

Addison-Wesley Professional Ruby Series



DESIGN PATTERNS IN RUBY

Foreword by **Obie Fernandez**
Professional Ruby Series Editor

RUSS OLSEN

Praise for Design Patterns in Ruby

“*Design Patterns in Ruby* documents smart ways to resolve many problems that Ruby developers commonly encounter. Russ Olsen has done a great job of selecting classic patterns and augmenting these with newer patterns that have special relevance for Ruby. He clearly explains each idea, making a wealth of experience available to Ruby developers for their own daily work.”

—Steve Metsker, Managing Consultant
with Dominion Digital, Inc.

“This book provides a great demonstration of the key ‘Gang of Four’ design patterns without resorting to overly technical explanations. Written in a precise, yet almost informal style, this book covers enough ground that even those without prior exposure to design patterns will soon feel confident applying them using Ruby. Olsen has done a great job to make a book about a classically ‘dry’ subject into such an engaging and even occasionally humorous read.”

—Peter Cooper

“This book renewed my interest in understanding patterns after a decade of good intentions. Russ picked the most useful patterns for Ruby and introduced them in a straightforward and logical manner, going beyond the GoF’s patterns. This book has improved my use of Ruby, and encouraged me to blow off the dust covering the GoF book.”

—Mike Stok

“*Design Patterns in Ruby* is a great way for programmers from statically typed object-oriented languages to learn how design patterns appear in a more dynamic, flexible language like Ruby.”

—Rob Sanheim, Ruby Ninja, Relevance

Design Patterns in Ruby (Adobe Reader)

Table of Contents

Contents

Foreword

Preface

Acknowledgments

About the Author

PART I: Patterns and Ruby

Chapter 1: Building Better Programs with Patterns

The Gang of Four

Patterns for Patterns

Fourteen Out of Twenty-Three

Patterns in Ruby?

Chapter 2: Getting Started with Ruby

Interactive Ruby

Saying Hello World

Variables

Fixnums and Bignums

Floats

There Are No Primitives Here

But Sometimes There Is No Object

Truth, Lies, and nil

Decisions, Decisions

Table of Contents

Loops
More about Strings
Symbols
Arrays
Hashes
Regular Expressions
A Class of Your Own
Getting at the Instance Variables
An Object Asks: Who Am I?
Inheritance, Subclasses, and Superclasses
Argument Options
Modules
Exceptions
Threads
Managing Separate Source Files
Wrapping Up

PART II: Patterns in Ruby

Chapter 3: Varying the Algorithm with the Template Method

Keeping Up with What Life Throws at You
Separate the Things That Stay the Same
Discovering the Template Method Pattern
Hook Methods
But Where Are All the Declarations?
Types, Safety, and Flexibility
Unit Tests Are Not Optional
Using and Abusing the Template Method Pattern
Templates in the Wild

Table of Contents

Wrapping Up

Chapter 4: Replacing the Algorithm with the Strategy

Delegate, Delegate, and Delegate Again

Sharing Data between the Context and the Strategy

Duck Typing Yet Again

Procs and Blocks

Quick-and-Dirty Strategies

Using and Abusing the Strategy Pattern

The Strategy Pattern in the Wild

Wrapping Up

Chapter 5: Keeping Up with the Times with the Observer

Staying Informed

A Better Way to Stay Informed

Factoring Out the Observable Support

Code Blocks as Observers

Variations on the Observer Pattern

Using and Abusing the Observer Pattern

Observers in the Wild

Wrapping Up

Chapter 6: Assembling the Whole from the Parts with the Composite

The Whole and the Parts

Creating Composites

Sprucing Up the Composite with Operators

An Array as a Composite?

An Inconvenient Difference

Pointers This Way and That

Using and Abusing the Composite Pattern

Table of Contents

Composites in the Wild

Wrapping Up

Chapter 7: Reaching into a Collection with the Iterator

External Iterators

Internal Iterators

Internal Iterators versus External Iterators

The Inimitable Enumerable

Using and Abusing the Iterator Pattern

Iterators in the Wild

Wrapping Up

Chapter 8: Getting Things Done with Commands

An Explosion of Subclasses

An Easier Way

Code Blocks as Commands

Commands That Record

Being Undone by a Command

Queuing Up Commands

Using and Abusing the Command Pattern

The Command Pattern in the Wild

Wrapping Up

Chapter 9: Filling in the Gaps with the Adapter

Software Adapters

The Near Misses

An Adaptive Alternative?

Modifying a Single Instance

Adapt or Modify?

Using and Abusing the Adapter Pattern

Adapters in the Wild

Table of Contents

Wrapping Up

Chapter 10: Getting in Front of Your Object with a Proxy

Proxies to the Rescue

The Protection Proxy

Remote Proxies

Virtual Proxies Make You Lazy

Eliminating That Proxy Drudgery

Using and Abusing Proxies

Proxies in the Wild

Wrapping Up

Chapter 11: Improving Your Objects with a Decorator

Decorators: The Cure for Ugly Code

Formal Decoration

Easing the Delegation Blues

Dynamic Alternatives to the Decorator Pattern

Using and Abusing the Decorator Pattern

Decorators in the Wild

Wrapping Up

Chapter 12: Making Sure There Is Only One with the Singleton

One Object, Global Access

Class Variables and Methods

A First Try at a Ruby Singleton

The Singleton Module

Lazy and Eager Singletons

Alternatives to the Classic Singleton

A Safety Harness or a Straitjacket?

Using and Abusing the Singleton Pattern

Table of Contents

Singletons in the Wild

Wrapping Up

Chapter 13: Picking the Right Class with a Factory

A Different Kind of Duck Typing

The Template Method Strikes Again

Parameterized Factory Methods

Classes Are Just Objects, Too

Bad News: Your Program Hits the Big Time

Bundles of Object Creation

Classes Are Just Objects (Again)

Leveraging the Name

Using and Abusing the Factory Patterns

Factory Patterns in the Wild

Wrapping Up

Chapter 14: Easier Object Construction with the Builder

Building Computers

Polymorphic Builders

Builders Can Ensure Sane Objects

Reusable Builders

Better Builders with Magic Methods

Using and Abusing the Builder Pattern

Builders in the Wild

Wrapping Up

Chapter 15: Assembling Your System with the Interpreter

The Right Language for the Job

Building an Interpreter

A File-Finding Interpreter

Table of Contents

Creating the AST

Using and Abusing the Interpreter Pattern

Interpreters in the Wild

Wrapping Up

PART III: Patterns for Ruby

Chapter 16: Opening Up Your System with Domain-Specific Languages

The Domain of Specific Languages

A File Backup DSL

Its a Data FileNo, Its a Program!

Building PackRat

Pulling Our DSL Together

Taking Stock of PackRat

Improving PackRat

Using and Abusing Internal DSLs

Internal DSLs in the Wild

Wrapping Up

Chapter 17: Creating Custom Objects with Meta-programming

Custom-Tailored Objects, Method by Method

Custom Objects, Module by Module

Conjuring Up Brand-New Methods

An Objects Gaze Turns Inward

Using and Abusing Meta-programming

Meta-programming in the Wild

Wrapping Up

Chapter 18: Convention Over Configuration

A Good User Interface for Developers

Table of Contents

A Message Gateway

Picking an Adapter

Loading the Classes

Adding Some Security

Getting the User Started

Taking Stock of the Message Gateway

Using and Abusing the Convention Over Configuration Pattern

Convention Over Configuration in the Wild

Wrapping Up

Chapter 19: Conclusion

Appendix A: Getting Hold of Ruby

Installing Ruby on Microsoft Windows

Installing Ruby on Linux and Other UNIX-Style Systems

Mac OS X

Appendix B: Digging Deeper

Design Patterns

Ruby

Regular Expressions

Blogs and Web Sites

Index