

Program Development in Java



Abstraction,
Specification, and
Object-Oriented Design

Barbara Liskov
with John Guttag

Program Development in Java

Program Development in Java: Abstraction, Specification, and Object-Oriented Design

Table of Contents

Contents

Preface

Acknowledgments

1 Introduction

1.1 Decomposition and Abstraction

1.2 Abstraction

1.2.1 Abstraction by Parameterization

1.2.2 Abstraction by Specification

1.2.3 Kinds of Abstractions

1.3 The Remainder of the Book

Exercises

2 Understanding Objects in Java

2.1 Program Structure

2.2 Packages

2.3 Objects and Variables

2.3.1 Mutability

2.3.2 Method Call Semantics

2.4 Type Checking

2.4.1 Type Hierarchy

2.4.2 Conversions and Overloading

2.5 Dispatching

Table of Contents

2.6 Types

2.6.1 Primitive Object Types

2.6.2 Vectors

2.7 Stream Input/Output

2.8 Java Applications

Exercises

3 Procedural Abstraction

3.1 The Benefits of Abstraction

3.2 Specifications

3.3 Specifications of Procedural Abstractions

3.4 Implementing Procedures

3.5 Designing Procedural Abstractions

3.6 Summary

Exercises

4 Exceptions

4.1 Specifications

4.2 The Java Exception Mechanism

4.2.1 Exception Types

4.2.2 Defining Exception Types

4.2.3 Throwing Exceptions

4.2.4 Handling Exceptions

4.2.5 Coping with Unchecked Exceptions

4.3 Programming with Exceptions

4.3.1 Reflecting and Masking

4.4 Design Issues

4.4.1 When to Use Exceptions

4.4.2 Checked versus Unchecked Exceptions

Table of Contents

4.5 Defensive Programming

4.6 Summary

Exercises

5 Data Abstraction

5.1 Specifications for Data Abstractions

5.1.1 Specification of IntSet

5.1.2 The Poly Abstraction

5.2 Using Data Abstractions

5.3 Implementing Data Abstractions

5.3.1 Implementing Data Abstractions in Java

5.3.2 Implementation of IntSet

5.3.3 Implementation of Poly

5.3.4 Records

5.4 Additional Methods

5.5 Aids to Understanding Implementations

5.5.1 The Abstraction Function

5.5.2 The Representation Invariant

5.5.3 Implementing the Abstraction Function and Rep Invariant

5.5.4 Discussion

5.6 Properties of Data Abstraction Implementations

5.6.1 Benevolent Side Effects

5.6.2 Exposing the Rep

5.7 Reasoning about Data Abstractions

5.7.1 Preserving the Rep Invariant

5.7.2 Reasoning about Operations

5.7.3 Reasoning at the Abstract Level

5.8 Design Issues

5.8.1 Mutability

Table of Contents

5.8.2 Operation Categories

5.8.3 Adequacy

5.9 Locality and Modifiability

5.10 Summary

Exercises

6 Iteration Abstraction

6.1 Iteration in Java

6.2 Specifying Iterators

6.3 Using Iterators

6.4 Implementing Iterators

6.5 Rep Invariants and Abstraction Functions for Generators

6.6 Ordered Lists

6.7 Design Issues

6.8 Summary

Exercises

7 Type Hierarchy

7.1 Assignment and Dispatching

7.1.1 Assignment

7.1.2 Dispatching

7.2 Defining a Type Hierarchy

7.3 Defining Hierarchies in Java

7.4 A Simple Example

7.5 Exception Types

7.6 Abstract Classes

7.7 Interfaces

7.8 Multiple Implementations

7.8.1 Lists

Table of Contents

7.8.2 Polynomials

7.9 The Meaning of Subtypes

7.9.1 The Methods Rule

7.9.2 The Properties Rule

7.9.3 Equality

7.10 Discussion of Type Hierarchy

7.11 Summary

Exercises

8 Polymorphic Abstractions

8.1 Polymorphic Data Abstractions

8.2 Using Polymorphic Data Abstractions

8.3 Equality Revisited

8.4 Additional Methods

8.5 More Flexibility

8.6 Polymorphic Procedures

8.7 Summary

Exercises

9 Specifications

9.1 Specifications and Specificand Sets

9.2 Some Criteria for Specifications

9.2.1 Restrictiveness

9.2.2 Generality

9.2.3 Clarity

9.3 Why Specifications?

9.4 Summary

Exercises

10 Testing and Debugging

Table of Contents

10.1 Testing

10.1.1 Black-Box Testing

10.1.2 Glass-Box Testing

10.2 Testing Procedures

10.3 Testing Iterators

10.4 Testing Data Abstractions

10.5 Testing Polymorphic Abstractions

10.6 Testing a Type Hierarchy

10.7 Unit and Integration Testing

10.8 Tools for Testing

10.9 Debugging

10.10 Defensive Programming

10.11 Summary

Exercises

11 Requirements Analysis

11.1 The Software Life Cycle

11.2 Requirements Analysis Overview

11.3 The Stock Tracker

11.4 Summary

Exercises

12 Requirements Specifications

12.1 Data Models

12.1.1 Subsets

12.1.2 Relations

12.1.3 Textual Information

12.2 Requirements Specifications

12.3 Requirements Specification for Stock Tracker

Table of Contents

12.3.1 The Data Model

12.3.2 Stock Tracker Specification

12.4 Requirements Specification for a Search Engine

12.5 Summary

Exercises

13 Design

13.1 An Overview of the Design Process

13.2 The Design Notebook

13.2.1 The Introductory Section

13.2.2 The Abstraction Sections

13.3 The Structure of Interactive Programs

13.4 Starting the Design

13.5 Discussion of the Method

13.6 Continuing the Design

13.7 The Query Abstraction

13.8 The WordTable Abstraction

13.9 Finishing Up

13.10 Interaction between FP and UI

13.11 Module Dependency Diagrams versus Data Models

13.12 Review and Discussion

13.12.1 Inventing Helpers

13.12.2 Specifying Helpers

13.12.3 Continuing the Design

13.12.4 The Design Notebook

13.13 Top-Down Design

13.14 Summary

Exercises

Table of Contents

14 Between Design and Implementation

14.1 Evaluating a Design

14.1.1 Correctness and Performance

14.1.2 Structure

14.2 Ordering the Program Development Process

14.3 Summary

Exercises

15 Design Patterns

15.1 Hiding Object Creation

15.2 Neat Hacks

15.2.1 Flyweights

15.2.2 Singletons

15.2.3 The State Pattern

15.3 The Bridge Pattern

15.4 Procedures Should Be Objects Too

15.5 Composites

15.5.1 Traversing the Tree

15.6 The Power of Indirection

15.7 Publish/Subscribe

15.7.1 Abstracting Control

15.8 Summary

Exercises

Glossary

A

B

C

D

Table of Contents

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

Index