COBOL Programmers Swing with Java

E. Reed Doke
Southwest Missouri State University

Bill C. Hardgrave
University of Arkansas

Richard A. Johnson
Southwest Missouri State University

Cambridge
University Press

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011-4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

© Cambridge University Press 2005

This book is in copyright. Subject to statutory exception
and to the provisions of relevant collective licensing agreements,
no reproduction of any part may take place without
the written permission of Cambridge University Press.

First published 2005

Printed in the United States of America

Typefaces ITC Berkeley Oldstyle 11/13.5 pt. and ITC Franklin Gothic *System* LaTeX 2_ε [TB]

A catalog record for this book is available from the British Library.

Library of Congress Cataloging in Publication Data

Doke, E. Reed.

COBOL programmers swing with Java / E. Reed Doke, Bill C. Hardgrave, Richard A. Johnson.

p. cm.

Includes bibliographical references and index.

ISBN 0-521-54684-2 (alk. paper)

1. Java (Computer program language) 2. COBOL (Computer program language)
I. Hardgrave, Bill C. II. Johnson, Richard A. (Richard Allen), 1953 – III. Title.

QA76.73.J38D645 2004
005.13′3 – dc22 2004048885

ISBN 0 521 54684 2 paperback

Contents

Preface xiii

Introduction 1

Chapter 1
Why You Should Learn Java 3

Objectives 3
History and Overview of Java 4
The Popularity of Java 6
What Makes Java Different? 8
Java Is Simple 8
Java Is Object-Oriented 9
Java Is Portable 10
Will Java Replace COBOL? 12
How to Use This Book 16
Summary of Key Points in Chapter 1 18
Bibliography 19

Chapter 2
An Introduction to Object-Oriented Programming 21

Objectives 21
The Community National Bank 22
History of OO 22
Objects 24

Classes 25
Diagramming Classes and Objects 26
Class Relationships 28
Inheritance 29
    Aggregation 32
    Association 34
Object Communication 35
Polymorphism 37
Dynamic Binding 38
Summary of Key Points in Chapter 2 39
Bibliography 40

Chapter 3
Java Structure 41

Objectives 41
A Class Program 42
    Listing 3.1: `Customer.java` 44
Java Column Restrictions 47
Writing Comments in Java 47
Naming Rules and Conventions 49
Creating Objects 50
    Listing 3.2: `Customer.java` 53
Invoking Methods 54
    Listing 3.3: `CustomerProcessor.java` 56
Working with Subclasses 58
    Listing 3.4: `Account.java` 58
    Listing 3.5: `CheckingAccount.java` 60
    Listing 3.6: `AccountProcessor.java` 62
Summary of Key Points in Chapter 3 64

Chapter 4
Defining Data 66

Objectives 66
COBOL Picture Clause 67
Defining Java Variables 68
Writing Java Literals 70
The Scope of Variables 71
Defining Java Constants 72

String Variables 72
Listing 4.1: StringDemo.java 74
Changing Variable Types 76
Listing 4.2: CastDemo.java 77
Variables for Community National Bank 79
Summary of Key Points in Chapter 4 80

Chapter 5

Computation 82

Objectives 82
Exceptions 83
Listing 5.1: try-catch Structure 85
Listing 5.2: ArithmeticExceptionDemo.java 86
Custom Exception Classes 87
Listing 5.3: CheckingAccount.java with NSFException 89
Listing 5.4: AccountProcessor.java with try-catch 91
A Review of Primitive Data Types 92
Wrapper Classes 93
Listing 5.5: WrapperDemo.java 96
Arithmetic Operators 98
The Math Class 99
Listing 5.6: MathClassDemo.java 103
The NumberFormat Class 104
Listing 5.7: NumberFormatDemo.java 105
Summary of Key Points in Chapter 5 107

Chapter 6

Decision Making 108

Objectives 108
Service Charges at Community National Bank 109
The if Statement 109
Using the else Clause 112
Nested if Statements 114
Writing Compound Conditions 115
Java's Conditional Operator 116

Condition Names 117

Computing the Service Charge with `if` Statements 118

Listing 6.1: COBOL Service Charge Computation Using `IF` Statements 119

Listing 6.2: ComputeServiceCharge method Using `if` Statements 120

Case Structure: COBOL `EVALUATE` and Java `switch` 121

Computing the Service Charge Using `switch` 123

Listing 6.3: COBOL Service Charge Computation Using `EVALUATE` 123

Listing 6.4: Java Service Charge Computation Using `switch` 125

Summary of Key Points in Chapter 6 126

Chapter 7
Loops 129

Objectives 129

Loop Structure 130

The COBOL `PERFORM` Statement 130

The Java `while` Statement 132

Listing 7.1: `WhileLoopDemo.java` 135

The Java `do` Statement 137

Listing 7.2: `DoLoopDemo.java` 139

The Java `for` Statement 140

Listing 7.3: `ForLoopDemo.java` 142

Nested Loops 143

Java `break` and `continue` Statements 145

Producing a Loan Amortization Schedule 146

Listing 7.4: `Amortizer.java` 147

Summary of Key Points in Chapter 7 149

Chapter 8
Arrays 151

Objectives 151

Declaring One-Dimensional Arrays 152

Populating One-Dimensional Arrays 155

Creating String Arrays 157

Listing 8.1: `OneDimArrayDemo.java` 158

Declaring Two-Dimensional Arrays 160
Populating Two-Dimensional Arrays 162
Listing 8.2: `TwoDimArrayDemo.java` 163
Passing Arrays as Arguments 166
Searching Arrays 167
Listing 8.3: `FindZipCode.java` 168
Listing 8.4: `ZipCodeProcessor.java` 169
Summary of Key Points in Chapter 8 171

Chapter 9
Data Access 173

Objectives 173
Java's I-O Class Library (`java.io`) 174
Object Persistence 175
Sequential File I-O 175
Listing 9.1: `SequentialFileDemo.java` 179
Database Access 181
Listing 9.2: COBOL SQL Example 182
Listing 9.3: `DatabaseDemo.java` 188
Object Serialization 189
Listing 9.4: `ObjectSerializationDemo.java` 193
Network Access 195
Summary of Key Points in Chapter 9 195

Chapter 10
Graphical User Interfaces 197

Objectives 197
Java's Swing Components 198
Event-Driven Programming 199
JFrame: Displaying and Closing a Window 201
Listing 10.1—`CustomerGUIOne.java` 201
JLabel: Adding Labels to a Window 203
Listing 10.2: `CustomerGUITwo.java` 203
JTextField: Adding Text Fields to a Window 205
Listing 10.3—`CustomerGUIThree.java` 206

JButton: Adding Buttons to a Window 209
Listing 10.4—`CustomerGUIFour.java` 209
How Java Handles Events 213
Improving the Window Layout 214
Listing 10.5: `CustomerGUIFive.java` 215
Creating Drop-Down Menus 220
Listing 10.6—`CustomerGUISix.java` 220
Writing Applets 225
Summary of Key Points in Chapter 10 227

Chapter 11
Object-Oriented Development Issues 229

Objectives 229
Developing Object-Oriented Systems 230
OO Methodologies 231
OO Analysis 232
OO Design 233
Three-Tier Design 235
Architecture Issues 246
Performance Issues 247
Summary of Key Points in Chapter 11 248
Bibliography 249

Glossary 251

Index 261

Introduction

We organized the chapters in this book to be read in sequence. However, each chapter begins with a clear statement of what we assume you know before reading the chapter, so you can jump around a little bit, depending on your background and experience. You can use the book as a reference and jump in anywhere once you have the fundamentals.

Chapter 1 presents reasons why you should learn Java and describes the many similarities between COBOL and Java.

Chapter 2 explains what OO is, and what it is not. OO terms and concepts are described using several everyday examples.

Chapter 3 describes the overall structure and format of a Java program. Several small programs are developed to show you how to create objects and call methods.

Chapter 4 shows you how to define Java data items and use them in a program. Java data definition is somewhat different than COBOL and these differences are clearly explained and demonstrated in the program examples.

Chapter 5 introduces Java computation and, again, several small programs are written to illustrate the ideas and concepts present. You will see that some Java computation is nearly identical to COBOL.

Chapter 6 illustrates how to use the Java decision-making statements. We develop programs using the Java `if` (sound familiar?) and the Java counterpart to the COBOL `EVALUATE` verb.

Chapter 7 describes how to write Java loops. As you will learn, Java looping is different from COBOL.

Chapter 8 shows you how to define and work with Java arrays, which are really the old COBOL tables with a more technical-sounding name. There are a lot of parallels between Java arrays and COBOL tables.

Chapter 9 explains how to access data in sequential files, relational databases, and networks. You will appreciate Java's approach to accessing relational databases using standard SQL statements.

Chapter 10 illustrates how to develop graphical user interfaces for user input and output using the Java Swing package. This is an interesting and important chapter, even though there are few COBOL similarities.

Chapter 11 discusses OO development in a broader context. Both software and hardware issues are explored. Three-tier software design is illustrated using a GUI front end and a relational database back end.

All code listings can be found on the Cambridge University Press web site: **http://publishing.cambridge.org/resources/0521546842/**