

INTERNATIONAL  
EDITION



# Java Software Structures

## *Designing and Using Data Structures*

FOURTH EDITION

John Lewis • Joseph Chase

ALWAYS LEARNING

PEARSON

*This page is intentionally left blank.*

# **eBook Instant Access - for Java Software Structures, International Edition**

## **Table of Contents**

Cover

Preface

Contents

Credits

Chapter 1: Introduction

1.1 Software Quality

Correctness

Reliability

Robustness

Usability

Maintainability

Reusability

Portability

Efficiency

Quality Issues

1.2 Data Structures

A Physical Example

Containers as Objects

Summary of Key Concepts

Chapter 2: Analysis of Algorithms

2.1 Algorithm Efficiency

2.2 Growth Functions and Big-Oh Notation

2.3 Comparing Growth Functions

2.4 Determining Time Complexity

Analyzing Loop Execution

Nested Loops

Method Calls

Summary of Key Concepts

Chapter 3: Introduction to Collections Stacks

3.1 Collections

Abstract Data Types

The Java Collections API

3.2 A Stack Collection

3.3 Crucial OO Concepts

Inheritance and Polymorphism

Generics

# Table of Contents

## 3.4 Using Stacks: Evaluating Postfix Expressions

Javadoc

## 3.5 Exceptions

## 3.6 A Stack ADT

## 3.7 Implementing a Stack: With Arrays

Managing Capacity

## 3.8 The ArrayStack Class

The Constructors

The push Operation

The pop Operation

The peek Operation

Other Operations

The EmptyCollectionException Class

Other Implementations

## Summary of Key Concepts

## Chapter 4: Linked Structures Stacks

### 4.1 References as Links

### 4.2 Managing Linked Lists

Accessing Elements

Inserting Nodes

Deleting Nodes

### 4.3 Elements without Links

Doubly Linked Lists

### 4.4 Stacks in the Java API

### 4.5 Using Stacks: Traversing a Maze

### 4.6 Implementing a Stack: With Links

The LinkedStack Class

The push Operation

The pop Operation

Other Operations

## Summary of Key Concepts

## Chapter 5: Queues

### 5.1 A Conceptual Queue

### 5.2 Queues in the Java API

### 5.3 Using Queues: Code Keys

### 5.4 Using Queues: Ticket Counter Simulation

### 5.5 A Queue ADT

### 5.6 A Linked Implementation of a Queue

The enqueue Operation



# Table of Contents

The dequeue Operation

Other Operations

## 5.7 Implementing Queues: With Arrays

The enqueue Operation

The dequeue Operation

Other Operations

## 5.8 Double-Ended Queues (Deque)

Summary of Key Concepts

## Chapter 6: Lists

### 6.1 A List Collection

### 6.2 Lists in the Java Collections API

### 6.3 Using Unordered Lists: Program of Study

### 6.4 Using Indexed Lists: Josephus

### 6.5 A List ADT

Adding Elements to a List

### 6.6 Implementing Lists with Arrays

The remove Operation

The contains Operation

The add Operation for an Ordered List

Operations Particular to Unordered Lists

The addAfter Operation for an Unordered List

### 6.7 Implementing Lists with Links

The remove Operation

Summary of Key Concepts

## Chapter 7: Iterators

### 7.1 What's an Iterator?

Other Iterator Issues

### 7.2 Using Iterators: Program of Study Revisited

Printing Certain Courses

Removing Courses

### 7.3 Implementing Iterators: With Arrays

### 7.4 Implementing Iterators: With Links

Summary of Key Concepts

## Chapter 8: Recursion

### 8.1 Recursive Thinking

Infinite Recursion

Recursion in Math

### 8.2 Recursive Programming

Recursion versus Iteration

# Table of Contents

Direct versus Indirect Recursion

## 8.3 Using Recursion

Traversing a Maze

The Towers of Hanoi

## 8.4 Analyzing Recursive Algorithms

Summary of Key Concepts

## Chapter 9: Searching and Sorting

### 9.1 Searching

Static Methods

Generic Methods

Linear Search

Binary Search

Comparing Search Algorithms

### 9.2 Sorting

Selection Sort

Insertion Sort

Bubble Sort

Quick Sort

Merge Sort

### 9.3 Radix Sort

Summary of Key Concepts

## Chapter 10: Trees

### 10.1 Trees

Tree Classifications

### 10.2 Strategies for Implementing Trees

Computational Strategy for Array Implementation of Trees

Simulated Link Strategy for Array Implementation of Trees

Analysis of Trees

### 10.3 Tree Traversals

Preorder Traversal

Inorder Traversal

Postorder Traversal

Level-Order Traversal

### 10.4 A Binary Tree ADT

### 10.5 Using Binary Trees: Expression Trees

### 10.6 A Back Pain Analyzer

### 10.7 Implementing Binary Trees with Links

The find Method

The iteratorInOrder Method

Summary of Key Concepts

# Table of Contents

## Chapter 11: Binary Search Trees

### 11.1 A Binary Search Tree

### 11.2 Implementing Binary Search Trees: With Links

The addElement Operation

The removeElement Operation

The removeAllOccurrences Operation

The removeMin Operation

Implementing Binary Search Trees: With Arrays

### 11.3 Using Binary Search Trees: Implementing Ordered Lists

Analysis of the BinarySearchTreeList Implementation

### 11.4 Balanced Binary Search Trees

Right Rotation

Left Rotation

Rightleft Rotation

Leftright Rotation

### 11.5 Implementing BSTs: AVL Trees

Right Rotation in an AVL Tree

Left Rotation in an AVL Tree

Rightleft Rotation in an AVL Tree

Leftright Rotation in an AVL Tree

### 11.6 Implementing BSTs: Red/Black Trees

Insertion into a Red/Black Tree

Element Removal from a Red/Black Tree

### Summary of Key Concepts

## Chapter 12: Heaps and Priority Queues

### 12.1 A Heap

The addElement Operation

The removeMin Operation

The findMin Operation

### 12.2 Using Heaps: Priority Queues

### 12.3 Implementing Heaps: With Links

The addElement Operation

The removeMin Operation

The findMin Operation

### 12.4 Implementing Heaps: With Arrays

The addElement Operation

The removeMin Operation

The findMin Operation

### 12.5 Using Heaps: Heap Sort

### Summary of Key Concepts

# Table of Contents

## Chapter 13: Sets and Maps

- 13.1 Set and Map Collections
- 13.2 Sets and Maps in the Java API
- 13.3 Using Sets: Domain Blocker
- 13.4 Using Maps: Product Sales
- 13.5 Using Maps: User Management
- 13.6 Implementing Sets and Maps Using Trees
- 13.7 Implementing Sets and Maps Using Hashing
- Summary of Key Concepts

## Chapter 14: Multi-Way Search Trees

- 14.1 Combining Tree Concepts
- 14.2 2-3 Trees
  - Inserting Elements into a 2-3 Tree
  - Removing Elements from a 2-3 Tree
- 14.3 2-4 Trees
- 14.4 B-Trees
  - B\*-Trees
  - B+-Trees
  - Analysis of B-Trees
- 14.5 Implementation Strategies for B-Trees
- Summary of Key Concepts

## Chapter 15: Graphs

- 15.1 Undirected Graphs
- 15.2 Directed Graphs
- 15.3 Networks
- 15.4 Common Graph Algorithms
  - Traversals
  - Testing for Connectivity
  - Minimum Spanning Trees
  - Determining the Shortest Path
- 15.5 Strategies for Implementing Graphs
  - Adjacency Lists
  - Adjacency Matrices
- 15.6 Implementing Undirected Graphs with an Adjacency Matrix
  - The addEdge Method
  - The addVertex Method
  - The expandCapacity Method
  - Other Methods
- Summary of Key Concepts



# **Table of Contents**

## **Appendix A: UML**

The Unified Modeling Language (UML)

UML Class Diagrams

UML Relationships

Summary of Key Concepts

## **Appendix B: Object-Oriented Design**

B.1 Overview of Object Orientation

B.2 Using Objects

Abstraction

Creating Objects

B.3 Class Libraries and Packages

The import Declaration

B.4 State and Behavior

B.5 Classes

Instance Data

B.6 Encapsulation

Visibility Modifiers

Local Data

B.7 Constructors

B.8 Method Overloading

B.9 References Revisited

The Null Reference

The this Reference

Aliases

Garbage Collection

Passing Objects as Parameters

B.10 The static Modifier

Static Variables

Static Methods

B.11 Wrapper Classes

B.12 Interfaces

The Comparable Interface

B.13 Inheritance

Derived Classes

The protected Modifier

The super Reference

Overriding Methods

B.14 Class Hierarchies

The Object Class

# Table of Contents

Abstract Classes

Interface Hierarchies

## **B.15 Polymorphism**

References and Class Hierarchies

Polymorphism via Inheritance

Polymorphism via Interfaces

## **B.16 Exceptions**

Exception Messages

The try Statement

Exception Propagation

The Exception Class Hierarchy

## **Summary of Key Concepts**

## **Appendix C: Java Graphics**

C.1 Pixels and Coordinates

C.2 Representing Color

C.3 Drawing Shapes

C.4 Polygons and Polylines

The Polygon Class

## **Appendix D: Graphical User Interfaces**

### **D.1 GUI Elements**

Frames and Panels

Buttons and Action Events

Determining Event Sources

### **D.2 More Components**

Text Fields

Check Boxes

Radio Buttons

Sliders

Combo Boxes

Timers

### **D.3 Layout Managers**

Flow Layout

Border Layout

Grid Layout

Box Layout

Containment Hierarchies

### **D.4 Mouse and Key Events**

Mouse Events

Key Events

Extending Adapter Classes

# Table of Contents

## D.5 Dialog Boxes

- File Choosers

- Color Choosers

## D.6 Some Important Details

- Borders

- Tool Tips and Mnemonics

## D.7 GUI Design

- Summary of Key Concepts

## Appendix E: Hashing

### E.1 Hashing

### E.2 Hashing Functions

- The Division Method

- The Folding Method

- The Mid-Square Method

- The Radix Transformation Method

- The Digit Analysis Method

- The Length-Dependent Method

- Hashing Functions in the Java Language

### E.3 Resolving Collisions

- Chaining

- Open Addressing

### E.4 Deleting Elements from a Hash Table

- Deleting from a Chained Implementation

- Deleting from an Open Addressing Implementation

### E.5 Hash Tables in the Java Collections API

- The Hashtable Class

- The HashSet Class

- The HashMap Class

- The IdentityHashMap Class

- The WeakHashMap Class

- LinkedHashSet and LinkedHashMap

- Summary of Key Concepts

## Appendix F: Regular Expressions

## Index