

●一本借鉴英语的泛读与精读学习方法的编程书

●一本“不是一步给出结果程序”

而是教你“如何才能构造出这样一个程序”的编程书

十十

实用编程技术 百例精编与妙解

周玉龙 编著

南开大学出版社

- 一本借鉴英语的泛读与精读学习方法的编程书
- 一本“不是一步给出结果程序”，而是教你“如何才能构造出这样一个程序”的编程书

C++实用编程技术

百例精编与妙解

周玉龙 编 著

南开大学出版社
天津

内容提要

本书是为提高 C++实用编程技术尤其是面向对象的编程方法与技术而编写的。其主要内容由两部分组成：一是 C++程序设计的使用技术要点及重要语法点；二是本书的主体，是专门针对 C++程序设计尤其是面向对象程序设计而精选的 100 多个应用程序实例，并对每一应用实例都按如下方式进行深入剖析与具体的求解过程，以便研讨怎样以及为什么要这样来“构建”它们：（1）实现要点与方法；（2）程序源码和结果；（3）难点剖析及评注。另外，每一实例都给出了“其他点评”，提出可进一步思考及改进的方向或者某些建议、忠告等。

本书非常适合于初学 C++程序设计的学生、科技工作者以及自学者，而且对学过 C++的读者也肯定会有很高的实用参考价值——可启发思路并提高实用编程技术与技巧。

本书既可作为大专院校学生学习 C++程序设计的教材来单独使用，又可与任何一本讲解 C++的其他教材或书籍配合使用。

图书在版编目(CIP)数据

C++实用编程技术百例精编与妙解 / 周玉龙编著.
天津:南开大学出版社, 2004.1
ISBN 7-310-02002-2

I. C... II. 周... III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 091453 号

出版发行 南开大学出版社

地址:天津市南开区卫津路 94 号 邮编:300071

营销部电话:(022)23508339 23500755

营销部传真:(022)23508542

邮购部电话:(022)23502200

出版人 肖占鹏

承印 河北昌黎人民胶印厂印刷

经销 全国各地新华书店

版次 2004 年 1 月第 1 版

印次 2004 年 1 月第 1 次印刷

开本 787mm×1092mm 1/16

印张 29.5

字数 760 千字

印数 1—3000

定价 41.00 元

前 言

面向对象的编程方法与技术代表着世界上目前最先进的计算机编程方法与技术。C++语言是目前最流行且应用最广泛的面向对象计算机编程语言之一。目前已有越来越多的人准备学习或正在学习与使用 C++ 语言。为了帮助读者尽快学会并提高自己的 C++ 语言使用技术与水平,笔者编写了这本《C++实用编程技术百例精编与妙解》,以奉献给广大的高校师生、科技人员以及计算机爱好者。

学过英语的人都知道,学习英语的难点在于,掌握了必要的语法与使用规则的情况下,怎样自己动手写出正确的英语句子,进而能够写出表达正确意思的文章乃至具有较高质量的文章。人们的经验是:多读范例与范文,而后进行仿效并多练多写,且总要力图对范例进行改进并有所创新,通过反复进行读写文章的实践理解并学会写出具有特色的文章来。学习 C++ 语言亦如此,首先要掌握必要的语法与使用规则,而后的学习难点在于怎样自己亲手编写出正确的 C++ 语句、进而能够编写出表达正确意思的程序(语句序列)乃至具有较高质量的程序。人们的经验仍然是:多读“范例”与“范文”,而后进行仿效并多练多编写,且总要力图对范例程序进行改进并有所创新,通过反复读写 C++ 程序的实践来理解并学会编写具有自我特色的 C++ 程序。

本书的主要内容由两部分组成:一是经高度概括、提炼、总结出的 C++ 程序设计所涉及的使用技术要点及重要语法点,代表上述提到的“首先要掌握必要的语法与使用规则”部分。该部分内容是学会、用好 C++ 语言所必需的,要充分理解、牢固掌握而且要能够灵活使用它们,在各章第二部分的编程实例中将对这些技术要点及重要语法点进行全面而多方位的参照与使用。二是本书的主体,是专门针对 C++ 程序设计尤其是面向对象程序设计所涉及的各部分内容及其相关使用技术而精选摘编出的 100 多个应用程序实例(代表着前面提到的所谓“范例”与“范文”),而且对每一个应用实例都按照如下的“从多层次分析解剖并处理问题”的方式进行深入剖析与具体的求解过程,研讨怎样以及为什么要这样来“构建”每一实例所对应的应用程序:(1)实现要点与方法;(2)程序源码和结果;(3)难点剖析及评注。另外,每一实例结束处都给出了“其他点评”,进而提出可进一步思考及改进的方向,或者给出与本实例的实现技术相关联的其他某些建议、忠告或注意点。即是说,对每一个应用实例的求解,都不是一步就给出“结果程序”,而是把重点放在解释说明“如何才能构造出这样一个程序”、程序内部实现逻辑的关联性以及实现该程序的特点与优缺点评价等方面。

全书共分为十二章,各章内容依次为:第一章 最简单的 C++ 程序设计;第二章 基本数据类型、运算符和表达式;第三章 分支与循环程序设计;第四章 数组的引入和使用;第五章 函数的设计及其应用;第六章 指针、结构体及引用;第七章 面向对象程序设计——类与对象;第八章 继承性与派生类;第九章 运算符重载;第十章 函数模板与类模板;第十一章 流类库与输入输出;第十二章 异常处理。

本书所精选的应用程序实例以及对它们的求解方法与过程可体现出如下多方面的特色:

1. 精选了一批通过自定义类来实现的可体现面向对象编程技术的应用实例,为今后开发面向对象的实用性软件奠定基础。能够编制出各种实用的自定义类、进而全面掌握面向对象

的编程方法与技术是学习 C++语言的重点与难点, 参照与模仿诸多的与自定义类相关的编程实例, 无疑会对这一难点的突破产生积极的作用与效果。

2. 递归函数以及对问题的递归求解方法是学习 C++语言时的另一个难点。本书精选了 10 多个不同问题类型的编程实例, 并着重从如何进行“问题分解”入手, 来学习这种较难掌握的递归求解问题的方法与技术。

3. 为灵活掌握与使用 C++语言的各种数据类型, 并熟悉对各种不同数据的存储与处理方法和手段, 往往要求对同一问题采用尽可能多的不同求解方法, 这样既可以提高对程序设计语言应用能力及其使用灵活性的理解, 同时又能启发学生的思维积极性与创造性。本书中包含有许多一题双解、一题多解的情况, 例如, 一个“反序输出问题”就设计并给出了 10 多种不同的求解方法。

4. 为体现 C++语言求解问题的广泛性与使用灵活性, 精选了一批使用综合编程技术来求解的应用性问题编程实例。如, 八皇后问题, 骑士巡游问题, “万年历”问题, n 元线性方程组求解问题, “算术表达式”求值问题, 学生数据文件管理问题等。

5. 含有若干个与绘图有关的实例及其相关的实现程序。在不同的操作系统与编译实现平台下, 绘图程序的编制方法会有所不同。本书给出的是在 VC6.0 环境下实现的绘图程序。

书中所有实例所对应的程序都在 VC6.0 集成环境下调试通过, 并附有可供参考的程序执行结果。

本书内容的选取与定夺是作者多年来 C++教学和科研经验的总结, 期望它能对广大读者起到较大的帮助与启发作用。

书中肯定会存在缺点与不足, 敬请阅读本书的老师 and 同学们予以批评指正。

电子邮件地址: zhouyl@nankai.edu.cn。

编 者

2003 年于南开大学

目 录

第一章 最简单的C++程序设计	1
1.1 使用技术要点及语法	1
1.2 精选实例剖析与求解	3
例 1-1 在屏幕上显示出若干行字符	3
例 1-2 对两个整数进行指定运算	4
例 1-3 计算圆柱体的表面积与体积	5
例 1-4 使用“简单变量方式”实现反序输出问题	6
第二章 基本数据类型、运算符和表达式	8
2.1 使用技术要点及语法	8
2.2 精选实例剖析与求解	12
例 2-1 对算术运算符的使用	12
例 2-2 关系运算符及逻辑运算符的使用	14
例 2-3 对位运算符的功能加以理解	16
例 2-4 理解运算符优先级并改写表达式	17
例 2-5 计算与正数 x 相关的多个表达式的值	20
第三章 分支与循环程序设计	22
3.1 使用技术要点及语法	22
3.2 精选实例剖析与求解	26
例 3-1 求 3 个数中的最大数以及它们的平均数	26
例 3-2 将百分制成绩转换为等级字母	27
例 3-3 if 语句及 switch 语句的执行效果	28
例 3-4 使用 for 语句来实现累加	30
例 3-5 计算某一参赛选手最后得分	31
例 3-6 判断整数 i 是否为素数	33
例 3-7 求任意 n 个数中的最大数和最小数	34
例 3-8 求 Fibonacci 数列前 20 项的解法	36
例 3-9 求正整数 n 与其反序数的和与差	37
例 3-10 对多种控制语句的使用	38
例 3-11 按所给公式求出两种累加和	40
例 3-12 按公式求出满足 eps 精度的累加和	42
例 3-13 从 m 的下一个数起连续找出 k 个素数	44
例 3-14 编写循环程序显示出指定的图形	45
例 3-15 在屏幕上显示符合要求的某种图形	46

例 3-16 求出多个不同近似程度的E 值	48
例 3-17 求满足所给条件的最小自然数 n	51
例 3-18 用户猜测藏物位置	52
例 3-19 计算机为小学生出题并判断对错	55

第四章 数组的引入和使用.....58

4.1 使用技术要点及语法.....58

4.2 精选实例剖析与求解.....60

例 4-1 使用“数组方式”实现反序输出问题	60
例 4-2 将八进制数化为十进制数	61
例 4-3 将十进制正整数化为八进制数	63
例 4-4 求Fibonacci 数列前 20 项的解法二	65
例 4-5 使用筛法求出 1000 以内的所有素数	66
例 4-6 求出最小的 4 个“完全数”并按指定格式输出结果	67
例 4-7 将多种控制语句与数组配合起来使用	69
例 4-8 对数组元素进行排序	71
例 4-9 在降序数组中按折半法查找某指定元素	73
例 4-10 对两个有序数组进行合并	74
例 4-11 对字符串进行某些指定处理	76
例 4-12 10 人围坐一圈按规则淘汰其中 9 人的数组实现法	77
例 4-13 对字符数组中各数字字符的出现情况进行统计	80
例 4-14 统计一批整数中各数的出现频率	81
例 4-15 把正整数转换为“混英表示字符串”	83
例 4-16 对二维数组的简单使用	85
例 4-17 寻找若干行(字符串)中的最长行并进行某些统计	87
例 4-18 计算某一批参赛选手最后得分	89
例 4-19 统计若干行(字符串)中的关键字各出现了多少次	91
例 4-20 计算机猜测用户“暗记”的某张扑克牌	93

第五章 函数的设计及其应用.....98

5.1 使用技术要点及语法.....98

5.2 精选实例剖析与求解.....105

例 5-1 使用自定义函数实现数学函数的求值过程	105
例 5-2 设计无参函数用于实现某种固定功能	107
例 5-3 设计一参函数使实现的功能可变化	108
例 5-4 设计二参函数使实现功能的变化更灵活	109
例 5-5 无参函数配合全局变量来实现可变功能	110
例 5-6 编写函数返回整数 x 某些指定位的数码值	111
例 5-7 将多种控制语句与数组及函数配合起来使用	113
例 5-8 计算两个日期之间的间隔天数	115

例 5-9	求 Fibonacci 数列前 20 项的解法三	120
例 5-10	理解函数递归调用并给出程序执行结果	121
例 5-11	使用“递归函数方式”实现反序输出问题	123
例 5-12	添加具有某种指定功能的递归函数	125
例 5-13	通过筛法求素数的自定义函数对偶数进行分解	128
例 5-14	将 a 数组中前 n 个整数累加以及求最大数的递归函数	129
例 5-15	理解并叙述递归函数的功能	131
例 5-16	间接递归函数的使用	133
例 5-17	对不同作用域中的同名标识符进行处理	134
例 5-18	文件级作用域和函数级作用域中变量的使用	136
例 5-19	全局变量和具有 static 存储类别的变量的使用	137
例 5-20	extern 型外部变量的使用	139
例 5-21	编写程序求解骑士巡游问题	142
例 5-22	编写程序对八皇后问题进行求解	144
第六章 指针、结构体及引用		150
6.1	使用技术要点及语法	150
6.2	精选实例剖析与求解	156
例 6-1	对字符指针进行使用	156
例 6-2	double 型指针及其相关运算	157
例 6-3	设置指针参数用于实现“双向传值”	159
例 6-4	被调函数中改变形参指针值后无法实现“双向传值”	161
例 6-5	使用“new 运算符与动态数组方式”实现反序输出问题	162
例 6-6	使用 new 与 delete 来进行内存动态分配与释放	163
例 6-7	对各分量均为指针的数组进行使用	165
例 6-8	对 n 元一次线性方程组进行求解	167
例 6-9	通过参数类型的改变实现不同的 swap 函数功能	173
例 6-10	对指向一维数组的指针 p 进行使用	176
例 6-11	指向一维数组的指针作为函数的参数	178
例 6-12	理解数组及其元素与指针的密切关系	181
例 6-13	将结构体与数组结合起来使用	183
例 6-14	计算参赛选手最后得分的函数与结构体实现方法	186
例 6-15	通过使用结构体数据类型对复数进行指定运算	189
例 6-16	使用“指针与链表的栈式用法”实现反序输出问题	191
例 6-17	10 人围坐一圈按规则淘汰其中 9 人的链表实现法	193
例 6-18	使用“指针与链表的队列式用法”实现反序输出问题	196
例 6-19	对函数指针进行使用	198
例 6-20	对返回指针值的函数进行使用	200
例 6-21	利用带参数的 main 函数来实现自定义运算	203
例 6-22	对引用进行简单使用	205

例 6-23 引用型参数以及返回引用的函数.....	207
----------------------------	-----

第七章 面向对象程序设计——类与对象.....	209
--------------------------------	------------

7.1 使用技术要点及语法.....	209
7.2 精选实例剖析与求解.....	218
例 7-1 对象的“诞生”与“死亡”及其相关的构造函数和析构函数.....	218
例 7-2 用户自定义 cylinder 类并对该类进行使用.....	220
例 7-3 使用“自定义类及其对象方式”实现反序输出问题.....	222
例 7-4 自定义的简单复数类 complex.....	224
例 7-5 自定义栈类型 stack 并对它进行使用.....	226
例 7-6 计算某一批参赛选手最后得分的类实现.....	230
例 7-7 通过自定义类在平面上画两条直线和一个圆.....	233
例 7-8 自定义类中的静态数据成员与静态函数成员.....	240
例 7-9 含有常量数据成员的自定义类.....	242
例 7-10 使用友元函数实现简单复数类 complex.....	243
例 7-11 处理字符串的自定义类 myString1.....	246
例 7-12 后缀式简单单步计算器类 Calculator.....	252
例 7-13 自定义的时间类 TimeType.....	259
例 7-14 自定义的日期类 DateType.....	263
例 7-15 含有对象成员的自定义类 line.....	269
例 7-16 系统调用拷贝构造函数的 3 种情况.....	271
例 7-17 情况 1 时显式拷贝构造函数的必要性.....	274
例 7-18 情况 2 时显式拷贝构造函数的必要性.....	276
例 7-19 情况 3 时显式拷贝构造函数的必要性.....	278
例 7-20 处理字符串的自定义类 myString2.....	280

第八章 继承性与派生类.....	285
-------------------------	------------

8.1 使用技术要点及语法.....	285
8.2 精选实例剖析与求解.....	293
例 8-1 基类成员在各自派生类中的存取权限.....	293
例 8-2 利用继承性与派生类来管理学生和教师档案.....	296
例 8-3 含有对象成员的派生类及其相关的初始化工作.....	301
例 8-4 利用派生于类 Matrix 的 Linequ 类来求解线性方程组.....	304
例 8-5 利用组合方式自定义日期时间类 DateTimeType.....	309
例 8-6 利用继承方式自定义日期时间类 DateTimeType.....	313
例 8-7 对虚函数进行简单使用.....	316
例 8-8 利用虚函数手段按不同算法求 Fibonacci 数列的某一项.....	318
例 8-9 在各图元类中使用虚函数 draw 并通过它们画出简单图形.....	324
例 8-10 与派生类对象相关的赋值兼容性以及重名成员处理.....	332

第九章 运算符重载	335
9.1 使用技术要点及语法.....	335
9.2 精选实例剖析与求解.....	340
例 9-1 使用友元方式重载运算符.....	340
例 9-2 使用类成员方式重载运算符.....	343
例 9-3 对自定义简单复数类 complex 中的运算符进行重载.....	346
例 9-4 在链表类 link 中重载下标运算符.....	348
例 9-5 对 new 和 delete 运算符进行重载.....	352
例 9-6 重载 new 与 delete 时全权控制分配与释放细节.....	356
例 9-7 显式赋值运算符重载函数的必要性.....	359
例 9-8 通过 operator<类型名>来自定义类型转换运算符.....	362
例 9-9 成员指针运算符.*与->*的使用.....	364
第十章 函数模板与类模板	367
10.1 使用技术要点及语法.....	367
10.2 精选实例剖析与求解.....	371
例 10-1 求一维数组元素最大值、最小值及平均值并输出的函数模板.....	371
例 10-2 函数模板与函数同名的重载使用方式.....	373
例 10-3 二函数模板同名的重载使用方式.....	374
例 10-4 只使用类型参数的类模板.....	376
例 10-5 使用了类型参数和普通参数的类模板.....	378
例 10-6 实现反序输出问题的类模板.....	380
例 10-7 以类模板作为基类派生出新的类模板.....	383
第十一章 流类库与输入输出	387
11.1 使用技术要点及语法.....	387
11.2 精选实例剖析与求解.....	397
例 11-1 对自定义的简单复数类 complex 重载输入输出运算符.....	397
例 11-2 按指定格式往 cout 对象即屏幕上输出数据.....	400
例 11-3 按指定格式往 text 型磁盘文件中输出数据.....	402
例 11-4 将 10 组数据保存到自定义磁盘文件中而后读出并处理的方式一.....	405
例 11-5 计算某一批参赛选手最后得分的 text 文件实现一.....	407
例 11-6 计算某一批参赛选手最后得分的 text 文件实现二.....	410
例 11-7 将 10 组数据保存到自定义磁盘文件中而后读出并处理的方式二.....	414
例 11-8 统计某源程序文件中各关键字出现的次数.....	416
例 11-9 使用“text 文件方式”实现反序输出问题.....	420
例 11-10 使用“binary 文件的随机访问方式”实现反序输出问题.....	421
例 11-11 对以结构体数据构成的文件进行随机读写处理.....	423
例 11-12 使用随机读写处理方式对学生数据文件进行管理.....	426

第十二章 异常处理	432
12.1 使用技术要点及语法	432
12.2 精选实例剖析与求解	434
例 12-1 无法打开文件的异常处理	434
例 12-2 内存不足获取动态空间失败的异常处理	435
例 12-3 使用 throw 抛掷不同类型的异常并进行处理	437
例 12-4 除以 0 或模 0 所产生的异常及其处理	441
例 12-5 对用户提供不恰当输入数据之异常的处理实例一	444
例 12-6 对用户提供不恰当输入数据之异常的处理实例二	447
例 12-7 使用缺省实参表达式的 throw 来重新抛掷当前的异常对象	451
例 12-8 对局部块作用域中的局部对象进行自动析构处理	453
例 12-9 对函数作用域中的局部对象进行自动析构处理	456
例 12-10 对 throw 所跳过的多个作用域层次中的局部对象进行自动析构	458

第一章 最简单的 C++ 程序设计

1.1 使用技术要点及语法

1. 任一个 C++ 程序都含有一个所谓的 main 函数。最常用的 main 函数的格式为：

```
void main() {  
    <语句 1>  
    :  
    <语句 n>  
}
```

每一语句均以分号作为结束。程序总是从 main 函数的第一个语句开始执行，main 执行结束则整个程序执行结束。

2. 最常用的 C++ 语句是：变量说明（定义）语句、赋值语句（可归为表达式语句）、条件分支语句、循环语句、复合语句、实现跳转的语句等。如，变量说明语句：int a, b, sum, prod; 就指定了 4 个变量名并说明了它们的数据类型为 int 型即整型。又如，赋值语句：sum=a+b; 可使系统负责算出 a+b 之值并把结果值赋给变量 sum（所拥有的存储空间中）。

3. 程序中涉及的数据，以常量和变量的形式出现，变量的值在程序执行中可以被改变。常量又分为字面常量和有名常量，如 3 与 15 等都为字面常量，而通过形如“const float pai=3.14;”的说明指出 pai 为一个 float 类型的有名常量，它的值恒等于字面常量 3.14，该有名常量 pai 的值在程序中不可改变。

常量和变量都具有各自的数据类型。C++ 预定义的基本数据类型有：int、float、double、char、void，它们分别代表整型（int）、浮点型（单精度浮点型 float 和双精度浮点型 double）、字符型（char）以及无值型（void）。程序中用到的所有变量及有名常量都必须在使用前先说明其数据类型（以便让系统为它们分配不同类型、不同大小的、用于存放其值的存储空间）。

4. 通过如下格式的语句来说明一批（如 n 个）变量的数据类型：

<数据类型名> <变量名 1>, ... , <变量名 n>;

变量名必须是满足如下规则的一个 C++ 标识符：标识符是由字母或下划线开头的后跟 0 到任意多个字母、数字或下划线所组成的符号串。除变量名外，程序员为有名常量、用户自定义类型、函数及其参数、类、对象等各种成分所起的名字也都必须是一个标识符。

对程序中说明的每一个变量，系统都为它们分配一个存储空间（用于存放该变量所具有的值）。但注意，若没有对变量进行显式初始化（说明的同时赋上初值），应该认为这些变量空间中并“没有值”（没有用户指定的确切值。但对某些特殊情况下的变量，系统会为它们预置一个隐含初始值），通过赋值语句或通过 cin 的输入可以使变量（拥有的存储空间中）具有了用户给定的确切值。

5. 数据的输入输出是每个实用程序必不可少的组成部分。“输入”向程序提供要加工的初始数据，而“输出”则将运算结果显示（或保存）到某个指定文件（如屏幕文件或磁盘文

件) 中。

C++ 含有两个特殊文件, 一个为标准流对象 `cin`, 它代表键盘, 是 C++ 的标准输入设备; 而另一个为标准流对象 `cout`, 它代表屏幕, 是 C++ 的标准输出设备。若要往标准流对象 `cout` (即屏幕) 上输出信息, 或要从标准流对象 `cin` (即键盘) 上输入信息, 必须在程序中使用编译预处理命令 “`#include <iostream.h>`” 来把系统预定义头文件 “`iostream.h`” 中的内容包含到当前程序中 (使用 `cin` 或 `cout` 二者之一时, 就必须这样做)。

往 `cout` (即屏幕) 上输出信息时, 可通过 “插入” 运算符 “`<<`” (也称输出运算符) 来实现, 其一般使用格式如下:

```
cout << 输出项 1 << 输出项 2 << ... << 输出项 n;
```

其中的各输出项通常为常量、变量、字符串 (更一般地说可为一个表达式) 以及格式控制符等。由 `cout` 来负责计算并依次输出它们 (各输出项) 的值, 当输出项为以双引号括起来的字符串时, 将会 “原样照印” 出该字符串。

从 `cin` (即键盘) 上输入信息时, 可通过 “提取” 运算符 “`>>`” (也称抽取或输入运算符) 来实现, 其一般使用格式如下:

```
cin >> 变量名 1 >> 变量名 2 >> ... >> 变量名 n;
```

实现由键盘一次接收 (输入) 多个值并将这些输入值 (多个输入值间通常以 “空格” 或 “换行” 符号分隔) 依次赋值给 `cin` 中的各变量 (注意 `cin` 中必须使用变量而不允许使用常量或非变量的表达式!)。

6. 字符串常量 (简称为字符串) 是由双引号括起来的字符序列。双引号为定界符, 不属于字符串常量。虽然字符串中的每个字符在内存中只占 1 个字节 (1 byte), 但整个字符串所占内存的字节数却为字符串的长度加 1。因为系统自动在每个字符串常量的末字符后加上了一个 “空字符” (作为结束标志)。系统使用转义字符 “`\0`” 来表示 “空字符”。注意, 字符 “`A`” 在内存占 1 个字节, 而字符串 “`A`” 则占 2 个字节。

往 `cout` (即屏幕) 上输出字符串常量可以对用户起到提示作用。对字符串的处理通常与字符数组和指针有关。

7. 每个 C++ 程序都是一个符号的序列。C++ 语言的基本符号由数字 (0~9)、大小写英文字母 (A~Z, a~z)、以及包括空格在内的特殊符号组成。由符号构成单词, 由单词构成语句, 由语句构成程序 (相当于文章)。

特殊符号可充当运算符 (如: `+`、`-`、`*`、`/`、`>`、`<`、`=`、`!` 等)、分隔符 (如: `,`、`;`、`{`、`}` 等)。另外, 下划线、空格、句点等也属于特殊符号。

C++ 语言的单词主要分为以下五种: 关键字 (也称保留字, 如, `int`、`float`、`const` 等), 标识符 (如, 变量名及有名常量名等各种名字), 字面常量 (如, 43、3.14 等), 运算符 (如, `+`、`=`、`*` 等), 分隔符 (如, `{`、`}`、`;` 等)。

每一个 C++ 关键字 (保留字) 都具有自己特定的使用含义, 用户程序中不可再将它们用作自定义标识符。C++ 预定义的关键字均由小写字母构成。注意, 对于某一个具体的编译实现来说, 可能还会再扩充规定并使用一些不同的关键字, 使用时请留意各扩充关键字的拼写与具体含义。

8. 每一个 C++ 程序从输入计算机到执行 (运行) 都要按照如下的步骤来进行:

编辑 → 编译 → 连接 → 运行。

编辑——借助某个编辑器把程序代码输入到计算机中。

编译——借助编译器把源程序编译成目标程序，通常生成扩展名为 .obj 的目标文件。

连接——用连接器将编译成功的目标文件与库文件等连接成一个可执行的 .exe 文件。

运行——执行上述连接成功的 .exe 可执行文件。

9. 在 VC6.0 集成开发环境下执行 C++ 程序的最简单步骤如下：

(1) 编辑——借助 VC6.0 的编辑器把程序代码输入到计算机中。

具体操作步骤为：File → New → Projects → Win32 Console Application → (右上处)“Project name”文本框中填入如“MyProj1” → OK → Finish → OK → Project → Add To Project → New → C++ Source File → (右中处)“File”文本框中填入如“Prog1”的文件名 → OK → 键入具体程序。

上述操作步骤的含义为：

选 File 菜单 → 选 New 菜单项 → 选 Projects 标签(也称选项卡) → 选 Win32 Console Application 类型 → (右上处)“Project name”文本框中填入如“MyProj1”的工程名 → 选 OK → 选 Finish(此时隐含创建的是一个空的工程) → 选 OK → 选 Project → 选 Add To Project → 选 New → 选 C++ Source File 类型 → (右中处)“File”文本框中填入如“Prog1”的文件名(此时系统将使用 Prog1.cpp 的文件来保存所键入的源程序) → 选 OK → 在当前的编辑窗口键入具体的源程序代码。

(2) 编译——编译 C++ 源程序。

具体操作步骤为：Build → Compiler ...。

上述操作步骤的含义为：

选 Build 菜单 → 选第一菜单项的 Compiler ... (其中的省略号部分为当前编辑窗口的文件名)。

(3) 连接——将编译成功的目标文件与库文件等连接成为一个可执行的 .exe 文件。

具体操作步骤为：Build → Build ...。

上述操作步骤的含义为：

选 Build 菜单 → 选第二菜单项的 Build ... (其中的省略号部分为将要生成的可执行文件名)。

(4) 运行——执行连接成功的 .exe 可执行文件。

具体操作步骤为：Build → Execute ...。

上述操作步骤的含义为：

选 Build 菜单 → 选第 8 菜单项的 Execute ... (其中的省略号部分为将要执行的文件名)。

1.2 精选实例剖析与求解

例 1-1 在屏幕上显示出若干行字符

编程序，执行后在屏幕上显示出如下几行字符：

Hello, world!

We are learning C++! 123ABC321.

OK! Good-by!

[解答]

(1) 实现要点与方法

1) 编写一个 main 函数。其中含有若干个往 cout (即屏幕) 上显示字符串的输出语句。每句负责显示出一行字符。

2) 往 cout (即屏幕) 上输出信息时, 可使用如下格式的所谓输出语句:

```
cout<< 输出项 1 << 输出项 2 << ... << 输出项 n;
```

其中的各输出项通常为常量、变量、字符串 (更一般地说可为一个表达式) 以及格式控制符等。由 cout 负责计算并依次在屏幕上显示出它们 (各输出项) 的值。

(2) 程序源码和结果

```
#include <iostream.h>                                //a
void main() {                                         //b
    cout<<"Hello, world!\n";                          //c
    cout<<"We are learning C++! 123ABC321."<<"\n";
    cout<<"OK! Good-by!"<<endl;                       //d
}
```

程序执行后 (在 VC6.0 环境下, 以下全同), 屏幕显示结果见题目处, 从略。

(3) 难点剖析及评注

1) 程序注解

a. 要往标准流对象 cout (即屏幕) 上输出信息, 必须包含头文件 “iostream.h”。以双斜杠 (//) 开头直到行末的全体 (符号) 称为注解。注解将被当作空格处理。

b. 任一个 C++ 程序都含有一个必须叫做 main 的主函数。

c. 显示出第一行字符, 其中的输出项为一个字符串, 靠最后的转义字符 (\n) 实现换行。

d. 显示最后一行, 格式控制符 endl 与转义字符 (\n) 所起的作用相同, 即实现换行。

2) 其他点评

① 注意每一语句均以分号作为结束。

② 该程序中没有使用任何变量, 根本不需要进行变量说明。

例 1-2 对两个整数进行指定运算

编程序, 从键盘输入两个正整数 i 和 j (int 型), 而后通过使用运算符 +、-、*、/、% 来计算 $2*i$ 与 $3*j$ 这两个数的和、差、积、商以及余数, 并将结果按一定的格式显示在屏幕上。如, 程序执行后可显示出如下形式的结果:

```
i, j = ? 20 2
```

```
(2*i) + (3*j) = 40 + 6 = 46
```

```
(2*i) - (3*j) = 40 - 6 = 34
```

```
(2*i) * (3*j) = 40 * 6 = 240
```

```
(2*i) / (3*j) = 40 / 6 = 6
```

```
(2*i) % (3*j) = 40 % 6 = 4
```

[解答]

(1) 实现要点与方法

1) 为了存放从键盘输入的两个整型 (int 型) 数据, 需要使用说明变量数据类型并可得到变量数据存储空间的说明语句。

2) 从 cin (即键盘) 上输入数据时, 可使用如下格式的所谓输入语句:

cin>> 变量名 1 >> 变量名 2 >> ... >> 变量名 n;

3) 可在 cout 语句中直接使用欲计算其结果的各表达式, 由 cout 负责计算并依次输出它们的值。

(2) 程序源码和结果

```
#include <iostream.h>
void main() {
    int i, j; //a
    cout<<"i, j = ? "; //b
    cin>>i>>j; //c
    cout<<"(2*i) + (3*j) = "<<2*i<<" + "<<3*j<<" = "<<(2*i)+(3*j)<<endl; //d
    cout<<"(2*i) - (3*j) = "<<2*i<<" - "<<3*j<<" = "<<(2*i)-(3*j)<<endl;
    cout<<"(2*i) * (3*j) = "<<2*i<<" * "<<3*j<<" = "<<(2*i)*(3*j)<<endl;
    cout<<"(2*i) / (3*j) = "<<2*i<<" / "<<3*j<<" = "<<(2*i)/(3*j)<<endl;
    cout<<"(2*i) % (3*j) = "<<2*i<<" % "<<3*j<<" = "<<(2*i)%(3*j)<<endl;
}
```

程序执行后的屏幕显示结果见题目处, 从略。

(3) 难点剖析及评注

1) 程序注解

a. 说明 i、j 为两个 int 型 (整型) 变量, 系统将为它们分配存储空间。变量名 i、j 由程序员任意起。

b. 屏幕提示进行数据输入。

c. 从键盘输入 2 个数据赋值给 i 与 j 变量 (2 个输入数据之间要以空格相分隔)。

d. 将结果按一定的格式显示在屏幕上。当输出项为表达式时, cout 会负责计算出表达式的值。

2) 其他点评

① 注意二整数相除的结果仍为整数, 是截断 (甩掉) 小数部分后取整的。

② 模运算 % 的结果是求出余数。

例 1-3 计算圆柱体的表面积与体积

编程序, 从键盘输入圆柱体的底圆半径 r 以及圆柱体的高 h, 而后计算出该圆柱体的表面积与体积并将结果显示在屏幕上。

[解答]

(1) 实现要点与方法

1) 说明 double 类型的变量 r 和 h 用来表示圆柱体的底圆半径和圆柱体的高, 并通过 cin

为它们输入值。

2) 圆柱体的表面积为二底圆面积加上圆柱体侧面积, 而圆柱体的体积为底圆面积乘以高。

3) 通过 cout 将计算结果值显示在屏幕上。

(2) 程序源码和结果

```
#include <iostream.h>

void main() {
    const double pai=3.1416;           //a
    double r, h, area, volume;         //b
    cout<<"Enter r and h:";
    cin>>r>>h;
    area = 2*pai*r*r + 2*pai*r*h;      //c
    volume = pai*r*r*h;                 //d
    cout<<"area="<<area<<"", volume="<<volume<<endl; //e
}
```

程序执行后, 屏幕显示结果为:

Enter r and h:1.1 10

area=76.7179, volume=38.0134

(3) 难点剖析及评注

1) 程序注解

a. 说明一个 double 型(双精度浮点型)有名常量 pai, 并将它的值设定为字面常量 3.1416, 该有名常量 pai 的值在程序中不可改变。

b. 说明四个 double 型变量, r 表示底圆半径, h 表示圆柱体的高, area 存放圆柱体的表面积, volume 存放圆柱体的体积。

c. 计算圆柱体的表面积 area: 为二底圆面积加上圆柱体侧面积。

d. 计算圆柱体的体积 volume: 等于底圆面积乘以高。

e. 通过 cout 将计算结果即变量 area 与 volume 的值显示在屏幕上。

2) 其他点评

也可以不设 area 与 volume 变量, 而像例 1-2 那样直接在 cout 语句中给出计算 area 与 volume 的表达式(如将本程序最后一行之 cout 句中的 area 与 volume 替换为对应的表达式)。另外, 为变量取名时, 虽然可以任意取, 但通常取作有标识意义的名字(如用 area 代表面积, 而用 volume 表示体积等), 以使所编写的程序具有更好的可读性与可维护性。

例 1-4 使用“简单变量方式”实现反序输出问题

从键盘输入 10 个 int 型数据, 先将它们存放在 10 个 int 型简单变量之中, 而后设法按输入的反顺序输出它们(以下简称“反序输出”)。

[解答]

(1) 实现要点与方法

1) 说明 10 个 int 型的简单变量并通过 cin 为各变量输入值。