

a SQL query that performs well — they just map Java classes to database tables.

- In a slow economy, be prepared to pass at least four interviews to get hired. Back in 1999 two good interviews would be enough; in 2001 it was very difficult to even get an interview, let alone a job!
- In 2010 a large portion of the development is done in India. Even though the total cost of development is not low despite the lower hourly rates of offshore developers, Western hiring managers unfortunately have an impression that local candidates must be flexible in their rates, too. Being a native English speaker is not enough for you to charge top dollar. Always be technically current, otherwise the more motivated guys from overseas will leave you in the dust.

Good knowledge of the business terminology of your potential employer is also important. I'm not sure about the Silicon Valley or Europe, but here in New York just being a techie may not be good enough to get you a senior job. Of course, interviewing with Google or Microsoft is an exception.

For example, if you're applying for a Java position in a financial brokerage company and don't know what a short sale is, this may be a deal breaker. If you are a senior developer you should be able to hit the ground running: Try to find out from your recruiter as many details as possible about the business. Do your homework, and you'll get the job! Recruiters are desperately looking for good programmers and you can be one of them.

To Get or Not to Get Certified?

Any certification program is a business for the vendor that wants to sell training to award certificates. Oracle is no exception: you can find various certification programs to rate your Java skills. The following website offers various certification programs in Java SE and EE tech-

nologies: http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=140#13.

Although most certification programs require you to pass a multiple-choice computer test, some of them (such as those for Java EE architects) will give you a challenging task to design and program.

Overall I think it's a good idea to prepare yourself and go through the certification process, because it will definitely improve your understanding of the Java language or a specific enterprise technology or framework. In some areas it may also improve the performance of your résumé. It'll also help you slip through computer screening tests that are used by job placement agencies. Just check job postings from your potential employers to see if they insist on having certificates.

But I also want to caution you against overestimating the importance of getting Java certification. When I see a résumé that starts with a list of certificates, it tells me that this candidate is good...at passing multiple-choice tests. Having practical hands-on experience working on open-source or enterprise projects is a lot more valuable than any certificate you might have earned. If you have certificates — fine, but keep them somewhere at the end of your résumé rather than trying to present them as a major achievement.

Technical Questions and Answers

Suggesting a list of technical questions for an interview is a risky business. Seasoned Java developers have different views on what's fair and what's not fair to ask. But junior developers who just completed a Java tutorial can definitely benefit from some guidance in preparing for Java job interviews.

The following are suggested technical interview questions and expected brief answers on various Java-related topics. The questions included in this section are those that I had to answer while working on

various projects. In no way is this a complete list of possible questions, but it will definitely give you an idea of what to expect at an interview. Most of the answers to these questions can be found in this book, but some will require additional research.

Q: What's the difference between an interface and an abstract class?

A: An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes you have to inherit your class from the abstract one because Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.

Q: How do you deploy a web application in the application server that you currently use?

A: In most Java EE application servers you create a web archive (war file) and copy it to the assigned directory per the application server documentation. (You should be able to explain the directory structure in a war file.)

Q: What's the usage of the keyword `static`?

A: It's used in declarations of methods and variables to make them available without creating an instance of the class. For example, the `main()` method is a `static` one. If a variable is `static`, its value is shared by all instances of the class.

Q: How can you force garbage collection?

A: You can't force garbage collection, but you can request it, because JVM does not guarantee that it'll be started immediately. Invoking `System.gc()` requests Java garbage collection.

Q: Explain the usage of event adapters.

A: Some `Swing` event listener interfaces declare multiple methods (for example, `WindowListener` has seven methods) and a class must implement all of them. Adapters already have all these methods predefined with empty bodies. So if a class needs to process just one of the `WindowListener` events, it has to override only the corresponding method of `WindowAdapter`.

Q: How do you decide if explicit casting is needed?

A: If you assign a superclass object to a variable of a subclass's data type, you need to use explicit casting. For example:

```
Object a;  
Customer b;  
b = (Customer) a;
```

Java generics eliminate the need for explicit casting. For subclass-to-superclass assignments the casting is performed automatically. You can also cast to interfaces that a class implements.

Q: Can you perform casting between objects of different types?

A: No, you can't. The objects can have a superclass-subclass relationship, or you can cast to interfaces implemented by a class.

Q: Can a Java class be inherited from two classes?

A: No, Java does not allow multiple inheritance, but a class can implement multiple interfaces, which, to some extent, can be used as a workaround.

Q: Both applets and Servlets are deployed on the server. What's the difference between the run-time environments of applets and Servlets?

A: Applets run in a JVM on the user's machine under control of a web browser, and they have security restrictions limiting access to a computer's devices. Servlets run on the server side under control of the Servlet container or Java EE Application Server.

Q: What's the difference between constructors and regular methods?

A: Constructors must have the same name as the class and cannot return a value. They are invoked only once, while regular methods can be invoked many times.

Q: What's the difference between the HTML methods `Get` and `Post`?

A: `Get` appends the parameters to the URL, and the resulting URL is bookmarkable. `Post` allows the sending of objects as well as text data, while `Get` works with text only.

Q: What's a cookie? Which Java components create them?

A: A cookie is an object that represents a name/value pair. Servlets, JSP, and JSF can create and send cookies to a web browser that saves them on the user's disk in a special directory. Cookies help a Servlet identify a user. For example, a bank can store your account number in a cookie file on your machine, so you do not need to enter it on a log-on screen.

Q: Can a non-abstract class have both abstract and concrete methods?

A: No, but only abstract classes can have abstract methods.

Q: Explain the use of Java packages.

A: Packages offer a way to organize multi-file projects. They also help in resolving naming conflicts when different packages have classes with the same names. Package access level also enables you to protect

data from being used by unauthorized classes, permitting only the classes from the same package to see each other's member variables.

Q: Explain the usage of the keyword `transient`.

A: The `transient` keyword indicates that the value of this member variable does not have to be serialized with the object. When the class gets de-serialized, `transient` variables will be initialized with the default values of the variable data types (such as `0` for integers).

Q: What do you know about thread synchronization? Explain the difference between

```
public void synchronized myMethod() { ... }
```

and

```
public void myMethod() {  
    ...  
    synchronized (some_object) {... }  
}
```

A: The keyword `synchronized` is used to prevent race conditions when more than one thread tries to update some values. Synchronized blocks are preferable to synchronized methods because they place locks for shorter periods. But before manually synchronizing blocks of code you should see if you can use one of the collection classes from the package `java.util.concurrent` as a better solution for dealing with race conditions.

Q: What's the difference between the methods `sleep()` and `wait()`?

A: The code `sleep(1000)` puts the thread aside for exactly one second. The code `wait(1000)` makes the thread wait for up to one second. A thread can

stop waiting earlier if it receives the `notify()` or `notifyAll()` call. The method `wait()` is defined in the class `Object`, but `sleep()` is defined in the class `Thread`.

Q: What do you know about the method `invokeLater()`?

A: This method is defined in the class `SwingUtilities` and is used to ensure that `Swing` windows will be updated through the event-dispatching thread. You can also do this with the `SwingWorker` class.

Q: What's the difference between creating threads subclassing the class `Thread` and implementing the `Runnable` interface?

A: Thread creation with the class `Thread` requires that your class be inherited from it. Then you need to create an instance of your class and call its `start()` method. If a class implements the `Runnable` interface the procedure is different — you have to create the instance of your class, and the instance of the `Thread` object, passing the `Runnable` class to the latter. Using `Runnable` enables you to create threads from classes that have to be inherited from classes other than `Thread` classes.

Q: How can you create a thread that can return a value from its method `run()`?

A: You need to implement `Callable` interface.

Q: What would you use to compare two `String` variables — the `equals()` method, or the `==` operator?

A: I'd use `equals()` to compare the values of the `Strings`, and the `==` operator to check if two variables point at the same `String` object in memory.

Q: What do you know about the MVC design pattern?

A: MVC stands for the model-view-controller design pattern, which is used to separate presentation modules from business logic and data

ones. The model part represents the data and the business logic of the application, the view is a visual representation (for example, a UI), and the controller accepts the data from the view and passes it to the model and vice versa. For example, a JSP page is a view, a Servlet is a controller, and a Java bean represents a model.

Q: How can you reduce the time spent by JVM on creating frequently used instances of some objects?

A: I'd create my own or configure existing object pools (e.g., DataSource or EJB objects can be configured to use object pools of a certain size).

Q: Will the following statement create the file `xyz.txt` on disk?

```
File a = new File("xyz.txt");
```

A: No, it just creates an object pointing to this file.

Q: How can a subclass invoke a method defined in a superclass?

A: Java has the keyword `super`. If a method has been overridden, but you'd like to invoke its version defined in the superclass, use the following syntax: `super.myMethod();`. To call a constructor of the superclass, just write `super();` in the first line of the subclass's constructor.

Q: What access level do you need to specify to ensure that only classes from the same directory can access the directory?

A: You do not need to specify any access level. In this case, Java will use the default package access level, which will guarantee visibility only to the classes located in the same directory (package).

Q: What's the use of JNDI?

A: JNDI is a Java API for naming and directory servers. It is used for finding Java objects by name in a distributed application. For example, you can use JNDI to get a reference to the `DataSource` or `JMS` objects.

Q: Do you have to invoke a method that reads a file inside the `try/catch` block?

A: You have to either put it in the `try/catch` block or declare that the calling method may throw an exception, for example:

```
void myMethod() throws IOException{...}
```

Q: If you need to catch more than one exception (such as `FileNotFoundException` and `IOException`), does the order of the `catch` statements matter?

A: Yes, it does. `FileNotFoundException` is inherited from `IOException`. The exception subclasses have to be caught first.

Q: Explain in detail the data workflow between a web page and a Servlet called `FindBooks` after the user presses Submit. (In this question interviewers can replace the web page and Servlet with any other client-server components. The main point is to find out if the job applicant understands the entire end-to-end data flow.)

A: The web browser connects to the server located at the specified URL, and if `FindBooks` has not been started yet, the Servlet container will start it, invoking the servlet's `init()` method followed by the `service()` method, which in turn calls `doGet()` or `doPost()`, depending on the method of HTTP request. The objects `HttpServletRequest` and `HttpServletResponse` are used for the interaction between the client and `FindBooks`. The Servlet's output can be sent back to the user by means of one of the methods of `HttpServletResponse`, for example `println()`.

Q: Explain the process of getting the data from a database table in a Java SE program using JDBC. Which classes and methods have to be used?

A: First you load the appropriate JDBC driver using `Class.forName()`. After that you get the `Connection` object using `DriverManager.getConnection()`. Then you create a `Statement` object and invoke one of its methods, like `executeQuery()` or `executeUpdate()`. Process the `ResultSet`, if any, and close the `Connection`, `Statement`, and `ResultSet` objects.

Q: What are the differences among the Java keywords `final`, `finalize`, and `finally`?

A: Depending on its position, the keyword `final` means either that the variable can be initialized only once, or that you cannot override a method, or that you cannot subclass a class.

The method `finalize()`, if defined in your class, is invoked by the garbage collector when it's ready to release a memory used by the instance of your class.

The keyword `finally` is used in a `try/catch` block to place code, such as code to close a stream, that must be executed whether the code in the `try` block succeeds or fails.

Q: Can you declare fields in an interface?

A: Yes you can, but they should be `final` and `static`.

Q: Can an inner class, declared inside a method, access local variables of this method?

A: It's possible only if the variables are `final`.

Q: Give an example of an anonymous inner class.

A: An event listener class can be created on the fly as an anonymous inner class, for example:

```
addActionListener (new ActionListener() {  
    public void actionPerformed (ActionEvent e) {  
        doSomething();  
    }  
});
```

Q: What could be used to keep track of sessions in web applications?

A: You can use cookies, URL rewriting, and the `HttpSession` object.

Q: Will session management with cookies always work?

A: No, if a user disables cookies in the web browser, it won't work. In such cases application servers usually automatically switch to URL rewriting.

Q: How can you stop a long-running thread?

A: The class `Thread` has a deprecated method called `stop()`, but this method does not guarantee that the thread will be stopped. Depending on the process that is run by this thread, you could try to close connections or open streams, if any, or use the method `interrupt()`.

Q: When could a Java class be called a bean?

A: You call the class a bean if it has a nonargument constructor, implements the `Serializable` interface, and has public setter/getter methods for its private properties.

Q: What are the advantages of using JSP rather than Servlets?

A: JSP enables you to separate presentation from business logic and the resulting web page can be modified by people who do not know

Java.

Q: Give an example that shows the difference between the use of the operators `&&` and `&`.

A: In the following code snippet the second expression in the `if` statement will not even be evaluated if variable `a` is `null`. If a single ampersand were used here we'd get a `NullPointerException`.

```
String a=null;
if (a==null && a.length()>10) {...}
```

Q: Name some predefined JSP variables.

A: `request`, `response`, `out`, `session`...

Q: How do you deploy a JSP?

A: You can place `.jsp` files in a document root directory of the Servlet container or application server, or create a war file.

Q: Why is JSF preferable to JSP from the MVC design pattern perspective?

A: JSF enforces separation of the presentation and the processing logic, while JSP doesn't.

Q: What's the default port number that web servers use for HTTP-based communications?

A: HTTP requests go through port 80, HTTPS ones through port 443.

Q: What method of the Servlet class is an equivalent of a constructor in a regular Java class?

A: The Servlet's method `init()` plays a similar role, but because all clients use the same instance of the Servlet, you should initialize only those variables that are allowed to have the same value for each user's request, such as the name of the database server.

Q: Which is faster, array or `ArrayList`?

A: Most likely array is faster. It can be used if you know in advance the number of elements in the array. `ArrayList` is an API on top of an array. You do not need to know its size in advance — new elements can be added as needed. Arrays work faster because JVM allocates memory only once for all elements. But to give you a better answer I'd have to write an appropriate benchmark in context to find out.

Q: What do you know about Java reflection?

A: Reflection is a way of finding information about a Java class during run time. For example, you can find out the constructors and the method signatures of a particular class. The class `Class` has such methods as `getConstructor()`, `getFields()`, `getMethods()`, and others.

Q: What would you do if your Java program crashed during the run time with an “out of memory” error?

A: I'd try to increase the size of the JVM's dynamic memory (heap) on program start-up using command-line parameters. For example, you can request the minimum heap size of 512 MB and the maximum heap size of 1024 MB as follows:

```
java -Xms64 -Xmx512 MyProgram
```

Q: How can you ensure that only one instance of some class can be created in your application?

A: You need to implement the singleton design pattern. Create a class with a private constructor and provide a public `Get` method that will return the only instance of this class, for example `MyClass.getInstance()`.

Q: What's the major difference between a `Hashtable` and a `HashMap`?

A: The `Hashtable` class is internally synchronized, while the `HashMap` is not.

Q: Name some design patterns.

A: Singleton, MVC, data transfer object, facade...

Q: The word stateless is used to describe a session EJB as well as HTTP protocol. What's the meaning of the word stateless in these two cases?

A: HTTP protocol is page-based, meaning that the web browser does not hold the connection between requests, but in the case of an EJB, stateless means that the session bean cannot be used to store the state of a particular client.

Q: Are Java objects passed by value or by reference?

A: Objects are passed by reference, but their reference variables are passed by value.

Q: How can a Java client access a message-driven bean?

A: Java clients don't access MDBs directly — they just place messages into a queue or publish them to topics, and the MDB retrieves them from there.

Epilogue

The book is over. Even though it's rather small in size for the number of the covered subjects, I tried to touch on a wide spectrum of topics that most Java practitioners have to know. You may still need to do ad-

ditional research on certain subjects, depending on your project needs, but at least now you know where to dig.

I really hope that you'll keep this book handy as a quick reference or for a refresher when it's time to hit the job market. No matter what your motivation is, have fun reading — I certainly had fun writing it. You can send thank-you notes and suggestions via e-mail to yakovfain@gmail.com. Thank you for reading my book!



Appendix

What's on the DVD?

This appendix provides you with information on the contents of the DVD that accompanies this book. For the most up-to-date information please refer to the ReadMe file located at the root of the DVD. Here is what you will find in this appendix:

- Using the DVD
- What's on the DVD
- Troubleshooting

Using the DVD on a PC

To access the content from the DVD, follow these steps:

1. Insert the DVD into your computer's DVD-ROM drive. The license agreement will appear.



The interface won't launch if you have autorun disabled. In that case click Start ⇒ Run (For Windows Vista, Start ⇒ All Programs ⇒ Accessories ⇒ Run). In the dialog box that appears, type **D:\Start.exe**. (Replace D with the proper letter if your DVD drive uses a different letter. If you don't know the letter, see how your CD drive is listed under My Computer.) Click OK.

2. Read through the license agreement, and click the Accept button if you want to use the DVD.

3. The DVD interface appears. Simply select the lesson video you want to view.

Using the DVD on a Mac

To install the items from the DVD to your hard drive, follow these steps:

1. Insert the DVD into your computer's DVD-ROM drive.

2. The DVD icon will appear on your desktop; double-click to open.

3. Double-click the Start button.

4. Read the license agreement and click the Accept button to use the DVD.

5. The DVD interface will appear. Here you can install the programs and run the demos.

What's on the DVD?

Most lessons in the book have a corresponding screencast that illustrates examples in the lesson and provides content beyond what is covered in print.

I recommend using the following steps when reading a lesson:

1. Read the lesson's text.

2. Watch the DVD and try to repeat all the instructor's actions on your computer.

3. Read the step-by-step instructions in the lesson's "Try It" section and complete the assignment on your own.

You can also download all the solutions to the "Try It" sections at the book's website. If you get stuck and don't know what to do next, visit the p2p forums (p2p.wrox.com), locate the forum for the book, and leave a post.

Troubleshooting

If you have difficulty installing or using any of the materials on the companion DVD, try the following solutions:

- **Turn off any antivirus software that you may have running:** Installers sometimes mimic virus activity and can make your computer incorrectly believe that it is being attacked by a virus. (Be sure to turn the antivirus software back on later.)
- **Close all running programs:** The more programs you're running, the less memory is available to other programs. Installers also typically update files and programs; if you keep other programs running, installation may not work properly.
- **Reference the ReadMe:** Please refer to the ReadMe file located at the root of the CD-ROM for the latest product information as of publication time.
- **Reboot if necessary:** If all else fails, rebooting your machine can often clear any conflicts in the system.

Customer Care

If you have trouble with the CD-ROM please call the Wiley Product Technical Support phone number, (800) 762-2974. Outside the United States call 1 (317) 572-3994. You can also contact Wiley Product Technical Support at <http://support.wiley.com>. John Wiley & Sons will provide technical support only for installation and other general quality-control issues. For technical support on the applications themselves, consult the program's vendor or author.

To place additional orders or to request information about other
Wiley products, please call (877) 762-2974.



Wiley Publishing, Inc.

Java® Programming 24-Hour Trainer

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2011 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-88964-0
ISBN: 978-1-118-05817-6 (ebk)
ISBN: 978-1-118-05818-3 (ebk)
ISBN: 978-1-118-05819-0 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accu-

racy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2010942181

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Wrox Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affili-

ates, in the United States and other countries, and may not be used without written permission. Java is a registered trademark of Oracle, Inc. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

In memory of my friend, Felix Rubinchik.

About the Author



Yakov Fain works as a software architect for Farata Systems, a company that provides consulting services in the field of development of enterprise applications. He has authored several technical books and lots of articles on software development. Sun Microsystems has awarded Mr. Fain with the title of Java Champion, which has been given to only 150 people in the world. He leads the Princeton Java Users Group. Yakov's Twitter ID is @yfain.

About the Technical Editor

Elliott Rusty Harold is originally from New Orleans, to which he returns periodically in search of a decent bowl of gumbo. However, he currently resides in the Prospect Heights neighborhood of Brooklyn with his wife Beth, dog Thor, and cat Marjorie. He is the author of numerous books including Refactoring HTML, Java Network Programming, Java I/O, and Processing XML with Java. His open source projects include the XOM Library for processing XML with Java and the Amateur media player.

Credits**Executive Editor**

Robert Elliott

Project Editor

Charlotte Kughen, The Wordsmithery LLC

Technical Editor

Elliote Rusty Harold

Production Editor

Kathleen Wisor

Copy Editor

Sadie Kleinman

Editorial Director

Robyn B. Siesky

Editorial Manager

Mary Beth Wakefield

Freelancer Editorial Manager

Rosemarie Graham

Marketing Manager

Ashley Zurcher

Production Manager

Tim Tate

Vice President and Executive Group Publisher

Richard Swadley

Vice President and Executive Publisher

Barry Pruett

Associate Publisher

Jim Minatel

Project Coordinator, Cover

Katie Crocker

Compositor

Jeff Lytle, Happenstance Type-O-Rama

Proofreaders

Nancy Carrasco

Jen Larsen, Word One New York

Indexer

Robert Swanson

Cover Designer

Michael E. Trent

Cover Image

© FotografiaBasica / iStockPhoto.com

Acknowledgments

First of all I want to thank my family for understanding that stealing time from family to write a computer book is OK.

I'd also like to thank Elliott Rusty Harold from Google for his valuable comments during a technical edit of this book. Elliott himself has authored more than 20 computer books (see <http://www.elharo.com>) and is respected in the Java community around the world. His comments on my lessons ranged from the philosophical to a simple "I don't agree with this solution." I have addressed all of them.

Three people volunteered to help me with the book: Nick Golubyev (Singapore), Dmitry Danilevko (Russia), and Viktor Gamov (USA). They wrote several code samples for the book, which really saved me time! Thank you, guys!

Big thanks to the Wiley editors for doing a great job of editing and for not cursing me for not meeting deadlines.



Introduction

It doesn't take a rocket scientist to notice the recent serious decline in the publishing of computer books, to say nothing of computer magazines. I enjoy reading books about software and visit the nearby Barnes and Noble at least twice a month. During the last three years the number of bookcases dedicated to computer books has shrunk substantially.

Today newly baked programmers prefer Google to books. They don't realize that Google is OK when you know what you are looking for, while books can give you a new perspective on how other people develop software.

During a recent visit to a major bookstore I noticed a series of books on digital photography. I'm one of the legions of people who are thinking of replacing a point-and-shoot camera with a digital SLR such as a Nikon D90. There were about 10 different thin books on that shelf, and I picked the one on the D90. It was about 65 pages long. I quickly skimmed the pages, learning that compared to a pocket camera one of the main advantages of D-SLR is the ability it gives you to change lenses. On top of that, these cameras use a small mirror that enables you to see exactly what you're shooting, plus you can take more shots per second. Not much information for a \$25 book.

The secret was that the book came with a DVD, which contained 80 minutes of instructional videos! The printed part of the book and the DVD complemented each other. The wannabe photographers don't want to read books.

The new generation of computer programmers doesn't want to read either. They want to watch videos. They can be YouTube videos on how to do something, or screencasts. The material has to be prepared for easy consumption.

But because not everything in the world of software can be video-taped, future computer books will still include several hundred pages — some code samples and short instructions will go there.

Remember those 1,500-page Bibles on software? It seems that their days are over. These days a 500-page computer book can be considered thick. Five years from now a 200-page software book will become the standard. But books will come with DVDs on which the authors walk you through all the examples mentioned on paper.

I like this new 24-Hour Trainer series from Wiley Publishing. This is not to say that you can learn the software covered in these books within 24 hours. It's about having a trainer that's with you 24 hours a day. Each book in this series is a set of short chapters containing a minimum of theory but accompanied by an instructional DVD with screencasts recorded by the author.

Who This Book Is For

This book is for anyone who wants to learn how to program with the Java language. No previous programming experience is expected.

This tutorial can be used by Java developers looking for simple working examples that use certain features of the language.

Accomplished Java developers can also use this book as a refresher while preparing for a technical job interview.

This tutorial can be used by university students who are interested in learning from a practitioner who has spent 20-plus years developing enterprise software for a living.

College professors should appreciate the fact that each lesson ends with a “Try It” section — a prepared home assignment for each lesson.

What This Book Covers

To be called a Java developer, a person has to know not only the core syntax of this programming language, but also the set of server-side technologies called Java EE (Enterprise Edition). This book covers both. At the time of this writing, the latest version of core Java is 6, and the latest release of Java EE is also 6.

Java is a general-purpose language — you can program applets running under control of a Web browser, stand-alone applications that run independently on the user’s computer, and applications that connect to remote servers. Java is used for writing applications running in mobile devices, cars, or in household appliances.

The first 25 lessons of this book cover a program that can run on users’ computers. The next 12 are about Java EE technologies used for Java programs that run on servers. The final lesson is dedicated to the process of getting prepared for Java technical job interviews.

This book doesn’t cover Java programming for mobile devices.

How This Book Is Structured

This book is a tutorial. Each lesson walks you through how to use certain elements and techniques of the Java language or gives an introduction to the server-side Java EE technologies. The “Try It” sections serve as continuations of materials explained in the lessons. The screencasts included on the DVD usually illustrate how the author compiles and runs code samples included in the lessons. You can choose to read the lesson and then try to run the example on your own, or you can read the lesson, watch the video, and then try to run the examples. After this part is done, start working on the assignments from the “Try It” section.

This book contains 38 lessons, which are not formally broken into sections. The first 25 lessons cover Java SE topics. Lessons 26 through 34 cover Java EE topics. Lessons 35 and 36 introduce you to popular third-party frameworks Spring and Hibernate. Lesson 37 demonstrates an application that utilizes several Java EE technologies. And, finally, Lesson 38 helps you to get prepared for the technical interview when you decide to look for a Java-related job.

The lessons are short and to the point. The goal is to explain the material quickly so you can start applying it hands-on as soon as possible. Some readers may feel that more explanation of certain subjects are required, and they are encouraged to do some extra research; there are lots and lots of online materials available on any Java-related subject. But the coverage of the material given in this book definitely helps in understanding what to focus on and what to look for.

What You Need to Use This Book

To do the examples in this book, you do not need to purchase any software—freely available software is used here. Installing Java Development Kit and Eclipse IDE is explained in the first two lessons, and this is all you need for Lessons 1 through 25. In Lesson 26, you install Java Application Server GlassFish, which is used for explanation of the materials in Lessons 26 through 34. If you decide to learn third-party frameworks Spring and Hibernate, you need to download and configure their software libraries as explained in Lessons 34 and 35. Whenever you need to download certain software, detailed instructions are given in the book and/or in the screencasts included on the DVD.

From the hardware perspective, you can use either PC running Windows or one of the Apple computers running Mac OS. Linux fans are also able to run all book samples. You should have at least 1GB of RAM on your computer to run all code examples from this book. But

adding more memory can make your Java compiler and Eclipse IDE work faster.

How to Read This Book

This book is a tutorial, and I assume in each lesson that the reader is already familiar with the materials from those that came before it. If you are new to Java, I highly recommend that you read this book sequentially.

Each lesson except the last one has a corresponding video screencast on the accompanying DVD, in which I either work on the assignment from the “Try It” section of the lesson or simply install and configure some software. Ideally, you should try to do all the assignments from the “Try It” sections on your own, and use the videos only if you get stuck or don’t understand the instructions. But if you prefer to learn by following the instructor, just watch the video first and then try to repeat the same actions on your own. Whatever works is fine.

Java is a multi-platform language and programs written for Microsoft Windows, say, should work the same way in Mac OS or on Linux computers. I’m using a Mac, but also have special software that enables me to run Microsoft Windows. In this book I use the open-source Eclipse Integrated Development Environment, which exists on all major platforms and looks pretty much the same on each. So regardless of your preferred operating system, you’ll be able to run all the code samples from this book.

Conventions

To help you get the most from the text and keep track of what’s happening, we’ve used a number of conventions throughout the book.



Notes, tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.



References like this one point you to the DVD or www.wrox.com to watch the instructional video that accompanies a given lesson.

As for styles in the text:

- We highlight new terms and important words when we introduce them.
- We show file names, URLs, and code within the text like so:

`persistence.properties.`

- We present code like the following:

```
We use a monofont type with no highlighting for most code examples.
```

Source Code

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All of the source code used in this book is available for download at www.wrox.com. You will find the code snippets from the source code are accompanied by a download icon and note indicating the name of the program, so you know it's available for download and can easily locate it in the download file. Once at the site, simply locate the book's title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book's detail page to obtain all the source code for the book.



Available for
download on
Wrox.com

Listings include the filename in the title. If it is just a code snippet, you'll find the filename in a code note such as this:

Code Filename



Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-470-88964-0.

After you download the code, just decompress it with your favorite compression tool. Alternately, you can go to the main Wrox code download page at www.wrox.com/dynamic/books/download.aspx to see the code available for this book and all other Wrox books.

The author will also maintain up-to-date code samples in the code repository hosted by Google at the following address:

<http://code.google.com/p/practicaljava/>.

This website also contains a sample online store application contributed by Nick Golubyev. Even though this application was not used in the book, it serves as a good example of an end-to-end application developed with Java EE 6 technologies. You can find the application by clicking the Source tab in the folder `java-ee`. You can find instructions on building the online store under the tab Wiki.

Errata

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata you may save another reader hours of frustration, and at the same time you will be helping us provide even higher quality information.

To find the errata page for this book, go to www.wrox.com and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors. A complete book list including links to each book's errata is also available at wrox.com/misc-pages/booklist.shtml.

If you don't spot "your" error on the Book Errata page, go to wrox.com/contact/techsupport.shtml and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

p2p.wrox.com

For author and peer discussion, join the P2P forums at p2p.wrox.com. The forums are a Web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At p2p.wrox.com you will find a number of different forums that will help you not only as you read this book, but also as you develop your

own applications. To join the forums, just follow these steps:

1. Go to p2p.wrox.com and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you wish to provide and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.



You can read messages in the forums without joining P2P but in order to post your own messages, you must join.

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.