

Java 推荐读物与源代码阅读

江苏 无锡 缪小东

1. Java 语言基础

谈到 Java 语言基础学习的书籍，大家肯定会推荐 Bruce Eckel 的《Thinking in Java》。它是一本写的相当深刻的技术书籍，Java 语言基础部分基本没有其它任何一本书可以超越它。该书的作者 Bruce Eckel 在网上被称为天才的投机者，作者的《Thinking in C++》在 1995 年曾获 SoftwareDevelopment Jolt Award 最佳书籍大奖，《Thinking in Java》被评为 1999 年 Java World “最爱读者欢迎图书”，并且赢得了编辑首选图书奖。作者从 1986 年至今，已经发表了超过 150 篇计算机技术文章，出版了 6 本书（其中 4 本是关于 C++ 的），并且在全世界做了数百次演讲。他是《Thinking in Java》、《Thinking in C++》、《C++ Inside & Out》《Using C++》和《Thinking in Patterns》的作者，同时还是《Black Belt C++》文集的编辑。他的书被读者称为“最好的 Java 参考书……绝对让人震惊”；“购买 Java 参考书最明智的选择”；“我见过的最棒的编程指南”。作者的非凡才华，极其跨越语言的能力，使作者被选为 Java 发展 10 年间与 Java 关系最密切的 10 个人物之一。

《Thinking in Java》讲述了 Java 语言的方方面面，很多 Java 语言的老手都评价“这是一本将 Java 语言讲得相当丑陋的书”。该书谈及了 java 语言的很多细节，每一个方面都是相当深刻的。通过本书你可以看到“丑陋的”java 语言。

网络上关于 java 语言讲解的视频很多很多，其中不凡有垃圾。《翁恺—JAVA 语言》可能是你学习 java 语言基础的唯一选择，该讲座基本按照《Thinking in Java》这本书讲解，其中不凡有翁老师的很多有意思的笑话。我很幸运学习就是从此视频开始的。内容包括 30 讲，我总共看了 3 遍。

不过，对于初学者我不太推荐使用《Thinking in Java》，我比较推荐 Prentice Hall PTR 的《Core Java 2》国内称为《Java 2 核心技术》，目前是第七版。网络上大家都可以下载到电子版。Oreilly 的《Java in a nutshell》也是一个不错的选择。读完以上两本后，你可以看看翁恺老师的视频，接着可以研究《Thinking in Java》了。

2. Java 数据结构

市面上关于 Java 数据结构的书本身就很少很少。大致有 Apress 的《Java Collections》，Jones 和 Bartlett 的《Data Structures in Java》、《Object-oriented Data Structures Using Java》以及 Prentice Hall 出版的《Data Structures and Algorithms in Java》（Dec 19, 2005）还有一本《Data Structures And Algorithms With Object-oriented Design Patterns In Java》。很幸运我的第一本英文书就是 Apress 的《Java Collections》（本书在国内可能根本就没有中文版——只能下载英文版了），很不错，讲得很有条理、很简单，是一本完完全全 Java Collections API 介绍的书籍，其中不凡有

扩展 API 的例子。这是我推荐你学习 java 数据结构的唯一一本好书。其它的 Jones 和 Bartlett 的那两本国内好像有一本中文版，想看你也可以看看。

在学习完 API 后，你可以看看 java.util 包中对应的类了。不过只有在学习过设计模式后你才有可能完全理解整个 Java Collections Framework。Java Collections Framework 使用了很多著名的设计模式如：迭代器（Iterator）模式，工厂方法模式、装饰器模式、适配器模式等等。通过研究 java.util 包中数据结构的源代码，你可以知道臭名昭著的 Properties 类的设计了，同时可能基本具备设计简单的数据结构的能力了。

所谓学习无止境，学习完 Sun 提供了 Java Collections Framework 后，你可以研究 Apache 的另一个 Java Collections Framework，很有意思哦。互为补充的两个 Framework。

在大家学习、研究 Java Collections 之前，我提示一下 Java Collections 主要包括以下三部分：接口（Interface）、实现（Implementation）和算法（Algorithm）。

1. 接口主要有 List、Set、Queue 和 Map。List、Set 和 Queue 是 Collection 接口的子接口。
2. 实现主要是实现这些接口的具体类。如实现 List 接口的 ArrayList、LinkedList、Stack 和 Vector；实现 Set 接口的 HashSet、TreeSet 和 LinkedHashSet；实现 Queue 接口的 PriorityQueue、SynchronousQueue 等等；实现 Map 接口的 HashMap、TreeMap、Hashtable、Properties、WeakHashMap 等等。
3. 算法主要是由 Arrays 类和 Collections 类提供的，它是整个 Java Collection Framework 算法的核心。支持各种类型的排序，查找等常用操作。

Java Collections 中包含两个版本的数据结构，主要是原先的支持同步的数据结构和后来不支持同步的数据结构。

Java Collection Framework 在使用 Comparator 和 Comparable 接口支持排序。同时提供新旧两个版本的迭代器 Iterator 和 Enumeraton，以及它们如何转换等等。

在 java.util 包中的 Observable 接口和 Observer 类是考察者模式的核心。

.....

3. Java IO

市面上关于 IO 的书籍也仅仅只有 O'Reilly 出版社的两本，都是 Elliotte Rusty Harold 的著作。两本书的风格基本一致，推荐阅读是第一版的《Jjava I/O》，讲得比较浅显，内容相对比较集中，实例也很多。第二版今年 5 月国外才出版，很有幸我在网络上下载了第二版，讲得极其详细——726 页的大块头（我化了两个星期），这次将 NIO 和 IO 和在一起，还包括 J2ME 部分的，不过串口、并口通信部分好像类库支持不够，自己不能实际操作。

与第一版的《Jjava I/O》一起的 O'Reilly 还有一本《Jjava NIO》，也是很不错的哦。

大家在依次阅读完《Jjava I/O》以及《Jjava NIO》后，可以研究 java.io 包中的源代码了。在大家研究源代码前我给点提示：

Java 的 io 包主要包括：

1. 两种流：字节流（byte Stream）和字符流（character stream），这两种流不存在所谓的谁代替谁、谁比谁高级之说，它们互为补充，只是侧重点不同而已。
2. 两种对称：1. 字节流、字符流的对称；2. 输入、输出的对称。
3. 一个桥梁：将字节流转变为字符流的 InputStreamReader 和 OutputStreamWriter。其中必须注意：

1. PipedInputStream 和 PipedOutputStream 是两个比较有趣的类。
2. 支持 Buffered 的流是我们经常使用的类。
3. 装饰器 (Decorator) 模式在 java 最著名的应用就是用于 io 的设计。仔细研究各个 Filter 流与具体流的关系，多看设计模式的书籍。相信你会有所收获。
4. 学习好 io 包，是研究 net 包，rmi 包……的基础哦！

4 . Java 数据库

数据库的书籍太多太多了，也是太烂太烂了！这方面的书我基本都研究过，推荐的就看看 Apress 的《JDBC Recipes A Problem Solution Approach》很不错，国外 2005 年底才出版，（国内好像没有中文版，不过出了中文版也不一定值得看——国内经常将国外的书翻译得一塌糊涂、不堪入目）不过我们真的很幸运，网络上有电子版的。值得一看。推荐我看的第一本比较满意的——Wiley 出版的《Java Database Bible》，讲得很不错！

Sun 公司自己的关于 JDBC API 介绍的那一本《JDBC API Tutorial and Reference》也不错。我第二本 JDBC 的就是研究的这套 API。

不过目前这些书都是一些相对比较浮浅的 API 应用的书籍。有机会我会给大家带来介绍 JDBC API 以及 JDBC 实现内部细节的书！我尽快努力，同时希望得到大家的支持！

顺便给学习 JDBC 的朋友一点提示：

JDBC 的学习和使用主要是这套 API，其使用过程也是极其简单，下面是使用 JDBC 的一般流程：

1. 加载某个数据库的驱动 (Driver 类)，通常使用 Class.forName(“驱动类名”)；
2. 连接数据库——


```

          Connection con =
          DriverManager.getConnection(url, username, password);
      
```
3. 得到会话——Statement stmt = con.createStatement();
4. 执行操作——Result rs = stmt.executeQuery(“SQL 查询语句”);
5. 处理结果——


```

          while(rs.next()) {
              String col1 = rs.getString(1);
              .....
          }
      
```

简单吧！整个 JDBC 中可以变化的一般是：

1. 可以由 Connection 对象创建 Statement、PreparedStatement 和 CallableStatement 创建三种类型的 Statement。
 2. 可以创建多种类型的 ResultSet：支持单向移动和个自由移动；可更新的和不可更新的；支持不同等级的交易的……
 3. 数据输入的批处理。
 4. 结果集中特殊类型 (Blob、Clob、Array 和 Ref、Struct) 列的操作。
 5. 这些特殊类型的录入数据库。
 6. javax.sql 包中特殊结果集 (CachedRowSet、JdbcRowSet、WebRowSet) 的操作。
 7. 其它的就是一个 DataSource 了，也很简单！一个 J2EE 中的被管理对象
- 简单吧！相信大家很快就会征服 JDBC。

5. Java 网络编程

网络编程——一个神秘的、充满挑战的方向。不过在谈 Java 网络编程之前首先感谢 Sun 公司的开发人员，因为它们天才的设想，充满智慧的架构，使广大 java 程序员学习 java 网络编程变得异常简单。

Java 网络编程方面的书，我推荐 O'Reilly 的《**Java Network Programming**》，目前已经第三版了，以前的版本市面上肯定有！网络上早有第三版的电子版，国外 2004 年出版，706 页哦！讲得很全，比较深入，太深入的可能由于 Sun 有些东西没有完全公开，所以也就不好讲了，有兴趣的可以下载看看！第二本还是 O'Reilly 1998 年出版的《Java distributed computing》，基础部分写得比较详细，后面的实例还是值得研究的。

在大家阅读这些书之前，给大家一点提示：

java 网络编程其实相对比较简单，入门也很快很快。java 网络编程主要包括两个部分：

1. Socket；2. URL 部分。不过第二部分也完全建立在第一部分的基础上。

1. Socket 包括客户端的 Socket 和服务端端的 ServerSocket。还有就是 DatagramSocket 和 DatagramPacket，它对应于 UDP 通信协议。总之，Socket 部分是建立其它高级协议的基础。
2. URL 类是一个网络资源定位器，通常和具体的网络协议如 HTTP，FTP，Telnet……相关。通过该类可以连接网络上的资源，通过其 openStream 可以以 io 包中的流(InputStream) 的形式读取网络资源；通过其 OpenConnection 方法，可以打开一个连接，在此连接上不仅可以完成读的操作，还可以完成写的操作。

Java 的网络编程大体包括以上两部分。网络编程和 IO 以及多线程部分非常密切，在学习此部分前大家一定对这两部分了解比较透彻。

学习了以上部分你可以研究 java.net 包中的与此相关的源代码了！研究所有的源代码还为时尚早。在整个 net 包中包含：ContentHandlerFactory、URLStreamHandlerFactory、URLStreamHandler、URLClassLoader 等辅助类，它们构成了 java.net 网络编程的框架，通过研究其源代码，你不仅可以快速理解 java.net 包，还可以为以后扩展该包打下基础，甚至可以将此思维方式运用到自己的项目中。

到此为止你对 java.net 包应该才了解 60%，还有一部分你可以使用 JD Decompiler 之类的反编译软件打开你 JDK 安装目录下\jdkxxx\jre\lib 目录中的 rt.jar，用 WinRAR 之类的软件打开它的 sun.net 包，反编译所有的文件，它是 URL 类工作的细节。当研究完该 sun.net 包，你就会对整个网络编程很熟悉很熟悉了。

一切看起来我们已经对网络编程很精通了。其实不然，刚刚开始而已，要想深入，请继续吧！网络上很多优秀的网络编程库甚至软件可以为我们“添加功力”。如 Apache 的 HttpCore 和 HTTPConnection 是两个和 HTTP 协议相关库；JGroups 是研究分布式通信、群组通信的必读库；接着我们可以研究 P2P 的软件包，如 Sun 公司的 JXTA，它可能是 java 平台点对点通信未来的标准哦！接着你可以研究成熟得不得了，使用极其广泛得 P2P 软件 Azureus！
www.sourceforge.net 可以下载到！

千里之行始于足下！Just do it！（目前我也只研究了 net 包，其它的会在不久的将来继续深入。Sun 公司因为某些原因没有公开 net 的其它实现细节，在其允许将其源代码以文字的形式加以研究，以及允许将其没有公开的实现写入书中时，我很希望能出一本 java 网络编程的书籍，以飨广大读者！！）

6. Servlet 和 JSP

Servlet、JSP 的书也是满地都是！值得推荐的也仅仅两三本。实推 Addison Wiley 的《Servlets and JavaServer pages : The J2EE Technology Web Tier》，又是一本很厚的哦！国外 2003 年出版、784 页，讲得比较全，例子也很多，特别是第八章 Filter，举了几个不错的例子。其它所有我看到的关于 Servlet 和 JSP 的书都没有如此深入的！（可能有我没有看到而已）。O'reilly 的《Java Servlet Programming》和《Java Server Pages》相对比较易懂一些，可以读读！

在大家学习 Servlet 和 Jsp 之前我还是要提醒一下：

本质上说 Servlet 就是一个实现 Servlet 接口的、部署于服务器端的服务器端的程序罢了！它可以象写其它任何 java 应用程序一样编写，它可以操作数据库、可以操作本地文件、可以连接本地 EJB……编写 Servlet 程序的一般流程为：

1. 继承一个 HttpServlet 类；
2. 覆盖其 doGet、doPost 方法；
3. 在覆盖方法的内部操作方法参数 HttpServletRequest 和 HttpServletResponse。
4. 读取请求利用 HttpServletRequest。利用 HttpServletRequest 你可以操作 Http 协议的协议头、可以得到请求的操作方法、可以得到请求的路径、可以得到请求的字符串、以及和请求客户相关的信息，更主要的你可以得到 Cookie 和 HttpSession 这两个对象。
5. 利用 Cookie 你可以操作“甜心”对象或者将其写入 HttpServletResponse 中。
6. 向客户输出信息可以使用 HttpServletResponse。使用 HttpServletResponse 可以写入各种类型的协议头、可以增加 Cookie、可以重定向其它 URL、可以向客户发送 Http 协议的状态码。
7. 利用 HttpSession 在会话内完成你想实现的任何功能。

同时 Servlet 还提供了一些事件和事件监听器（简单的观察者模式而已）。还有就是过滤器（Filter）和包装器（ServletRequestWrapper、ServletResponseWrapper）——简单的流的使用和装饰器模式的使用。

学习 Servlet、JSP 必然要部署到服务器中，记住通常文件部署的步骤和参数的设置以及在程序中如何使用就可以了。

完全理解 Servlet 后，学习 jsp 相对比较容易了！Jsp 完全建立在 Servlet 的基础上，它是为了迎合那些喜欢在 Html 文档中嵌入脚本（如：PHP 之类的网页编程语言）的程序员的需要罢了！学起来也相当的容易！

一切看起来似乎那么的风平浪静，简单好学！简单的表象背后有其复杂的机理。要想对 Servlet 和 Jsp 彻底研究，你得研究 Tomcat 等开源软件的具体实现。它无非就是一个服务器，在客户利用网页通过 HTTP 协议向服务器发送请求后，服务器将此 HTTP 请求转化为相应的 HttpServletRequest 对象，调用你编写的 Servlet 罢了，在你的 Servlet 中你肯定操作了此 HttpServletRequest 了吧，同时操作了 HttpServletResponse 了吧，服务器就将此 HttpServletResponse 按照 HTTP 协议的要求利用 HTTP 协议发送给你的浏览器了！在服务器端的 Jsp 网页在被客户请求后，Tomcat 会利用编译软件，使用 javax.servlet.jsp 包中的模板，编译此 jsp 文件，编译后就是一个 Servlet！以后的操作和 Servlet 完全一样哦！

在 Servlet 和 Jsp 的基础上出现了，所谓的高级技术：JSTL，Struts……无非就是一些标签和 MVC 模式的使用。

继续前进吧！胜利就在前方！！

7. 多线程

一个看起来很神秘，却很容易上手、很难精通的方向！

我推荐两本我感觉很好的书籍。首先是我第一本能上手看的这方面的书，Sams 1998 年出版的《Java Thread Programming》，写得暴好，很容易读懂，我有空还时常看当时的笔记！要知道怎么好你自己看吧！第二本 OReilly 三次出版的《Java Threads》，最新是 2004 版，国内好像有中文版，推荐你还是看英文版的吧！书中谈到了与多线程相关的 N 个方向，如 IO、Swing、Collection 等等。

给大家一点提示吧！java 类库中与多线程相关的类不是很多，主要有：Thread、ThreadGroup 以及 ThreadLocal 和 InheritableThreadLocal 四个类和一个 Runnable 接口；关键字 synchronize、volatile；以及 Object 对象的 wait、notify、notifyAll 方法！

- 1 Thread 是多线程的核心类，提供了一系列创建和操作多线程的方法。
- 2 ThreadGroup 是一个管理 Thread 的工具类。
- 3 ThreadLocal 和 InheritableThreadLocal 为 Thread 提供了一个类似保险箱功能的存储线程对象的类！
- 4 Runnable 不用说了吧！
- 5 synchronize 是同步方法和同步块的核心哦！多个线程调用此方法时，只有一个线程可以使用此方法，其它方法阻塞，从而保证被操作对象内部状态完整性。某个线程调用带有 synchronize 的方法或块时会得到该对象的对象锁，完成块中的操作后释放此对象锁，从而其它对象可以继续操作。
- 6 wait、notify、notifyAll 提供了有效的等待/通知机制。Java 语言中每一个对象都有一个休息室，任何线程在其操作的对象的状态不满足的情况下，在该对象的休息室中休息，释放对象锁；当其它线程操作该对象后，唤醒休息室中的线程，它们再检查条件，当条件满足后，执行相应的操作。

多线程大致就这么多基础的！简单吗！这对于一个真正的程序员应该是不够的，真正对多线程要有所掌握，请您研究 java.util.concurrent 包吧！大师 Doug Lea 的作品，原先是一个开源的一致性编程的库，后来被 Sun 公司并入 java 类库。作者的网站上也有另外一个版本的该类库！值得研究的好东西！Hibernation、OpenJMS 等开源软件都使用了此包！

8. 设计模式

谈到设计模式很多人多会推荐 GOF 的那本，该书在 Amzon 上是五星级的推荐书籍。不过对于学习 java 没多久的、特别是 java 初学者，我可不推荐这本书。主要是该书的例子基本都是 C++ 的，很多细节没有讲述得足够清楚。

我给大家推荐的第一本是阎宏博士的《Java 与模式》，它是第一本中国人自己写的关于设计模式的书籍，写的比较有趣，融合了很多中华民族的文化和观念，例子、类图都比较多，

且相对简单！非常不错的入门书籍——又是大块头哦！

其次我推荐 Wiley 出版社出版的《Pattern In Java》一套三本，我才看了第一本，好像第二本不怎么样，第三本还不错！

第三本是中文翻译版的关于多线程模式的（很难得的中文翻译版）中国铁道出版社 2003 年出版的《Java 多线程设计模式》，将多线程模式讲得非常浅显，配有大量的图例，每章都有习题，最后有答案！我研究多线程模式就是由它开始的！

第四本，今年出版的 Head First 系列的《Head First Design Pattern》，秉承 Head First 系列图书的优点，大量的类图、丰富的实例、有趣的注解，值得购买！

其次在 J2EE 方向你可以研究阅读 Addison Wesley 2002 年出版的《Patterns of Enterprise Application Architecture》，众多大腕的作品，讲企业消息集成的！Sun 提供的《J2EE PATTERNS SL500》也很好！晚了推荐那一本 Amzon 4 星半的《Holub on patterns》，大师的作品，提供了，很值得研究的例子，不过对上面四本不是很熟悉的读者，最好不要读它！可能会让你比较累！

我学习设计模式经过一段很曲折的路线，前前后后大约看了 20 本，阎宏博士的《Java 与模式》我看了 4 遍，还排除我第一次基本没看懂的看！记得研一时老师给我们讲了 GOF 的那本，作为选修课，我和它们计算机系的硕士、博士们一起，到最后一个班 40—50 个人，不超过 3 个人明白，我也没有明白任何一点（基础差吧——主要我对 C++ 语言一点都不了解），凭我不伏输的性格，我认为我对 java 语言理解还可以，我就借了《Java 与模式》，结果还是基本没看懂。很有幸的是读研三时，听过了上交大饶若楠老师关于 Java OOP 语言的讲座，我懂了组合书籍模式等三种设计模式后，对其它模式有了强烈的兴趣和要征服它的愿望！工作后我买的第一本就是《Java 与模式》，第一遍花了 2 个月研究了 this 1000 多页的大块头，后来第三遍 15 天左右就可以搞定，笔记记了一大本！从此一发不可收拾。

选对书、埋头研究。相信很快就会入门的！

学习 Java 语言 8 个简单的部分，这只是我们研究 Java 语言的开始！这些都懂了充其量一个 java 程序员而已，后面的路很长很长！我们可以继续研究数据库实现的源代码、Servlet 服务器的源代码、RMI、EJB、JNDI、面向方面编程、重构、ANT 工具、Eclipse 工具、Spring 工具、JBoss、JOnAS、Apache Geronimo 等 J2EE 服务器！研究了这些你可能会成为一个出色的 J2EE Architecture！你可以继续研究剖析器、编译器、JNODE（java 写的操作系统）……

感谢大家有此耐心，听我罗罗嗦嗦大半天！感谢大家的阅读，感谢群里的朋友！这篇文章主要应群里朋友的呼声——不知道如何选书、不知道从何看起！大半天的功夫完成赶此文章，字句上难免有失误，同时由于能力有限不凡有错误！请阅读后批评指正！

上面基本是我研究 java 语言的顺序，以上书籍都是我阅读过的，不存在替任何出版社宣传的成分！有的方法可能不适合你，假如你能收获一点，两点甚至更多，请你不要吝啬推荐给你的朋友——共同学习！

感谢大家的阅读；感谢互联网的设计者；感谢 java 的设计师；感谢 www.open-open.com 和 www.sourceforge.net 网站！

2006-11-28

更多精彩请来：

<http://blog.163.com/miaoxiaodong78/>