

GLOBAL
EDITION



Building Java™ Programs

A Back to Basics Approach

FOURTH EDITION

Stuart Reges • Marty Stepp



Fourth Edition
Global Edition

Building Java Programs

A Back to Basics Approach

Stuart Reges
University of Washington

Marty Stepp
Stanford University



Pearson

Boston Columbus Indianapolis New York San Francisco Hoboken
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City Sao Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Building Java Programs: A Back to Basics Approach, Global Edition

Table of Contents

Cover

Title Page

Copyright Page

Preface

Brief Contents

Contents

Chapter 1: Introduction to Java Programming

1.1. Basic Computing Concepts

Why Programming?

Hardware and Software

The Digital Realm

The Process of Programming

Why Java?

The Java Programming Environment

1.2. And NowJava

String Literals (Strings)

System.out.println

Escape Sequences

print versus println

Identifiers and Keywords

A Complex Example: DrawFigures1

Comments and Readability

Table of Contents

1.3. Program Errors

- Syntax Errors

- Logic Errors (Bugs)

1.4. Procedural Decomposition

- Static Methods

- Flow of Control

- Methods That Call Other Methods

- An Example Runtime Error

1.5. Case Study: DrawFigures

- Structured Version

- Final Version without Redundancy

- Analysis of Flow of Execution

Chapter 2: Primitive Data and Definite Loops

2.1. Basic Data Concepts

- Primitive Types

- Expressions

- Literals

- Arithmetic Operators

- Precedence

- Mixing Types and Casting

2.2. Variables

- Assignment/Declaration Variations

- String Concatenation

- Increment/Decrement Operators

- Variables and Mixing Types

2.3. The for Loop

- Tracing for Loops

- for Loop Patterns

Table of Contents

Nested for Loops

2.4. Managing Complexity

Scope

Pseudocode

Class Constants

2.5. Case Study: Hourglass Figure

Problem Decomposition and Pseudocode

Initial Structured Version

Adding a Class Constant

Further Variations

Chapter 3: Introduction to Parameters and Objects

3.1. Parameters

The Mechanics of Parameters

Limitations of Parameters

Multiple Parameters

Parameters versus Constants

Overloading of Methods

3.2. Methods That Return Values

The Math Class

Defining Methods That Return Values

3.3. Using Objects

String Objects

Interactive Programs and Scanner Objects

Sample Interactive Program

3.4. Case Study: Projectile Trajectory

Unstructured Solution

Structured Solution

Supplement 3G: Graphics (Optional)

Table of Contents

3G.1. Introduction to Graphics

DrawingPanel

Drawing Lines and Shapes

Colors

Drawing with Loops

Text and Fonts

Images

3G.2. Procedural Decomposition with Graphics

A Larger Example: DrawDiamonds

3G.3. Case Study: Pyramids

Unstructured Partial Solution

Generalizing the Drawing of Pyramids

Complete Structured Solution

Chapter 4: Conditional Execution

4.1. if/else Statements

Relational Operators

Nested if/else Statements

Object Equality

Factoring if/else Statements

Testing Multiple Conditions

4.2. Cumulative Algorithms

Cumulative Sum

Min/Max Loops

Cumulative Sum with if

Roundoff Errors

4.3. Text Processing

The char Type

char versus int

Table of Contents

Cumulative Text Algorithms

System.out.printf

4.4. Methods with Conditional Execution

Preconditions and Postconditions

Throwing Exceptions

Revisiting Return Values

Reasoning about Paths

4.5. Case Study: Body Mass Index

One-Person Unstructured Solution

Two-Person Unstructured Solution

Two-Person Structured Solution

Procedural Design Heuristics

Chapter 5: Program Logic and Indefinite Loops

5.1. The while Loop

A Loop to Find the Smallest Divisor

Random Numbers

Simulations

do/while Loop

5.2. Fencepost Algorithms

Sentinel Loops

Fencepost with if

5.3. The boolean Type

Logical Operators

Short-Circuited Evaluation

boolean Variables and Flags

Boolean Zen

Negating Boolean Expressions

5.4. User Errors

Table of Contents

Scanner Lookahead

Handling User Errors

5.5. Assertions and Program Logic

Reasoning about Assertions

A Detailed Assertions Example

5.6. Case Study: NumberGuess

Initial Version without Hinting

Randomized Version with Hinting

Final Robust Version

Chapter 6: File Processing

6.1. File-Reading Basics

Data, Data Everywhere

Files and File Objects

Reading a File with a Scanner

6.2. Details of Token-Based Processing

Structure of Files and Consuming Input

Scanner Parameters

Paths and Directories

A More Complex Input File

6.3. Line-Based Processing

String Scanners and Line/Token Combinations

6.4. Advanced File Processing

Output Files with PrintStream

Guaranteeing That Files Can Be Read

6.5. Case Study: Zip Code Lookup

Chapter 7: Arrays

7.1. Array Basics

Constructing and Traversing an Array

Table of Contents

Accessing an Array

A Complete Array Program

Random Access

Arrays and Methods

The For-Each Loop

Initializing Arrays

The Arrays Class

7.2. Array-Traversal Algorithms

Printing an Array

Searching and Replacing

Testing for Equality

Reversing an Array

String Traversal Algorithms

Functional Approach

7.3. Reference Semantics

Multiple Objects

7.4. Advanced Array Techniques

Shifting Values in an Array

Arrays of Objects

Command-Line Arguments

Nested Loop Algorithms

7.5. Multidimensional Arrays

Rectangular Two-Dimensional Arrays

Jagged Arrays

7.6. Arrays of Pixels

7.7. Case Study: Benfords Law

Tallying Values

Completing the Program

Table of Contents

Chapter 8: Classes

8.1. Object-Oriented Programming

Classes and Objects

Point Objects

8.2. Object State and Behavior

Object State: Fields

Object Behavior: Methods

The Implicit Parameter

Mutators and Accessors

The toString Method

8.3. Object Initialization: Constructors

The Keyword this

Multiple Constructors

8.4. Encapsulation

Private Fields

Class Invariants

Changing Internal Implementations

8.5. Case Study: Designing a Stock Class

Object-Oriented Design Heuristics

Stock Fields and Method Headers

Stock Method and Constructor Implementation

Chapter 9: Inheritance and Interfaces

9.1. Inheritance Basics

Nonprogramming Hierarchies

Extending a Class

Overriding Methods

9.2. Interacting with the Superclass

Calling Overridden Methods

Table of Contents

- Accessing Inherited Fields
- Calling a Superclasss Constructor
- DividendStock Behavior
- The Object Class
- The equals Method
- The instanceof Keyword

9.3. Polymorphism

- Polymorphism Mechanics
- Interpreting Inheritance Code
- Interpreting Complex Calls

9.4. Inheritance and Design

- A Misuse of Inheritance
- Is-a Versus Has-a Relationships
- Graphics2D

9.5. Interfaces

- An Interface for Shapes
- Implementing an Interface
- Benefits of Interfaces

9.6. Case Study: Financial Class Hierarchy

- Designing the Classes
- Redundant Implementation
- Abstract Classes

Chapter 10: ArrayLists

10.1. ArrayLists

- Basic ArrayList Operations
- ArrayList Searching Methods
- A Complete ArrayList Program
- Adding to and Removing from an ArrayList

Table of Contents

Using the For-Each Loop with ArrayLists

Wrapper Classes

10.2. The Comparable Interface

Natural Ordering and compareTo

Implementing the Comparable Interface

10.3. Case Study: Vocabulary Comparison

Some Efficiency Considerations

Version 1: Compute Vocabulary

Version 2: Compute Overlap

Version 3: Complete Program

Chapter 11: Java Collections Framework

11.1. Lists

Collections

LinkedList versus ArrayList

Iterators

Abstract Data Types (ADTs)

LinkedList Case Study: Sieve

11.2. Sets

Set Concepts

TreeSet versus HashSet

Set Operations

Set Case Study: Lottery

11.3. Maps

Basic Map Operations

Map Views (keySet and values)

TreeMap versus HashMap

Map Case Study: WordCount

Collection Overview

Table of Contents

Chapter 12: Recursion

12.1. Thinking Recursively

A Nonprogramming Example

An Iterative Solution Converted to Recursion

Structure of Recursive Solutions

12.2. A Better Example of Recursion

Mechanics of Recursion

12.3. Recursive Functions and Data

Integer Exponentiation

Greatest Common Divisor

Directory Crawler

Helper Methods

12.4. Recursive Graphics

12.5. Recursive Backtracking

A Simple Example: Traveling North/East

8 Queens Puzzle

Solving Sudoku Puzzles

12.6. Case Study: Prefix Evaluator

Infix, Prefix, and Postfix Notation

Evaluating Prefix Expressions

Complete Program

Chapter 13: Searching and Sorting

13.1. Searching and Sorting in the Java Class Libraries

Binary Search

Sorting

Shuffling

Custom Ordering with Comparators

13.2. Program Complexity

Table of Contents

Empirical Analysis

Complexity Classes

13.3. Implementing Searching and Sorting Algorithms

Sequential Search

Binary Search

Recursive Binary Search

Searching Objects

Selection Sort

13.4. Case Study: Implementing Merge Sort

Splitting and Merging Arrays

Recursive Merge Sort

Complete Program

Chapter 14: Stacks and Queues

14.1. Stack/Queue Basics

Stack Concepts

Queue Concepts

14.2. Common Stack/Queue Operations

Transferring Between Stacks and Queues

Sum of a Queue

Sum of a Stack

14.3. Complex Stack/Queue Operations

Removing Values from a Queue

Comparing Two Stacks for Similarity

14.4. Case Study: Expression Evaluator

Splitting into Tokens

The Evaluator

Chapter 15: Implementing a Collection Class

15.1. Simple ArrayIntList

Table of Contents

Adding and Printing

Thinking about Encapsulation

Dealing with the Middle of the List

Another Constructor and a Constant

Preconditions and Postconditions

15.2. A More Complete ArrayList

Throwing Exceptions

Convenience Methods

15.3. Advanced Features

Resizing When Necessary

Adding an Iterator

15.4. ArrayList<E>

Chapter 16: Linked Lists

16.1. Working with Nodes

Constructing a List

List Basics

Manipulating Nodes

Traversing a List

16.2. A Linked List Class

Simple LinkedList

Appending

The Middle of the List

16.3. A Complex List Operation

Inchworm Approach

16.4. An IntList Interface

16.5. LinkedList<E>

Linked List Variations

Linked List Iterators

Table of Contents

Other Code Details

Chapter 17: Binary Trees

17.1. Binary Tree Basics

Node and Tree Classes

17.2. Tree Traversals

Constructing and Viewing a Tree

17.3. Common Tree Operations

Sum of a Tree

Counting Levels

Counting Leaves

17.4. Binary Search Trees

The Binary Search Tree Property

Building a Binary Search Tree

The Pattern $x = \text{change}(x)$

Searching the Tree

Binary Search Tree Complexity

17.5. SearchTree<E>

Chapter 18: Advanced Data Structures

18.1 Hashing

Array Set Implementations

Hash Functions and Hash Tables

Collisions

Rehashing

Hashing Non-Integer Data

Hash Map Implementation

18.2. Priority Queues and Heaps

Priority Queues

Introduction to Heaps

Table of Contents

- Removing from a Heap
- Adding to a Heap
- Array Heap Implementation
- Heap Sort

Chapter 19: Functional Programming with Java 8

19.1. Effect-Free Programming

19.2. First-Class Functions

- Lambda Expressions

19.3. Streams

- Basic Idea

- Using Map

- Using Filter

- Using Reduce

- Optional Results

19.4. Function Closures

19.5. Higher-Order Operations on Collections

- Working with Arrays

- Working with Lists

- Working with Files

19.6. Case Study: Perfect Numbers

- Computing Sums

- Incorporating Square Root

- Just Five and Leveraging Concurrency

Appendix A: Java Summary

Appendix B: The Java API Specification and Javadoc Comments

Appendix C: Additional Java Syntax

Index

Table of Contents

Credits

Back Cover