

程序员的 数学思维修炼 (趣味解读)

周颖 等编著

一本专门为程序员而写的数学书，训练数学思维，增强职场竞争力

书中没有罗列晦涩难懂的数学公式和推导，而是代之以生动有趣的数学实例
读者不必精通高深的数学知识，只需要具备四则运算和基本的逻辑思维即可阅读
通俗易懂，风格活泼，趣谈110个实例，并给出了33个具体的程序代码实现



清华大学出版社

014037245

TP311.1
159

程序员的 数学思维修炼 (趣味解读)

周颖 等编著



TP311.1
159

清华大学出版社
北京



北航

C1725427

内 容 简 介

本书是一本专门为程序员而写的数学书，介绍了程序设计中常用的数学知识。本书门槛不高，不需要读者精通很多高深的数学知识，只需要读者具备基本的四则运算、乘方等数学基础知识和日常生活中的基本逻辑判断能力即可。本书拒绝枯燥乏味的讲解，而是代之以轻松活泼的风格。书中列举了大量读者都很熟悉，而且非常有趣的数学实例，并结合程序设计的思维和算法加以剖析，可以训练读者的数学思维能力和程序设计能力，进而拓宽读者的视野，增强职场竞争力。

本书共 11 章，分别介绍了数据的表示、神奇的素数、递归、排列组合、用余数进行数据分组、概率、复利、数理逻辑、推理、几何图形构造、统筹规划等程序设计中常用的数学知识，从而引导读者深入理解编程中的数学方法和思路。

本书适合广大程序设计人员及数学爱好者阅读，尤其适合有一定程序设计经验，但还需要进一步加深对程序设计理解的人员阅读。本书对 IT 求职人员、信息学竞赛和大学生程序设计竞赛等参赛学员也有很好的参考价值。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目 (CIP) 数据

程序员的数学思维修炼 (趣味解读) / 周颖等编著. —北京: 清华大学出版社, 2014
ISBN 978-7-302-35060-6

I. ①程… II. ①周… III. ①电子计算机—数学基础 IV. ①TP301.6

中国版本图书馆 CIP 数据核字 (2014) 第 006758 号

责任编辑: 夏兆彦

封面设计: 欧振旭

责任校对: 胡伟民

责任印制: 刘海龙

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 19.75 字 数: 496 千字

版 次: 2014 年 4 月第 1 版 印 次: 2014 年 4 月第 1 次印刷

印 数: 1~4000

定 价: 45.00 元

产品编号: 056332-01

前 言

数学在人类文明的发展过程中起着非常重要的作用。数学推动了重大的科学技术进步。从远古的“结绳记事”，到现代计算机技术的快速发展，都与数学这门学科的发展密不可分。

无论是日常生活中简单的商品交易计算，还是神舟飞船设计中复杂的计算，都离不开数学。生活即数学。没有二进制，就不会有现在的计算机；没有几何学，就没有现在的高楼大厦……。

对于程序员来说更需要知道：数学是计算机科学的基础。在我国，绝大部分大学的计算机科学系都是从数学系中分出来的。由此也可以看出，计算机科学与数学的关系非常紧密。

数学是一门化繁为简的学科。通过数学，可以对现实生活中的很多不同事物进行高度抽象，从而能找出不同事物的共性。不过，由于数学的这种高度抽象，又使数学变得很难学。特别是一些复杂的公式推导，看起来就头痛。

本书面向程序员介绍了程序设计中常用的数学基础知识。通过阅读本书，可以训练程序员的数学思维能力和程序设计能力，进而拓宽视野，增强职场竞争力。

本书特点

- ❑ **简单易懂** 用通俗易懂的语言讲解知识点，尽量避免复杂的公式推导过程，让读者能够轻松阅读并掌握相关的数学知识。
- ❑ **门槛很低** 阅读本书的读者不需要精通很多高深的数学知识，只需要具备基本的四则运算、乘方等数学基础知识和日常生活中的基本逻辑判断能力即可。
- ❑ **生动有趣** 本书拒绝枯燥乏味的讲解，而是代之以轻松活泼的风格，讲解时列举了大量我们都很熟悉，而且非常有趣的数学实例。
- ❑ **内容丰富** 本书从最简单的数据的表示开始，对素数、递归、排列组合、逻辑推理、几何构造、统筹规划等方面都会逐一介绍，涵盖了程序员需要掌握的数学知识。
- ❑ **图文并茂** 讲解每个知识点和实例时，都给出了简单易懂的图示和必要分析，让读者理解起来清晰明了，没有任何障碍，也让读者感觉到学习数学并不困难。

本书内容概述

第1章通过一则童话故事导入了数据大小的知识，然后逐步介绍了十进制、二进制、八进制、十六进制以及其他常用进制的知识，还介绍了不同进制的转换方法。

第2章从素数的判断开始，逐步介绍了与素数相关的数学知识，包括孪生素数、梅森素数、哥德巴赫猜想、RSA 的应用等内容。

第3章介绍递归这种自己调用自己的方法，通过阶乘、汉诺塔、斐波那契数列等经典实例，练习从复杂事物中发现递归结构的方法。

第4章的主题是排列组合，从乘法原理、加法原理入手，介绍了排列与组合的概念和关系，并研究了计算机中的字符编码、密码长度等相关内容。

第5章讨论余数。主要介绍使用余数对数据进行分组，如日历、一些小魔术都是通过余数分组的规则进行的；本章还讨论了计算机中的奇偶校验及两个有趣的问题（座位安排和智叟分牛）。

第6章介绍概率的相关知识，首先从两个常见的事例导入概率的概念，接着从军事故事、赌场游戏、中奖概率等方面介绍了概率的实际应用。

第7章学习翻番的知识。首先介绍翻番和翻倍的概念、计算方式。接着进一步通过复利的威力、对折纸张、舍罕王的赏赐等实例，展示了翻番这个令数据快速增长的数学概念。最后还介绍翻番的逆运算——折半的应用。

第8章学习数理逻辑的相关知识，介绍了逻辑、命题逻辑、布尔逻辑、逻辑的重叠与遗漏等概念，最后介绍了通过卡诺图化简逻辑表达式的方法。

第9章则在第8章的基础上进一步讨论了逻辑推理，包括演绎推理中的三段论、选言推理、假言推理、关系推理，以及归纳推理中的完全归纳推理和不完全归纳推理。

第10章介绍了几何图形构造的基础知识，从花盆摆放、残缺棋盘、丢失的线条等有趣实例，初步了解几何图形构造，最后还介绍了几何图形的分割与拼接。

第11章讨论统筹规划相关知识，首先从田忌赛马这个古老故事中看出统筹规划的重要性，然后通过生活中的两个简单例子认识统筹规划，最后还讨论了“背包问题”及其程序设计方法。

本书读者对象

本书可适用以下各类人员阅读：

- 计算机专业的学生；
- 数学专业的学生；
- 程序设计人员；
- 数学爱好者；

□ 编程爱好者。

本书作者

本书由周颖主笔编写。其他参与编写的人员有韩先锋、何艳芬、李荣亮、刘德环、孙姗姗、王晓燕、杨平、杨艳艳、袁玉健、张锐、张翔、陈明、邓睿、巩民顺、吉燕、水淼、宗志勇、安静、曹方、曾苗苗、陈超。

编写本书的过程中，虽然编者竭尽全力，不敢有丝毫疏忽，但恐百密一疏，书中仍难免存在不足之处，望广大读者批评指正。

编著者

目 录

第 1 章 数据的表示	1
1.1 一则童话	1
1.1.1 0 和 1 的故事	1
1.1.2 0 是什么都没有?	2
1.1.3 0 的位置	3
1.1.4 程序中的 0	4
1.2 司空见惯的十进制数	8
1.2.1 远古的结绳记事	9
1.2.2 什么是十进制计数	10
1.2.3 为啥人类习惯十进制	11
1.2.4 十进制运算规则	11
1.2.5 十进制数的分解	13
1.2.6 $20!$ 等于多少	14
1.2.7 大整数构想	16
1.3 为啥要用二进制	18
1.3.1 人脑与电脑	18
1.3.2 二进制计数规则	20
1.3.3 简单的二进制运算规则	22
1.3.4 二进制数的分解	25
1.3.5 十进制数转换为二进制数	25
1.4 还有哪些进制	26
1.4.1 神奇的八卦: 八进制	26
1.4.2 钟表使用的十二进制	28
1.4.3 半斤八两: 十六进制	29
1.4.4 60 年一个甲子: 六十进制	30
1.4.5 各种进制之间的转换	30
1.4.6 二进制与八进制、十六进制的转换	33
第 2 章 神奇的素数	35
2.1 怎么判断素数	35
2.1.1 什么是素数	35

2.1.2	验证素数	36
2.1.3	寻找素数的算法	38
2.1.4	已被证明的素数定理	41
2.2	孪生素数	43
2.2.1	什么是孪生素数	43
2.2.2	孪生素数的公式	44
2.2.3	中国剩余定理	44
2.2.4	孪生素数分布情况	45
2.3	使用素数的 RSA 算法	47
2.3.1	什么是 RSA	47
2.3.2	RSA 算法基础	48
2.3.3	RSA 算法实践	50
2.3.4	RSA 应用：数字签名	51
2.3.5	RSA 被破解的可能性	52
2.4	哥德巴赫猜想	53
2.4.1	哥德巴赫猜想是什么	53
2.4.2	数值验证	55
2.5	梅森素数	57
2.5.1	什么是梅森素数	57
2.5.2	已知的梅森素数列表	58
第 3 章	递归——自己调用自己	61
3.1	从前有座山，山里有座庙	61
3.1.1	老和尚讲的故事	61
3.1.2	德罗斯特效应	61
3.1.3	什么是递归	62
3.1.4	用递归能解决哪些问题	63
3.1.5	一个简单例子：求最大公约数	64
3.2	用递归计算阶乘	66
3.2.1	阶乘该怎么计算	66
3.2.2	阶乘的递归计算方法	70
3.2.3	递归的过程	71
3.2.4	递归的本质：缩小问题规模	74
3.3	汉诺塔	75
3.3.1	古老的传说	75
3.3.2	从两个盘考虑	76
3.3.3	找出递归结构	78
3.3.4	实现程序	80
3.3.5	究竟需要移动多少次	82
3.4	斐波那契数列	83

3.4.1	兔子的家族	83
3.4.2	从最初几月数据中找规律	83
3.4.3	斐波那契数列	85
3.4.4	神奇的魔八方	87
第 4 章	排列组合——让数选边站队	90
4.1	把所有情况都列出来	90
4.1.1	从 0 还是 1 开始	90
4.1.2	赛程安排	92
4.2	乘法原理	94
4.2.1	行程安排的问题	94
4.2.2	乘法原理适用条件	95
4.2.3	棋盘上棋子的放法	96
4.2.4	买彩票保证中奖的方法	98
4.3	加法原理	99
4.3.1	仍然是行程问题	99
4.3.2	总结出的加法原理	99
4.3.3	骰子出现偶数的次数	100
4.4	排列与组合的关系	101
4.4.1	排列	101
4.4.2	组合	106
4.4.3	排列与组合的联系	109
4.4.4	可重排列	110
4.5	计算机中的字符编码	113
4.5.1	ASCII 码能表示的字符数量	114
4.5.2	能表示更大范围的编码	117
4.6	密码的长度	119
4.6.1	容易破解的密码	119
4.6.2	多长的密码才安全	120
4.6.3	密码中使用的字符数量也很关键	120
第 5 章	余数——数据分组	122
5.1	复习小学的余数	122
5.1.1	自然数的余数	122
5.1.2	余数的性质	123
5.1.3	用余数进行分组	126
5.2	日历中的数学	127
5.2.1	n 天后是星期几	127
5.2.2	下月的今天是星期几	129

5.2.3	10 年后的“今天”是星期几	130
5.3	心灵感应魔术	132
5.3.1	一个小魔术	132
5.3.2	魔术师是怎么猜出来的	135
5.4	奇偶校验	139
5.4.1	不可靠的网络传输	139
5.4.2	用奇偶校验检查错误	139
5.5	吕洞宾不能坐首位	140
5.5.1	座位安排	141
5.5.2	试排座位找规律	142
5.5.3	西方的约瑟夫环	144
5.5.4	用数学方法解约瑟夫环	147
5.6	智叟分牛	150
5.6.1	遗产分配难题	150
5.6.2	智叟给出的分配方案	151
5.6.3	分配原理	151
第 6 章	概率——你运气好吗	154
6.1	初中学习过的概率	154
6.1.1	谁先开球	154
6.1.2	用程序模拟抛硬币	155
6.1.3	什么是概率	158
6.1.4	必然事件与不可能事件	159
6.1.5	概率的基本性质	160
6.2	百枚钱币鼓士气	161
6.2.1	狄青的计谋	162
6.2.2	全为正面的概率是多少	162
6.2.3	必然还是偶然	165
6.3	庄家的胜率是多少	165
6.3.1	一个看似公平的游戏	165
6.3.2	庄家能赢钱吗	166
6.3.3	庄家盈利比率	168
6.3.4	游戏参与者获胜的概率	170
6.4	你能中奖吗	171
6.4.1	想中大奖吗	171
6.4.2	计算中奖概率	172
6.5	渔塘中有多少条鱼	177
6.5.1	该怎么估算渔塘中的鱼	177
6.5.2	用概率来估算	178

6.5.3 用概率方法求 π 值	179
第 7 章 翻一番是多少	182
7.1 翻番的概念	182
7.1.1 什么是翻番	182
7.1.2 翻倍的概念	183
7.1.3 计算倍数和番数	184
7.2 复利的威力	184
7.2.1 利润——投资回报	185
7.2.2 认识单利	185
7.2.3 认识复利	187
7.2.4 计算投资回报的程序	190
7.2.5 忘还钱的信用卡	191
7.2.6 爱因斯坦的 72 法则	193
7.3 对折纸张	194
7.3.1 有趣的问题：纸张对折	194
7.3.2 100 米长的纸能对折几次	195
7.3.3 计算对折次数的程序	198
7.4 一棋盘的麦子	200
7.4.1 舍罕王的赏赐	200
7.4.2 需要多少麦粒	201
7.5 折半法的运用	203
7.5.1 翻番的逆运算	203
7.5.2 找出假硬币	203
7.5.3 编写程序找出假硬币	207
7.5.4 折半法在查找中的应用	209
第 8 章 数理逻辑——非此即彼	212
8.1 逻辑的重要性	212
8.1.1 模棱两可的表述	212
8.1.2 肯定或否定	213
8.1.3 程序中的逻辑判断	213
8.2 命题逻辑	214
8.2.1 什么是命题	214
8.2.2 命题的逻辑形式	216
8.2.3 简单命题	217
8.2.4 复合命题	217
8.2.5 复合命题的联结词	218
8.3 布尔逻辑	224

8.3.1	逻辑或	225
8.3.2	逻辑与	227
8.3.3	逻辑非	228
8.3.4	逻辑异或	229
8.3.5	二进制位运算	230
8.4	考虑到各种可能了吗	233
8.4.1	逻辑重叠的实例	233
8.4.2	逻辑遗漏的实例	235
8.4.3	用数轴确定边界	236
8.5	用卡诺图简化逻辑函数	237
8.5.1	什么是卡诺图	237
8.5.2	三变量卡诺图	239
8.5.3	四变量卡诺图	240
8.5.4	卡诺图化简	242
8.5.5	卡诺图中的相邻	244
第 9 章	推理——逻辑的应用	246
9.1	演绎推理	246
9.1.1	认识演绎推理点	246
9.1.2	三段论	247
9.1.3	选言推理	249
9.1.4	假言推理	252
9.1.5	关系推理	256
9.1.6	演绎推理综合实例	257
9.2	归纳推理	258
9.2.1	什么是归纳推理	258
9.2.2	完全归纳推理	260
9.2.3	不完全归纳推理	261
9.3	足球比赛的得分	265
9.3.1	粗心的记分员	265
9.3.2	从已有数据推算出比分	267
第 10 章	几何图形构造	271
10.1	花盆摆放问题	271
10.1.1	10 盆花摆成 5 行，每行 4 盆	271
10.1.2	转变思路，找出答案	272
10.1.3	升级问题（10 盆花摆 10 行，每行 3 盆）	274
10.2	残缺的棋盘能补上吗？	275
10.2.1	被切割的棋盘	275

10.2.2	能拼接出残缺棋盘吗	276
10.3	线条哪里去了?	278
10.3.1	神奇的魔术	278
10.3.2	解析丢失的线条	279
10.4	图形剪拼	280
10.4.1	均分三角形	281
10.4.2	拼接正方形	282
第 11 章	统筹规划	286
11.1	认识统筹规划	286
11.1.1	田忌赛马	286
11.1.2	为什么会赢	287
11.2	生活中的统筹规划	288
11.2.1	匆忙的早晨	288
11.2.2	如何节约运输成本	290
11.3	著名的背包问题	292
11.3.1	什么是背包问题	292
11.3.2	用递归程序解决背包问题	294
11.3.3	用穷举法解决背包问题	298

第1章 数据的表示

数学古称算学，是中国古代科学中一门重要的学科。根据中国古代数学发展的特点，可以分为5个时期，分别是萌芽、体系的形成、发展、繁荣和中西方数学的融合。

在数学的不同发展阶段，对于数据的表示都有一些不同的形式。从远古的结绳记数，到现在用计算机等现代科技设计记数，数的表示形式也在逐步演化。

本章主要介绍数据的各种表示形式，包括各种进制及进制之间的转换。

1.1 一则童话

根据我们所学的知识可知道，数据通常是用0、1、2、3、4、5、6、7、8、9这些数来表示，由这些数的不同组合表示现实生活中各种各样的数据。首先来看这个数列中的前两个数：0和1，从通常意义来说，0就是什么也没有，真的是这样吗？对程序员来说不应该这样理解。

先来看这样一个问题，0和1谁大？

$0 > 1?$ $0 < 1?$

1.1.1 0和1的故事

在数学王国里，胖子0与瘦子1常常为了谁大而争执不休。瞧！今天，这两个小冤家狭路相逢，彼此之间又展开了一场舌战。

瘦子1抢先发言：“哼！胖胖的0，你有什么了不起？就像100，如果没有我这个瘦子1，你这两个胖0有什么用？”

胖子0不服气了：“你也甭在我面前耍威风，想想看，要是没有我，你就只是一个光杆呢？”

“哟！”1不甘示弱，“你再神气也不过是表示什么也没有，看！ $1+0$ 还不等于我本身，你哪儿派得上用场啦？”

“去！ 1×0 结果也还不是我，你1不也同样没用！”0针锋相对。

“你……”1顿了顿，随机应变道，“不管怎么说，你0就是表示什么也没有！”

“这就是你见识少了。”0不慌不忙地说，“你看，日常生活中，气温0度，难道是没有温度吗？再比如，直尺上没有我作为起点，哪有你1呢？”

“再怎么比，我始终比你大。”1信心十足地说。

听了这话，0更显得理直气壮地说：“嘿嘿，你的大小还得我说了算，我站你左边，你就成0.1，我站你右边你就是10。怎么样？我可让你放大10倍，也可让你缩小10倍！”



眼看着胖子0与瘦子1争得脸红耳赤，谁也不让谁，一旁观战的其他数字们都十分着急。

这时，9灵机一动，上前做了个暂停的手势：“你俩都别争了，瞧你们，1、0有哪个数比我大？”

“这……”胖子0、瘦子1哑口无言。

这时，9才心平气和地说：“1、0，其实，只要你们站在一块，不就比我大了吗？”

1、0面面相觑，半晌才搔搔头笑了。“这才对嘛！把自己的位置放正，就能起到应有的作用”。9语重心长地说。

从以上故事可看出以下两点：

- 0并不表示什么都没有。
- 数的大小与所处的位置有关系。

下面就来讨论这两个问题。

1.1.2 0是什么都没有？

通常意义上，0表示“没有”的意思。例如，“2012年过去了，可我的收获为零！”这就表示在2012年没有收获。

但是，0真表示什么都没有吗？

其实，0不仅表示什么都没有，它还有更丰富的内涵。例如，0度并不是没有温度，而是表示温度为0度，比零下1度高，比1度低，如图1-1所示。

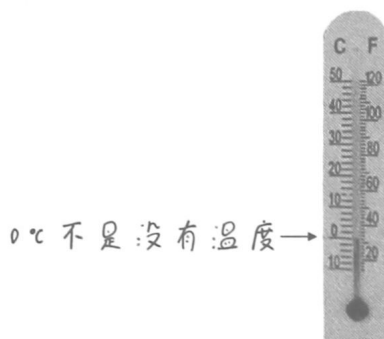


图 1-1

在日常生活的常用语中，也有很多用 0 来表示的，如“很多女孩子都喜欢吃零食”，这里的“零食”并不是表示没有“食”，如图 1-2 所示。



图 1-2

“为了增加收入，改善生活，很多程序员在业余时间都会接点零活来做。”这里的“零活”并不是没有“活”。

其实，在数学上，0 也并不是表示没有。例如，8 和 8.0 相等吗？其含义相同吗？

$$8 \neq 8.0$$

看起来在小数点后添加一个 0 是没有意义的，不过，其含义实际是不相同的。在近似数表示中，数字 8 表示数据只精确到个位，如 7.9、8.2 等数精确到个位都表示为 8。而 8.0 表示数据精确到十分位，如 8.02、7.99 等数精确到十分位都表示为 8.0。所以，从这个角度来看，8 和 8.0 是不相等的。

1.1.3 0 的位置

从“0 和 1 的故事”可看出，当 0 所处的位置不同时，其含义也不一样。如前面说的 8 和 8.0，当把 0 放在小数点后面时，从绝对值方面来看，两个数是相等的，但从近似数来看，小数点后多了一个 0，其表示的含义也就不一样了。

那么，在小数点左侧添加 0 呢？如果在数的最左侧添加 0，无论添加多少个 0，数的大小都不变。

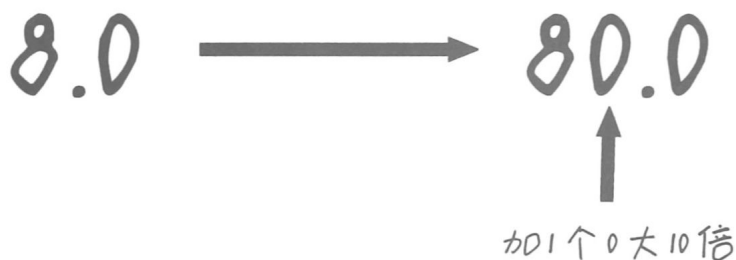
$$8 = 08 \quad 80 = 080$$

但是，如果在数的中间插入 0，数的位置与数的大小关系就很明显了，如在 18 的中间插入一个 0，得到的是 108，很明显，其大小差别很大。

$$18 \neq 108$$

对于 18，表示十位为 1，个位为 8，也就是说，表示 18 这个数有 1 个 10，8 个 1。而 108，表示百位为 1，十位为 0，个位为 8，即表示有 1 个 100，0 个 10，8 个 1，这时的 0 是一个占位符，把 1 从十位挤到百位。

而如果在紧邻小数点的左侧添加 0，则数据会扩大 10 倍。



1.1.4 程序中的 0

在电子技术中，0 一般表示低电平，1 为高电平。在逻辑计算中，0 一般表示逻辑假 (False)，1 为逻辑真 (True)。在数值运算中，0 与平常数学中 0 的含义相同。

在程序中，数据 0 有什么含义呢？

1. 未赋值的变量为 0？

在不同的程序设计语言中，对于未赋值变量的处理不一样。

对于 Basic 类的程序语言，如 QB (Quick Basic, 简环 QB)、VB (Visual Basic, 简称 VB)，如果数值型变量未赋初值，则其初始值为 0。例如，有以下 VB 程序代码：

```
Private Sub Test
    Dim i As Integer
    MsgBox "变量 i=" & i, , "变量初始值"
End Sub
```

在以上 VB 代码中，声明了变量 *i*，但未对其进行赋值。虽然未进行变量赋值初始化，但 VB 编译器会自动将这类数值型变量初始化为 0。因此，执行以上代码将显示如图 1-3 所示的对话框。

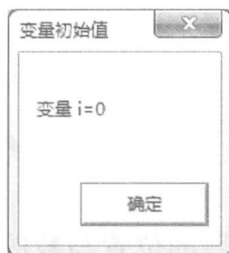


图 1-3