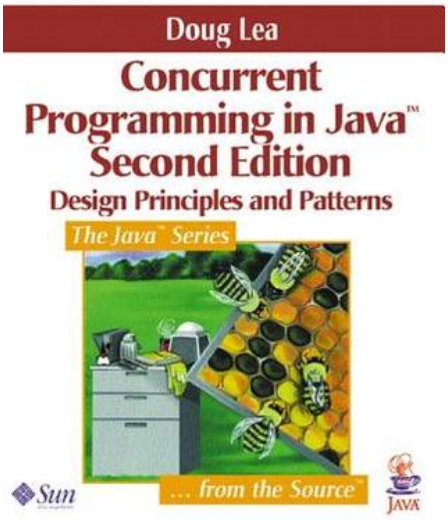

Concurrent Programming in Java?: Design Principles and Pattern

Doug Lea

Concurrent Programming in Java?: Design Principles and Pattern

Doug Lea

Concurrent Programming in Java?: Design Principles and Pattern Doug Lea
In this second edition, you will find thoroughly updated coverage of the Javao 2 platform and new or expanded coverage of: * Memory model * Cancellation * Portable parallel programming * Utility classes for concurrency control The Java platform provides a broad and powerful set of APIs, tools, and technologies. One of its most powerful capabilities is the built-in support for threads. This makes concurrent programming an attractive yet challenging option for programmers using the Java programming language. This book shows readers how to use the Java platforms threading model more precisely by helping them to understand the patterns and tradeoffs associated with concurrent programming. You will learn how to initiate, control, and coordinate concurrent activities using the class java.lang.Thread, the keywords synchronized and volatile, and the methods wait, notify, and notifyAll. In addition, you will find detailed coverage of all aspects of concurrent programming, including such topics as confinement and synchronization, deadlocks and conflicts, state-dependent action control, asynchronous message passing and control flow, coordinated interaction, and structuring web-based and compu

Concurrent Programming in Java?: Design Principles and Pattern Details

Date	:	Published November 4th 1999 by Addison-Wesley Professional (first published November 1996)
ISBN	:	9780201310092
Author	:	Doug Lea
Format	:	Paperback 432 pages
Genre	:	Computer Science, Programming, Science, Technical, Technology, Software, Nonfiction

From Reader Review Concurrent Programming in Java?: Design Principles and Pattern for online ebook

Dan Boeriu says

Humble pie for the Java developer. It doesn't get better.

Martin Chalupa says

A very good overview of concurrent programming in Java. I was worried that the book might be now obsolete, but most topics are covered in timeless manner. The book goes over basic concurrency constructs in Java such as synchronized, wait, notify. Then it uses them to show useful design pattern. And those pattern are generalised into higher level concurrency utilities. I like that everything is clearly explained. The author provide details which leads to certain design decision. What is useful in what context.

Chad Rhyner says

In this book, it covers a lot of different concurrent programming constructs. Some of these constructs are commonly known for Computer Scientists (things such as Binary/Counting Semaphores, fork/join, acquire/release constructs, latches.). Going into this book, I thought this would be more of a review of some of my Computer Science studies. I was pleasantly surprised to learn new terms and techniques to handle concurrency issues that I never would have thought of without reading this book. These examples all pleasantly surprised me, such as "Oh yea, I guess that would be useful to do in a situation like when you want to make sure that all systems in a distributed system are shut off."

One thing it was also nice to read is the trade-offs between using solution x versus solution y. In the real world (and in Computer Science, in general), there are constant tradeoffs. The age old example is "use more memory for the algo -or- wait more time for the algo to complete" is a common paradigm that software engineers need to make on a day-to-day basis. It is nice to hear from experience some of the trade-offs that need to be considered when using a technique described in the book.

I would have given this book a 5-star rating if it was not for one thing that continually irked me. As you are reading along, instead of referencing a concept in the book that may have already been mentioned or is going to be mentioned...it always referenced the Section Numbers. It was annoying and tedious to read. A sentence might be like this:

"While technique is used extensively, if you are looking for more flexibility in the framework that you use, refer to Section 2.3.2.3." Instead of giving me some pedantic section number, why don't you just say the section name? Say something like "or you could use techniques described in the Guarded Methods section to achieve the same results is a less complicated way.". The section numbers instead of the section names bugged me enough to remove 1 star from the review. I was thinking about removing 2 stars, but I think that was probably a bit too drastic.

Borys says

Well, it took me a very long time to finish this one :) A couple of months or so. Style of the book is very academic, and I had to re-read some passages many times (also I've got to mention that it is somewhat hard to read this book while riding the subway :)

I never had any problem with cross references, as other reviewer mentions. And I don't think it's not worth a read because it's too old and outdated, I think it's as useful as ever. It explains some basic stuff, it elaborates on Java memory model, on monitors, on how wait and notify really work internally, and also covers large number of design patterns and paradigms (and therefore it will be relevant as long as those patterns are used in practice).

So, my final word is: very useful book to read if you really want to understand how to write good and live and safe concurrent programs.

P.S. I first heard about this book while reading this review: http://steve-yegge.blogspot.com/2008/...

Christophe Addinquy says

Ma note de lecture complète en français ici

Jack.liu says

Concurrent Programming in Java(TM): Design Principles and Patterns (3rd Edition)

Amiruddin Nagri says

One of my favorite topics, and the legendary book every developer should read

---

Otis Chandler says

If you deal with any concurrency issues in Java this is a must - as is anything by Doug Lea

---

Michael says

Strangely seems to assume you already know all the relevant issues with concurrent programming.
