

## **Elogios a Use a Cabeça! Padrões de Projetos (Design Patterns)**

“Recebi o livro ontem e comecei a ler no caminho para casa... e não consegui parar. Levei-o para a academia e imagino que as pessoas tenham me visto sorrindo bastante enquanto eu me exercitava e lia. O livro é muito legal. É divertido e ao mesmo tempo eles cobrem vários tópicos e vão direto ao assunto. Estou realmente impressionado.”- **Erich Gamma, Engenheiro Laureado da IBM, e co-autor de Padrões de Projetos**

“Use a Cabeça! Padrões de Projetos ´ consegue misturar diversão, grandes risadas, profundidade técnica e ótimos conselhos práticos em uma leitura divertida e inspiradora. Independentemente de ser um novato nos padrões de projetos, ou os ter usado por anos, você certamente sairá ganhando ao visitar a Objetolândia”- **Richard Helm, co-autor de “Padrões de Projetos” com o resto da Gang of Four – Erich Gamma, Ralph Johnson e John Vlissides**

“Me sinto como se uma pilha de mil livros tivesse acabado de ser tirada de cima da minha cabeça”- **Ward Cunningham, inventor do Wiki e fundador do Hillside Group**

“Este livro chega quase à perfeição, por causa da maneira como ele combina o conhecimento e a facilidade de leitura. Ele fala com autoridade e é ótimo de ler. É um dos muito poucos livros sobre software que já li e considero indispensáveis. (Eu colocaria talvez 10 livros nessa categoria, no máximo.)”- **David Gelerner, Professor de Ciência da Computação, Universidade de Yale e autor de “Mirror Worlds” e “Machine Beauty”**.

“Um mergulho de cabeça no reino dos padrões, um lugar onde coisas complexas se tornam simples, mas onde coisas simples podem também se tornar complexas. Não consigo pensar em guias melhores do que os Freemans.- **Miko Matsumura, Analista da Indústria, The Middleware Company; Ex-Pregador Chefe do Java, Sun Microsystems**

“Eu ri, eu chorei, o livro me emocionou”- **Daniel Steinberg, Editor-em-Chefe, java.net**

“Minha primeira reação foi rolar de rir. Depois de me recompor, percebi que o livro não somente é tecnicamente preciso, como também é a introdução aos padrões de projetos mais fácil de entender que eu já vi.” -**Dr. Timothy A. Budd, Professor Associado de Ciência da Computação na Universidade Estadual de Oregon e autor de mais de uma dúzia de livro, incluindo “C++ para Programadores Java”**

“Jerry Rice executa padrões melhor do que qualquer jogador da NFL, mas os Freemans acabaram de superá-lo. Mas falando sério... este é um dos livros mais engraçados e inteligentes sobre projetos de software que eu já li.”- **Aaron LaBerge , VP Technology, ESPN.com**

## **Mais elogios a Use a Cabeça! Padrões de Projetos**

“Um ótimo projeto de código é, antes de mais nada, um ótimo projeto de informação. Um projetista de código está ensinando um computador como fazer algo, e não é surpresa que a editora seja uma grande professora de programadores. A clareza, o humor e as substanciais doses de inteligência deste livro o tornam o tipo que ajuda até mesmo não-programadores a pensar bem sobre como solucionar problemas.”- **Cory Doctorow, co-editor de Boing Boing e autor de “Down and Out in the Magic Kingdom” e “Someone Comes to Town, Someone leaves Town”**

“Há um velho ditado na indústria de computadores e videogames – bem, não pode ser tão velho porque a própria indústria não é velha assim – que diz mais ou menos o seguinte: O Projeto é Vida. O mais curioso sobre essa frase é que mesmo hoje em dia quase ninguém que trabalha com a arte de criar jogos eletrônicos consegue chegar a um acordo sobre o que significa “projetar” um jogo. O projetista é um engenheiro de software? Um diretor de arte? Um contador de histórias? Um arquiteto ou um pedreiro? Uma pessoa com os pés no chão ou um visionário? É possível um mesmo indivíduo ter algo de todos esses papéis? E, mais importante, quem liga para essa %\$#&\*?”

Já foi dito que o crédito “projetado por” em entretenimento interativo é similar ao “dirigido por” na indústria dos filmes, o que de fato permite ao nosso crédito compartilhar DNA com aquele que é talvez o crédito mais controverso, exagerado, e freqüentemente sem nenhuma humildade já propagado na arte comercial. Boa companhia, hein? E mesmo assim, o Projeto é Vida, então talvez seja hora de despendermos algum tempo pensando sobre o que ele é. Eric e Elisabeth Freeman intrepidamente se ofereceram para explorar os mistérios do código para nós em *Use a Cabeça! Padrões de Projetos*. Não estou certo de que qualquer um deles se importe tanto assim com o PlayStation ou o X-Box, e nem eles deveriam. Mesmo assim, eles abordam a noção de projeto em um nível significativamente honesto, de forma tal que qualquer um procurando por elogios aos seus próprios talentos como escritor deve ficar avisado de que terá uma boa lição de humildade ao ler este livro”.- **Ken Goldstein, Vice-Presidente Executivo & Diretor Gerente, Disney Online**

“O livro tem o tom exato para o guru nerd e casualmente legal em todos nós. A referência correta para estratégias práticas de desenvolvimento – faz o meu cérebro funcionar sem ter de empurrar um monte de conversa velha e mofada de professor.”- **Travis Kalanick, Fundador do Scour and Red Swoosh, membro do MIT TR100**

“Este livro combina bom humor, ótimos exemplos e conhecimento aprofundado dos Padrões de Projetos de forma tal que torna o aprendizado divertido. Trabalhando na indústria de tecnologia de entretenimento, eu tenho dúvidas sobre o Princípio Hollywood e o Padrão Facade dos home theaters, entre outros. O entendimento dos Padrões de Projetos não apenas nos ajudam a criar bom software reutilizável e fácil de manter, mas também ajuda a melhorar as nossas habilidades de resolução de problemas em várias áreas. Este livro é indispensável para todos os profissionais e estudantes de computação.” - **Newton Lee, Fundador e Editor-Chefe, Association for Computing Machinery’s (ACM) Computers int Entertainment (acmcie.org)**

## **Elogios à abordagem Use a Cabeça!**

“A tecnologia Java está em toda parte – em telefones celulares, câmeras, impressoras, jogos, PDAs, caixas eletrônicos, smart cards, bombas de gasolina, estádios esportivos, dispositivos médicos, web cams, servidores, entre outros. Se você desenvolve software e ainda não aprendeu Java, definitivamente está na hora de mergulhar – Use A Cabeça!” – **Scott McNealy, Presidente e CEO da Sun Microsystems**

“É rápido, irreverente, divertido e envolvente. Cuidado – você pode acabar realmente aprendendo algo!” – **Ken Arnold, ex-Engenheiro Sênior da Sun Microsystems Co-autor (com James Gosling, criador do Java) de The Java Programming Language**

“Use A Cabeça! Java é mais ou menos o que acontece quando o Monty Python encontra a gang of four... o texto é tão bem dividido entre quebra-cabeças e histórias, testes e exemplos, que você aprende com uma profundidade que nenhum outro livro sobre computação oferece.” – **Douglas Rowe, Grupo de Usuários Java de Columbia**

“ ‘Use A Cabeça! Java’... dá um novo significado à frase de marketing deles ‘Há um O’Reilly para isso’. Eu escolhi este aqui porque várias pessoas que respeito o tinham descrito em termos como ‘revolucionário’ e ressaltando a abordagem radicalmente diferente do livro-texto. Elas estavam (estão) certas... De modo tipicamente O’Reilly, eles usaram uma abordagem científica e bem pensada. O resultado é engraçado, irreverente, dividido em tópicos, interativo e brilhante... Ler este livro é como estar no saguão de uma conferência técnica, aprendendo – e rindo – com os seus pares... Se você quiser ENTENDER Java, vá comprar este livro.” – **Andrew Pollack, www.thenorth.corn**

“Se você quer *aprender* Java, achou o que procurava: bem-vindo ao primeiro livro técnico com uma GUI! Este formato perfeitamente executado e inovador oferece benefícios que outros textos sobre Java simplesmente não conseguem... Prepare-se para uma viagem realmente notável pela Javalândia.” – **Neil R. Bauman, Capitão e CEO, Geek Cruises (www.GeekCruises.corn)**

“Que fantástica maneira de aprender!!! NÃO CONSIGO FECHAR ESTE LIVRO!!! Meu filho de 3 anos acordou à 1h40 da madrugada e eu o coloquei de volta na cama com o livro e uma lanterna na mão, para poder continuar lendo por cerca de mais uma hora.” – **Ross Goldberg**

“Este material é tão absurdamente bom que me faz querer CHORAR! Estou impressionado.” – **Floyd Jones, Escritor Técnico Sênior, BEA**

# Use a Cabeça!

## Padrões de Projetos

(Design Patterns)

Não seria maravilhoso se houvesse um livro sobre Padrões de Projetos que fosse mais divertido que ir ao dentista e mais revelador do que um formulário do Imposto de Renda? Mas provavelmente isso é só uma fantasia...



2ª Edição Revisada

*Eric Freeman*  
*Elisabeth Freeman*

com  
*Kathy Sierra*  
*Bert Bates*



# Use a Cabeça!

# Padrões de Projetos

(Design Patterns)

2ª Edição Revisada

*Eric Freeman*  
*Elisabeth Freeman*

com  
*Kathy Sierra*  
*Bert Bates*



Rio de Janeiro.2007

## Use a Cabeça! Padrões de Projetos (Design Patterns) 2ª Edição Revisada

Do original **Head First Desing Paterns** Copyright © 2005, 2007 da Editora Alta Books Ltda.

*Authorized translation from English language edition, entitled **Head First Desing Paterns**, ISBN 0-596-00712-4, by Eric and Elisabeth Freeman with Kathy Sierra and Bert Bates, published by O'Reilly Media, Inc. Copyright © 2004 by O'Reilly Media, In.. PORTUGUESE language edition published by Editora Alta Books, Copyright © 2007 by Editora Alta Books.*

Todos os direitos reservados e protegidos pela Lei 5988 de 14/12/73. Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônico, mecânico, fotográfico, gravação ou quaisquer outros.

Todo o esforço foi feito para fornecer a mais completa e adequada informação, contudo a editora e o(s) autor(es) não assumem responsabilidade pelos resultados e usos da informação fornecida. Recomendamos aos leitores testar a informação, bem como tomar todos os cuidados necessários (como o backup), antes da efetiva utilização. Este livro não contém CD-ROM, disquete ou qualquer outra mídia.

**Erratas e atualizações:** Sempre nos esforçamos para entregar a você, leitor, um livro livre de erros técnicos ou de conteúdo; porém, nem sempre isso é conseguido, seja por motivo de alteração de software, interpretação ou mesmo quando alguns deslizes constam na versão original de alguns livros que traduzimos. Sendo assim, criamos em nosso site, [www.altabooks.com.br](http://www.altabooks.com.br), a seção Erratas, onde relataremos, com a devida correção, qualquer erro encontrado em nossos livros.

**Avisos e Renúncia de Direitos:** Este livro é vendido como está, sem garantia de qualquer tipo, seja expressa ou implícita.

**Marcas Registradas:** Todos os termos mencionados e reconhecidos como Marca Registrada e/ou comercial são de responsabilidade de seus proprietários. A Editora informa não estar associada a nenhum produto e/ou fornecedor apresentado no livro. No decorrer da obra, imagens, nomes de produtos e fabricantes podem ter sido utilizados, e desde já a Editora informa que o uso é apenas ilustrativo e/ou educativo, não visando ao lucro, favorecimento ou desmerecimento do produto/fabricante.

**Produção Editorial: Editora Alta Books**

**Coordenação Editorial: Fernanda Silveira**

**Tradução: Andreza Gonçalves, Marcelo Soares e Pedro César de Conti**

**Revisão: Carolina Menegassi**

**Revisão Técnica:**

**(1ª Edição) Késsia Nina - Licenciatura em Computação pela Universidade de Brasília (UNB)**

**(2ª Edição) Hélder Pereira Borges - Professor do Departamento Acadêmico de Informática (DAI) do CEFET-MA**

**Diagramação: Flávio Damasceno**

Impresso no Brasil

*O código de propriedade intelectual de 1º de Julho de 1992 proíbe expressamente o uso coletivo sem autorização dos detentores do direito autoral da obra, bem como a cópia ilegal do original. Esta prática generalizada nos estabelecimentos de ensino, provoca uma brutal baixa nas vendas dos livros a ponto de impossibilitar os autores de criarem novas obras.*



Rua Viúva Claudio, 291 - Jacaré  
Rio de Janeiro - RJ.  
CEP: 20970-031  
Tel: 21 3278-8069/ Fax: 3277-1253  
[www.altabooks.com.br](http://www.altabooks.com.br)  
e-mail: [altabooks@altabooks.com.br](mailto:altabooks@altabooks.com.br)

## Autores/Desenvolvedores de Use a Cabeça Padrões de Projetos

**Eric Freeman** é um cientista da computação apaixonado por arquiteturas de mídia e de software. Ele acabou de fechar quatro anos em um emprego dos sonhos – dirigindo trabalhos de Internet banda larga e wireless na Disney – e agora está voltado a programar, criando softwares legais e hackeando Java e Macs.

Eric passou boa parte dos anos 90 trabalhando em alternativas à metáfora do desktop com David Gelernter (e os dois *ainda* estão fazendo a pergunta “por que eu tenho de dar um nome ao meu arquivo?”). Com base nesse trabalho, Eric obteve um Ph.D. na Yale University em 97. Foi também co-fundador da Mirror Worlds Technologies (agora adquirida para criar uma versão comercial do trabalho da sua tese, o Lifestreams).

Em uma vida anterior, Eric criava software para redes e supercomputadores. Talvez você o conheça de livros como *JavaSpaces Principles Patterns and Practice*. Eric tem boas lembranças de implementar sistemas tuple-space em Thinking Machine CM-5s e de criar alguns dos primeiros sistemas de informação Internet para a NASA, no final dos anos 80.

Eric atualmente mora no deserto perto de Santa Fe. Quando não está escrevendo texto nem código, você o encontrará passando mais tempo ajustando seus programas do que assistindo TV no seu home theater ou tentando consertar um videogame do Dragon's Lair dos anos 80. Ele também não se importaria de passar a noite como um DJ de música eletrônica. Escreva para ele (em inglês) em [eric@wickedlysmart.com](mailto:eric@wickedlysmart.com) ou visite o seu blog em <http://www.ericfreeman.com>.

**Elisabeth Freeman** é escritora, desenvolvedora de software e artista digital. Ela esteve envolvida com a Internet desde o princípio, sendo co-fundadora do The Ada Project (TAP), um premiado site voltado às mulheres que trabalham com computação, adotado agora pela ACM. Mais recentemente, Elisabeth liderou trabalhos de pesquisa e desenvolvimento em mídia digital na Walt Disney Company, onde co-inventou o Motion, um sistema de conteúdo que distribui terabytes de vídeo todos os dias aos usuários da Disney, da ESPN e de Movies.com.

Elisabeth é cientista da computação por formação e tem diplomas universitários em Ciência da Computação pela Yale University e pela Indiana University. Tem trabalhado em uma série de áreas, incluindo linguagens visuais, transmissões RSS e sistemas Internet. Tem sido também uma ativista pelas mulheres na computação, desenvolvendo programas que encorajam as mulheres a se envolverem na área. Hoje em dia, você a encontra bebendo café ou achocolatado ao lado do seu Mac, embora ela sonhe com o dia em que todo o mundo usará o Scheme.

Elisabeth sempre adorou caminhadas ao ar livre desde a sua infância na Escócia. Quando está ao ar livre, sua câmera está sempre ao seu alcance. Ela é também uma ávida ciclista, vegetariana e amante dos animais. Você pode enviar um e-mail para ela (em inglês) em [beth@wickedlysmart.com](mailto:beth@wickedlysmart.com).

### Criadores da série Use a Cabeça! (e co-conspiradores neste livro)

**Kathy** se interessa por teoria da aprendizagem desde os seus dias como designer de jogos (ela escreveu jogos para a Virgin, a MGM e a Amblin). Ela desenvolveu a maior parte do formato Use a Cabeça! enquanto ensinava Autoração de Novas Mídias no programa de extensão Estudos de Entretenimento da UCLA. Mais recentemente, ela tem trabalhado como instrutora máster para a Sun Microsystems, ensinando aos instrutores Java da Sun sobre como ensinar as tecnologias Java mais recentes e desenvolvendo vários dos exames de certificação da Sun. Junto com Bert Bates, ela tem usado ativamente os conceitos Use a Cabeça! para ensinar a milhares de desenvolvedores. Kathy é a fundadora do [javaranch.com](http://javaranch.com), que recebeu em 2003 e 2004 o prêmio Jolt Cola Productivity Award, da revista Software Development. Também é possível encontrá-la dando aulas de certificação Java no site Java Jam Geek Cruise ([geekcruises.com](http://geekcruises.com)).

Recentemente se mudou da Califórnia para Boulder, Colorado, onde teve que aprender novas palavras, como “limpador de gelo” e “pele”, mas pelo menos lá é possível esquiar. Gosta de: correr, esquiar, andar de skate, brincar com seu cavalo islandês e de ciência esquisita. Não gosta de: entropia. Um artigo de uma revista uma vez a descreveu como “persuasiva”. Ela ainda está tentando descobrir se eles disseram isso no *bom* sentido. Se você souber, ou se tiver qualquer outro comentário, escreva para ela (em inglês) em [kathy@wickedlysmart.com](mailto:kathy@wickedlysmart.com).

**Bert** é desenvolvedor e arquiteto de software, mas uma passagem de dez anos no campo de inteligência artificial levou-o a se interessar por teoria da aprendizagem e pelo treinamento baseado em tecnologia. Ele tem ensinado programação a clientes desde então. Recentemente, tornou-se membro da equipe de desenvolvimento de vários dos exames de Certificação Java da Sun, incluindo o novo SCBCD.

Ele passou a primeira década da sua carreira em software viajando pelo mundo para ajudar clientes broadcast como a Rádio Nova Zelândia, o Weather Channel e a rede Arts & Entertainment (A & E). Um dos seus projetos favoritos foi a criação de um sistema completo de simulação de malha ferroviária para a Union Pacific Railroad.

Bert é um jogador viciado de *go* de longa data e tem trabalhado em um programa de *go* por tempo demais. Ele é um guitarrista razoavelmente bom e agora está tentando aprender banjo. Escreva para ele (em inglês) em [terrapi@wickedlysmart.com](mailto:terrapi@wickedlysmart.com).



**Bert Bates**

**Kathy Sierra**

## Tabela de Conteúdo (sumário)

Introdução	XV
1 Bem-vindo aos Padrões de Projetos: <i>uma introdução</i>	25
2 Mantendo os seus Objetos atualizados: <i>o Padrão Observer</i>	51
3 Decorando Objetos: <i>o Padrão Decorator</i>	81
4 Cozinhando com a polpa da programação OO: <i>o Padrão Factory</i>	103
5 Objetos Únicos: <i>o Padrão Singleton</i>	149
6 Encapsulando a Chamada de Métodos: <i>o Padrão Command</i>	167
7 Como Ser Adaptável: <i>os padrões Adapter e Facade</i>	201
8 Encapsulando Algoritmos: <i>o Padrão Template Method</i>	229
9 Coleções bem administradas: <i>os Padrões Iterador e Composite</i>	259
10 O Estado das Coisas: <i>o Padrão State</i>	313
11 Controlando o Acesso de Objetos: <i>o Padrão Proxy</i>	349
12 Padrões de Padrões: <i>Padrões Compostos</i>	397
13 Padrões no Mundo Real: <i>Vivendo Melhor com os Padrões</i>	459
14 Apêndice: <i>Padrões Restantes</i>	483

## Tabela de Conteúdo (de verdade)

### Introdução

**Sua mente concentrada nos Padrões de Projetos.** Aqui *you* está tentando *aprender* algo, enquanto o seu *cérebro* está lhe fazendo um favor certificando-se de que o aprendizado não se *manterá*. Seu cérebro está pensando: “é melhor deixar espaço para coisas mais importantes, como quais animais selvagens evitar e fazer esqui na neve pelado é uma má idéia”. Então, *como* você engana o seu cérebro para fazê-lo pensar que a sua vida depende de aprender os Padrões de Projetos?

A quem se destina este livro?	16
Sabemos o que o seu cérebro está pensando	16
Metacognição	18
Faça o seu cérebro obedecê-lo	19
Revisores técnicos	21
Agradecimentos	22

### Introdução aos Padrões de Projetos

#### 1 Bem-vindo aos Padrões de Projetos

**Alguém já resolveu os seus problemas.** Neste capítulo, você aprenderá por que (e como) pode se aproveitar da sabedoria e das lições aprendidas por outros desenvolvedores que tenham passado pelo mesmo campo minado dos problemas de projetos e tenham sobrevivido à viagem. Antes de terminarmos, veremos o uso e os benefícios dos padrões de projetos, veremos alguns princípios chaves dos projetos OO, e o guiaremos através de um exemplo de como um padrão funciona. A melhor maneira de usar os padrões é *carregando-os no seu cérebro* e depois *reconhecendo lugares* de seu projeto e das suas aplicações existentes onde pode *aplicá-los*. Em vez de reutilização de código, com padrões você tem uma reutilização de *experiência*.

Lembre-se: conhecer conceitos como abstração, herança e polimorfismo não o torna um bom projetista orientado a objetos. Um guru dos projetos pensa sobre como criar projetos flexíveis que sejam fáceis de manter e modificar.

Começou com um simples aplicativo SimUDuck	26
Joe pensa na herança...	28
Que tal uma interface?	28
A constante no desenvolvimento de software	30
Separando o que muda do que permanece igual	31
Desenvolvendo os comportamentos Duck	32
Testando o código Duck	37
Configurando o comportamento de forma dinâmica	39
Uma Visão Geral dos comportamentos encapsulados	40
TEM-UM pode ser melhor do que É-UM	41
O Padrão Estratégia	42
O poder de um vocabulário de padrões compartilhado	45
Como usar os Padrões de Projetos?	45
Ferramentas para a sua Caixa de Ferramentas de Projetos	47
Soluções dos Exercícios	50

## O Padrão Observer

### 2 Mantendo os seus Objetos informados

**Não durma no ponto quando algo interessante acontecer!** Temos um padrão que mantém os seus objetos informados quando algo que possa lhes interessar acontece. Os objetos podem até mesmo decidir, no momento da execução, se querem ser mantidos informados ou não. O Padrão Observer é um dos mais usados do JDK e é incrivelmente útil. Antes de terminarmos, veremos também as relações de um para muitos e o acoplamento solto (sim, isso mesmo, nós dissemos acoplamento). Com o Observer, você será a alma da Festa dos Padrões.

A aplicação Monitoração do Tempo	52
Conheça o Padrão Observer	56
Editores + Assinantes = Padrão Observer	56
Cinco minutos de Teatro: um assunto para observação	59
O Padrão Observer definido	60
O poder da Ligação Leve	62
Desenvolvendo a Estação do Tempo	64
Implementando a Estação do Tempo	64
Usando o Padrão Observer do Java	69
Retrabalhando a Estação Metereológica	72
Ferramentas para a sua Caixa de Ferramentas de Projetos	76
Soluções dos Exercícios	79

## O Padrão Decorator

### 3 Decorando Objetos

**Chame este capítulo de “Olho de design para o garoto da herança”.** Examinaremos novamente o típico uso excessivo da herança e você aprenderá a decorar suas classes no tempo de execução usando uma forma de composição de objeto. Por quê? Depois que você descobrir as técnicas de decoração, poderá dar novas responsabilidades a seus objetos (ou de outra pessoa) *sem fazer nenhuma mudança de código nas classes básicas*.

Eu achava que homens de verdade subclassificavam tudo. Isso até eu aprender o poder da extensão no tempo de execução, em vez de no tempo de compilação. Olhe só para mim agora!

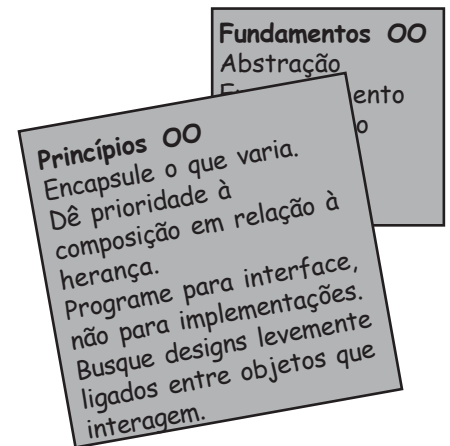
Bem-vindo ao Café Starbuzz	82
O Princípio Aberto-Fechado	87
Siga o Padrão Decorator	88
Construindo um Pedido de Bebida com Decorators	88
O Padrão Decorator Definido	89
Decorando as nossas Bebidas	90
Escrevendo o código Starbuzz	93
Decorators no Mundo Real: Java I/O	97
Escrevendo o seu próprio Decorator Java I/O	98
Ferramentas para a sua Caixa de Ferramentas de Projetos	100
Soluções dos Exercícios	101

## O Padrão Factory

### 4 Cozinhando com a polpa da programação OO

**Prepare-se para cozinhar alguns projetos OO com acoplamento solto.** Para se criar objetos, é preciso mais do que apenas usar o operador **new**. Você aprenderá que a chamada é uma atividade que não deve ser feita sempre em público e pode frequentemente levar a *problemas de acoplamento*. E você não quer que *isso* aconteça, quer? Descubra como os Padrões Factory podem ajudar a salvá-lo de dependências embaraçosas.

Quando vir “new”, pense em “concreto”	104
O que há de errado com o new	105
Encapsulando a criação de objetos	106
Construindo uma fábrica simples de pizza	107



O Simple Factory definido	109
Franqueando a pizzaria	110
Deixando as sub-classes decidirem	111
Vamos criar uma PizzaStore	113
Declarando um método de fábrica	114
Conheça o Padrão Factory Method	119
Hierarquias de Classes Paralelas	119
O Padrão Factory Method definindo	121
Uma PizzaStore muito dependente	123
Analisando as dependências dos objetos	124
O Princípio da Inversão da Dependência	124
Enquanto isso, de volta à PizzaStore...	128
Famílias de ingredientes...	129
Construindo as nossas fábricas de ingredientes	129
Observando a Abstract Factory	132
Nos bastidores	135
O Padrão Abstract Factory definido	137
Factory Method e Abstract Factory comparados	140
Ferramentas para a sua Caixa de Ferramentas de Projetos	142
Soluções dos Exercícios	144

## O Padrão Singleton

### 5 Objetos Únicos

**O Padrão Singleton: o seu ingresso para criar objetos únicos, para os quais há apenas uma instância.** Talvez você fique contente de saber que, de todos os padrões, o Singleton é o mais simples em termos do diagrama das suas classes; na verdade, o diagrama possui apenas uma classe! Mas não se sinta à vontade demais; apesar da sua simplicidade de um ponto de vista do design das classes, encontraremos algumas ondulações e alguns buracos na pista da sua implementação. Então aperte o cinto – este aqui não é tão simples quanto parece...

Um e apenas um objeto	150
O Pequeno Singleton	151
Dissecando o Padrão Singleton clássico	152
Confissões de um Singleton	153
A Fábrica de Chocolate	154
O Padrão Singleton definido	156
Hershey, PA	156
Houston, temos um problema...	157
SEJA o JVM	158
Lidando com vários segmentos	158
Podemos Melhorar os Vários Segmentos	162
Ferramentas para a sua Caixa de Ferramentas de Projetos	164
Soluções dos Exercícios	166

## O Padrão Command

### 6 Encapsulando a Chamada de Métodos

**Neste capítulo, levaremos o encapsulamento a um novo nível: vamos encapsular a chamada de métodos.** É isso mesmo – quando encapsulamos a chamada de métodos, podemos cristalizar partes da computação para que o objeto que está invocando esse processo não precise saber como as coisas são feitas. Ele só precisa usar o método cristalizado para executar a tarefa. O encapsulamento das chamadas de métodos também nos permite fazer algumas coisas muito engenhosas, como salvar os métodos encapsulados para fins de registro ou reutilizá-los para implementar recursos de “desfazer” no nosso código.

Automação doméstica ou Bust	168
O Controle Remoto	169
Examinando as classes de vendedor	169
Enquanto isso, de volta ao Restaurante	171
Vamos estudar a interação do Restaurante	172
Os Papéis e Responsabilidades no Restaurante da Objetolândia	172
Do Restaurante ao Padrão Command	174

Nosso primeiro objeto command	175
A definição do Padrão Command	178
Definição do Padrão Comando: o diagrama de classes	179
Implementando o Controle Remoto	180
Testando o Controle Remoto	182
Hora de escrever aquela documentação	185
Hora de testar aquele botão Desfazer!	188
Todo controle remoto precisa de um Modo de Festa	191
Usando um Macro Command	192
Mais usos do Padrão Command: enfileirando solicitações	194
Mais usos do Padrão Command: registrando solicitações	195
Ferramentas para a sua Caixa de Ferramentas de Projetos	196
Soluções dos Exercícios	198

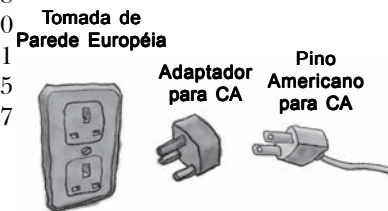
Singleton – Verifique se uma classe tem apenas uma instância e forneça um ponto global de acesso a ela.

## Os Padrões Adapter e Facade

### 7 Como ser Adaptável

Neste capítulo, iremos tentar feitos tão impossíveis quanto colocar um objeto quadrado em um buraco redondo. Parece impossível? Não quando temos os Padrões de Projetos. Lembra-se do Padrão Decorator? Nós **embrulhamos os objetos** para dar a eles novas responsabilidades. Agora iremos embrulhar alguns objetos com um propósito diferente: fazer com que as suas interfaces aparentem ser algo que não são. Por que faríamos algo assim? Para podermos adaptar um projeto que espera uma determinada interface a uma classe que implementa uma interface diferente. E não é só isso: enquanto estamos com a mão na massa, veremos também um outro padrão que embrulha objetos para simplificar as suas interfaces.

Estamos cercados de adaptadores	202
Adaptadores Orientados a Objetos	202
Explicando o Padrão Adapter	206
Defindo o Padrão Adapter	207
Adaptadores de Objetos e Classes	208
O assunto de hoje: o Objeto Adaptador e o Adaptador de Classes	210
Adaptadores no Mundo Real	211
Adaptando uma Enumeração a um Iterator	211
O assunto de hoje: o Padrão Decorator e o Padrão Adaptador	214
Todo mundo quer um Home Theater	215
Luzes, Câmera e Facade	217
Construindo a Fachada do seu Home Theater	218
Definição do Padrão Facade	220
O Princípio do Conhecimento Mínimo	221
Ferramentas para a sua Caixa de Ferramentas de Projetos	225
Soluções dos Exercícios	227



## O Padrão Template

### 8 Encapsulando Algoritmos

Já encapsulamos a criação de objetos, métodos, interfaces complexas, patos, pizzas... o que virá depois? Vamos encapsular *algoritmos* para que as subclasses possam se conectar diretamente a uma computação a qualquer momento que quiserem. Iremos aprender até mesmo um princípio de projetos inspirado em Hollywood.

Preparando algumas classes de café e chá	230
Senhor, posso abstrair seu café e seu chá ?	232
Aprimorando o Design	233
Abstraindo prepareRecipe()	233
O que nós fizemos?	236
Conheça o Template Method	236
Vamos fazer um pouco de chá	237
O que o Template Method pode fazer por nós?	238
Definindo o Padrão Template Method	238
Aprovitando o gancho do Template Method	240

Usando o gancho	241
Fazendo um Test-drive	242
O Princípio Hollywood	244
O Princípio Hollywood e o Template Method	244
O Método Gabarito a Olho Nu	246
Ordenando dados com o Template Method	246
Temos alguns patos para ordenar	247
Comparando patos com patos	250
A construção da máquina de ordenar patos	250
O Swing dos Quadros	251
Applets	252
O assunto de hoje: Template Method e Strategy	253
Ferramentas para a sua Caixa de Ferramentas de Projetos	256
Soluções dos Exercícios	257

## Os Padrões Iterator e Composite

### 9 Coleções bem gerenciadas

**Existem várias maneiras de inserir objetos em uma coleção.** Colocá-los em um Array, um Stack, uma Lista, um Mapa... você escolhe. Cada um tem as suas vantagens e desvantagens. Mas, quando o seu cliente quiser iterar através dos seus objetos, você vai deixá-lo ver a sua implementação? Certamente esperamos que não! Isso simplesmente *não* seria profissional. Não se preocupe – neste capítulo você verá como permitir aos seus clientes *iterar* através dos seus objetos sem jamais verem como você *armazena* os objetos. Você aprenderá também como criar *super coleções* de objetos capazes de saltar sobre algumas estruturas de dados impressionantes em uma única passada. Você aprenderá também algumas coisas sobre a responsabilidade dos objetos.

Fusão do Restaurante Objetolândia com a Panquecaria Objetolândia	260
Examinando os itens do Menu	260
Podemos encapsular a iteração?	265
Conheça o Padrão Iterator	267
Incorporando o Iterator ao Menu do Restaurante	268
Testando nosso código	270
Aprimorando o processo com o java.util.Iterator	273
Definindo o Padrão Iterator	276
Responsabilidade Única	278
Iterators e Coleções	285
Iterators e Coleções em Java 5	286
Justamente quando achávamos que estava pronto.	289
Definindo o Padrão Composite	291
Redesenhando os Menus com o Padrão Composite	293
Implementando o Componente de Menu	294
Voltando ao Iterator	302
O Iterator Nulo	305
A mágica de Iterator & Composite juntos...	306
Ferramentas para a sua Caixa de Ferramentas de Projetos	310
Soluções dos Exercícios	311

## O Padrão State

### 10 O Estado das Coisas

**Um fato pouco conhecido: os Padrões State e Strategy são irmãos gêmeos que foram separados ao nascer.** Como você sabe, o Padrão Strategy acabou se dando muito bem no negócio dos algoritmos intercambiáveis. O Padrão State, por sua vez, tomou o caminho talvez mais nobre de ajudar objetos a controlarem seu comportamento através de mudanças no seu estado interno: “Repita comigo: eu sou bastante bom, eu sou bastante esperto, e isso é o que importa...”.

Quebra queixos em java	314
Curso elementar de máquinas de estado	315
Teste Preliminar	319
Você já estava prevendo isso... um pedido de modificação !	322
O ESTADO confuso das coisas...	323

Definindo as interfaces e classes Estado	325
Implementando as nossas Classes Estado	326
Refazendo a Máquina de Gomas	327
Definindo o Padrão State	334
State x Strategy	340
Quase nos esquecemos!	341
Ferramentas para a sua Caixa de Ferramentas de Projetos	343
Soluções dos Exercícios	344

## O Padrão Proxy

### 11 Controlando o Acesso de Objetos

**Já brincou de tira bonzinho e tira malvado?** Você é o tira bonzinho e fornece todos os seus serviços de forma agradável e amistosa, mas você não quer que *todo mundo* peça os seus serviços, então você faz o tira malvado *controlar o acesso* a você. É isso que os proxies fazem: controlam e gerenciam o acesso. Como você verá, existem *várias* maneiras pelas quais os proxies substituem os seus respectivos objetos. Os proxies têm a fama de carregar chamadas a métodos inteiros através da Internet para os objetos que substituem; eles também têm a fama de pacientemente substituir alguns objetos bem preguiçosos.

Codificando o Monitor	350
O papel do 'proxy remoto'	352
Acrescentando um proxy remoto....	354
O desvio RMI	354
O proxy remoto nos bastidores	370
Definição do Padrão Proxy	372
Prepare-se para o proxy virtual	373
Projetando o proxy virtual para capa de CD	374
O proxy virtual nos bastidores	378
Usando o proxy da API Java	381
Peça teatral em cinco minutos: a proteção dos objetos	383
Criando um proxy dinâmico	384
O Zoológico dos Proxies	390
Ferramentas para a sua Caixa de Ferramentas de Projetos	392
Soluções dos Exercícios	393

## Padrões Compostos

### 12 Padrões de Padrões

**Quem poderia adivinhar que os Padrões conseguem trabalhar juntos?** Tendo presenciado as exaltadas Conversas ao Pé do Fogo (e você ainda nem viu as páginas do Combate Mortal entre Padrões, que o editor nos forçou a remover do livro), você certamente não imaginaria que ocasionalmente os padrões podem até conviver em paz. Bem, acredite ou não, alguns dos esquemas mais poderosos da programação orientada a objetos baseiam-se na utilização simultânea de vários padrões. Portanto, prepare-se para subir mais um degrau no seu domínio dos padrões: chegou a hora dos padrões compostos. Mas cuidado – os seus colegas podem ter que sacrificá-lo se você pegar a Febre dos Padrões

Trabalhando Juntos	398
Reunião de patos	398
O que nós fizemos?	417
Visão panorâmica: o diagrama de classes	418
O Rei dos Padrões: Canção Modelo-Vizualização-Controlador	420
Os Padrões de Projetos são a sua chave para o MVC	421
Examinando o MVC através de lentes coloridas por padrões	424
Usando o MVC para controlar a batida...	426
Conheça a Visualização do Java DJ	426
A Visualização	429
E agora o Controlador	432
Explorando a estratégia	435
Adaptando o modelo	435
Agora estamos prontos para um HeartController	436
MVC e a Internet	438



Padrões de Projetos e o Modelo 2	443
Ferramentas para a sua Caixa de Ferramentas de Projetos	446
Soluções dos Exercícios	447

## Vivendo Melhor com os Padrões

### 13 Padrões no Mundo Real

**Ahh, agora você está pronto para um brilhante mundo novo cheio de Padrões de Projetos.** Mas, antes de abrir todas essas novas portas de oportunidades, precisamos abordar alguns detalhes que você encontrará no mundo real – as coisas ficam um pouco mais complexas *lá fora* do que elas são na Objetolândia. Siga-nos, temos um bom guia para ajudá-lo na transição...

Guia da Objetolândia	460
Definindo o Padrão de Projetos	460
Examinando mais atentamente a definição de Padrão de Projetos	462
Que a força esteja com você	463
Então você quer ser um escritor de Padrões de Projetos?	466
Organizando os Padrões de Projetos	467
Pensando em Padrões	471
Padrões da Mente	473
Não se esqueça do poder do vocabulário compartilhado	474
As cinco melhores maneiras de compartilhar o seu vocabulário	475
Rodando a Objetolândia com a Gang of Four	475
Sua jornada só está começando...	476
Outros recursos sobre os Padrões de Projetos	477
O Zoológico dos Padrões	477
Aniquilando o mal com Anti-Padrões	478
Ferramentas para a sua Caixa de Ferramentas de Projetos	480
Deixando a Objetolândia...	481
Soluçãodo Exercícios	482

### 14 Apêndice: Padrões Restantes

**483**

**Nem todos podem ser os mais populares.** Muitas coisas mudaram nos últimos dez anos. Desde que saiu a primeira edição de *Padrões de Projetos: Elementos de Software Reutilizado Orientado a Objetos*, os desenvolvedores vêm aplicando esses padrões milhares de vezes. Os padrões que resumimos neste apêndice são padrões GoF (padrões fundamentais originais) oficiais completos e de carteirinha, mas não são sempre usados com a mesma frequência que os padrões que exploramos até aqui. Mas os padrões deste apêndice são eles mesmos excelentes e, se a situação exigir, você deve ser capaz de aplicá-los com orgulho. O nosso objetivo neste apêndice é lhe dar uma idéia geral do que são estes padrões.