疯狂Java

你是否在求职时被鄙视“基本功不扎实”而万分惭愧？

本书能检验你的基本功，并让你弥补不足。

本书归纳了Java学习者、工作者在学习、工作过程中最欠缺的技术短板。

形成内存管理、常见的陷阱、数据结构、程序调试、软件测试等16堂必修课！

你是否深入研究过它们，是否深入掌握它们？

谨以此书献给打算以编程为职业，并愿意为之疯狂的人。

突破程序员基本功的16课

李刚 编著

人民邮电出版社
POSTS & TELECOM PRESS

疯狂Java

突破程序员基本功的16课

李刚 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

疯狂Java：突破程序员基本功的16课 / 李刚编著
-- 北京 ：人民邮电出版社，2010.5
ISBN 978-7-115-22168-1

Ⅰ. ①疯… Ⅱ. ①李… Ⅲ. ①JAVA语言－程序设计
Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2010)第012230号

内 容 简 介

本书是著名 Java 领域研究专家、Java 语言培训大师、“疯狂 Java”创始人李刚老师的又一倾心力作。

在本书中，李刚老师一改枯燥的教学方式，专门面向 Java 初学者可能会遇到的各种学习问题，由点及面，详细讨论了 Java 内存管理、Java 编程过程中常遇陷阱、常用数据结构的 Java 实现和 Java 程序开发的方法与经验等内容。

这些问题，看似“司空见惯”，实际上却是很多 Java 初学者在初学阶段都会遇到的问题和疑难。李刚老师在本书中，正是试图为读者们展现出这些疑点、难点的实质，让读者能在瞬息之间，彻底掌握住这门语言的“内功心法”。

这不是一本包容了所有技术细节的手册，而是一本 Java 前辈对于晚辈们的提点和教导。书中很多内容，是李刚老师和他的众多学子曾亲身体验过的桎梏，非常具有参考意义。本书承载了无数前辈的谆谆教导之言，向你展示着一个痛并快乐着的 Java 世界。

疯狂 Java：突破程序员基本功的 16 课

◆ 编　　著　李　刚
责任编辑　蒋　佳

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
中国铁道出版社印刷厂印刷

◆ 开本：800×1000　1/16
印张：29.25
字数：670 千字　　2010 年 5 月第 1 版
印数：1－4 000 册　　2010 年 5 月北京第 1 次印刷

ISBN 978-7-115-22168-1

定价：59.00 元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前 言

Java 语言的开发人群越来越大了，大量程序员已经进入或正打算进入 Java 编程领域。这当然和 Java 语言本身的优秀不无关系，却也和 Java 语言入门简单有关。一个毫无编程基础的初学者，只要有点数据库和 SQL 的基础，大概花不到一个月时间就可以学会编写 JSP 页面，说不定这样就可以找到一份 Java 编程的工作了。如果他肯再多下点功夫，学习一下编写 Struts Action 类、配置 Action，编写 Spring Bean 类和配置 Bean，他甚至可以自我感觉很不错了。

问题是：这种“快餐式”、“突击式”的程序员真的能满足要求吗？如果仅仅满足于这些简单的、重复式的开发，他们也许没有太多的问题，但他们可能很少有突破的机会。究其根本原因，很大程度上是因为他们的基本功不够扎实。对他们而言，与其说 Java 是一种面向对象的语言，不如说更像一种脚本；他们从源代码层次来看程序运行（甚至只会从 Eclipse 等集成开发环境中看程序运行），完全无法从底层内存分配的角度来看程序运行；他们天天在用 Java 类库、用 Struts、用 Spring，但对这些东西的实现却知之甚少——这又如何突破自己、获得更好的提高呢？

鉴于此种现状，笔者在“疯狂 Java 实训营”的训练中，除了采用大量实际项目来驱动教学之外，往往会花很多时间、精力来培养学员的基本功。比如讲授 Spring 框架时，如果只关注如何编写 Bean 类、配置 Bean，那么一天的时间就足够了。而笔者往往会深入 Spring 框架的底层实现，带领学生从工厂模式、策略模式、门面模式、代理模式、命令模式的角度来深度分析 Spring 框架实现，然后进行对比，总结 Spring 框架的优势到底在哪里？不使用 Spring 框架是否有替换解决方案？从而感受设计模式对实际开发的帮助。

上面这些内容，看似“高深”，但其实质依然离不开 Java 编程的基本功。也可以这样说：一旦读者真正把基本功打扎实了，将可以看得更高、更透彻。

在这样的背景下，偶然之间我获得了这样一个想法：把这些容易被 Java 程序员所忽视的内容整理成一本书，也许可以帮助这些需要突破基本功的准程序员、初级程序员。本书的写法对笔者来说，是一种新的尝试：笔者以前所写之书，通常都会围绕某方面知识“画一个圆”——把这个方面的相关知识，全面、详细地向读者介绍出来。本书并没有采用这种写法，本书的每课围绕一个相对独立的专题进行讲解，形式上有点类似于“针对性地补课”。当读者感觉自己在某些方面的基本功不够扎实时，可以直接选择相应的内容进行学习，完全无须从头到尾地按顺序阅读。

本书创作过程

本书创作得有些困难，当初刚刚获得这个想法之时，我也曾感到有些力不从心，程序员的基

本功看似非常简单，但覆盖范围却非常广泛，因此似乎很难形成一个完整、系统的知识体系。

但随着写作过程的不断推进，此时笔者心中才渐渐明朗起来：该书即是一本无所不包的解惑之书，它即是将 Java 学习、实际开发中最容易被大家忽视、又非常重要的内容收集起来，便足够了。

期间，ACS（天津）公司邀请笔者为其公司员工进行了一次技术内训，内容主要围绕 Java 程序的内存管理问题。内训结束时该员工反应很好，这给笔者带来了很大的触动。Java 程序的内存管理既是 Java 程序员容易忽视的地方，但又是 Java 编程的重点。后来，笔者即将此次内训的知识进行整理，并最终形成了本书第 1 部分的内容。

本书第 3 部分所介绍的常见数据结构、排序算法的 Java 实现，则是笔者一直想介绍的内容。在笔者的“疯狂 Java 实训营”内，常见数据结构、排序算法是必须掌握的——也许你初涉编程时感受不到这些经典算法的用途，因为你可以直接利用别人的实现，但如果你希望突破自己，上升到另外一个高度时，你就不可避免地需要自己开发类库，而不是总使用别人的类库，那这些经典算法的作用就显现出来了。

本书第 2 部分和第 4 部分则主要来自于参加“疯狂 Java 实训营”的学生，正如每个动手编程的初学者，他们都曾经遭遇着各种各样的陷阱，因此笔者总是提醒他们应该将这些陷阱收集起来，以免再次陷进去。本书第 2 部分收集了 Java 编程中各种常见的陷阱，第 4 部分的内容则解决了他们进入实际开发之前的诸多困扰，包括程序开发的基本方法，有效进行程序调试的方法，如何看待、使用 IDE 工具，软件测试等相关内容。

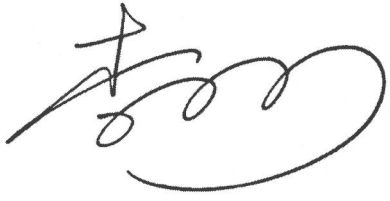
衷心感谢

本书得以成书，离不开人民邮电出版社诸位编辑的热心帮助，他们用心地整理出本书的知识结构，并为本书制定了初步大纲，这些工作给笔者提供很大的帮助。

感谢造物主和妻子在本书交稿之际赐给笔者一个非常可爱的宝宝，他的出现稍稍延迟了本书的部分工作，但却给笔者的生活添加了无限的乐趣。

本书写给谁看

如果你是一名 Java 语言的初学者，并已经学会了 Java 基本的语法，但却经常在动手编程时感到困难重重，或者你已经是一个 Java 程序员了，但在实际开发中却经常感觉力不从心，那么本书将非常适合你。本书会帮助你找出自己的技术短板，迅速突破 Java 编程的基本功，使你成为一名真正的 Java 达人。



2009-12-20

疯狂Java

目 录

第 1 课

数组与内存控制

一家国际著名软件企业的面试。

“你的简历我看了，你会使用 Java？”面试官面无表情地问道。

“是的。”参加面试的人，成竹在胸地回答。

“那好，你给我叙述一下，在 Java 中，声明一个数组的过程中，是如何分配内存的。”

“……”

“那，Java 数组的初始化一共哪有几种方式，你能说一说吗？”

“……”

“那你知道基本类型数组和应用类型数组之间，在初始化时的内存分配机制有什么区别吗？”

“……”

过了一会儿之后，房间的门打开了，可怜的面试者，狼狈地走了出来。

离开的时候，他喃喃自语：“原来，小小的数组，也有这么多的知识。”

本课要点：

- **Java 数组的基本语法；**
- **Java 数组的静态特性；**
- **Java 数组的内存分配机制；**
- **初始化 Java 数组的两种方式；**
- **初始化基本类型数组的内存分配；**
- **初始化应用类型数组的内存分配；**
- **数组引用变量和数组对象；**
- **何时是数组引用变量，何时是数组对象；**
- **数组元素等同于变量；**
- **多维数组的内存分配。**

Java 数组并不是什么很难的知识，如果单从用法的角度来看，数组的用法并不难，只是很多程序员虽然一直使用 Java 数组，但他们往往对 Java 数组的内存分配把握并不准确。本章正是为了弥补程序员的这部分基本功而做的深入探讨。

本课将会深入探讨 Java 数组的静态特征。使用 Java 数组之前必须先对数组对象进行初始化。当数组的所有元素都被分配了合适的内存空间，并指定了初始值时，数组初始化完成。程序以后将不能重新改变数组对象在内存中的位置和大小。从用法角度来看，数组元素相当于普通变量，程序既可把数组元素的值赋给普通变量，也可把普通变量的值赋给数组元素。

本课还将深入分析多维数组的实质，深入讲解多维数组和一维数组之间的关联，并通过程序示范如何将一维数组扩展成多维数组。

1.1 数组初始化

数组是大多数编程语言都提供一种的复合结构，如果程序需要多个类型相同的变量时，就可以考虑定义一个数组。Java 语言的数组变量是引用类型的变量，因此具有 Java 独有的特性。

1.1.1 Java 数组是静态的

Java 语言是典型的静态语言，因此 Java 的数组是静态的，即当数组被初始化之后，该数组的长度是不可变的。Java 程序中的数组必须经初始化才可使用。所谓初始化，就是为数组对象的元素分配内存空间，并为每个数组元素指定初始值。

数组的初始化有以下两种方式。

- 静态初始化：初始化时由程序员显式指定每个数组元素的初始值，由系统决定数组长度。
- 动态初始化：初始化时程序员只指定数组长度，由系统为数组元素分配初始值。

不管采用哪种方式初始化 Java 数组，一旦初始化完成，该数组的长度就不可改变，Java 语言允许通过数组的 length 属性来访问数组的长度。示例如下。

程序清单：codes\01\1.1\ArrayTest.java

```
public class ArrayTest
{
        public static void main(String[] args)
    {
        //采用静态初始化方式初始化第 1 个数组
        String[] books = new String[]
        {
            "疯狂 Java 讲义",
            "轻量级 Java EE 企业应用实战",
            "疯狂 Ajax 讲义",
            "疯狂 XML 讲义"
        };
        //采用静态初始化的简化形式初始化第 2 个数组
        String[] names =
        {
            "孙悟空",
            "猪八戒",
            "白骨精"
        };
```