completo de classes para implementação de interfaces gráficas.

Vale a pena lembrar que estas bibliotecas são padrão de Java – qualquer máquina virtual Java permite o uso destas bibliotecas, sem a necessidade de instalar pacotes adicionais – e que, mesmo que o compilador usado não tenha interface gráfica similar à de linguagens visuais, os programas criados com este compilador podem ter interfaces gráficas complexas.

Além das bibliotecas padrão da máquina virtual Java é possível usar uma ampla gama de bibliotecas adicionais que permitem o desenvolvimento de aplicações complexas.

## Categorias de softwares em Java

Existem basicamente três categorias de software que podem ser criados em Java e que serão apresentadas neste livro:

**Classes para representação de modelos.** Classes são usadas para representar, em linguagem de programação, modelos e abstrações de dados como os mostrados no capítulo 1. Classes para representação de modelos não podem ser executadas diretamente, mas instâncias destas classes podem ser usadas dentro de aplicações



e *applets*. Detalhes e mecanismos de criação de classes em Java serão vistos no capítulo 2.

**Aplicações ou Programas.** Aplicações ou programas podem ser executadas por um sistema operacional, podendo envolver ou não maior interação com o usuário, e podendo ou não ter uma interface gráfica. Informações sobre a criação de aplicações em Java (sem uso de interface gráfica) serão mostradas no capítulo 3. Aplicações que são executadas em um servidor (*servlets*, *Java Server Pages*, *web services* e similares) não serão cobertas neste livro.

**Classes como conjuntos de rotinas.** Classes podem conter somente métodos (rotinas) ao invés de representar dados. Estas classes funcionam como *bibliotecas* de métodos ou funções que tem algo em comum. Detalhes sobre a criação deste tipo de classes serão vistos no capítulo 5.

Esta lista não considera aplicações com interfaces gráficas, aplicações que são executadas em um servidor, aplicações embarcadas ou para dispositivos móveis como *smartphones*, que não são cobertas neste livro.

## Modificações na Segunda Edição

Todos os capítulos do livro foram revisados para a segunda edição, com melhorias no conteúdo e apresentação e correções no texto. Alguns exemplos foram melhorados para ficar mais realistas, enquanto outros foram removidos por serem redundantes ou não corresponder adequadamente a novas características da linguagem.

Alguns conceitos que não foram incluídos na primeira edição o foram na segunda: enumeradores (seção 5.6), laços para iteração em arrays e coleções (seção 10.2.1), argumentos variáveis (seção 10.6), tipos genéricos (seção 12.1.1) e *autoboxing* (seção 12.1.2).

O capítulo sobre pacotes foi removido, e seu conteúdo apresentado no capítulo 2. Classes foram organizadas em pacotes correspondentes aos capítulos do livro, e são declaradas como públicas logo desde os primeiros exemplos.

As listagens foram separadas das figuras, com um índice à parte. O código também foi revisado para melhor padronização.

Alguns apêndices foram removidos do texto, por tratar de conceitos tecnológicos que mudam muito com o tempo (por exemplo, ambientes integrados de desenvolvimento). O conteúdo destes apêndices será apresentado em material à parte, que será disponibilizado no site de suporte.

## Dedicatória e Agradecimentos

Este livro é dedicado a minha esposa Yukari, pela enorme paciência e suporte, e a nossos filhos Miyuki e Hiroshi, pela constante alegria que proporcionam. Dedico também o livro aos meus pais, Elias e Mariza, pelas várias lições de vida.

Sou grato aos estudantes dos cursos da Faculdade de Ciência da Computação da Universidade do Vale do Paraíba (Univap) e do antigo IBTA (Instituto Brasileiro de Tecnologia Avançada) que contribuíram com ideias e perguntas que resultaram em exemplos e exercícios, e em especial, aos estudantes que participaram na revisão do livro.

Sou também grato a vários colegas da Univap, IBTA e INPE (Instituto Nacional de Pesquisas Espaciais) por sugestões e esclarecimentos necessários para alguns exercícios, à professora Claudia Maria Bauzer Medeiros da Unicamp, por várias sugestões para o texto, e aos editores e funcionários da Elsevier, por acreditarem na ideia apesar de todos os atrasos que causei.

# Sumário

<b>Prefácio para a segunda edição</b>	<b>v</b>
<b>Sumário</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xiv</b>
<b>Lista de Listagens</b>	<b>xv</b>
<b>Lista de Tabelas</b>	<b>xxii</b>
<b>1 Introdução à programação orientada a objetos</b>	<b>1</b>
1.1 O que é um programa de computador? . . . . .	1
1.2 O que são modelos? . . . . .	2
1.3 O que é programação orientada a objetos? . . . . .	5
1.4 Encapsulamento . . . . .	6
1.5 Mais exemplos de modelos . . . . .	7
1.5.1 Exemplo 1: Uma lâmpada incandescente . . . . .	8
1.5.2 Exemplo 2: Uma conta bancária simplificada . . . . .	10
1.5.3 Exemplo 3: Uma data . . . . .	12
1.5.4 Exemplo 4: Um registro acadêmico de aluno . . . . .	14
1.6 Classes, objetos, instâncias, referências, atributos e métodos . . . . .	16
1.7 Orientação a objetos é necessária? . . . . .	18
1.8 Papéis de programadores que utilizam orientação a objetos . . . . .	18
<b>2 Criando classes em Java</b>	<b>20</b>
2.1 Introdução . . . . .	20
2.2 Sintaxe básica para a criação de classes em Java . . . . .	20
2.3 Atributos de classes em Java . . . . .	23
2.3.1 Usando dados nativos em Java . . . . .	24
2.3.2 Declarando atributos em classes em Java . . . . .	26
2.3.3 Que tipos de dados devem ser usados? . . . . .	28
2.4 Métodos em classes em Java . . . . .	29
2.5 Escopo . . . . .	33
2.6 Modificadores de acesso . . . . .	34

<b>3</b>	<b>Criando aplicações em Java</b>	<b>43</b>
3.1	Introdução . . . . .	43
3.2	Criando aplicações em Java . . . . .	43
3.3	A palavra-chave <b>new</b> . . . . .	47
3.4	A palavra-chave <b>null</b> . . . . .	53
3.5	Escrevendo classes e aplicações mais interativas . . . . .	55
<b>4</b>	<b>Construtores e sobrecarga</b>	<b>57</b>
4.1	Introdução . . . . .	57
4.2	O que são construtores? . . . . .	60
4.3	Sobrecarga de métodos . . . . .	64
4.3.1	A palavra-chave <b>this</b> . . . . .	69
4.3.2	Cuidados com sobrecarga de métodos . . . . .	72
<b>5</b>	<b>Atributos e métodos estáticos</b>	<b>75</b>
5.1	Introdução . . . . .	75
5.2	Atributos estáticos em classes . . . . .	75
5.3	Métodos estáticos em classes . . . . .	82
5.4	Atributos e métodos estáticos em aplicações . . . . .	85
5.5	Fábricas de instâncias . . . . .	86
5.6	Enumeradores . . . . .	89
<b>6</b>	<b>Estruturas de decisão e controle – Condicionais</b>	<b>94</b>
6.1	Introdução . . . . .	94
6.2	Operadores lógicos em Java . . . . .	95
6.3	Estruturas de decisão e execução seletiva . . . . .	100
6.3.1	As instruções <b>if-else</b> . . . . .	100
6.3.2	O operador condicional ? . . . . .	104
6.3.3	A instrução <b>switch</b> . . . . .	105
<b>7</b>	<b>Estruturas de decisão e controle – Repetição</b>	<b>110</b>
7.1	Estruturas de repetição ou iteração . . . . .	110
7.1.1	Contadores . . . . .	111
7.1.2	O laço <b>while</b> . . . . .	113
7.1.3	O laço <b>do-while</b> . . . . .	117
7.1.4	O laço <b>for</b> . . . . .	121
7.2	Introdução à recursão . . . . .	125
<b>8</b>	<b>Reutilização de classes</b>	<b>134</b>
8.1	Introdução . . . . .	134
8.2	Delegação ou composição . . . . .	135
8.2.1	Delegação e modificadores de acesso . . . . .	141
8.2.2	Delegação e construtores . . . . .	145



8.3	Herança . . . . .	145
8.3.1	A palavra-chave <b>super</b> . . . . .	152
8.3.2	Sobreposição e ocultação . . . . .	160
8.3.3	O modificador <b>protected</b> . . . . .	165
8.3.4	Polimorfismo . . . . .	170
8.4	Comentários finais sobre composição e herança . . . . .	178
<b>9</b>	<b>Classes abstratas e interfaces</b>	<b>179</b>
9.1	Introdução . . . . .	179
9.2	Classes e métodos abstratos . . . . .	180
9.3	Interfaces . . . . .	185
9.4	Herança múltipla usando interfaces . . . . .	194
9.4.1	Conflitos de herança múltipla . . . . .	203
<b>10</b>	<b>Arrays em Java</b>	<b>208</b>
10.1	Introdução . . . . .	208
10.2	Arrays unidimensionais . . . . .	209
10.2.1	Laços para iteração em arrays . . . . .	216
10.3	Arrays de instâncias de classes . . . . .	217
10.4	Arrays multidimensionais . . . . .	222
10.4.1	Arrays irregulares . . . . .	227
10.5	Processando argumentos da linha de comando . . . . .	230
10.6	Argumentos variáveis . . . . .	233
<b>11</b>	<b>Classes para manipulação de strings</b>	<b>236</b>
11.1	Introdução . . . . .	236
11.2	A classe <b>String</b> . . . . .	236
11.2.1	Construindo instâncias da classe <b>String</b> . . . . .	237
11.2.2	Métodos básicos da classe <b>String</b> . . . . .	238
11.2.3	Métodos para comparação de instâncias da classe <b>String</b> . . . . .	241
11.2.4	Métodos para procura em instâncias da classe <b>String</b> . . . . .	243
11.2.5	Métodos para modificação de instâncias da classe <b>String</b> . . . . .	244
11.2.6	Métodos de conversão da classe <b>String</b> . . . . .	246
11.2.7	Separando uma string em componentes . . . . .	246
11.3	A classe <b>StringBuilder</b> . . . . .	248
11.3.1	Métodos básicos da classe <b>StringBuilder</b> . . . . .	248
<b>12</b>	<b>Coleções de objetos</b>	<b>252</b>
12.1	Introdução . . . . .	252
12.1.1	Tipos Genéricos . . . . .	253
12.1.2	<i>Autoboxing</i> . . . . .	254
12.2	Conjuntos ( <i>sets</i> ) de objetos . . . . .	255
12.2.1	Diferenças adicionais entre as classes <b>TreeSet</b> e <b>HashSet</b> . . . . .	263

12.3	Listas de objetos . . . . .	264
12.3.1	Pilhas e filas implementadas como instâncias de listas . . . . .	271
12.4	Mapas (ou arrays associativos) de objetos . . . . .	277
12.5	Algoritmos em coleções . . . . .	282
12.6	Comparadores e ordem natural de objetos . . . . .	283
<b>A</b>	<b>A classe Keyboard</b> . . . . .	<b>291</b>
A.1	Introdução . . . . .	291
A.2	A classe <b>Keyboard</b> . . . . .	292
A.2.1	Exemplos de uso dos métodos da classe <b>Keyboard</b> . . . . .	303
A.2.2	Sumário dos métodos da classe <b>Keyboard</b> . . . . .	305
A.3	Usando arquivos de respostas . . . . .	307
	<b>Índice Remissivo</b> . . . . .	<b>309</b>

## Lista de Figuras

1.1	Comanda do Restaurante Caseiro Hipotético. . . . .	3
1.2	Representação de encapsulamento. . . . .	7
1.3	O modelo <b>Lampada</b> , seus atributos e operações. . . . .	8
1.4	O modelo <b>ContaBancariaSimplificada</b> , seus atributos e operações. . . . .	11
1.5	O modelo <b>Data</b> , seus atributos e operações. . . . .	13
1.6	O modelo <b>RegistroAcademico</b> , seus atributos e operações. . . . .	15
3.1	Referências e instâncias criadas na classe <b>UsaDataEncapsulada</b> . . . . .	48
3.2	Coordenadas no espaço cartesiano de duas dimensões. . . . .	49
5.1	Caixas de bancos com e sem fila única. . . . .	76
7.1	Ilustração do problema das Torres de Hanói. . . . .	128
7.2	Passos na solução do problema das Torres de Hanói. . . . .	129
7.3	Chamadas recursivas ao método <b>fibonacci</b> . . . . .	133
8.1	Diagrama de herança envolvendo várias classes. . . . .	151
10.1	Exemplo de array irregular de duas dimensões. . . . .	228
10.2	Triângulo de Pascal com seis linhas. . . . .	229



# Lista de Listagens

1.1	O modelo <b>Lampada</b> , em pseudocódigo. . . . .	9
1.2	O modelo <b>ContaBancariaSimplificada</b> , em pseudocódigo. . . . .	11
1.3	O modelo <b>Data</b> , em pseudocódigo. . . . .	13
1.4	O modelo <b>RegistroAcademico</b> , em pseudocódigo. . . . .	15
2.1	Uma classe vazia em Java. . . . .	22
2.2	A classe <b>RegistroAcademicoSimples</b> , que demonstra declarações de alguns atributos e seus tipos. . . . .	26
2.3	A classe <b>DataSemMetodos</b> , que demonstra mais declarações de alguns atributos e seus tipos. . . . .	27
2.4	A classe <b>DataSimples</b> , que apresenta declarações de atributos e métodos. . . . .	30
2.5	A classe <b>Triangulo</b> , que encapsula os atributos e métodos para representar triângulos. . . . .	33
2.6	A classe <b>UsaDataSimples</b> , que pode ser executada como uma aplicação e que demonstra o uso de instâncias da classe <b>DataSimples</b> . . . . .	35
2.7	A classe <b>DataEncapsulada</b> , que implementa uma política de ocultação de atributos. . . . .	38
2.8	A classe <b>UsaDataEncapsulada</b> , que pode ser executada e que demonstra o uso de instâncias da classe <b>DataEncapsulada</b> . . . . .	41
3.1	A classe <b>UsaDataEncapsulada</b> , que demonstra outros usos de instâncias da classe <b>DataEncapsulada</b> . . . . .	44
3.2	A classe <b>Ponto2D</b> , que encapsula um ponto no espaço cartesiano de duas dimensões. . . . .	49
3.3	A classe <b>UsaPonto2D</b> , que demonstra instâncias da classe <b>Ponto2D</b> . . . . .	51
3.4	A classe <b>OutrosExemplosDataEncapsulada</b> , que demonstra mais exemplos de referências. . . . .	53
3.5	A classe <b>UsaPontoTeclado</b> , que demonstra instâncias da classe <b>Ponto2D</b> que serão inicializadas com valores lidos do teclado. . . . .	55
4.1	A classe <b>RegistroAcademicoSemConstrutor</b> , que encapsula alguns dados de um registro acadêmico. . . . .	57
4.2	A classe <b>UsaRegistroAcademicoSemConstrutor</b> , que demonstra o uso de instâncias da classe <b>RegistroAcademicoSemConstrutor</b> . . . . .	59
4.3	A classe <b>EventoAcademico</b> , que representa um evento acadêmico. . . . .	61

4.4	A classe <b>UsaEventoAcademico</b> , que demonstra o uso de instâncias da classe <b>EventoAcademico</b> . . . . .	62
4.5	A classe <b>RoboSimples</b> , que encapsula dados e comportamento de um robô móvel simulado. . . . .	65
4.6	A classe <b>UsaRoboSimples</b> , que demonstra a criação e o uso de instâncias da classe <b>RoboSimples</b> . . . . .	68
4.7	A classe <b>RoboSimplesRevisto</b> , que encapsula dados e comportamento de um robô móvel simulado (com melhorias). . . . .	69
4.8	Trecho de código que exemplifica uso da palavra-chave <b>this</b> para diferenciar argumentos de atributos. . . . .	72
4.9	A classe <b>UsaRoboSimplesComErros</b> , que demonstra o uso de chamadas com argumentos incorretos a construtores e métodos da classe <b>RoboSimples</b> . . .	73
5.1	A classe <b>SimuladorDeCaixaDeBanco</b> , que simula o mecanismo de atendimento de um caixa de banco. . . . .	77
5.2	A classe <b>UsaSimuladorDeCaixaDeBanco</b> , que demonstra usos de instâncias da classe <b>SimuladorDeCaixaDeBanco</b> . . . . .	77
5.3	Resultado da execução da classe <b>UsaSimuladorDeCaixaDeBanco</b> . . . . .	78
5.4	A classe <b>SimuladorDeCaixaDeBancoRevisto</b> , que simula o mecanismo de atendimento de um caixa de banco (com senhas unificadas para o atendimento). . . . .	79
5.5	Resultado da execução da classe <b>UsaSimuladorDeCaixaDeBanco</b> , reescrita para usar instâncias de <b>SimuladorDeCaixaDeBancoRevisto</b> . . . . .	80
5.6	A classe <b>ConstantesMatematicas</b> , que contém algumas constantes matemáticas úteis. . . . .	81
5.7	A classe <b>UsaConstantesMatematicas</b> , que demonstra o uso dos atributos estáticos na classe <b>ConstantesMatematicas</b> . . . . .	81
5.8	A classe <b>ConversaoDeUnidadesDeComprimento</b> , que contém métodos estáticos para a conversão de unidades de comprimento. . . . .	83
5.9	A classe <b>UsaConversaoDeUnidadesDeComprimento</b> , que demonstra o uso dos métodos estáticos na classe <b>ConversaoDeUnidadesDeComprimento</b> . . .	84
5.10	A classe <b>CalculoDePrecoDeTerreno</b> , que calcula o preço de um terreno baseado em sua área e localização, usando um método estático. . . . .	85
5.11	A classe <b>DataComFabrica</b> , que contém uma fábrica de instâncias da própria classe. . . . .	87
5.12	A classe <b>UsaDataComFabrica</b> , que demonstra usos da classe <b>DataComFabrica</b> . . . . .	88
5.13	O enumerador <b>EstadosDoBrasil</b> , que declara constantes para todos os estados do Brasil. . . . .	89
5.14	A classe <b>UsaEstadosDoBrasil</b> , que demonstra o uso do enumerador <b>EstadosDoBrasil</b> . . . . .	90
5.15	O enumerador <b>EstadosDoBrasilRevisto</b> , que associa várias informações às constantes de um enumerador. . . . .	91
5.16	A classe <b>UsaEstadosDoBrasilRevisto</b> , que demonstra o uso do enumerador <b>EstadosDoBrasilRevisto</b> . . . . .	93



6.1	A classe <b>Comparavel</b> , que encapsula um valor e contém métodos para compará-lo com outros. . . . .	96
6.2	A classe <b>UsaComparavel</b> , que demonstra usos de instâncias e métodos da classe <b>Comparavel</b> . . . . .	99
6.3	Pseudocódigo usado para calcular o preço de uma entrada de cinema. . . .	100
6.4	A classe <b>EntradaDeCinema</b> , que calcula o preço de uma entrada de cinema.	101
6.5	Trecho de código que demonstra <b>ifs</b> aninhados. . . . .	102
6.6	Trecho de código que demonstra <b>ifs</b> aninhados e blocos de comandos. . . .	102
6.7	A classe <b>DataIf</b> , com um método que demonstra o uso da instrução <b>if</b> em cascata. . . . .	103
6.8	A classe <b>ComparaSimples</b> , que demonstra o uso do operador condicional <b>?</b> .	104
6.9	A classe <b>DataSwitch</b> , com métodos que demonstram a instrução <b>switch</b> . .	105
6.10	A classe <b>UsaEstadosDoBrasilComSwitch</b> , que demonstra como as constantes do enumerador <b>EstadosDoBrasil</b> podem ser usadas com a instrução <b>switch</b> . . . . .	108
7.1	A classe <b>ExemplosWhile</b> , que demonstra vários usos do laço <b>while</b> . . . . .	113
7.2	A classe <b>EscolhaComWhile</b> , que encapsula um mecanismo de escolha de valores (usando um bloco <b>while</b> ). . . . .	114
7.3	A classe <b>EscolhaComWhileEContinue</b> , que encapsula um mecanismo de escolha de valores (usando um bloco <b>while</b> e o comando <b>continue</b> ). . . . .	116
7.4	A classe <b>EscolhaComDoWhile</b> , que encapsula um mecanismo de escolha de valores (usando um bloco <b>do-while</b> ). . . . .	117
7.5	A classe <b>JogoDeAdivinhacao</b> , que implementa um jogo simples de adivinhação de números. . . . .	119
7.6	A classe <b>UsaJogoDeAdivinhacao</b> , que demonstra o uso de uma instância da classe <b>JogoDeAdivinhacao</b> . . . . .	120
7.7	A classe <b>ExemplosFor</b> , que demonstra vários usos do laço <b>for</b> . . . . .	121
7.8	A classe <b>Combinatoria</b> , que contém alguns métodos estáticos para cálculos básicos de combinatória. . . . .	123
7.9	A classe <b>UsaCombinatoria</b> , que demonstra o uso dos métodos estáticos da classe <b>Combinatoria</b> . . . . .	124
7.10	A classe <b>ExemplosForEnumeradores</b> , que demonstra como percorrer as constantes do enumerador <b>EstadosDoBrasil</b> com um laço <b>for</b> . . . . .	125
7.11	Pseudocódigo para calcular somatória recursivamente. . . . .	126
7.12	A classe <b>SomatoriaRecursiva</b> , que demonstra o cálculo da somatória recursiva dos primeiros <i>N</i> números inteiros. . . . .	126
7.13	Resultado da execução da classe <b>SomatoriaRecursiva</b> com o valor 6. . . .	128
7.14	A classe <b>TorresDeHanoi</b> , que demonstra a solução recursiva para o problema das Torres de Hanói. . . . .	130
7.15	A classe <b>FibonacciRecursiva</b> , que demonstra o cálculo da série de Fibonacci recursivamente. . . . .	131

8.1	A classe <b>DataHora</b> , que reutiliza as classes <b>Data</b> e <b>Hora</b> através do mecanismo de delegação. . . . .	135
8.2	A classe <b>RegistroAcademicoDeGraduacao</b> , que reutiliza a classe <b>Data</b> através de delegação. . . . .	137
8.3	O enumerador <b>CursosDeGraduacao</b> , que contém constantes e métodos para representar alguns cursos de graduação. . . . .	139
8.4	A classe <b>UsaRegistroAcademicoDeGraduacao</b> , que demonstra a criação e o uso de instâncias da classe <b>RegistroAcademicoDeGraduacao</b> . . . . .	140
8.5	A classe <b>PessoaSimple</b> s, que encapsula os dados de identificação de uma pessoa. . . . .	141
8.6	A classe <b>FuncionarioSimple</b> s, que encapsula os dados básicos de um funcionário de uma empresa. . . . .	142
8.7	A classe <b>UsaFuncionarioSimple</b> s, que demonstra o uso de uma instância da classe <b>FuncionarioSimple</b> s. . . . .	143
8.8	A classe <b>Pessoa</b> , que encapsula os dados de identificação de uma pessoa. . . . .	146
8.9	A classe <b>Funcionario</b> , que encapsula os dados básicos de um funcionário de uma empresa e herda da classe <b>Pessoa</b> . . . . .	147
8.10	A classe <b>ChefeDeDepartamento</b> , que encapsula os dados básicos de um chefe de um departamento de uma empresa e que herda da classe <b>Funcionario</b> . . . . .	149
8.11	A classe <b>ObjetoSimulado</b> , que encapsula os dados de um objeto genérico em um simulador. . . . .	154
8.12	A classe <b>ObjetoMovelsimulado</b> , que encapsula os dados de um objeto simulado que pode se mover. . . . .	156
8.13	O enumerador <b>DirecoesSimulador</b> , que lista os possíveis valores para a direção para movimento de um objeto simulado. . . . .	157
8.14	A classe <b>ObjetoRoboSimulado</b> , que herda da classe <b>ObjetoMovelsimulado</b> e que representa um robô simulado. . . . .	157
8.15	A classe <b>SimuladorSimple</b> s, que cria e usa algumas instâncias de classes que herdam de <b>ObjetoSimulado</b> . . . . .	159
8.16	A classe <b>ObjetoRoboPesadoSimulado</b> , que herda de <b>ObjetoRoboSimulado</b> . . . . .	161
8.17	A classe <b>SimuladorSimpleRobos</b> , que cria e usa algumas instâncias de <b>ObjetoRoboSimulado</b> e <b>ObjetoRoboPesadoSimulado</b> . . . . .	162
8.18	A classe <b>Pessoa</b> (segunda versão). . . . .	165
8.19	A classe <b>Funcionario</b> (segunda versão). . . . .	166
8.20	A classe <b>ChefeDeDepartamento</b> (segunda versão). . . . .	168
8.21	A classe <b>SimuladorPolimorfismo</b> , que demonstra polimorfismo com instâncias de <b>ObjetoSimulado</b> e classes herdeiras. . . . .	170
8.22	A classe <b>EmprestimoBancario</b> , que demonstra implementação de métodos com polimorfismo e sobrecarga. . . . .	172
8.23	A classe <b>EmprestimoBancarioComCast</b> , que demonstra implementação de métodos com polimorfismo e <i>cast</i> de instâncias. . . . .	175
8.24	Trecho de código que demonstra algumas regras de <i>cast</i> de instâncias de classes. . . . .	177



9.1	A classe abstrata <b>ContaBancaria</b> , que representa uma conta bancária simples. . . . .	181
9.2	A classe <b>ContaSimples</b> , que representa uma conta bancária simples, e que pode ser instanciada. . . . .	182
9.3	A classe <b>ContaEspecial</b> , que representa uma conta bancária especial. . . . .	183
9.4	A classe <b>UsaContasBancarias</b> , que demonstra o uso de classes que herdam de <b>ContaBancaria</b> . . . . .	184
9.5	A interface <b>ObjetoGeometrico</b> , que declara que métodos um objeto geométrico genérico deve implementar. . . . .	185
9.6	A classe <b>Ponto2D</b> , que representa um ponto em duas dimensões. . . . .	186
9.7	A classe <b>Circulo</b> , que implementa a interface <b>ObjetoGeometrico</b> e representa um círculo. . . . .	187
9.8	A classe <b>Retangulo</b> , que implementa a interface <b>ObjetoGeometrico</b> e representa um retângulo. . . . .	189
9.9	A classe <b>UsaObjetosGeometricos</b> , que demonstra instâncias de classes que implementam a interface <b>ObjetoGeometrico</b> . . . . .	191
9.10	A classe <b>UsaObjetosGeometricosComPolimorfismo</b> , que demonstra efeitos de polimorfismo em classes que implementam uma interface comum. . . . .	192
9.11	A classe <b>Escalavel</b> , que define que métodos um objeto que seja escalável (possa ter seu tamanho modificável) deve implementar. . . . .	194
9.12	A classe <b>CirculoEscalavel</b> , que implementa as interfaces <b>ObjetoGeometrico</b> e <b>Escalavel</b> . . . . .	195
9.13	A classe <b>UsaCirculoEscalavel</b> , que demonstra o uso dos métodos da classe <b>CirculoEscalavel</b> . . . . .	196
9.14	A classe <b>ItemDeBiblioteca</b> , que declara que atributos e métodos devem ser usados para representar itens em uma biblioteca. . . . .	197
9.15	O enumerador <b>Estado</b> , que representa os estados de empréstimo de um item de biblioteca. . . . .	198
9.16	A classe <b>Livro</b> , que encapsula os dados de um livro. . . . .	199
9.17	A classe <b>LivroDeBiblioteca</b> , que encapsula os dados de um livro de uma biblioteca. . . . .	200
9.18	A classe <b>UsaLivroDeBiblioteca</b> , que demonstra o uso de instâncias da classe <b>LivroDeBiblioteca</b> . . . . .	202
9.19	A classe <b>ItemRaroDeBiblioteca</b> , que declara os atributos e métodos que itens raros de uma biblioteca devem ter. . . . .	203
9.20	A classe <b>Mapa</b> , que encapsula os dados de um mapa. . . . .	204
9.21	A classe <b>MapaDeBiblioteca</b> , que encapsula os dados de um mapa de uma biblioteca. . . . .	205
10.1	Trecho de código que demonstra a declaração de arrays em Java. . . . .	209
10.2	Trecho de código que demonstra a declaração e inicialização de arrays em Java. . . . .	209
10.3	A classe <b>AvaliaNotas</b> , que manipula arrays de notas para criar um array de enumeradores. . . . .	212

10.4	O enumerador <b>Situacao</b> , que manipula arrays de notas para criar um array de enumeradores. . . . .	212
10.5	A classe <b>ArrayDeFloats</b> , que encapsula um array de valores do tipo nativo <b>float</b> . . . . .	213
10.6	A classe <b>UsaArrayDeFloats</b> , que demonstra o uso de instâncias da classe <b>ArrayDeFloats</b> . . . . .	215
10.7	Trecho de código que demonstra as duas formas de iteração em arrays. . .	217
10.8	A classe <b>Equipe</b> , que cria e usa um array de instâncias de classes que herdam de <b>Funcionario</b> . . . . .	217
10.9	A classe <b>ArrayDeObjetosGeometricos</b> , que encapsula um array de instâncias de classes que implementam a interface <b>ObjetosGeometricos</b> . . . . .	219
10.10	A classe <b>UsaArrayDeObjetosGeometricos</b> , que demonstra o uso de instâncias da classe <b>ArrayDeObjetosGeometricos</b> . . . . .	221
10.11	Trecho de código que demonstra a declaração de arrays multidimensionais em Java. . . . .	223
10.12	A classe <b>MatrizDeDoubles</b> , que encapsula um array bidimensional (matriz) de valores do tipo <b>double</b> . . . . .	223
10.13	A classe <b>TabuleiroDeJogoDeDamas</b> , que encapsula um tabuleiro de jogo de damas. . . . .	225
10.14	O enumerador <b>PecasDoJogoDeDamas</b> , que representa as peças em um jogo de damas. . . . .	226
10.15	Trecho de código que demonstra a declaração e uso de arrays irregulares em Java. . . . .	228
10.16	A classe <b>TrianguloDePascal</b> , que implementa um triângulo de Pascal como um array irregular. . . . .	229
10.17	A classe <b>CalculadoraDeLinhaDeComando</b> , que implementa uma calculadora simples usando argumentos passados pela linha de comando. . . . .	231
10.18	A classe <b>ArgumentosVariaveis</b> , que demonstra a criação e uso de métodos com argumentos variáveis. . . . .	233
11.1	Trecho de código que demonstra a criação de instâncias de <b>String</b> . . . . .	237
11.2	A classe <b>JogoDaForca</b> , que encapsula o mecanismo de um jogo da forca simples. . . . .	238
11.3	A classe <b>UsaJogoDaForca</b> , que demonstra o uso da classe <b>JogoDaForca</b> . . .	241
11.4	Trecho de código que exemplifica a ordem de chamada de concatenação de strings. . . . .	246
11.5	A classe <b>SeparacaoDeStrings</b> , que demonstra como usar algumas expressões regulares para separar strings em componentes. . . . .	247
11.6	Trecho de código que ilustra métodos de modificação da classe <b>StringBuilder</b> . . . . .	249
11.7	Trecho de código que exemplifica uso do método <b>delete</b> da classe <b>StringBuilder</b> . . . . .	250
11.8	Trecho de código que exemplifica uso do método <b>replace</b> da classe <b>StringBuilder</b> . . . . .	250



12.1	Trecho de código que demonstra <i>autoboxing</i> . . . . .	254
12.2	Trecho de código que demonstra métodos da classe <b>HashSet</b> . . . . .	256
12.3	A classe <b>MegaSena</b> , que contém um método estático que demonstra o uso de um <i>Set</i> para sortear números sem repetições. . . . .	257
12.4	A classe <b>OperacoesEmConjuntos</b> , que demonstra algumas operações em conjuntos ( <i>sets</i> ). . . . .	259
12.5	A classe <b>ConjuntoDePalavras</b> , que encapsula um conjunto de palavras sem repetição. . . . .	261
12.6	A classe <b>UsaConjuntoDePalavras</b> , que demonstra o uso da classe <b>ConjuntoDePalavras</b> . . . . .	262
12.7	Trecho de código que demonstra o uso de métodos de <b>TreeSet</b> . . . . .	263
12.8	Trecho de código que demonstra uso de instâncias da classe <b>LinkedList</b> . . . . .	265
12.9	A classe <b>MovimentoDeRobo</b> , que representa um único movimento de um robô (que consiste em um número de passos dados em uma certa direção). . . . .	266
12.10	O enumerador <b>DirecoesDoRobo</b> , que contém constantes para representar os pontos cardeais. . . . .	266
12.11	A classe <b>RoboComMemoria</b> , que representa um robô que armazena os movimentos feitos por ele. . . . .	267
12.12	A classe <b>UsaRoboComMemoria</b> , que demonstra o uso de uma instância da classe <b>RoboComMemoria</b> . . . . .	269
12.13	Trecho de código que demonstra outros métodos de classes que implementam <b>List</b> . . . . .	270
12.14	Trecho de código que demonstra como implementar uma fila com a classe <b>LinkedList</b> . . . . .	272
12.15	Trecho de código que demonstra como implementar uma pilha com a classe <b>LinkedList</b> . . . . .	272
12.16	A classe <b>PilhaDeDoubles</b> , que representa uma pilha onde podemos armazenar e recuperar valores do tipo <b>double</b> . . . . .	274
12.17	A classe <b>AvaliadorDeExpressoes</b> , que contém um método estático que permite a avaliação de expressões matemáticas do tipo <i>postfix</i> . . . . .	275
12.18	A classe <b>UsaAvaliadorDeExpressoes</b> , que demonstra o uso do método <b>avaliaExpressãoPostfix</b> da classe <b>AvaliadorDeExpressoes</b> . . . . .	276
12.19	Trecho de código que demonstra métodos básicos de classes que implementam <b>Map</b> . . . . .	278
12.20	A classe <b>DicionarioDeIPs</b> , que cria e permite a consulta a um minidicionário de nomes de servidores e seus IPs. . . . .	279
12.21	A classe <b>ArrayEsparsodeDoubles</b> , que encapsula um array esparsode valores de ponto flutuante. . . . .	280
12.22	A classe <b>UsaArrayEsparsodeDoubles</b> , que demonstra o uso de uma instância da classe <b>ArrayEsparsodeDoubles</b> . . . . .	282
12.23	Trecho de código que exemplifica o uso de alguns métodos da classe <b>Collections</b> para reorganizar listas. . . . .	283
12.24	A classe <b>DisciplinaCursada</b> , que encapsula dados referentes a uma disciplina cursada. . . . .	284

12.25	O enumerador <b>CodigoDisciplina</b> , que representa as disciplinas que podem ser cursadas em uma universidade. . . . .	285
12.26	A classe <b>AlunoDeGraduacao</b> , que encapsula alguns dados de um aluno de graduação. . . . .	285
12.27	A classe <b>ComparaPorNomeCrescente</b> , que permite a comparação de instâncias de <b>AlunoDeGraduacao</b> pelos nomes. . . . .	287
12.28	A classe <b>ComparaPorNotaDecrescente</b> , que permite a comparação de instâncias de <b>AlunoDeGraduacao</b> pelas médias de notas. . . . .	288
12.29	A classe <b>TurmaDeAlunos</b> , que demonstra operações em uma turma (lista) de alunos. . . . .	289
A.1	A classe <b>Keyboard</b> , que permite a leitura de valores simples via teclado. . .	292
A.2	A classe <b>UsaKeyboard</b> , que demonstra o uso da classe <b>Keyboard</b> . . . . .	303
A.3	Um exemplo de arquivo de respostas para uso com a classe <b>UsaKeyboard</b> . .	308

## Lista de Tabelas

2.1	Palavras reservadas em Java. . . . .	22
2.2	Tipos básicos de Java. . . . .	25