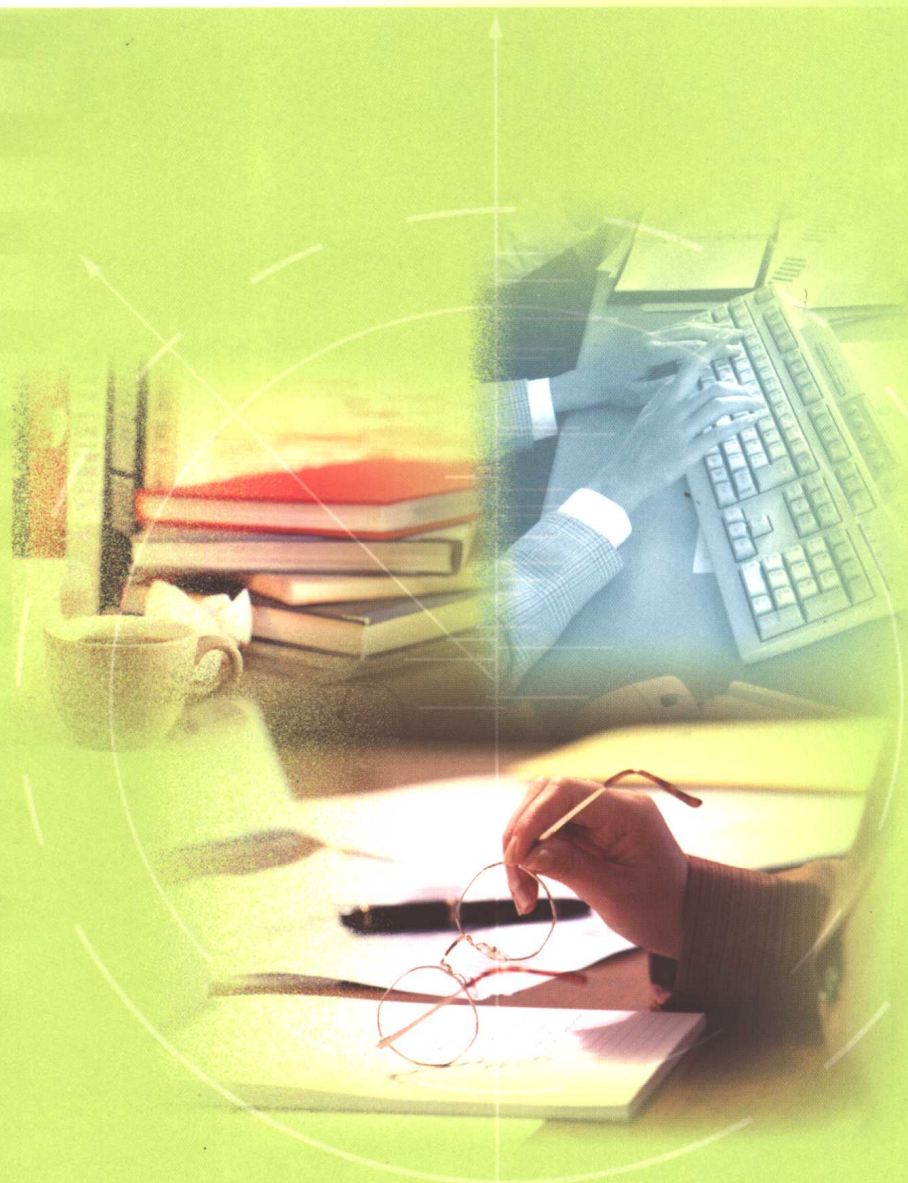


● 编程之路系列教材

Java

程序设计导学

张洪斌 编著



北京科海电子出版社
<http://www.KHP.com.cn>



► 编程之路系列教材

Java 程序设计导学

张洪斌 编著



[北京科海电子出版社]
<http://www.KHP.com.cn>



内 容 提 要

本书作者采用独特的先进教育思想,结合朴实的叙述风格,集中全面地讲述了 Java 程序设计语言的基本概念与编程思想,具体阐述了一般程序设计语言的基础知识、面向对象编程技术,以及 Java 语言的强大功能。

书中充分考虑初学者经常遇到的疑难困惑,强调学习过程的编程实践与设计思路,每章均配有大量习题,所有习题都给出参考答案,并辅以书后覆盖面广的综合实习题,使您能学以致用,迅速提升编程能力。

本书适用于没有或缺乏程序设计经验的初学者,可供大专院校计算机或非计算机专业学生使用,也可作为 Java 语言的自学教材。

盘 名: Java 程序设计导学
作 者: 张洪斌
责任编辑: 何武
排 版: 卞雨桂
光盘制作: 明卫军
咨询电话: (010) 82896445-8407



出 品: 北京科海电子出版社
印 刷 者: 北京市朝阳区科普印刷厂
开 本: 787×1092 1/16 印张: 22.25 字数: 541 千字
版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷
印 数: 0001~5000
盘 号: ISBN 7-900107-50-9
定 价: 32.00 元 (1CD)

北京科海电子出版社
http://www.KHP.com.cn

《编程之路系列教材》序

随着计算机技术在我国各个领域的广泛应用,以及计算机软件平台的不断提升,计算机编程不再仅限于计算机专业人员,越来越多的计算机爱好者通过专项培训或自学,已成为计算机编程的“行家里手”;特别是我国已经入世,国内IT企业及非IT企业对人才的需求将超过40万,其中一半是软件技术人才,这是传统学历教育远远满足不了的,它需要通过各种途径为这一行业的发展提供大批的IT技术人材。本丛书就是为此目的而编写的,它以计算机编程为核心,涵盖了从基础到专业应用的一些重要课目。本套丛书包括:

1. 《C++语言程序设计导学》
2. 《数据结构导学》
3. 《Visual Basic 6 程序设计导学》
4. 《PowerBuilder 7.0 程序设计导学》
5. 《Visual C++程序设计导学》
6. 《Delphi 6 程序设计导学》
7. 《C++ Builder 程序设计导学》
8. 《Java 程序设计导学》

本丛书具有以下特点:

○ 力求通俗易懂

本丛书不仅面向计算机专业人员,更立足于计算机编程爱好者,因此,在文字叙述和内容的安排上尽量通俗易懂,力求讲出问题的来龙去脉,把编程的“过程”讲透。

○ 强化编程的概念

作为一个编程人员,必须深入领会编程的实质,这样才能做到举一反三,融会贯通,达到编制自己的应用程序的目的。所以本丛书不同于一般的软件系统使用手册,而是针对读者学习中可能遇到的问题诠释了编程思路和编程技巧,便于读者提升编程能力。

○ 编程思想与开发工具运用相结合

学习编程,不仅要在编程思想上有所突破,还应学会如何更好地运用编程的开发工具,只有两者的结合才是真正的理论联系实际,事半功倍的学习方法。本丛书精选了目前流行的软件开发工具(如Visual Basic, PowerBuilder, Visual C++, Delphi, C++ Builder),这些工具中提供了许多编程技巧和功能,

对编程者具有实际的应用价值。

○ 内容表述与习题、实习训练并重

本丛书提供了大量的习题和实习题，而且给出了这些习题和实习题的参考答案，便于读者练习、仿效，达到快速掌握编程方法和技巧的目的。

由于时间仓促，本书疏漏之处在所难免。但我们相信本丛书一定会成为计算机编程爱好者的良师益友。

读者选购指南

选购一本电脑书从来就不是件轻松的事，因为同类书实在是琳琅满目，难以取舍。不过，以作者所见，本书可能是书店里惟一一本把Java作为入门程序设计语言的著作，或者说，如果你想学习程序设计语言，并且想学到最新的Java语言技术而不是“老式”的Qbasic、Pascal等，本书很可能是你惟一的选择。当然，如果你学过程序设计语言，但碰巧感觉不佳，那么本书也可以作为你的选择之一。如果你已经有相当好的程序设计基础，本书可能不是你的最佳选择。

其实，上面所述就是读者选购电脑书时要注意的第一项：了解书的起点或者读者对象，不要买了对自己来说太浅或者太深的书。

选购电脑书的第二项是了解书中的内容。本书的重点为第0章到第5章。

第0章讲述程序设计的一些基本背景知识，由于本书的起点是不要求读者具备任何程序设计基础，所以就用“俗话”来讲述。这些内容对所有的程序设计语言都适用。

第1章讲述编写和运行Java程序的完整过程。这一过程对绝大多数程序设计语言都是相同或者相似的。

第2章和第3章讲述面向过程程序设计和Java基础知识，与前面提到的老式程序设计语言的内容大体相似。

第4章和第5章讲述面向对象技术。目前主流程序设计语言都是以面向对象为基础的，而老式程序设计语言不具备此一功能。如果你不懂面向对象技术，几乎可以断定你无法以程序员为职业。反过来说，掌握了面向对象技术，尽管本书讲解的是Java语言，也能让你很容易地学会其他程序设计语言，例如C++和C#。

尽管本书讲述的是Java，但前5章的内容从原理和形式两方面，都与其他主流程序设计语言相同或者近似。

从第6章开始，本书讲述的内容从形式上可能与其他语言有较大的差别，但原理仍然相同或者类似。例如第7章讲述的数组运算、数组、字符串等，是其他任何程序设计语言都具备的功能。第8章的异常和第9章的输入输出则与C++至少在原理上相同。第10章的绘图也是任何程序设计语言必备的功能。

上面讲述的是本书的读者对象和内容概貌，接着作者要解释的是本书如何组织和讲解内容，即本书的写作原则或者指导思想。这就是选购电脑书的第三项：了解作者是怎样讲解书中知识的。

任何一本书的作者，在写书时都有指导思想，无论他是否觉察到或者这个指导思想是否恰当。本书除了具备科技著作的起码要求，即技术讲解正确、内容全面完整外，全书的写作思想和具体的写作方法尤其具有与众不同的特色，使读者不仅看得懂、看得轻松，还看得愉快。本书的指导思想是Know Unknown Through Known（通过已知知未知），这是法国教师Piggysong的名言和教育思想，意思是人类了解未知的最好方式是通过已知。

本书的具体写作策略是站在读者的角度，按照普通人的学习习惯而不是知识本身的固有体系布局谋篇，全书章节都是按照“基本—常用—深入”来安排内容结构，突破了传统电脑书不分主次和轻重缓急的字典式写作惯例。因为每当提到一个名词、概念或者其他知识点，作者要问自己三个问题：“读者可能学过吗？前面讲过吗？应该在这里讲吗？”，尽量保证每一句话、每一个段落的知识只涉及前面的内容而不涉及后面的内容，使读者从第一页看起，基本没有看不懂的地方，这就是不用未知解释另外一个未知。万一要提及后面才能详细讲解的知识，就使用这样的说法：“先照葫芦画瓢，后面还要详细讲解”，以避免读者产生没有讲清楚的错觉，仅此一点，就可保证读者看得不累。

本书对知识点，不管是一个专业名词还是一章一节的内容，大多评价其实用效果，以便读者知道如何在实践中运用知识，使读者为应用而学习，而不是为知识而学习。

本书对知识点基本采用先举例、从已知知识切入或者先比喻的方式引入，这些都是读者已知或者很容易“已知”的，作者以为，这是最好的知识点引入方式，类似的例子也不胜枚举。

本书在语言方面，尽量地运用比喻来讲解知识点，这在科技著作中极为罕见，看过本书内容的读者都可以很容易地在书中找到证据。

除了运用比喻外，本书的语言叙述尽量平白、简洁、生动又不乏幽默，因为作者厌恶以“专业”术语唬人或者自我炫耀，反喜欢“化神奇为腐朽”，把艰深干涩的电脑知识通俗化、生动化、趣味化，使读者在阅读时，不仅读得懂、读得轻松，还读得愉快，甚至会发出“Java不过如此！”的感叹。

上面所述仅是本书的几个突出特点而已，其他特色就不必一一列出了，读者在阅读时，自然能够体会到。作者相信，读者只要在书店里站着读几分钟，就能够发现本书的几个特色，如果不是全部特色的话。顺便说一句，本书的写作思想和写作方法与作者已经出版的其他著作完全相同，这些书得到了读者的高度评价，读者不妨先看看，再决定是否选择本书。

本书英文书名为“Bestbook”，作者已经出版的著作也是用同一名字。作者在刚写作时，就希望能够创作出最好的电脑书奉献给读者，因此以“Bestbook”作为英文书名，鞭策自己精心构思，认真写作，不辱此名。

读者可能会注意到作者没有用“通俗易懂”、“循序渐进”、“快速入门”、“轻松上手”等来形容本书，因为所有的电脑书几乎都这样自称，而尽管本书当之无愧，却只用一个简单的“Bestbook”。为了支持作者的这句“广告词”，作者还拿出诸多可判定的证据来证明本书为什么是Bestbook。

本书的全部程序和有关软件在本书所附的光盘上。如果读者对本书有什么批评和建议，既可以给作者发送电子邮件，也可以在作者的个人网站上进行交流。

张洪斌

bestbooks@netease.com

<http://www.BestPapers.org>

自序兼谈本书应用的教育思想

Know Unknown Through Known是19世纪法国教师Piggysong的名言，意思是人类了解未知的最好方式是通过已知，即The best way to know what is unknown is through what is known。我是在上大学时，从图书馆借的一本英文名人传中知道这句话的，记得那是一本发黄的书，好像是解放前的馆藏。我能把这句话记到现在不是因为我当时明白了其中的深邃思想，而是这句话表现出的英文的优美特点：只有4个单词，并且其中的3个都带有know。我曾竭力地想把它翻译成旗鼓相当的中文，可是至今都没有完全成功，我所做的最好译文是“通过已知知未知”，意思表现出来了，而且与英文一样也有3个“知”，但却用了7个汉字（一般认为汉语译文比英文原文篇幅要少）。

那本英文传记介绍说，作为教师，Piggysong非常善于教学，他的学生总是能以最快的速度掌握知识，他用的就是Through Known的原则。例如，他大量地使用比喻，把生活中学生已经了解的事物与要学习的专业知识比较。在做试验时不是像现在的教师那样先讲原理，而是先做试验，再讲原理。在教学顺序上，他按照内容的深浅程度而不是知识固有的体系安排，虽然被同行批评教授的知识显得缺少体系和章法，但是他的学生学得快学得好却是不争的事实，不然怎么上了名人传，让我这个几百年以后的学生也知道他的大名呢？

当时我在看传记时，对他的思想并没有留下太多的感触，因为自己当初并非教育工作者，只是刚上大学的学生而已，况且他的教学方法就是由简到繁、由浅入深、由具体到抽象之类，这些大家不都知道吗？很plain（平白）吗！我学得是计算机专业，得听课还得看书，自然地感到对于同样的内容，有些老师和有些书就比别的老师和教材讲得好、讲得更容易懂。现在已经工作了，主要是通过看书来提升自己，逐渐感到现在的电脑书越来越难看了，一开始认为也许是电脑技术进步的“恶果”，可是后来逐渐发现并非完全如此。例如很多书的第一章都是告诉你这个软件有什么强大的功能、新版本比老版本有什么改进，其中会涉及一大堆后面的章节才能讲解的术语，看得人头皮发麻。试问，还没有学它如何明白它的强大之处？没有用过旧版本何以明白新版本高明的地方？

笔者至今还记得自己刚学习“文件”这个概念时的艰难历程。笔者刚学电脑时，书中告诉我：文件用于记录电脑信息(大意)。我不知道文件是什么，当然也不知道信息是什么，用我不理解的名词来解释另一个不理解的名词岂不是雪上加霜？接着开始讲解文件的命名法，我不认识老虎却要认老虎这二字干吗？然后书中又告诉我，文件有可执行文件、文本文件等类型，既然我不知道什么是老虎又怎么知道什么是东北虎、华南虎呢？书中讲完了全部的文件概念才开始学习DOS命令，这时才明白什么是文件，可是都已经“学”了很多页了。如果一边学习DOS命令一边讲解文件的概念岂不是轻松得多，例如要讲解可执行文件，只要运行一个游戏文件就可以了。作者也注意到目前很多Windows的书也是先讲文件的概念，再讲文件操作的。

我在专业上学习的同时顺便也在研究电脑书的写作方法，慢慢地发现电脑书写得差的根本原因就是：应该在后面讲解的知识在讲解前出现，就像笔者在前面举的文件例子一样。于是我就想起了Piggysong和他的名言，尽管我当时认为多么plain、多么simple，可是不是有很多教育工作者包括他们写的书都没有运用吗？作者已经出版的著作全部采用Piggysong的思想创作，全部以bestbook作为英文书名（即The Bestbook For XXX），结果也是广受好评，这至少可以证明Piggysong的思想good（不错），如果不是best（最好）的话。所以作者依然像以前一样“大胆”，还把这本书叫bestbook，同时作者也希望Piggysong的教育思想对其他电脑书作者和教育工作者能有参考价值，促进书店里出现更多看得懂、看得轻松、看得愉快的好书。

Piggysong不是个太有名的人，甚至不能作为教育家流传后世，也许是因为他不太“善于”总结自己的思想，如果把自己的思想叫做“XXX思想”或者“XXX主义”（例如目前最流行的“建构主义”教育理论），名字最好如雷贯耳又莫名其妙并且拌有枯燥加令人困惑的长篇大论，这样后人才会不知疲倦、兴致勃勃地研究和探讨。他的Know Unknown Through Known太白了、太容易理解了，根本不需要什么人做进一步的发挥和解释。不过，笔者认为，这正是一个教育工作者的追求：授业的目的不正是解惑，并且解得越快越好吗？

张洪斌

bestbooks@netease.com

<http://www.BestPapers.org>

目 录

第0章 俗话程序设计	1
0.1 软件、程序和程序员.....	1
0.2 程序设计语言和生活中的语言.....	1
0.3 程序设计语言概况.....	2
0.4 程序员的工作和回报.....	2
0.4.1 程序员的工作.....	2
0.4.2 不同程序设计语言的“回报率”.....	3
0.5 Java程序设计语言.....	3
0.5.1 Java作为入门程序设计语言是否好学.....	3
0.5.2 Java的故事.....	3
第1章 编程全过程概览	5
1.1 编写程序的完整过程.....	5
1.2 JDK的安装和设置.....	5
1.3 使用文本编辑器输入和编辑源程序.....	6
1.4 编译源程序.....	7
1.5 运行程序.....	11
1.6 Java程序框架.....	13
1.7 JDK中的演示程序欣赏.....	14
1.8 编写源程序时常见的问题.....	17
1.9 编译和运行程序时需要注意的问题.....	18
1.10 本章小结.....	19
第1章习题.....	19
第2章 Java基本语法	21
2.1 注释.....	21
2.1.1 单行注释.....	21
2.1.2 多行注释.....	21
2.1.3 文档注释.....	22
2.2 标识符.....	22
2.3 语句、空格和语句块.....	27
2.4 基本数据类型.....	28
2.4.1 整型和整数变量的声明.....	28

2.4.2 浮点型和浮点变量的声明	31
2.4.3 字符型和字符变量的声明	32
2.4.4 布尔型和布尔变量的声明	33
2.4.5 常量	34
2.5 运算符、表达式和返回值	35
2.6 数学运算	35
2.7 赋值运算	39
2.8 类型转换	41
2.9 自增与自减运算	42
2.10 比较运算	43
2.11 逻辑运算	44
2.12 位运算和移位运算	46
2.13 本章小结	47
第2章习题	47
第3章 程序流程设计和控制	51
3.1 算法、伪代码和流程图	51
3.2 if 语句	53
3.3 switch语句	59
3.4 while语句	61
3.5 do while语句	65
3.6 for语句	66
3.7 循环的综合示例	67
3.8 多重循环	68
3.9 break、continue和标号语句在循环中的作用	70
3.10 本章小结	75
第3章习题	75
第4章 面向对象技术的基础知识	80
4.1 类的定义和使用	80
4.1.1 最简单的类	81
4.1.2 类的数据	81
4.1.3 类的函数	81
4.1.4 完整的类定义示例	83
4.1.5 创建类的对象和运行程序	83
4.1.6 对象之间的关系	85
4.1.7 程序代码的书写格式	87
4.1.8 小结	87
4.2 有返回值的方法	87

4.3 改变类的结构	91
4.4 类中成员间的访问机制.....	94
4.4.1 在一个类中的访问机制.....	94
4.4.2 一个类访问另外一个类.....	97
4.5 类中的变量与方法中的变量.....	99
4.5.1 全局变量和局部变量的访问规则.....	99
4.5.2 全局变量和局部变量的初始化.....	100
4.5.3 变量的作用域.....	102
4.6 类的构造方法和重载.....	105
4.6.1 类的构造方法(Constructor).....	105
4.6.2 重载(overload)方法与默认构造方法.....	107
4.7 类的继承与方法重置.....	110
4.7.1 类的继承和重置方法.....	110
4.7.2 在一个文件中继承类.....	111
4.7.3 为每个类添加main方法.....	113
4.7.4 类与其超类的构造方法的调用.....	115
4.7.5 不用继承使用另外一个类的功能.....	116
4.7.6 滥用重置.....	118
4.7.7 重置与重载.....	119
4.8 static和final关键字.....	121
4.8.1 static关键字.....	121
4.8.2 final关键字.....	123
4.9 this和super关键字.....	123
4.10 本章小结.....	126
第4章习题.....	126
第5章 面向对象技术的高级主题.....	136
5.1 超类与继承类的关系.....	136
5.2 通过classpath访问另一个目录中的类.....	137
5.3 通过软件包访问另一个目录中的类.....	138
5.3.1 建立软件包.....	138
5.3.2 使用软件包.....	139
5.4 存取(访问)控制.....	141
5.5 抽象类和抽象方法.....	144
5.6 接口interface与多重继承.....	146
5.6.1 接口基础.....	146
5.6.2 多重继承.....	148
5.6.3 接口的类型转换(upcasting).....	150

5.7 内部类基础	151
5.8 Java的存储结构	153
5.9 本章小结	153
第5章习题.....	154
第6章 通过Java文档了解类的使用.....	159
6.1 Java文档的使用介绍.....	159
6.1.1 Overview页面	160
6.1.2 Package页面	161
6.1.3 Class页面	162
6.1.4 Use页面	166
6.1.5 Tree页面	166
6.1.6 Deprecated API页面	167
6.1.7 Index页面.....	167
6.2 怎样使用软件包.....	168
6.3 使用JavaDoc生成自己的程序文档	169
6.4 本章小结	173
第6章习题.....	173
第7章 几个基本的Java类.....	175
7.1 Math类.....	175
7.2 数组	178
7.2.1 数组基础.....	179
7.2.2 数组的建立方式.....	181
7.2.3 数组的长度.....	183
7.2.4 对象数组.....	183
7.2.5 多维数组.....	185
7.2.6 与数组有关的运行错误	188
7.3 String类.....	189
7.3.1 字符串的声明和赋初值.....	190
7.3.2 String类中的方法	191
7.4 main方法	195
7.5 包裹类	196
7.6 本章小结	199
第7章习题.....	199
第8章 程序的异常处理.....	201
8.1 异常入门	201
8.2 传统的错误处理方法.....	203

8.3 基本的异常处理.....	205
8.3.1 try和catch.....	205
8.3.2 Exception类及其子类.....	207
8.3.3 得到异常类的信息.....	209
8.4 必须抓住异常的情况.....	210
8.5 本章小结.....	213
第8章习题.....	213
第9章 输入输出.....	215
9.1 输入与输出.....	215
9.2 流在功能上的分类.....	215
9.2.1 Node stream.....	216
9.2.2 Processing stream.....	216
9.3 InputStream类.....	217
9.4 处理流的使用.....	219
9.5 OutputStream类.....	220
9.6 Reader和Writer类.....	221
9.7 System类提供的输入和输出功能.....	222
9.8 本章小结.....	224
第9章习题.....	224
第10章 在Windows中绘图.....	226
10.1 建立Windows窗口.....	226
10.1.1 建立Windows窗口的方法之一.....	226
10.1.2 建立Windows窗口的方法之二.....	228
10.1.3 建立Windows窗口的方法之三.....	228
10.1.4 关闭Windows窗口.....	229
10.1.5 设置Windows窗口在屏幕上的位置.....	230
10.2 图形绘制入门.....	230
10.3 显示文字.....	232
10.4 选择颜色.....	233
10.5 设置字体.....	234
10.6 画直线.....	236
10.7 绘制平面矩形.....	237
10.8 绘制立体矩形.....	238
10.9 画圆.....	239
10.10 绘制圆弧.....	240
10.11 绘制圆角矩形.....	241
10.12 绘制多边形和多折线.....	241

10.13 显示图像	243
10.14 本章小结	245
第10章习题	245
附录A Java运行环境设置和DOS基本操作	249
A.1 autoexec.bat的设置方法	249
A.2 DOS基本操作方法	250
A.2.1 进入DOS窗口	250
A.2.2 基本DOS操作命令	251
A.2.3 窗口的设置	255
附录B Windows程序设计入门	257
B.1 Windows程序中的几个基本概念	257
B.2 容器、构件和布局	258
B.2.1 在窗口中添加构件	258
B.2.2 使用布局	260
B.2.3 文本框和文本区	264
B.2.4 JPanel类(面板)	266
B.2.5 没有交互功能的聊天室程序	268
B.3 Windows事件处理	270
B.3.1 事件处理的运行机制	270
B.3.2 具有交互功能的聊天室程序	272
B.3.3 关闭窗口事件	274
附录C 各章习题参考答案	275
第1章习题答案	275
第2章习题答案	275
第3章习题答案	281
第4章习题答案	286
第5章习题答案	288
第6章习题答案	289
第7章习题答案	289
第8章习题答案	298
第9章习题答案	300
第10章习题答案	309
附录D 综合实习题	311
D.1 勾股定理(Pythagorean三角形)	311
D.2 加密与解密	314
D.3 猜数游戏	317

D.4 斐波纳契数列 (Fibonacci Series)	319
D.5 随机运动图形.....	323
D.6 条状图	326
D.7 用于反恐的敏感词汇分析	330
附录E Java中部分中文术语之我见	336

第0章 俗话程序设计

由于本书的阅读对象是没有学过程序设计的读者，所以在本章，作者用通俗的语言讲解程序设计的一些基本概念，并且尽量不涉及过多的技术名词。

0.1 软件、程序和程序员

大家都使用过电脑，多数情况下我们使用现成的软件（Software）或者程序（Program）。在某些场合，可以认为软件和程序是一个概念，例如常见的Windows、Office和游戏等。这些程序当然不是天上掉下来的，它们是人做出来的。这些人一般被称为“程序员”（Programmer），即“做程序的人”。程序员必须学习怎样做程序，这是高科技的活。做程序的工具是程序设计语言（Programming Language），这是计算机才懂的语言，不是我们人类使用的语言。程序员主要学习这些语言，用这些语言告诉计算机应该做什么。

0.2 程序设计语言和生活中的语言

生活中的语言有规则，要按照语法规则把字、词组成句子、段落、文章，否则别人听不懂或者看不懂。程序设计语言在很多方面也和生活中的语言相似，它也有语法等规则，只是更强，一点错误都不能有。所谓编程就是按照程序设计语言的规则来“说话”，命令计算机做某件事情。

比如做一道 $1+2$ 的数学题，在生活中我们用汉语这样说：

如果 x 等于1， y 等于2， z 是 x 和 y 之和，那么 z 等于多少。

如果想让计算机解决这个问题，就要按照程序设计语言的规则做，下面是一个例子：

```
x=1;  
y=2;  
z=x+y;
```

这是Java语言的例子，可以认为它由3句话组成，这些句子称为“语句”（Statement），以分号结尾。

计算机语言的要求比生活中的语言严格，例如：

1. 上面的每一句话后面都有分号，少了分号就是错误。
2. 语句的顺序不能随便颠倒。