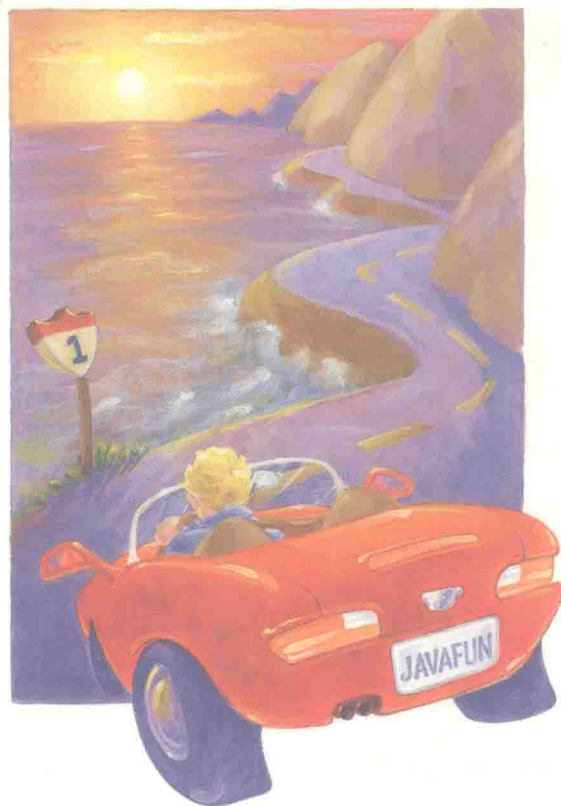




just JTMAVA 2



FIFTH EDITION

▼ Java programming language
and basic packages

▼ Server-side libraries and enterprise
development

▼ Networking, Beans, and collections

▼ GUI components and more

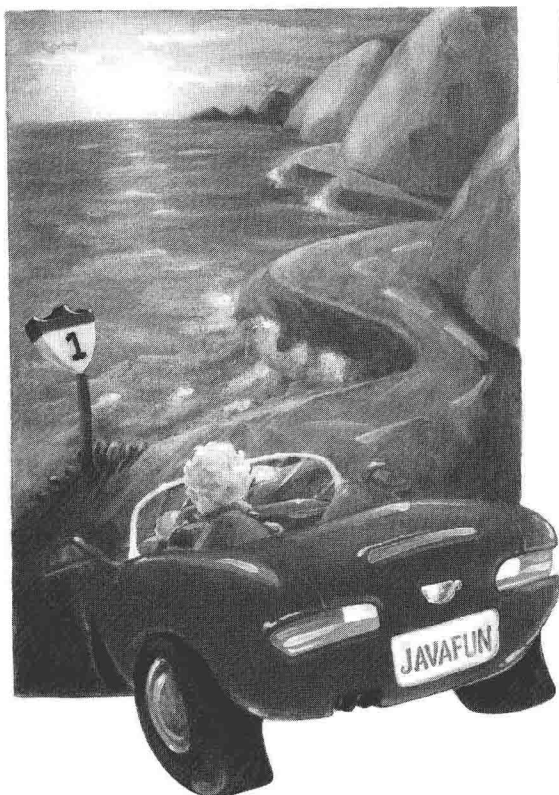
THE SUN MICROSYSTEMS PRESS
JTMAVA SERIES



van der LINDEN

just JAVATM 2

FIFTH EDITION



SUN MICROSYSTEMS PRESS
A Prentice Hall Title

PETER van der LINDEN

© 2002 Sun Microsystems, Inc.—
Printed in the United States of America.
901 San Antonio Road, Palo Alto, California
94303 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The products described may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS—HotJava, Java, Java Development Kit, Solaris, SPARC, SunOS, and Sunsoft are trademarks of Sun Microsystems, Inc.

The publisher offers discounts on this book when ordered in bulk quantities.

For more information, contact Corporate Sales Department, Prentice Hall PTR,
One Lake Street, Upper Saddle River, NJ 07458. Phone: 800-382-3419; FAX: 201-236-7141.
E-mail: corpsales@prenhall.com.

Editorial/production supervision: *Patti Guerrieri*

Cover design director: *Jerry Votta*

Cover designer: *Nina Scuderi*

Cover illustration: *Karen Strelecki*

Manufacturing manager: *Alexis R. Heydt-Long*

Marketing Manager: *Debby vanDijk*

Acquisitions editor: *Gregory G. Doench*

Sun Microsystems Press publisher: *Michael Alread*

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-032072-2

Sun Microsystems Press
A Prentice Hall Title

just
JAVA[™] 2
FIFTH EDITION

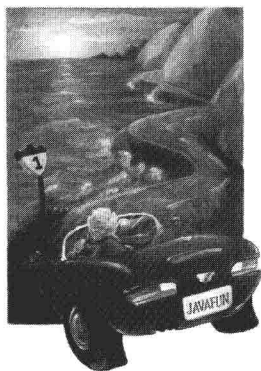
ISBN 0-13-032072-2



9 780130 320728

9 0000





Acknowledgments

I would like to express my thanks to Bob Lynch and Peter Jones who read all the new chapters for this edition and kept me on the straight and narrow. Other chapters were improved by suggestions from Chris Riesbeck and Jon Skeet.

I take responsibility for remaining errors, except the ones to do with quotation marks or source listing layout—blame FrameMaker for those. If you notice an error that isn't in the errata sheet at www.afu.com, please let me know by email (my address is at the end of the errata sheet).

I am grateful to all the programmers who were willing to let me showcase their excellent work, and in many cases add it to the CD: William Brogden, Toby Downer, Wilson Fletcher, Carl Ginnow, Marcus Green, Karl Hörnell, Dean Jones, Karsten Lentzsch, the Limewire team, Walter Pullen, Burkhard Ratheiser, Kerry Shetline, Robin Southgate.

The layout and design of this book was improved by Patti Guerrieri and previous editors who have earned my warmest appreciation. Thanks go to the copy editor, Barbara, who blue-pencilled out my anachronisms like “whence,” “especial,” and “hitherto.” Barbara also got the ‘ell out of “marvellous.”



Thanks to the hardworking people at Sun Microsystems Press and Prentice Hall: Greg Doench, Rachel Borden, Michael Alread, Eileen Clark, Debbie Van Dijk, and Brandt Kenna.

I would also like to thank the following people who provided significant help, code for the CD, moral support, beer recipes, programming assistance, etc.:

Josh Bloch,

Matthew Burtch,

John F. Dumas,

Jane Erskine,

Roedy Green,

Steve Griffiths,

Roger Lindsjö,

Nicholas J. Morrell (what a taste he has for expensive Scotch),

Alec Muffet,

Jef Poskanzer,

Kathy Stark,

Al Sutton,

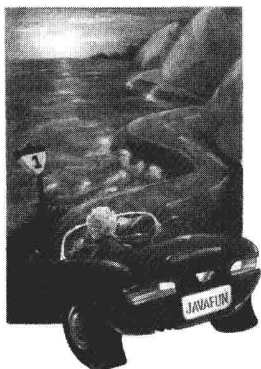
Greg Turner,

Tom van der Linden (who instilled in me his love for reading, writing, and consolidated statements of stockholder equity),

Wendy van der Linden (who pointed out that Citrus was a subclass of Fruit, and conceived the “Wendy Wand” pointing device),

Sean Willard.

We’re doing something right.



Using the Just Java CD-ROM

About the CD-ROM

Welcome to the *Just Java* CD-ROM—a disk packed with Java tools and source code discussed in the book and lots more.

This CD is for any reasonably modern system that can read a CD with a UDF file system. That includes Unix, Windows, and MacOS but may exclude some vintage PCs.

There is a huge amount of useful, entertaining, or educational material on this CD. Some of the content (and there is a *lot* more) is:

- **Useful** Java Programmer's FAQ and Glossary.
 Decompilers and obfuscators.
- **Educational** Translators for Perl, TCL, Eiffel, C, C++, Python, etc.
 CIA World Fact Book—your tax dollars at work.
 Java Digital Simulator.
- **Entertaining** Java program to solve crossword puzzles.
 The Jargon File
 The Sherlock Holmes books.
 Java Bible Code software. Look for hidden messages.

Explore the CD-ROM using a browser. Put the CD in your computer and point your browser at the `index.html` file in the root directory with a URL like `"file:/e:/index.html"` (substitute the right letter for the CD drive if it is not 'e').

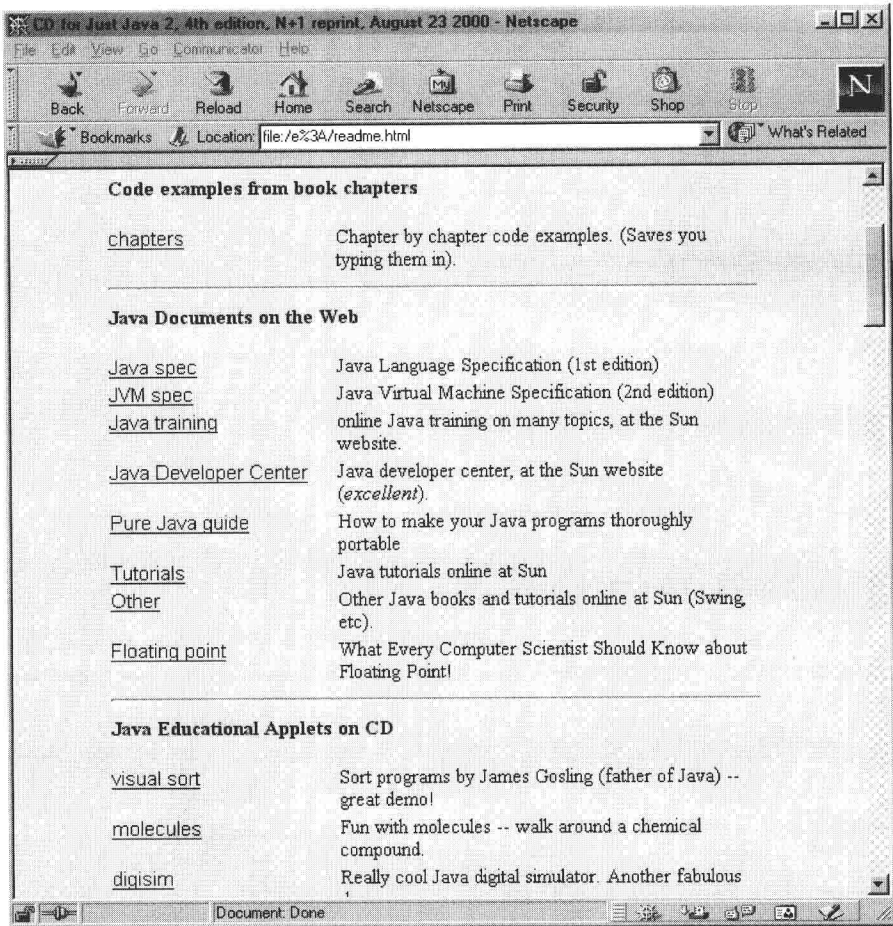


Figure P-1 Browsing the CD.

You'll see a display like that shown in Figure P-1.

If you click on, for example, the "digisim" link at the bottom of the page, the browser will take you to that directory, and the display will now look like Figure P-2.

Some of the content on the CD is in applet form, and you can run the programs as you browse them. The program shown in Figure P-2 is a digital simulator applet, allowing you to drag and drop icons representing electronic components to make circuits. It comes with a demo LED circuit (shown).

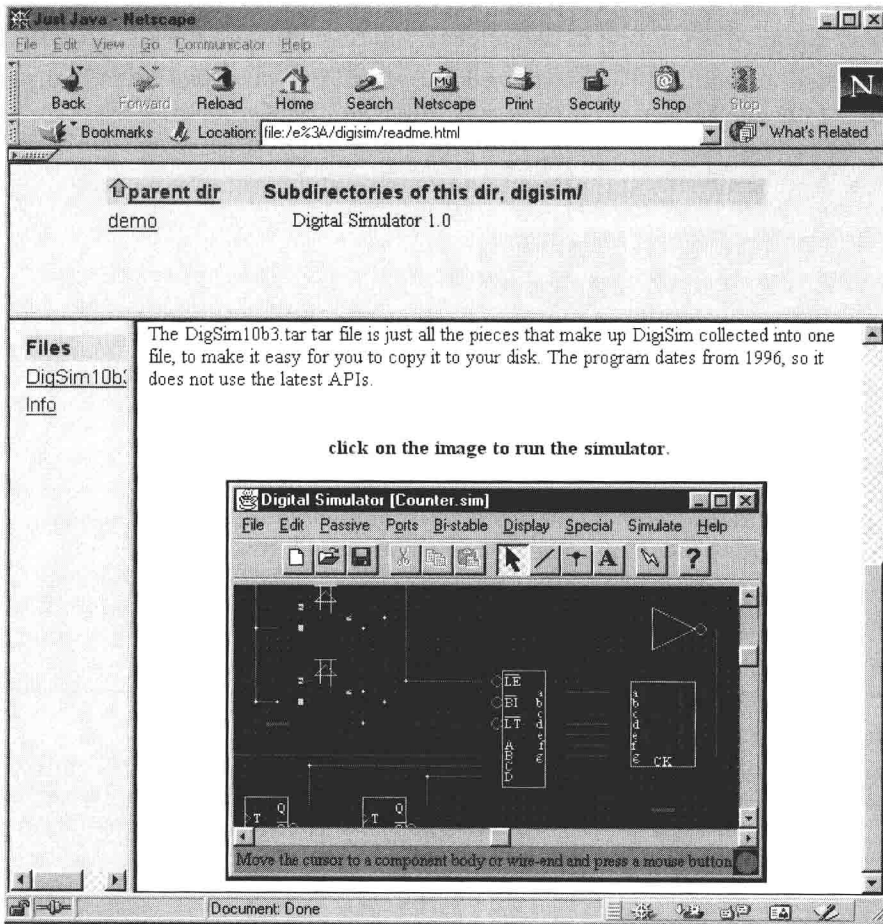


Figure P-2 Java Digital Simulator.

You can carry on exploring the CD, and also use the “back” and “forward” buttons in the browser. There are hundreds of megabytes of data of interest to a professional programmer on this CD, including many freeware or shareware compilers for other programming languages.

Your next step should be to download and install a Java Development Kit. That used to be called a “JDK,” but recently Sun has started calling it the “Java 2 Standard Edition SDK.” The JDK software is not on the CD, but there are links to the download sites on the CD. Choose the latest (highest version number) JDK for



your system. You should also download and install the HTML documentation for the release. You will be referring to that a lot as you read through these chapters.

The J2 SDK is about a 40 Mbyte download, which will take about two hours to download on a 56 Kb dial-up connection. The Java 2 SDK documentation is about a 30 Mbyte download, taking about one and a half hours to download.

Follow the instructions at the download site to install these downloads.

Running the Java Tools

Many programmers today do their development using an IDE. Other programmers, often those who learned programming on non-Windows systems, prefer to use command line tools.

There are many excellent IDEs available that support Java—some free and even some open source. Sun allows free download of the “Forte for Java” IDE. It runs on Solaris, Linux, and Windows, and can be found at www.sun.com/forte/ffj/buy.html.

My recommendation is that you write your first few Java programs using only an editor and the command line tools from an MS DOS window (on Windows). The Windows Notepad editor (under Start-> Programs -> Accessories) works fine. That way, you can focus on just Java, and you don’t have to figure out an IDE at the same time

Editing Command Lines in Windows 9x

There is a very useful, but easily-overlooked, utility in Windows 9x that remembers the last few dozen commands you typed and lets you recall and edit them. The command is known as “doskey,” and you get its benefits by putting a line like this in your `\autoexec.bat` file:

```
doskey /insert /keysize:64
```

Once you have added that line and rebooted, the up and down cursor keys will move you backwards and forwards through the list of commands you previously entered. The left and right cursor keys will move you along the line. The insert key will let you add characters by typing, and delete or backspace will remove them.

This lets you easily modify and re-issue long command lines. You can type “doskey /?” at the command line to see some of the additional options that the program supports.



Pathnames Used Throughout This Book. The source code for the Java runtime library is supplied as part of the JDK. It's in a file called `src.jar` that you need to unpack using `winzip` or other utility. Jar files have the exact same format as zip files.

From time to time, I will refer you to particular files in the runtime library. The first part of the pathname will depend on where you installed JDK, and I'll represent this by “`$JAVAHOME`.”

The separators in a pathname are different on Unix than on Windows. For example, I may recommend you look at a file located here:

`$JAVAHOME/src/java/awt/Window.java`

If you installed Java on your PC at `C:\jdk1.4`, then the file to review is here:

`C:\jdk1.4\src\java\awt\Window.java`.

Enough of the administrative details! Let's go on to look at our first Java program.

Contents

Acknowledgments xxix

Using the Just Java CD-ROM xxxi

About the CD-ROM xxxi

Running the Java Tools xxxiv

Part One Language

Chapter 1

What Is Java? 3

Compiling and Executing a Sample Program 5

The Biggest Java Benefit: Future-Proofing 9

Why Software Portability Matters 9

J2EE, J2ME, J2SE Editions 11

Java Virtual Machine 14

The “Java Platform” 15

The Java Language 16



Java Libraries 18

How New Libraries are Developed 20

Three Execution Environments 23

Applets 23

Server-Side Java 24

Some Light Relief—A Java Desktop Application 25

Chapter 2

The Story of O: Object-Oriented Programming 27

Abstraction 30

Encapsulation 31

Comparing Primitive Types to Class Types 32

Defining a Type 32

Declaring Variables 33

Newsflash: Object vs. Reference to an Object! 34

Operations on Primitive Types/Methods on Objects 35

The One-Minute Object Manager 37

Creating New Objects: Constructors 39

A First Glimpse at Inheritance 41

How Constructors Are Invoked 42

No Destructors 44

Methods 45

You Always Get to an Object with a “Reference Variable” 47

The “null” Reference 49

Dynamic Data Structures 51

Per-Instance and Per-Class Members 52

Static Data 53

Static Methods 54

Static Blocks 55

Static Classes 56

The “final” Modifier 57

The “Blank Final Variable” 57

Access Modifiers 59

Exercises 61

Some Light Relief—It’s Not Your Father’s IBM 62



Chapter 3

Explanation of a Sample Program 65

Explanation of the Example Program 67

Applications vs. Applets vs. Servlets 69

Where an Application Starts 70

Runtime Internals: Stack and Heap 72

The Class “Object” 75

Reading the Java API 79

Some Light Relief—Napster and LimeWire 82

Chapter 4

Identifiers, Keywords, and Types 85

Java and Unicode 86

Scanning 88

Identifiers 89

Comments 90

Keywords 92

Used for built-in types: 92

Used for objects: 92

Used in statements: 92

Used to modify declarations (visibility, sharing, etc.): 93

Used for other method or class-related purposes: 93

Used for larger-than-a-single-class building blocks: 93

Reserved for possible future use: 93

The Primitive Types 94

boolean 95

int 96

long 97

byte 98

short 99

double 100

float 103

char 104

String 106



String Concatenation 107
String Comparison and `intern()` 108
Some Light Relief—Hatless Atlas 111

Chapter 5

Names, Arrays, Operators, and Accuracy 113

Names 114

When Can an Identifier Be Forward-Referenced? 115

Expressions 116

Arrays 117

Creating an Array 119

Initializing an Array 120

Arrays of Arrays of ... 121

Have Array Brackets, Will Travel 124

Operators 126

Java Operators 128

The ++ and -- Operators 128

The % and / Operators 129

The << >> and >>> Operators 129

The instanceof Operator 130

The & | and ^ Operators 130

The && and || Operators 131

The ? ... : Operator 131

The Assignment Operators 132

The Comma Operator Is Gone 132

Associativity 133

How Accurate Are Calculations? 134

Floating-Point Extension 136

Widening and Narrowing Conversions 137

What Happens on Overflow? 139

Integer Overflow 139

Floating Point Overflow 139

Arithmetic That Cannot Overflow 140



- The Math Package 141
- Further Reading 143
- Some Light Relief—Too Much Bread 144

Chapter 6

More OOP—Extending Classes 147

- Inheritance 149
 - A Java Example of Inheritance* 150
 - What Happens When Names Collide?* 157
 - Compatibility of Superclass and Subclass* 158
 - Casting* 158
- Polymorphism 161
 - Overloading* 161
 - Overriding* 161
 - Inheriting from Object* 164
 - Forcing Overriding off: Final* 165
 - Forcing Overriding: Abstract* 167
- The Class Whose Name Is `class` 170
- Exercises 173
- Some Light Relief—The Nerd Detection System 174

Chapter 7

Java Statements 175

- “Organizing” Statements 177
- Expression Statements 178
- Selection Statements 179
 - Statement Notes* 179
 - Statement Notes* 180
- Iteration Statements 181
 - Statement Notes* 181
 - Statement Notes* 182
 - Statement Notes* 182
- Transfer of Control Statements 185
 - Statement Notes* 185



Exceptions 187

The Purpose of Exceptions 187

How to Cause an Exception (Implicitly and Explicitly) 188

Triggering an Exception 189

*How to Handle ("Catch") an Exception Within the Method
Where It Was Thrown 192*

Handling Groups of Related Exceptions 194

*How the Exception Propagates If Not Handled in the Method Where It
Was Thrown 194*

*How and Why Methods Declare the Exceptions
That Can Propagate Out of Them 195*

Fancy Exception Stuff 196

Summary of Exceptions 197

The Assert Statement 198

Further Reading 203

Exercises 203

Some Light Relief—MiniScribe: The Hard Luck Hard Disk 204

Chapter 8

Interfaces 207

What Problem Does an Interface Solve? 208

Comparable 212

Interfaces vs. Abstract Classes 216

Granting Permission Through an Interface—Cloneable 216

What Protected Really Means 218

Interfaces in the Java Runtimes 220

Call Backs 221

Using Interfaces for Named Constants 223

The Class Double 224

Exercises 225

Some Light Relief—The Odyssey of Naming Hal 226

Chapter 9

Packages and Visibility 227

What Are Packages For? 229

How to Choose a Package Name 230