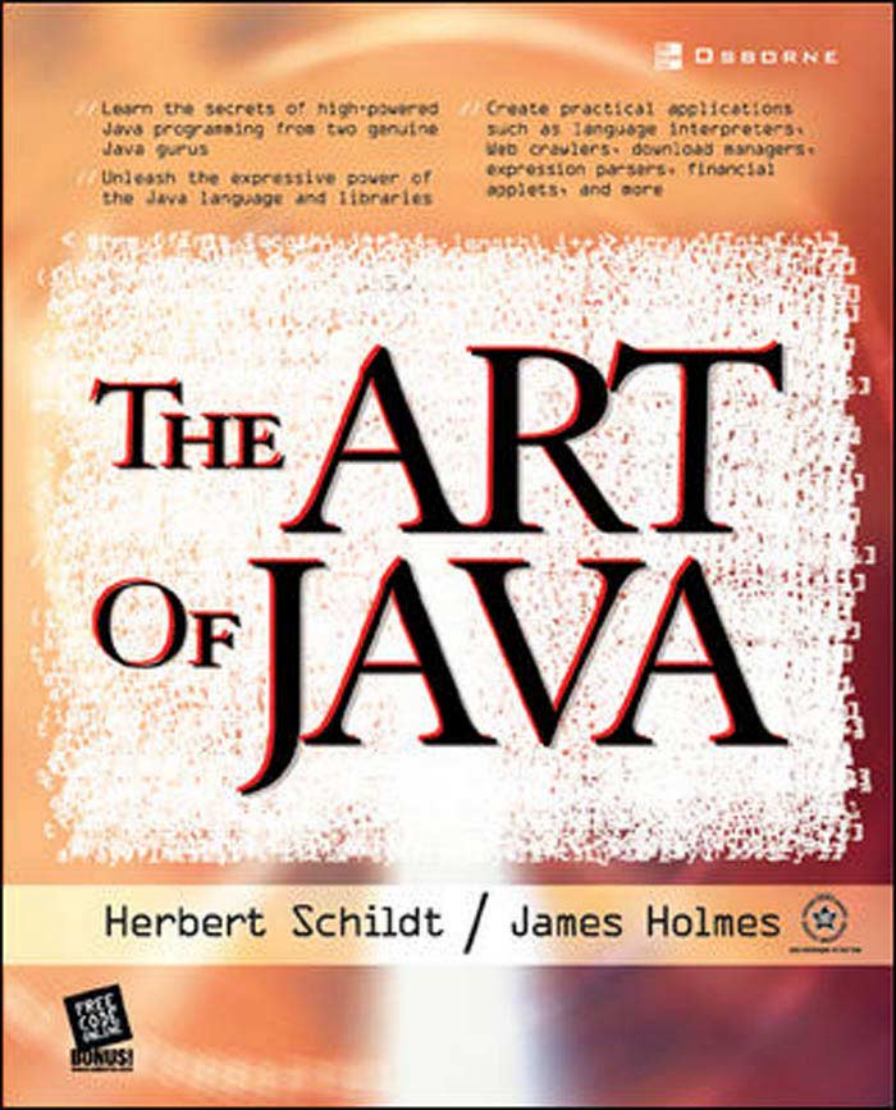
OSBORNE
Learn the secrets of high-powered Java programming from two genuine Java gurus
Unleash the expressive power of the Java language and libraries
Create practical applications such as language interpreters, Web crawlers, download managers, expression parsers, financial applets, and more
THE ART OF JAVA
Herbert Schildt / James Holmes
FREE CODE INSIDE
BONUS!

The Art of Java

Herbert Schildt, James Holmes

McGraw-Hill/Osborne

New York Chicago San Francisco
Lisbon London Madrid Mexico City Milan
New Delhi San Juan Seoul Singapore Sydney Toronto

The McGraw·Hill Companies

McGraw-Hill/Osborne
2100 Powell Street, 10th Floor
Emeryville, California 94608
U.S.A.

To arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact **McGraw-Hill**/Osborne at the above address. For information on translations or book distributors outside the U.S.A., please see the International Contact Information page immediately following the index of this book.

The Art of Java

1234567890 FGR FGR 019876543
ISBN 0-07-222971-3

Publisher	Brandon A. Nordin
Vice President & Associate Publisher	Scott Rogers
Editorial Director	Wendy Rinaldi
Project Editor	Jennifer Malnick
Acquisitions Coordinator	Athena Honore
Technical Editor	James Holmes
Copy Editor	Emily Rader
Proofreader	Emily Hsuan
Indexer	Sheryl Schildt
Composition	Tara A. Davis, Lucie Ericksen
Illustrators	Kathleen Fay Edwards, Melinda Moore Lytle, Lyssa Wald
Series Designer	Roberta Steele
Cover Designer	Jeff Weeks

This book was composed with Corel VENTURA™ Publisher.

Contents

Preface ix

Chapter 1 **The Genius of Java** **1**
Simple Types and Objects: The Right Balance 2
Memory Management Through Garbage Collection 3
A Wonderfully Simple Multithreading Model 4
Fully Integrated Exceptions 5
Streamlined Support for Polymorphism 5
Portability and Security Through Bytecode 6
The Richness of the Java API 6
The Applet 7
The Continuing Revolution 8

Chapter 2 **A Recursive-Descent Expression Parser** **9**
Expressions 10
Parsing Expressions: The Problem 11
Parsing an Expression 12
Dissecting an Expression 13
A Simple Expression Parser 17
Understanding the Parser 24
Adding Variables to the Parser 25
Syntax Checking in a Recursive-Descent Parser 35
A Calculator Applet 36
Some Things to Try 38

Chapter 3 **Implementing Language Interpreters in Java** **39**
What Computer Language to Interpret? 40
An Overview of the Interpreter 42
The Small BASIC Interpreter 42
The Small BASIC Expression Parser 64
Small BASIC Expressions 64
Small BASIC Tokens 65
The Interpreter 70
The InterpreterException Class 70
The SBasic Constructor 70

The Keywords . . . 72
The run( ) Method . . . 73
The sbInterp( ) Method . . . 74
Assignment . . . 75
The PRINT Statement . . . 76
The INPUT Statement . . . 78
The GOTO Statement . . . 79
The IF Statement . . . 82
The FOR Loop . . . 82
The GOSUB . . . 85
The END Statement . . . 87
Using Small BASIC . . . 87
More Small BASIC Sample Programs . . . 88
Enhancing and Expanding the Interpreter . . . 90
Creating Your Own Computer Language . . . 90

Chapter 4 **Creating a Download Manager in Java** . . . **91**
Understanding Internet Downloads . . . 92
An Overview of the Download Manager . . . 93
The Download Class . . . 94
The Download Variables . . . 98
The Download Constructor . . . 98
The download( ) Method . . . 98
The run( ) Method . . . 99
The stateChanged( ) Method . . . 102
Action and Accessor Methods . . . 103
The ProgressRenderer Class . . . 103
The DownloadsTableModel Class . . . 104
The addDownload( ) Method . . . 106
The clearDownload( ) Method . . . 107
The getColumnClass( ) Method . . . 107
The getValueAt( ) Method . . . 108
The update( ) Method . . . 108
The DownloadManager Class . . . 109
The DownloadManager Variables . . . 115
The DownloadManager Constructor . . . 115
The verifyUrl( ) Method . . . 116
The tableSelectionChanged( ) Method . . . 117
The updateButtons( ) Method . . . 117
Handling Action Events . . . 119
Compiling and Running the Download Manager . . . 119
Enhancing the Download Manager . . . 120

Chapter 5 **Implementing an E-mail Client in Java** **121**
E-mail Behind the Scenes 122
POP3 123
IMAP 123
SMTP 123
The General Procedure for Sending and Receiving E-mail 123
The JavaMail API 124
An Overview of Using JavaMail 124
A Simple E-mail Client 125
The ConnectDialog Class 126
The DownloadingDialog Class 132
The MessageDialog Class 134
The MessagesTableModel Class 141
The EmailClient Class 145
Compiling and Running the E-mail Client 163
Expanding Beyond the Basic E-mail Client 165

Chapter 6 **Crawling the Web with Java** **167**
Fundamentals of a Web Crawler 168
Adhering to the Robot Protocol 169
An Overview of the Search Crawler 171
The SearchCrawler Class 172
The SearchCrawler Variables 190
The SearchCrawler Constructor 190
The actionSearch( ) Method 191
The search( ) Method 193
The showError( ) Method 196
The updateStats( ) Method 196
The addMatch( ) Method 197
The verifyUrl( ) Method 198
The isRobotAllowed( ) Method 199
The downloadPage( ) Method 202
The removeWwwFromUrl( ) Method 203
The retrieveLinks( ) Method 203
The searchStringMatches( ) Method 210
The crawl( ) Method 211
Compiling and Running the Search Web Crawler 214
Web Crawler Ideas 215

Chapter 7 **Rendering HTML with Java** **217**
Rendering HTML with JEditorPane 218
Handling Hyperlink Events 219
Creating a Mini Web Browser 220
The MiniBrowser Class 221

The MiniBrowser Variables . . . 226
The MiniBrowser Constructor . . . 227
The actionBack( ) Method . . . 227
The actionForward( ) Method . . . 228
The actionGo( ) Method . . . 228
The showError( ) Method . . . 229
The verifyUrl( ) Method . . . 229
The showPage( ) Method . . . 230
The updateButtons( ) Method . . . 232
The hyperlinkUpdate( ) Method . . . 232
Compiling and Running the Mini Web Browser . . . 233
HTML Renderer Possibilities . . . 234

Chapter 8 **Statistics, Graphing, and Java** . . . **235**
Samples, Populations, Distributions, and Variables . . . 236
The Basic Statistics . . . 237
The Mean . . . 237
The Median . . . 238
The Mode . . . 239
Variance and Standard Deviation . . . 240
The Regression Equation . . . 242
The Correlation Coefficient . . . 243
The Entire Stats Class . . . 246
Graphing Data . . . 250
Scaling Data . . . 250
The Graphs Class . . . 251
The Graphs final and Instance Variables . . . 255
The Graphs Constructor . . . 257
The paint( ) method . . . 258
The bargraph( ) Method . . . 262
The scatter( ) Method . . . 262
The regplot( ) Method . . . 263
A Statistics Application . . . 263
The StatsWin Constructor . . . 268
The itemStateChanged( ) Handler . . . 269
The actionPerformed( ) Method . . . 270
The shutdown( ) Method . . . 270
The createMenu( ) Method . . . 271
The DataWin Class . . . 271
Putting Together the Pieces . . . 272
Creating a Simple Statistical Applet . . . 274
Some Things to Try . . . 276

Chapter 9 **Financial Applets and Servlets** **277**
Finding the Payments for a Loan 278
The RegPay Fields 283
The init( ) Method 283
The actionPerformed( ) Method 286
The paint( ) Method 286
The compute( ) Method 287
Finding the Future Value of an Investment 287
Finding the Initial Investment Required to Achieve a Future Value 292
Finding the Initial Investment Needed for a Desired Annuity 296
Finding the Maximum Annuity for a Given Investment 301
Finding the Remaining Balance on a Loan 305
Creating Financial Servlets 310
Using Tomcat 310
Converting the RegPay Applet into a Servlet 311
The RegPayS Servlet 311
Some Things to Try 316

Chapter 10 **AI-Based Problem Solving** **317**
Representation and Terminology 318
Combinatorial Explosions 320
Search Techniques 322
Evaluating a Search 322
The Problem 322
A Graphic Representation 323
The FlightInfo Class 325
The Depth-First Search 325
An Analysis of the Depth-First Search 336
The Breadth-First Search 336
An Analysis of the Breadth-First Search 338
Adding Heuristics 339
The Hill-Climbing Search 340
An Analysis of Hill Climbing 345
The Least-Cost Search 346
An Analysis of the Least-Cost Search 347
Finding Multiple Solutions 348
Path Removal 349
Node Removal 350
Finding the “Optimal” Solution 356
Back to the Lost Keys 361

Index **367**

About the Authors

Herbert Schildt is a leading authority on the Java, C, C++, and C# languages, and is a master Windows programmer. His programming books have sold more than three million copies worldwide and have been translated into all major foreign languages. He is the author of numerous bestsellers, including *Java 2: The Complete Reference, Java 2: A Beginner's Guide, Java 2 Programmer's Reference*, *C++: The Complete Reference*, *C: The Complete Reference*, and *C#: The Complete Reference*. Schildt holds a master's degree in computer science from the University of Illinois. He can be reached at his consulting office at (217) 586-4683.

James Holmes is a recognized leader in Java programming. He was named 2002 Oracle Magazine Java Developer of the Year and is a Committer on the Jakarta Struts open source project. He is currently an independent Java consultant, Sun Certified Java Programmer, and Sun Certified Web Component Developer. James can be reached via e-mail at james@jamesholmes.com. You can also visit his Web site at http://www.JamesHolmes.com.

Preface

by Herbert Schildt

Beginning in 1991 at Sun Microsystems, James Gosling, along with Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan, began work on a new language that would eventually rock the foundations of programming. Originally called Oak, this new language was renamed Java in 1995—and computing hasn't been the same since.

Java changed the course of programming in two important ways. First, Java incorporated features that facilitated the creation of Internet-enabled applications. Thus, Java was the world's first truly Internet-ready language. Second, Java advanced the state of the art in computer language design. For example, it redefined the object paradigm, streamlined exceptions, fully integrated multithreading into the language, and created a portable object code called bytecode that enabled programs to run on a variety of different platforms.

Java's importance to computing, therefore, lies firmly on two pillars: its built-in support for the Internet, and its advances in computer language design. Either one of these would have made Java a good language, but it is the combination that made Java a great language and ensured its place in computing history.

This book shows some of the reasons why Java is such an extraordinary language.

What's Inside

This book is different from most other books on Java. Whereas other books teach the basics of the language, this book shows how to apply it to some of computing's most interesting, useful, and, at times, mysterious programming tasks. In the process, it displays the power, versatility, and elegance of the Java language. Thus, it is through the *art* of Java that the *artistry* of Java's design is displayed.

As you might expect, several of the applications, such as the download manager in Chapter 4 or the e-mail subsystem in Chapter 5, relate directly to the Internet. However, many of the chapters develop code that illustrates the expressiveness of Java independently of the Internet. For example, the language interpreter in Chapter 3, or the AI-based search routines in Chapter 10, are what we call "pure code" examples. Neither of these applications relies on the Internet or uses a GUI interface. They are the type of code that in the past one might have expected to find written in C++. The ease by which these types of programs can be written in Java demonstrates the versatility and agility of the language.