Software Recycling p. 359
Software Maintenance as an Engineering Discipline p. 368
Software Reverse Engineering: A Case Study p. 374
Program Translation via Abstraction and Reimplementation p. 390
Annotating and Documenting Existing Programs p. 415
A Model for Assembly Program Maintenance p. 416
Using Function Abstraction to Understand Program Behavior p. 446
Documentation in a Software Maintenance Environment p. 455
Recognizing Design Decisions in Programs p. 463
Reengineering for Reuse p. 475
Software Reuse and Reengineering p. 476
Identifying and Qualifying Reusable Software Components p. 485
Software Reclamation: Improving Post-Development Reusability p. 495
Software Reclamation p. 510
Reverse Engineering and Design Recovery p. 519
Design Recovery for Maintenance and Reuse p. 520
Recognizing a Program’s Design: A Graph-Parsing Approach p. 534
Creating Specifications from Code: Reverse-engineering Techniques p. 542
Object Recovery p. 563
Re-engineering of Old Systems to an Object-Oriented Architecture p. 564
Saving a Legacy with Objects p. 575
Software Reuse in an Industrial Setting: A Case Study p. 582
Program Understanding p. 595
Program Understanding: Challenge for the 1990s p. 596
Approaches to Program Comprehension p. 609
Program Recognition p. 615
Knowledge-Based Program Analysis p. 631
SRE: A Knowledge-Based Environment for Large Scale Software Re-engineering Activities p. 632
A Knowledge-Based Approach to Software System Understanding p. 642
Knowledge-Based Program Analysis p. 651
Annotated Bibliography p. 659
About the Author p. 675
Table of Contents provided by Blackwell's Book Services and R.R. Bowker. Used with permission.