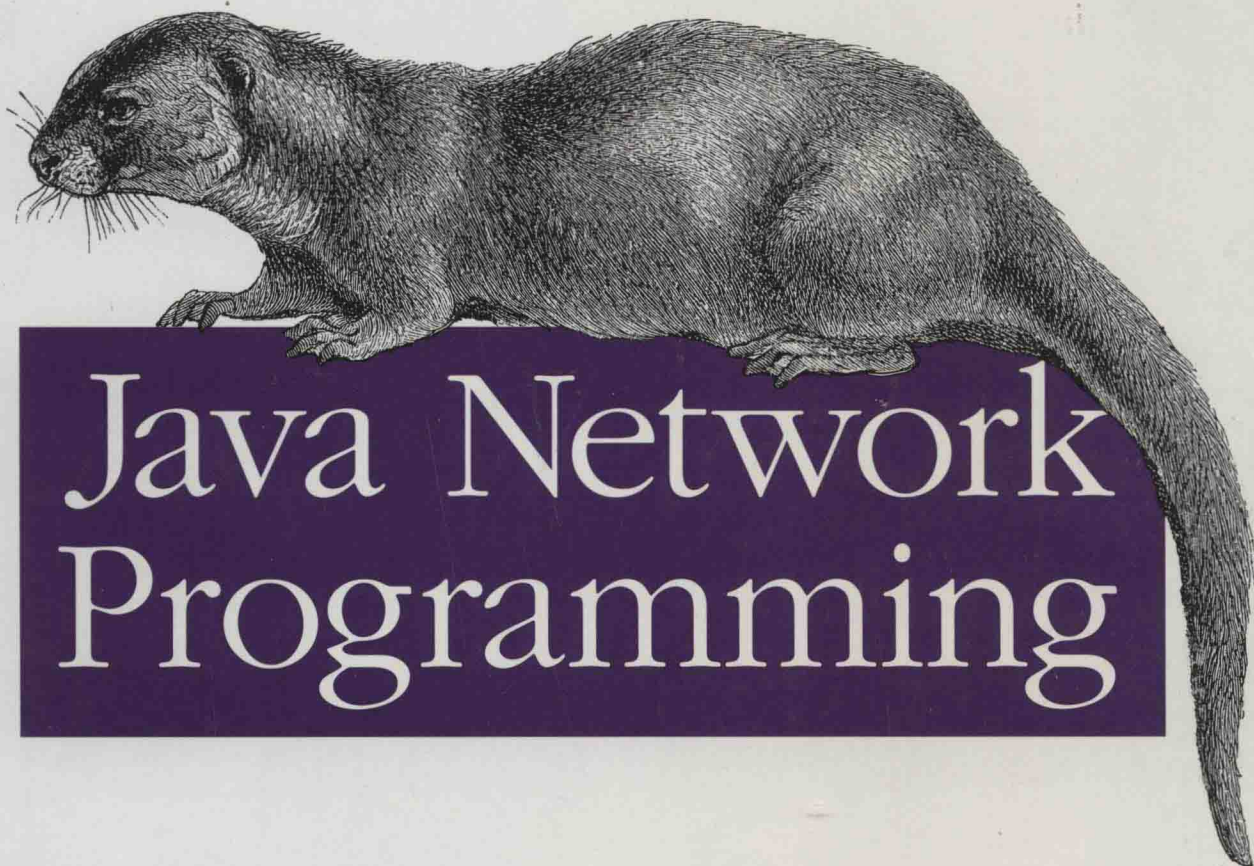


Java网络编程 (影印版)

第4版



O'REILLY®

東南大學出版社

Elliotte Rusty Harold 著

第4版

Java网络编程 (影印版)

Java Network Programming

Elliote Rusty Harold 著

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

南京 东南大学出版社

图书在版编目(CIP)数据

Java 网络编程:第4版:英文/(美)哈罗德(Harold,E. R.)

著. —影印本. —南京:东南大学出版社, 2014.9

书名原文: Java Network Programming, 4E

ISBN 978-7-5641-4960-4

I. ① J… II. ① 哈… III. ① JAVA 语言—程序设计—
英文 IV. ① TP312

中国版本图书馆CIP数据核字(2014)第102966号

江苏省版权局著作权合同登记

图字: 10-2014-145 号

©2013 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2014. Authorized reprint of the original English edition, 2014 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2013。

英文影印版由东南大学出版社出版 2014。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

Java 网络编程 第4版(影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼2号 邮编: 210096

出 版 人: 江建中

网 址: <http://www.seupress.com>

电子邮件: press@seupress.com

印 刷: 常州市武进第三印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 31.5

字 数: 617 千字

版 次: 2014 年 9 月第 1 版

印 次: 2014 年 9 月第 1 次印刷

书 号: ISBN 978-7-5641-4960-4

定 价: 82.00 元

本社图书若有印装质量问题, 请直接与营销部联系。电话(传真): 025-83791830

This book is dedicated to my dog, Thor.

Preface

Java's growth over the past 20 years has been nothing short of phenomenal. Given Java's rapid rise to prominence and the even more spectacular growth of the Internet, it's a little surprising that network programming in Java remains so mysterious to so many. It doesn't have to be. In fact, writing network programs in Java is quite simple, as this book will show. Readers with previous experience in network programming in a Unix, Windows, or Macintosh environment will be pleasantly surprised at how much easier it is to write equivalent programs in Java. The Java core API includes well-designed interfaces to most network features. Indeed, there is very little application layer network software you can write in C or C++ that you can't write more easily in Java. *Java Network Programming*, Fourth Edition, endeavors to show you how to take advantage of Java's network class library to quickly and easily write programs that accomplish many common networking tasks. Some of these include:

- Browsing the Web with HTTP
- Writing multithreaded servers
- Encrypting communications for confidentiality, authentication, and guaranteed message integrity
- Designing GUI clients for network services
- Posting data to server-side programs
- Looking up hosts using DNS
- Downloading files with anonymous FTP
- Connecting sockets for low-level network communication
- Multicasting to all hosts on the network

Java is the first (though no longer the only) language to provide such a powerful cross-platform network library for handling all these diverse tasks. *Java Network Programming* exposes the power and sophistication of this library. This book's goal is to enable

you to start using Java as a platform for serious network programming. To do so, this book provides a general background in network fundamentals, as well as detailed discussions of Java's facilities for writing network programs. You'll learn how to write Java programs that share data across the Internet for games, collaboration, software updates, file transfer, and more. You'll also get a behind-the-scenes look at HTTP, SMTP, TCP/IP, and the other protocols that support the Internet and the Web. When you finish this book, you'll have the knowledge and the tools to create the next generation of software that takes full advantage of the Internet.

About the Fourth Edition

In 1996, in the first edition of this book's opening chapter, I wrote extensively about the sort of dynamic, distributed network applications I thought Java would make possible. One of the most exciting parts of writing subsequent editions has been seeing virtually all of the applications I foretold come to pass. Programmers are using Java to query database servers, monitor web pages, control telescopes, manage multiplayer games, and more, all by using Java's native ability to access the Internet. Java in general and network programming in Java in particular has moved well beyond the hype stage and into the realm of real, working applications.

This book has come a long way, too. The fourth edition focuses even more heavily on HTTP and REST. HTTP has gone from being one of many network protocols to almost *the* network protocol. As you'll see, it is often the protocol on which other protocols are built, forming its own layer in the network stack.

There have been lots of other small changes and updates throughout the `java.net` and supporting packages in Java 6, 7, and 8, and these are covered here as well. New classes addressed in this edition include `CookieManager`, `CookiePolicy`, `CookieStore`, `HttpCookie`, `SwingWorker`, `Executor`, `ExecutorService`, `AsynchronousSocketChannel`, `AsynchronousServerSocketChannel`, and more. Many other methods have been added to existing classes in the last three releases of Java, and these are discussed in the relevant chapters. I've also rewritten large parts of the book to reflect the ever-changing fashions in Java programming in general and network programming in particular. I hope you'll find this fourth edition an even stronger, longer-lived, more accurate, and more enjoyable tutorial and reference to network programming in Java than the previous one.

Organization of the Book

Chapter 1, *Basic Network Concepts*, explains in detail what a programmer needs to know about how the networks and the Internet work. It covers the protocols that underlie the Internet, such as TCP/IP and UDP/IP.

The next two chapters throw some light on two parts of Java programming that are critical to almost all network programs but are often misunderstood and misused: I/O and threading. Chapter 2, *Streams*, explores Java's classic I/O which—despite the new I/O APIs— isn't going away any time soon and is still the preferred means of handling input and output in most client applications. Understanding how Java handles I/O in the general case is a prerequisite for understanding the special case of how Java handles network I/O. Chapter 3, *Threads*, explores multithreading and synchronization, with a special emphasis on how they can be used for asynchronous I/O and network servers.

Experienced Java programmers may be able to skim or skip these two chapters. However, Chapter 4, *Internet Addresses*, is essential reading for everyone. It shows how Java programs interact with the Domain Name System through the `InetAddress` class, the one class that's needed by essentially all network programs. Once you've finished this chapter, it's possible to jump around in the book as your interests and needs dictate.

Chapter 5, *URLs and URIs*, explores Java's `URL` class, a powerful abstraction for downloading information and files from network servers of many kinds. The `URL` class enables you to connect to and download files and documents from a network server without concerning yourself with the details of the protocol the server speaks. It lets you connect to an FTP server using the same code you use to talk to an HTTP server or to read a file on the local hard disk. You'll also learn about the newer `URI` class, a more standards-conformant alternative for identifying but not retrieving resources.

Chapter 6, *HTTP*, delves deeper into the HTTP protocol specifically. Topics covered include REST, HTTP headers, and cookies. Chapter 7, *URLConnections*, shows you how to use the `URLConnection` and `HttpURLConnection` classes not just to download data from web servers, but to upload documents and configure connections.

Chapter 8 through Chapter 10 discuss Java's low-level socket classes for network access. Chapter 8, *Sockets for Clients*, introduces the Java sockets API and the `Socket` class in particular. It shows you how to write network clients that interact with TCP servers of all kinds including whois, dict, and HTTP. Chapter 9, *Sockets for Servers*, shows you how to use the `ServerSocket` class to write servers for these and other protocols. Finally, Chapter 10, *Secure Sockets*, shows you how to protect your client-server communications using the Secure Sockets Layer (SSL) and the Java Secure Sockets Extension (JSSE).

Chapter 11, *Nonblocking I/O*, introduces the new I/O APIs specifically designed for network servers. These APIs enable a program to figure out whether a connection is ready before it tries to read from or write to the socket. This allows a single thread to manage many different connections simultaneously, thereby placing much less load on the virtual machine. The new I/O APIs don't help much for small servers or clients that don't open many simultaneous connections, but they may provide performance boosts for high-volume servers that want to transmit as much data as the network can handle as fast as the network can deliver it.

Chapter 12, *UDP*, introduces the User Datagram Protocol (UDP) and the associated `DatagramPacket` and `DatagramSocket` classes that provide fast, unreliable communication. Finally, Chapter 13, *IP Multicast*, shows you how to use UDP to communicate with multiple hosts at the same time.

Who You Are

This book assumes you are comfortable with the Java language and programming environment, in addition to object-oriented programming in general. This book does not attempt to be a basic language tutorial. You should be thoroughly familiar with the syntax of Java. You should have written simple applications. It also wouldn't hurt if you're familiar with basic Swing programming, though that's not required aside from a few examples. When you encounter a topic that requires a deeper understanding for network programming than is customary—for instance, threads and streams—I'll cover that topic as well, at least briefly.

However, this book doesn't assume that you have prior experience with network programming. You should find it a complete introduction to networking concepts and network application development. I don't assume that you have a few thousand networking acronyms (TCP, UDP, SMTP, etc.) at the tip of your tongue. You'll learn what you need to know about these here.

Java Versions

Java's network classes have changed a lot more slowly since Java 1.0 than other parts of the core API. In comparison to the AWT or I/O, there have been almost no changes and only a few additions. Of course, all network programs make extensive use of the I/O classes and some make heavy use of GUIs. This book is written with the assumption that you are coding with at least Java 5.0. In general, I use Java 5 features like generics and the enhanced for loop freely without further explanation.

For network programming purposes, the distinction between Java 5 and Java 6 is not large. Most examples look identical in the two versions. When a particular method or class is new in Java 6, 7, or 8, it is noted by a comment following its declaration like this:

```
public void setFixedLengthStreamingMode(long contentLength) // Java 7
```

Java 7 is a bit more of a stretch. I have not shied away from using features introduced in Java 7 where they seemed especially useful or convenient—for instance, try-with-resources and multithreading are both very helpful when trying to fit examples into the limited space available in a printed book—but I have been careful to point out my use of such features.

Overall, though, Java's networking API has been relatively stable since Java 1.0. Very little of the post-1.0 networking API has ever been deprecated, and additions have been

relatively minor. You shouldn't have any trouble using this book after Java 8 is released. New APIs, however, have been somewhat more frequent in the supporting classes, particularly I/O, which has undergone three major revisions since Java 1.0.

About the Examples

Most methods and classes described in this book are illustrated with at least one complete working program, simple though it may be. In my experience, a complete working program is essential to showing the proper use of a method. Without a program, it is too easy to drop into jargon or to gloss over points about which the author may be unclear in his own mind. The Java API documentation itself often suffers from excessively terse descriptions of the method calls. In this book, I have tried to err on the side of providing too much explication rather than too little. If a point is obvious to you, feel free to skip over it. You do not need to type in and run every example in this book; but if a particular method does give you trouble, you should have at least one working example.

Each chapter includes at least one (and often several) more complex programs that demonstrate the classes and methods of that chapter in a more realistic setting. These often rely on Java features not discussed in this book. Indeed, in many of the programs, the networking components are only a small fraction of the source code and often the least difficult parts. Nonetheless, none of these programs could be written as easily in languages that didn't give networking the central position it occupies in Java. The apparent simplicity of the networked sections of the code reflects the extent to which networking has been made a core feature of Java, and not any triviality of the program itself. All example programs presented in this book are available online, often with corrections and additions. You can download the source code from <http://www.cafeaulait.org/books/jnp4/>.

I have tested all the examples on Linux and many on Windows and Mac OS X. Most of the examples given here *should* work on other platforms and with other compilers and virtual machines that support Java 5 or later. The most common reasons an example may not compile with Java 5 or 6 are `try-with-resources` and `multicatch`. These examples can easily be rewritten to support earlier Java versions at the cost of increased verbosity.

I do feel a little guilty about a couple of compromises necessitated by the needs of space in a printed book. First, I rarely check preconditions. Most methods assume they are passed good data, and dispense with null checks and similar principles of good code hygiene. Furthermore, I have reduced the indentation to two characters per block and four characters per continuation line, as opposed to the Java standard of four and eight, respectively. I hope these flaws will not be too distracting. On the positive side, these compromises have aided me in making this edition considerably shorter (by several hundred pages) than the previous edition.

Conventions Used in This Book

Body text is Minion Pro, normal, like you're reading now.

A monospaced typewriter font is used for:

- Code examples and fragments
- Anything that might appear in a Java program, including keywords, operators, data types, method names, variable names, class names, and interface names
- Program output
- Tags that might appear in an HTML document

A **bold monospaced font** is used for:

- Command lines and options that should be typed verbatim on the screen

An *italicized* font is used for:

- New terms where they are defined
- Pathnames, filenames, and program names (however, if the program name is also the name of a Java class, it is given in a monospaced font, like other class names)
- Host and domain names (*www.hpmor.com*)
- URLs (*http://www.cafeaulait.org/slides/*)
- Titles of other books (*Java I/O*)



Indicates a tip, suggestion, or general note.



Indicates a warning or caution.

Significant code fragments and complete programs are generally placed into a separate paragraph, like this:

```
Socket s = new Socket("java.oreilly.com", 80);  
if (!s.getTcpNoDelay()) s.setTcpNoDelay(true);
```

When code is presented as fragments rather than complete programs, the existence of the appropriate `import` statements should be inferred. For example, in the preceding code fragment you may assume that `java.net.Socket` was imported.

Some examples intermix user input with program output. In these cases, the user input will be displayed in bold, as in this example from Chapter 9:

```
% telnet rama.poly.edu 7
Trying 128.238.10.212...
Connected to rama.poly.edu.
Escape character is '^]'.
This is a test
This is a test
This is another test
This is another test
9876543210
9876543210
^]
telnet> close
Connection closed.
```

Finally, although many of the examples used here are toy examples unlikely to be reused, a few of the classes I develop have real value. Please feel free to reuse them or any parts of them in your own code. No special permission is required. They are in the public domain (although the same is most definitely not true of the explanatory text!).

Request for Comments

I enjoy hearing from readers, whether with general comments about this book, specific corrections, other topics they would like to see covered, or just war stories about their own network programming travails. You can reach me by sending an email to elharo@ibiblio.org. Please realize, however, that I receive several hundred pieces of email a day and cannot personally respond to each one. For the best chance of getting a personal response, please identify yourself as a reader of this book. If you have a question about a particular program that isn't working as you expect, try to reduce it to the simplest case that reproduces the bug, preferably a single class, and paste the text of the entire program into the body of your email. Unsolicited attachments will be deleted unopened. And please, please send the message from the account you want me to reply to and make sure that your Reply-to address is properly set! There's nothing quite so frustrating as spending an hour or more carefully researching the answer to an interesting question and composing a detailed response, only to have it bounce because my correspondent sent her feedback from a public terminal and neglected to set the browser preferences to include her actual email address.

I also adhere to the old saying "If you like this book, tell your friends. If you don't like it, tell me." I'm especially interested in hearing about mistakes. This is the fourth edition. I've yet to make it perfect, but I keep trying. As hard as I and the editors at O'Reilly

worked on this book, I'm sure there are mistakes and typographical errors that we missed here somewhere. And I'm sure that at least one of them is a really embarrassing whopper of a problem. If you find a mistake or a typo, please let me know so I can correct it. I'll post it on the O'Reilly website at http://oreil.ly/java_np_errata. Before reporting errors, please check one of those pages to see if I already know about it and have posted a fix. Any errors that are reported will be fixed in future printings.

Using Code Examples

This book is here to help you get your job done. In general, if this book includes code examples, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Java Network Programming*, Fourth Edition, by Elliotte Rusty Harold (O'Reilly). Copyright 2014 Elliotte Rusty Harold, 978-1-449-35767-2."

If you feel your use of code examples falls outside fair use or the permission given here, feel free to contact us at permissions@oreilly.com.

Safari® Books Online



Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technol-

ogy, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at <http://oreil.ly/java-network-programming>.

To comment or ask technical questions about this book, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Acknowledgments

Many people were involved in the production of this book. My editor, Mike Loukides, got things rolling, and provided many helpful comments along the way that substantially improved the book. Dr. Peter “Peppar” Parnes helped out immensely with Chapter 13. The technical editors all provided invaluable assistance in hunting down errors and omissions. Simon St. Laurent provided crucial advice on which topics deserved more coverage. Scott Oaks lent his thread expertise to Chapter 3, proving once again by the many subtle bugs he hunted down that multithreading still requires the attention of an expert. Ron Hitchens shone light into many of the darker areas of the new I/O APIs. Marc Loy and Jim Elliott reviewed some of the most bleeding edge material in the book. Timothy F. Rohaly was unswerving in his commitment to making sure I closed all my sockets and caught all possible exceptions, and in general wrote the cleanest, safest, most exemplary code I could write. John Zukowski found numerous errors of omission, all now filled thanks to him. And the eagle-eyed Avner Gelb displayed an astonishing ability to spot mistakes that had somehow managed to go unnoticed by myself, all the other

editors, and the tens of thousands of readers of the first edition. Alex Stangl and Ryan Cuprak provided further assistance with spotting both new and lingering mistakes in this latest edition.

It isn't customary to thank the publisher, but the publisher does set the tone for the rest of the company, authors, editors, and production staff alike; and I think Tim O'Reilly deserves special credit for making O'Reilly Media absolutely one of the best houses an author can write for. If there's one person without whom this book would never have been written, it's him. If you, the reader, find O'Reilly books to be consistently better than most of the dreck on the market, the reason really can be traced straight back to Tim.

My agent, David Rogelberg, convinced me it was possible to make a living writing books like this rather than working in an office. The entire crew at *ibiblio.org* over the last several years has really helped me to communicate better with my readers in a variety of ways. Every reader who sent in bouquets and brickbats for previous editions has been instrumental in helping me write this much-improved edition. All these people deserve much thanks and credit. Finally, as always, I'd like to offer my largest thanks to my wife, Beth, without whose love and support this book would never have happened.

—Elliotte Rusty Harold
elharo@ibiblio.org
July 5, 2013

Table of Contents

Preface.....	xiii
1. Basic Network Concepts.....	1
Networks	2
The Layers of a Network	4
The Host-to-Network Layer	7
The Internet Layer	8
The Transport Layer	9
The Application Layer	10
IP, TCP, and UDP	10
IP Addresses and Domain Names	11
Ports	13
The Internet	14
Internet Address Blocks	15
Network Address Translation	15
Firewalls	15
Proxy Servers	16
The Client/Server Model	18
Internet Standards	19
IETF RFCs	20
W3C Recommendations	22
2. Streams.....	25
Output Streams	26
Input Streams	31
Marking and Resetting	34
Filter Streams	35
Chaining Filters Together	37
Buffered Streams	38

PrintStream	39
Data Streams	41
Readers and Writers	44
Writers	45
OutputStreamWriter	47
Readers	47
Filter Readers and Writers	49
PrintWriter	51
3. Threads.....	53
Running Threads	55
Subclassing Thread	56
Implementing the Runnable Interface	58
Returning Information from a Thread	60
Race Conditions	61
Polling	63
Callbacks	63
Futures, Callables, and Executors	68
Synchronization	70
Synchronized Blocks	72
Synchronized Methods	74
Alternatives to Synchronization	75
Deadlock	77
Thread Scheduling	78
Priorities	78
Preemption	79
Thread Pools and Executors	89
4. Internet Addresses.....	93
The InetAddress Class	95
Creating New InetAddress Objects	95
Getter Methods	100
Address Types	102
Testing Reachability	106
Object Methods	106
Inet4Address and Inet6Address	107
The NetworkInterface Class	108
Factory Methods	108
Getter Methods	110
Some Useful Programs	111
SpamCheck	111