

21世纪高职高专规划教材

计算机应用系列

C语言程序设计案例教程

吴绍根 主 编
邓伟林 张赞波 孙素云 副主编

清华大学出版社



21 世纪高职高专规划教材

计算机应用系列

C语言程序设计案例教程

吴绍根 主编

邓伟林 张赞波 孙素云 副主编



清华大学出版社

北 京

内 容 简 介

本书采用“以项目为中心、案例驱动”的编写方式,围绕案例的实施过程逐步引入和介绍 C 语言程序设计的相关知识,并使用这些知识来解决案例中的问题。为了达到较好的学习效果,本书同时配有练习用案例,使读者边学边练,以期达到更好的效果。本书主要内容包括:用流程图表示业务过程、综合案例场景及目标、显示程序的基本信息、构建程序功能菜单、构建程序功能框架、处理简单数据、处理复合数据、处理动态数据及保存程序数据等。

本书结构合理,语言通俗易懂,并具有理论内容适当、侧重技术应用的特点。该教材主要面向高职高专的学生,同时也可作为相关技术人员的培训教材。

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

C 语言程序设计案例教程/吴绍根主编. —北京:清华大学出版社,2010.9

(21 世纪高职高专规划教材. 计算机应用系列)

ISBN 978-7-302-23319-0

I. ①C… II. ①吴… III. ①C 语言—程序设计—高等学校:技术学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 150172 号

责任编辑:张龙卿(sdzlq123@163.com)

责任校对:袁 芳

责任印制:孟凡玉

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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:北京国马印刷厂

经 销:全国新华书店

开 本:185×260

印 张:12.5

字 数:295 千字

版 次:2010 年 9 月第 1 版

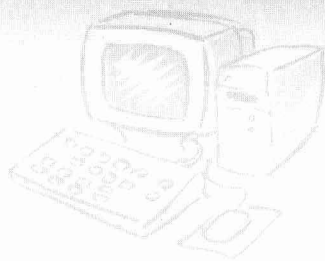
印 次:2010 年 9 月第 1 次印刷

印 数:1~3000

定 价:23.00 元

产品编号:037166-01

前 言



学习知识的目的是为了运用所学的知识来解决问题,这是学习的根本出发点和最终归宿。本书通过对案例进行分解,由浅入深地介绍了 C 语言的相关知识。与本书主要案例并行的是一个练习用案例,学习者通过参考该案例,达到边学边练、学以致用目的。

本书基于“工学结合、以项目为中心、案例驱动教学、边讲边练”的教学思想,以综合案例为依托,介绍了 C 语言程序设计的关键技术。

本课程建议授课时间为 90 课时,其中理论课为 50 课时,练习为 40 课时,并要求先修计算机导论课程。

本书共 11 章,其中第 1~3 章、第 7 章、第 10 章和第 11 章由吴绍根老师编写,第 4 章和第 5 章由邓伟林老师编写,第 6 章由张赞波老师编写,第 8 章和第 9 章由孙素云老师编写。各章具体内容如下:第 1 章介绍了流程图的基本概念、图符的使用及如何使用流程图表达业务过程。第 2 章对书中所使用的教学用案例和练习用案例的场景及目标进行了介绍。第 3 章引入、介绍了 C 语言程序的基本结构,并编写了显示案例程序基本信息的最简单的 C 语言程序。第 4 章通过构建程序功能菜单的需求和过程,介绍了 C 语言的变量、常量、数据类型、流程控制语句等 C 语言最基本也是最重要的内容。第 5 章介绍了 C 语言的函数及其使用方法。第 6 章以处理案例程序中学生的姓名为目标,详细介绍了 C 语言的数组及其用法。第 7 章以处理完整的学生信息为需求,介绍了 C 语言的结构类型及其应用。第 8 章介绍了 C 语言的指针及其应用。第 9 章介绍了文件操作的基本过程、方法及应用。第 10 章为最终的学生信息管理系统完整代码。第 11 章以一个案例程序的实施来演示使用 C 语言开发应用程序的基本步骤和方法,以强化前面各章所介绍的知识。

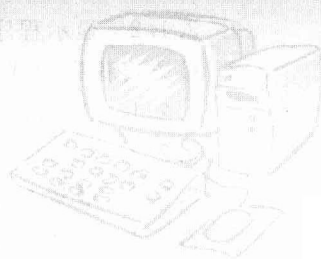
本书的顺利出版,要感谢广东轻工职业技术学院的领导和老师给予的大力支持和帮助,李洛、汪清明、古凌岚几位老师都提出了许多富有启发性的建议,作者在此表示衷心的感谢。

限于作者水平,书中难免存在不妥之处,敬请读者原谅,并提出宝贵意见。

编 者

2010 年 6 月

目 录



第 1 章 用流程图表示业务过程	1
1.1 案例 1-1: 去图书市场买书	1
1.1.1 分析及描述“去图书市场买书”的过程	1
1.1.2 流程图及流程图最基本图符	2
1.1.3 学生练习: 用流程图表示“去商场买东西”的过程	3
1.2 案例 1-2: 计算从 1 到 n 的整数和	3
1.2.1 用流程图描述“计算从 1 到 n 的整数和”	3
1.2.2 流程图高级图符	4
1.2.3 学生练习: 用流程图表示“计算从 1 到 n 的乘积”	5
1.3 案例 1-3: 判断任一年份是否为闰年	5
1.3.1 用流程图描述“判断任一年份是否为闰年”	5
1.3.2 学生练习: 用流程图表示“计算二次方程的根”	6
1.4 拓展	6
1.5 本章小结	7
第 2 章 综合案例场景及目标	9
2.1 教学案例介绍: 学生信息管理系统	9
2.1.1 学生信息管理系统场景	9
2.1.2 学生信息管理系统最终目标	10
2.1.3 分析学生信息管理系统	10
2.1.4 学生信息管理系统运行过程	11
2.2 边讲边练案例介绍: 图书信息管理系统	13
2.2.1 图书信息管理系统场景	13
2.2.2 图书信息管理系统最终目标	13
2.2.3 分析图书信息管理系统	14
2.3 本章小结	14
第 3 章 显示程序的基本信息	15
3.1 案例 3-1: 显示“学生信息管理系统”的基本信息	15
3.1.1 目标与要求	15



3.1.2	显示程序基本信息的最简单的 C 语言程序	15
3.2	C 语言程序的基本结构及运行过程	16
3.2.1	C 语言程序的基本结构	16
3.2.2	C 语言程序的运行过程	18
3.2.3	最基本的输出信息函数 printf	19
3.2.4	C 语言的历史	20
3.3	编辑、编译及运行 C 语言程序	21
3.3.1	下载及安装 CodeBlocks 集成开发环境	21
3.3.2	编辑 C 语言程序	22
3.3.3	编译 C 语言程序	28
3.3.4	运行 C 语言程序	29
3.4	学生练习：显示“图书信息管理系统”的基本信息	30
3.5	拓展	30
3.6	本章小结	30
第 4 章	构建程序功能菜单	31
4.1	案例 4-1：构建“学生信息管理系统”的功能菜单	31
4.2	显示功能菜单	32
4.3	菜单的选择	32
4.3.1	变量及常量	32
4.3.2	C 语言基本数据类型	34
4.3.3	运算符和表达式	38
4.3.4	scanf 及 getchar 的使用	43
4.4	处理菜单选择	44
4.4.1	检测菜单选择	44
4.4.2	基本分支语句	44
4.4.3	多分支语句	48
4.4.4	循环控制语句	50
4.5	案例实施：本阶段案例代码	54
4.6	学生练习：构建“图书信息管理系统”的功能菜单	55
4.7	拓展	56
4.8	本章小结	56
第 5 章	构建程序功能框架	57
5.1	案例 5-1：构建“学生信息管理系统”的功能框架	57
5.2	使用函数构建程序功能框架	57
5.2.1	函数及其分类	58
5.2.2	函数的定义	59
5.2.3	函数声明	60



5.2.4 函数的调用	61
5.3 案例实施: 构建“学生信息管理系统”的功能框架	64
5.4 学生练习: 构建“图书信息管理系统”的功能框架	67
5.5 拓展	67
5.5.1 main()函数的参数	67
5.5.2 函数的递归调用	69
5.6 本章小结	70
第6章 处理简单数据	71
6.1 案例 6-1: 处理“学生信息管理系统”学生的姓名信息	71
6.2 使用数组表示学生姓名	72
6.2.1 定义数组	72
6.2.2 访问数组元素	73
6.2.3 字符数组的特性及其应用	73
6.2.4 常用 C 库字符串操作函数	74
6.2.5 二维数组	77
6.2.6 全局变量	78
6.3 案例实施: 处理“学生信息管理系统”学生的姓名信息	79
6.3.1 定义全局变量	79
6.3.2 录入功能	79
6.3.3 查询功能	80
6.3.4 修改功能	81
6.3.5 删除功能	81
6.3.6 列表功能	83
6.4 学生练习: 处理“图书信息管理系统”的图书名称	84
6.5 拓展	84
6.5.1 scanf 函数和 gets 函数	84
6.5.2 全局变量的一些讨论	84
6.5.3 strcpy 函数与 memmove 函数	84
6.5.4 数组名作为函数参数	85
6.6 本章小结	86
第7章 处理复合数据	87
7.1 案例 7-1: 处理“学生信息管理系统”中学生的完整信息	87
7.2 使用结构类型表示学生的完整信息	88
7.2.1 定义结构类型	88
7.2.2 定义结构类型变量	89
7.2.3 访问结构类型变量的成员及结构变量的初始化	91
7.2.4 使用结构类型数组	91



7.2.5	结构类型变量作为函数参数	92
7.3	案例实施：处理“学生信息管理系统”中学生的完整信息	93
7.3.1	添加新的工程文件	93
7.3.2	工程各个源文件的内容	100
7.3.3	修改代码以管理学生的完整信息	105
7.3.4	编译和运行修改后的程序	109
7.4	调试程序	110
7.4.1	设置程序为可调试状态	111
7.4.2	设置程序的断点	111
7.4.3	启动程序调试	112
7.4.4	单步跟踪程序	113
7.4.5	结束调试	116
7.5	学生练习：处理“图书信息管理系统”中的图书的完整信息	116
7.6	拓展	116
7.6.1	union 共同体	116
7.6.2	enum 枚举类型	117
7.7	本章小结	119
第8章	处理动态数据	120
8.1	案例 8-1：处理“学生信息管理系统”学生人数的变动	120
8.2	使用指针处理学生人数的变动	121
8.2.1	指针及其作用	121
8.2.2	定义指针变量	122
8.2.3	指针操作符的使用	123
8.2.4	结构类型指针变量的使用	124
8.2.5	指针变量作为函数参数	128
8.2.6	内存空间的分配和释放函数 malloc 及 free 函数的使用	130
8.3	案例实施：处理“学生信息管理系统”学生人数的变动	131
8.4	学生练习：处理“图书信息管理系统”的动态图书数据	132
8.5	拓展	133
8.5.1	指向一维数组的指针变量	133
8.5.2	指向二维数组的指针变量	134
8.5.3	用指针处理链表	136
8.6	本章小结	143
第9章	保存程序数据	144
9.1	案例 9-1：将“学生信息管理系统”的学生信息进行永久保存	144
9.2	使用文件永久保存学生信息	145
9.2.1	文件及其作用	145



9.2.2	文件操作的基本过程	145
9.2.3	打开文件	146
9.2.4	读写文件	148
9.2.5	关闭文件	151
9.3	案例实施：将“学生信息管理系统”的学生信息进行永久保存	151
9.4	学生练习：将“图书信息管理系统”中的图书信息永久保存	153
9.5	拓展	153
9.6	本章小结	156
第 10 章	最终的学生信息管理系统完整代码	157
10.1	Student.h	157
10.2	function.h	157
10.3	StudentManager.c	158
10.4	function.c	160
第 11 章	名片管理系统	166
11.1	名片管理系统案例场景及功能规划	166
11.2	在 Code::Blocks 中创建名片管理系统工程	166
11.3	定义名片基本数据信息	167
11.4	编写程序功能主菜单	168
11.5	编写功能实现函数	171
11.6	名片管理系统运行效果	177
11.7	本章小结	179
附录 1	常用 ASCII 码表	180
附录 2	常用 C 语言库函数	181
参考文献		187



第1章 用流程图表示业务过程

在做任何一件事情以前,首先会构思应该如何完成这件事情,然后将完成这件事情的步骤或方法记忆在头脑中或记录在某种介质上,例如纸上。如果所要完成的事情比较复杂,一般的方法是采用某种表达形式,例如文字、图表等表达形式将完成事情的步骤或方法首先记录下来,然后按照预定的方法进行具体的实施。流程图是表达工作步骤或方法的一种图形方法,它可以比较清晰、明了、无歧义地表达工作步骤。

程序流程图是人们对解决问题的方法、思路或策略的一种描述。流程图的优点包括:

- (1) 采用简单规范的符号,画法简单;
- (2) 结构清晰,逻辑性强;
- (3) 便于描述,容易理解。

本章将通过具体的例子介绍流程图的基本图符和用法。

本章要求:

- (1) 理解流程图图符的含义及其作用;
- (2) 能够使用流程图描述解决问题的过程或步骤。

1.1 案例 1-1: 去图书市场买书

首先采用“去图书市场买书”这个最简单的也是最通俗的例子来引入“流程图”的概念,然后再介绍“流程图”这个基本工具及其使用方法。

1.1.1 分析及描述“去图书市场买书”的过程

读书人都曾经到书店或图书市场买过书。虽然设有在纸上采用流程图形式记录下买书的过程或步骤,但是一定经过了如下几个步骤。

第一步,有买书的意愿;

第二步,确定要买什么书;

第三步,确定到哪个书店或图书市场才可以买到需要的书;

第四步,坐车或步行到目的书店或目的图书市场;

第五步,找到需要的图书;

第六步,付款;



第七步,得到需要的图书;

第八步,结束。

以上描述形式其实就是“去图书市场买书”这个过程的文字描述。你或许会问:用文字方式描述做事情的过程不也是挺好吗,为什么还要引入“流程图”这个工具呢?对这个问题的回答很简单:流程图可以更加简单、清晰、直观和明了地描述过程,对于比较复杂的过程,用流程图表示将更加方便!为了说明这个问题,在具体介绍流程图使用以前,试着采用流程图来描述“去图书市场买书”的过程,如图 1-1 所示。

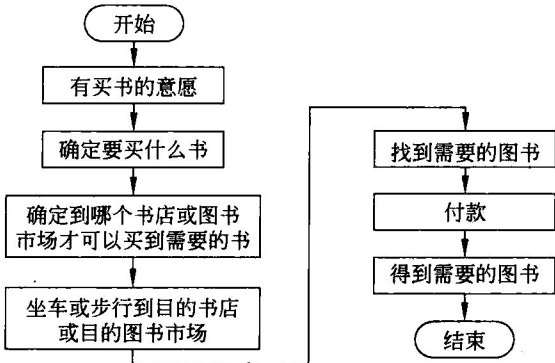


图 1-1 去图书市场买书的流程图

同样表示“去图书市场买书”这个过程,对比采用文字形式的描述和采用流程图形式的描述可以发现:采用流程图形式的描述更能体现做事情的过程性。从图 1-1 可以清晰地看出,为了完成“去图书市场买书”这件事情,在做任何一件事情,都需要有做这些事的先后顺序:有开始,有各种处理,有结束。为了具有一般性,在下面的论述中,将采用“解决问题”这个规范的字眼来代替“做事情”这个通俗的字眼。

1.1.2 流程图及流程图最基本图符

虽然现在还没有介绍流程图的基本用法,但是在描述“去图书市场买书”这个流程图中,已经采用流程图的一些符号来表示解决问题的过程。所谓流程图,通俗来讲,就是使用一些基本图形符号来描述解决问题的过程及方法。在描述“去图书市场买书”这个流程图中使用的流程图图符及其含义如表 1-1 所示。

表 1-1 流程图基本图符表

图 符	含 义
	起止符号。用于指明处理的开始或结束,通常包括 start、begin、stop、end 等词语
	处理符号。指明需要进行的处理,通常包括需要进行的处理的名称,处理的名称可以自由设定,但最好能表达处理的内容
	流程方向线。用于指明处理的先后顺序,处理过程随箭头方向移动



1.1.3 学生练习：用流程图表示“去商场买东西”的过程

大家都曾有去商场买东西的经历,例如买衣服、买电器等。利用已经介绍的流程图的基本用法描述“去商场买东西”的过程。要求:

- (1) 过程要清晰、明了;
- (2) 使用已经介绍的流程图基本图符进行描述;
- (3) 流程图要画得正确、工整。

1.2 案例 1-2: 计算从 1 到 n 的整数和

在这个例子中,为了简单起见,规定 n 必须大于等于 0。

从中学数学中我们了解到,有两种方法可以计算从 1 到 n 的整数和。

方法 1: 也就是直接累加, $1+2+3+\dots+n$;

方法 2: 采用等差数列的和计算公式进行计算,即, $n \times (n+1)/2$ 。这个例子与“去图书市场买书”这个例子不同点在于: 其一,必须首先知道计算从 1 到 n 的整数和中的 n 到底是多少;其二,必须将结果输出。为此,需要引入几个新的流程图图符。

1.2.1 用流程图描述“计算从 1 到 n 的整数和”

为了采用流程图描述计算从 1 到 n 的整数和的计算过程,首先必须要知道这个 n 到底是多少,最简单的方法就是让使用者通过输入的方式输入这个 n 。幸运的是,流程图工具提供了用于表示输入数据的图符,同时也提供了用于表示输出结果的图符和其他一些高级图符,包括:判断、注释等。用流程图描述计算从 1 到 n 的整数和的过程,如图 1-2 和图 1-3 所示。

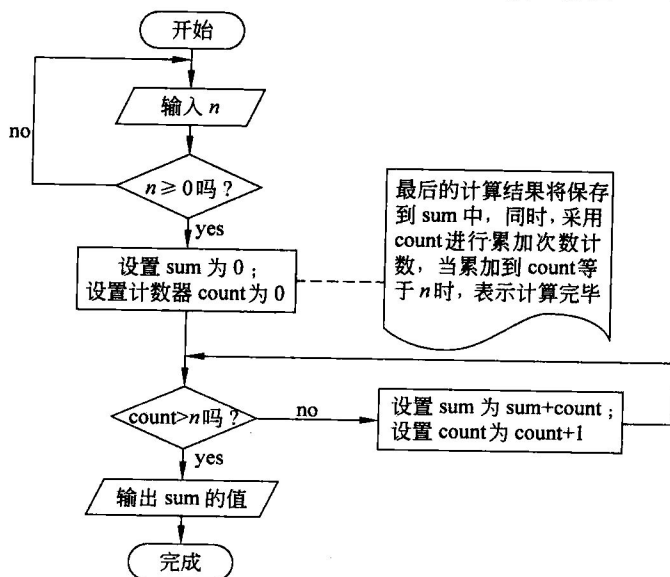


图 1-2 计算从 1 到 n 的整数和的流程图

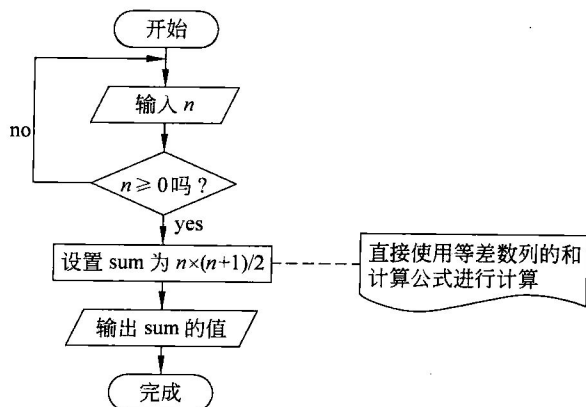


图 1-3 计算从 1 到 n 的整数和的等差数列和公式方式的流程图

从上面的流程图可以看出,可以有多种方法解决同一个问题,在实际生活中情况也是如此。实际上,在用于解决同一问题的可选方法中,总是从可用方法中选取最优的方法,在程序设计中,情况更是如此。例如,在用于解决“计算从 1 到 n 的整数和”中,采用等差数列的和公式比采用从 1 到 n 的累加方式来计算从 1 到 n 的累加和要更加有效,关于如何设计最优方法也是程序设计的一门专门课程——“算法设计”要讨论的问题。所谓算法,简而言之,就是解决某一问题的方法。在这里将不对“算法”做更多的介绍。

回到“计算从 1 到 n 的整数和”问题,可以采用了两种方法解决这个问题,并给出了相应的流程图,下面对这两个流程图进行简单的说明。

在图 1-2 中,首先要求输入 n 的值,然后判断所输入的 n 是否大于等于 0,如果不是,则要求重新输入。如果输入的 n 大于等于 0,则设置结果和 sum 为 0,同时设置一个计算器为 0,然后通过条件“ $count > n$ 吗”来判断是否已经完成从 1 到 n 的共 n 个数的累加。如果 $count$ 大于 n ,说明已经完成了 1 到 n 的 n 个数的累加,并将最后累加结果输出;否则,若 $count$ 小于、等于 n ,则通过“设置 sum 为 $sum + count$ ”将第 $count$ 个数累加到 sum 中,同时,通过“设置 $count$ 为 $count + 1$ ”使 $count$ 指向下一个要累加的数。当 $count$ 大于 n 时,说明已经完成从 1 到 n 的 n 个整数的累加,并将最后的结果输出。

在图 1-3 中,首先要求输入 n 的值,然后判断所输入的 n 是否大于等于 0,如果不是,则要求重新输入;否则,直接采用等差数列的和计算公式直接计算出从 1 到 n 的整数和并输出。太简单了,这也是“算法”的魅力!

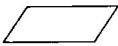
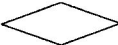
细心的读者可能发现,在上面的说明中,我们没有对在图 1-2 及图 1-3 中的用虚线连接的类似一张纸的一个特殊的框做过说明:这个框其实就起到解释说明的作用,也就是说,它用于对其他流程图图符的解释说明。

1.2.2 流程图高级图符

现在对图 1-2 和图 1-3 中用到的流程图图符进行介绍,其中包括:输入/输出图符、判断图符、注释图符,这些图符的形状及功能如表 1-2 所示。



表 1-2 流程图高级图符-1

图 符	含 义
	输入/输出图符。指明需要获取的数据或是要记录下来的结果。该图符通常用于接收数据输入或将结果输出
	判定图符。根据图符中的条件会产生两个不同的控制走向。这一图符通常包含一个问题,控制的走向是通过标记在分支中的答案来决定的
	注释图符。用于说明其他条目,虚线表明注释的位置

1.2.3 学生练习：用流程图表示“计算从 1 到 n 的乘积”

利用已经介绍的流程图的使用法描述“计算从 1 到 n 的乘积”的计算过程。要求：

- (1) 过程要清晰、明了；
- (2) 使用已经介绍的流程图基本图符进行描述；
- (3) 流程图要画得正确、工整；
- (4) 在认为必须的地方加上注释图符，以使流程图更容易理解。

1.3 案例 1-3：判断任一年份是否为闰年

要判断任一年份是否为闰年，必须首先搞清楚什么是“闰年”。

通常所说的一年 365 天，其实是个约数，准确的数字应是 365.2422 日。那么一年 365 天，就与实际的一年相差 0.2422 日，这样四年之后就比实际的一年少了近 1 天。为了弥补这个差值，历法中规定，4 年设一闰，即能被 4 整除的年份为闰年，另附加规定，凡遇世纪年（末尾数字为两个零的年份），必然被 400 所整除才算闰年。如 1996 年即闰年，2000 年也是闰年，而 1700 年则不是闰年。阳历闰年的 2 月有 29 天，2 月 29 日为闰日，阳历闰年有 366 天。从上面的介绍可以知道，闰年的年份必须同时满足如下的条件：

- (1) 年份能被 4 整除；
- (2) 年份若是 100 的整数倍，需被 400 整除，否则就不是闰年。

根据这些条件就可以判断任一年份是否为闰年。

1.3.1 用流程图描述“判断任一年份是否为闰年”

判断任一年份是否为闰年，与“计算从 1 到 n 的整数和”一样，必须知道要判断的年份，这个年份是任一输入的。判断“任一年份是否为闰年”的流程图如图 1-4 所示。

以上用于“判断任一年份是否为闰年”的流程图是严格按照闰年的条件进行设计的。这个例子也说明，流程图只是解决实际问题的方法的一种形式描述，而不是解决问题的方法，也就是说，流程图只是一种描述手段，而不是方法本身。

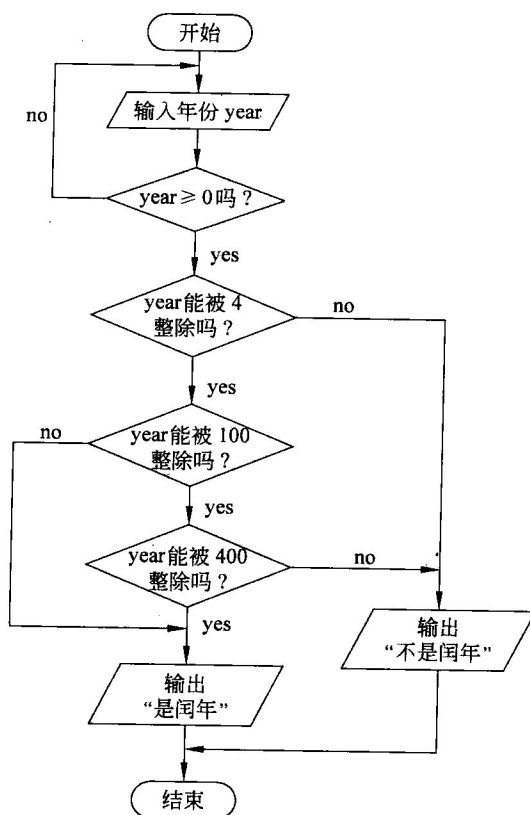


图 1-4 判断任一年份是否为闰年的流程图

1.3.2 学生练习：用流程图表示“计算二次方程的根”

从中学数学了解到,求二次方程的根,就是求 $ax^2+bx+c=0$ 方程的根,其中, a 、 b 、 c 都是实数。这里,为了简单起见,只要求求出方程的实根。要求:

- (1) 过程要清晰、明了;
- (2) 使用已经介绍的流程图基本图符进行描述;
- (3) 流程图要画得正确、工整;
- (4) 在认为必须的地方加上注释图符,以使流程图更容易理解。

 **提示:** 这里,需要首先输入方程的三个系数,然后根据 $\Delta=b^2-4a\times c$ 来判断是否存在实根,进而进一步计算出方程的根。

1.4 拓展

作为一套完整的符号描述系统,流程图除了包含已经介绍的图符以外,还包括高级图符,包括:连接图符、特定处理图符。这两个图符的形式及其含义如表 1-3 所示。



表 1-3 流程图高级图符-2

图 符	含 义
	连接图符。用以表明转向流程图的别处,或从流程图别处转入。它是流线的断点,在图内注明某一标识符,表明该流线将在具有相同标识符的另一连接符处继续下去
	特定处理图符。带有双纵边线的矩形表示已命名的特定处理,该处理已在另外的地方得到详细说明的一个操作或一组操作,矩形内可注明特定处理名或其简要功能

以一个假想的示例说明这两个图符的使用,如图 1-5 所示。

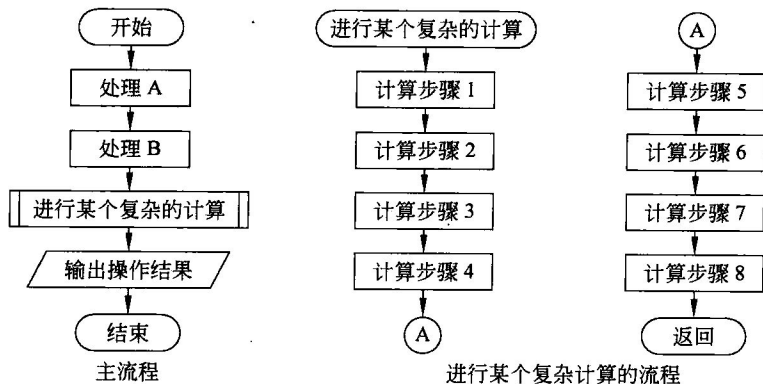


图 1-5 流程图高级图符的使用

在图 1-5 的示例流程图中,由于“进行某个复杂的计算”的流程图已经事先编制好了,因此,可在“主流程”流程图中通过“特定处理图符”直接对其引用。同时,在“进行某个复杂的计算”流程图中,由于该流程图太长,可以使用“连接图符”进行分段描述,从而使得流程图更加整洁、明了。

1.5 本章小结

本章介绍了流程图及如何使用流程图描述解决问题的过程和步骤。作为一个通用的过程描述工具,流程图在实际生产生活中得到了广泛的应用,特别是在程序设计中,流程图是描述“算法”的一种有力的工具。

本章通过三个通俗易懂的例子介绍了流程图及其常用图符的使用。通过本章的学习,读者不仅要能够使用流程图工具描述解决问题的过程,也要求读者能够看懂别人绘制的流程图。

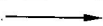
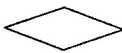
为了便于查阅,我们将流程图常用图符汇总于表 1-4 中。

表 1-4 流程图常用图符汇总

图 符	含 义
	起止符号。用于指明处理的开始或结束,通常包括 start、begin、开始、stop、end、结束等词语
	处理符号。指明需要进行的处理,通常包括需要进行处理的名称,处理的名称可以自由设定,但最好能表达处理的内容



续表

图 符	含 义
	流程方向线。用于指明处理的先后顺序,处理过程随箭头方向移动
	输入/输出图符。指明需要获取的数据或是要记录下来的结果。该图符通常用于接收数据输入或将结果输出
	判定图符。根据图符中的条件会产生两个不同的控制走向。这一图符通常包含一个问题,控制的走向是通过标记在分支中的答案来决定的
	注释图符。用于说明其他条目,虚线表明注释的位置
	连接图符。用以表明转向流程图的别处,或从流程图别处转入。它是流线的断点,在图内注明某一标识符,表明该流线将在具有相同标识符的另一连接符处继续下去
	特定处理图符。带有双纵边线的矩形表示已命名的特定处理,该处理已在另外的地方得到详细说明的一个操作或一组操作,矩形内可注明特定处理名或其简要功能