
Refatoração

Aperfeiçoando o design de códigos existentes

Segunda Edição

Martin Fowler

com contribuições de Kent Beck

◆ Addison-Wesley

Novatec

Authorized translation from the English language edition, entitled REFACTORING: IMPROVING THE DESIGN OF EXISTING CODE, 2nd Edition by MARTIN FOWLER, published by Pearson Education, Inc, publishing as Addison-Wesley Professional, Copyright © 2019 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. PORTUGUESE language edition published by NOVATEC EDITORA LTDA., Copyright © 2019.

Tradução autorizada da edição original em inglês, intitulada REFACTORING: IMPROVING THE DESIGN OF EXISTING CODE, 2nd Edition by MARTIN FOWLER, publicada pela Pearson Education, Inc, publicando como Addison-Wesley Professional, Copyright © 2019 por Pearson Education, Inc.

Todos os direitos reservados. Nenhuma parte deste livro pode ser reproduzida ou transmitida por qualquer forma ou meio, eletrônica ou mecânica, incluindo fotocópia, gravação ou qualquer sistema de armazenamento de informação, sem a permissão da Pearson Education, Inc. Edição em Português publicada pela NOVATEC EDITORA LTDA., Copyright © 2019.

Editor: Rubens Prates

Tradução: Lúcia A. Kinoshita

Revisão gramatical: Tássia Carvalho

Editoração eletrônica: Carolina Kuwabata

ISBN: 978-85-7522-724-4

Histórico de impressões:

Abril/2020 Primeira edição

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

Email: novatec@novatec.com.br

Site: www.novatec.com.br

Twitter: twitter.com/novateceditora

Facebook: facebook.com/novatec

LinkedIn: linkedin.com/in/novatec

FC20200420

Sumário

<i>Apresentação da primeira edição</i>	9
<i>Prefácio</i>	11
Capítulo 1 ■ Refatoração: primeiro exemplo	18
Ponto de partida.....	18
Comentários sobre o programa inicial.....	21
Primeiro passo na refatoração	22
Decompondo a função statement	23
Status: muitas funções aninhadas	43
Separando as fases de cálculo e de formatação	44
Status: separado em dois arquivos (e fases).....	53
Reorganizando os cálculos por tipo.....	55
Status: criando os dados com a calculadora polimórfica	64
Considerações finais.....	66
Capítulo 2 ■ Princípios da refatoração	68
Definindo a refatoração	68
Dois chapéus	69
Por que devemos refatorar?	70
Quando devemos refatorar?	73
Problemas com a refatoração	79
Refatoração, arquitetura e Yagni	86
Refatoração e o processo mais amplo de desenvolvimento de software.....	88
Refatoração e desempenho	89
De onde veio a refatoração?	92
Refatorações automatizadas	94
Indo além	96
Capítulo 3 ■ “Maus cheiros” no código	97
Nome misterioso.....	98
Código duplicado.....	98
Função longa	99
Lista longa de parâmetros.....	100
Dados globais	101

Dados mutáveis.....	101
Alteração divergente	102
Cirurgia com rifle	103
Inveja de recursos	104
Agrupamentos de dados	104
Obsessão por primitivos	105
Switches repetidos	106
Laços	106
Elemento ocioso.....	106
Generalidade especulativa	107
Campo temporário.....	107
Cadeias de mensagens	108
Intermediário	108
Trocas escusas	109
Classe grande.....	109
Classes alternativas com interfaces diferentes	110
Classe de dados.....	110
Herança recusada.....	111
Comentários	111
Capítulo 4 ■ Escrevendo testes	113
Importância de um código autotestável	113
Código de exemplo para testar	115
Um teste inicial	119
Acrescente outro teste	122
Modificando o fixture	124
Sondando os limites	125
Muito além disso.....	128
Capítulo 5 ■ Apresentação do catálogo	130
Formato das refatorações	130
Escolha das refatorações	132
Capítulo 6 ■ Primeiro conjunto de refatorações	133
Extrair função (Extract Function)	134
Internalizar função (Inline Function)	144
Extrair variável (Extract Variable).....	147
Internalizar variável (Inline Variable).....	152
Mudar declaração de função (Change Function Declaration)	153
Encapsular variável (Encapsulate Variable)	160
Renomear variável (Rename Variable).....	165
Introduzir objeto de parâmetros (Introduce Parameter Object).....	168
Combinar funções em classe (Combine Functions into Class).....	172
Combinar funções em transformação (Combine Functions into Transform).....	177
Separar em fases (Split Phase)	183

Capítulo 7 ■ Encapsulamento.....	189
Encapsular registro (Encapsulate Record).....	190
Encapsular coleção (Encapsulate Collection).....	198
Substituir primitivo por objeto (Replace Primitive with Object).....	203
Substituir variável temporária por consulta (Replace Temp with Query).....	207
Extraír classe (Extract Class).....	211
Internalizar classe (Inline Class).....	215
Ocultar delegação (Hide Delegate).....	218
Remover intermediário (Remove Middle Man).....	220
Substituir algoritmo (Substitute Algorithm).....	223
Capítulo 8 ■ Movendo recursos.....	225
Mover função (Move Function).....	225
Mover campo (Move Field).....	235
Mover instruções para uma função (Move Statements into Function).....	241
Mover instruções para os pontos de chamada (Move Statements to Callers).....	245
Substituir código internalizado por chamada de função (Replace Inline Code with Function Call).....	251
Deslocar instruções (Slide Statements).....	252
Dividir laço (Split Loop).....	257
Substituir laço por pipeline (Replace Loop with Pipeline).....	261
Remover código morto (Remove Dead Code).....	267
Capítulo 9 ■ Organizando dados.....	269
Separar variável (Split Variable).....	269
Renomear campo (Rename Field).....	273
Substituir variável derivada por consulta (Replace Derived Variable with Query).....	277
Mudar referência para valor (Change Reference to Value).....	281
Mudar valor para referência (Change Value to Reference).....	284
Capítulo 10 ■ Simplificando lógicas condicionais.....	288
Decompor condicional (Decompose Conditional).....	288
Consolidar expressão condicional (Consolidate Conditional Expression).....	291
Substituir condicional aninhada por cláusulas de guarda (Replace Nested Conditional with Guard Clauses).....	294
Substituir condicional por polimorfismo (Replace Conditional with Polymorphism).....	299
Introduzir caso especial (Introduce Special Case).....	318
Introduzir asserção (Introduce Assertion).....	333
Capítulo 11 ■ Refatorando APIs.....	336
Separar consulta de modificador (Separate Query from Modifier).....	337
Parametrizar função (Parameterize Function).....	340
Remover argumento de flag (Remove Flag Argument).....	344
Preservar objeto inteiro (Preserve Whole Object).....	349

Substituir parâmetro por consulta (Replace Parameter with Query).....	354
Substituir consulta por parâmetro (Replace Query with Parameter).....	357
Remover método de escrita (Remove Setting Method).....	361
Substituir construtor por função de factory (Replace Constructor with Factory Function).....	363
Substituir função por comando (Replace Function with Command).....	366
Substituir comando por função (Replace Command with Function).....	373
Capítulo 12 ▀ Lidando com herança	378
Subir método (Pull Up Method)	378
Subir campo (Pull Up Field)	381
Subir corpo do construtor (Pull Up Constructor Body).....	383
Descer método (Push Down Method).....	387
Descer campo (Push Down Field)	388
Substituir código de tipos por subclasses (Replace Type Code with Subclasses)	389
Remover subclasse (Remove Subclass).....	397
Extrair superclasse (Extract Superclass).....	402
Condensar Hierarquia (Collapse Hierarchy).....	408
Substituir subclasse por delegação (Replace Subclass with Delegate)	409
Substituir superclasse por delegação (Replace Superclass with Delegate).....	429
<i>Bibliografia</i>.....	436
<i>Índice remissivo</i>.....	439