



高等学校财经管理类专业计算机基础与应用规划教材

# 经济管理中C++程序设计

韩冬梅 主编 / 赵龙强 张勇 王炳雪 参编

非  
外  
借



清华大学出版社

高等学校财经管理类专业计算机基础与应用规划教材

教育部文科计算机基础教学指导委员会立项教材(2009年度)

# 经济管理中C++程序设计

韩冬梅 主编  
赵龙强 张勇 王炳雪 参编

清华大学出版社  
北京

## 内 容 简 介

本书在介绍计算机编程的思想、方法及算法分析的基础上,以 C++ 程序设计语言为教学实践语言,结合财经类院校的特点阐述了程序设计方法。本书面向实用、注重实践,采用深入浅出、通俗易懂的案例,着重培养读者程序设计能力,提高读者数据获取、数据处理及数据分析的能力。

全书内容分为 C++ 面向过程程序设计、面向对象程序设计及附录三大部分。第 1 章至第 6 章为面向过程程序设计,阐述了 C++ 语言基础、程序控制结构、函数、数组、构造数据类型、指针和引用。第 7 章至第 12 章为面向对象程序设计,阐述了类和对象、继承和组合、多态和虚函数、运算符重载、输入/输出流、C++ 的异常处理。

本书在阐述案例程序算法分析及编写程序的基础上,着重进行了程序分析和思考拓展,配有相关的练习题。本书适合高校非计算机专业教学用书,尤其是财经管理类专业教学使用,本书也适合作为从事 C++ 程序设计领域相关工作人员的参考书。

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

经济管理 C++ 程序设计/韩冬梅主编. —北京:清华大学出版社,2019

(高等学校财经管理类专业计算机基础与应用规划教材)

ISBN 978-7-302-51347-6

I. ①经… II. ①韩… III. ①C++语言—程序设计—应用—经济管理—高等学校—教材 IV. ①F2-39

中国版本图书馆 CIP 数据核字(2018)第 231190 号

责任编辑:黄 芝

封面设计:常雪影

责任校对:梁 毅

责任印制:杨 艳

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:24

字 数:583 千字

版 次:2019 年 1 月第 1 版

印 次:2019 年 1 月第 1 次印刷

印 数:1~1500

定 价:59.50 元

产品编号:075506-01



人类进入了信息时代,信息技术和数据技术的核心是计算机技术。计算机的迅速发展改变了人们学习和生活的方式,各行各业都在利用计算机进行信息管理,信息化进程促使人们正确掌握计算机的编程思想和方法。程序设计成为当代高校大学生必须掌握的一项基本技能,财经院校的学生从事金融投资等相关研究领域通常需进行数据分析,而全面深入的数据分析会涉及一些重要的模型,模型计算与计算机编程紧密相关,所以掌握计算机编程思想和方法能够更方便地拓展研究来深入分析相关数据。计算机程序设计教学是高校财经管理类专业必须掌握的一门基本技能,随着信息化的深入,计算机编程的重要性也逐步得到了财经院校财经管理类专业的重视,使财经院校的学生成为金融知识与计算机技术兼备的复合型创新型人才。

本书在阐述计算机编程的思想、方法及算法分析的基础上,以 C++ 程序设计语言作为教学实践语言。C++ 是面向计算机硬件基于机器系统软件开发的一种非常流行的高级语言,其功能强大,既支持面向过程的程序设计又支持面向对象的程序设计。本书培养读者熟练掌握 C++ 程序设计方法去解决实际问题的能力,使读者掌握相关数据类型及算法的基本逻辑,掌握程序编程的基本思想和方法,提高数据分析的能力,尤其是数据获取、数据处理、数据分析和信息决策的能力,掌握在财经管理类专业课程学习中使用编程方法,为后继学习计算机相关课程以及熟练使用专业软件包夯实基础。

本书用 12 章内容阐述了 C++ 程序设计。第 1 章至第 6 章为面向过程程序设计。第 1 章 C++ 语言基础,阐述了数据类型、表达式及程序的基本结构。第 2 章程序控制结构,阐述了程序设计三种控制方法(顺序结构、分支结构和循环结构)及模块化程序控制流程与算法。第 3 章函数,阐述了函数的调用及数据存储的性质及过程封装、递归思想与递归设计技术。第 4 章数组,阐述了数据的批量存储和批量处理。第 5 章构造数据类型,阐述了结构体数据类型和使用。第 6 章指针和引用,阐述了数据的直接访问与间接访问。第 7 章至第 12 章为面向对象程序设计。第 7 章类和对象,阐述了类的定义和对象构造、对象的析构及面向对象程序的特点和基本设计方法。第 8 章继承和组合,阐述了面向对象的重要特征代码重用。第 9 章多态和虚函数,阐述了面向对象程序设计中的多态性两种实现方法:静态绑定编译时的多态性和动态绑定运行时的多态性。第 10 章运算符重载,阐述了对类类型的对象执行内置运算符的操作。第 11 章输入/输出流,阐述了输入与输出是一连串字节流方式进行的,面向对象的输入与输出流通过输入与输出库来实现。第 12 章 C++ 的异常处理,阐述了程序

异常的处理机制。

本书由具有丰富计算机程序设计教育经验的教师编写。理论与实践相结合,面向实用、注重实践,采用深入浅出、通俗易懂的案例,着重培养读者的编程能力和实际应用能力。本书在阐述程序算法分析及编写程序的基础上着重进行程序分析和思考拓展,配有相关的练习题。本书适合作为高校非计算机专业教学用书,尤其是财经管理类专业教学使用。阅读本书不要求读者有任何编程经验。本书也适合作为从事C++程序设计领域相关工作人员的参考书。

本书第1~3章由韩冬梅编写;第4~6章由赵龙强编写;第7~9章由王炳雪编写;第10~12章由张勇编写。全书由韩冬梅教授审定和统稿。由于学识浅陋,水平有限,书中的不当之处恳请广大读者批评指正。

编 者

2018年5月



### 第 1 部分 面向过程程序设计

<b>第 1 章 C++ 语言基础</b> .....	3
1.1 引言 .....	3
1.2 编写 C++ 程序 .....	3
1.3 常量 .....	5
1.4 变量 .....	11
1.5 数据类型 .....	18
1.6 运算符和表达式 .....	20
本章小结 .....	29
思考题 .....	30
练习题 .....	30
<b>第 2 章 程序控制结构</b> .....	32
2.1 顺序结构 .....	32
2.2 分支结构 .....	33
2.2.1 if 条件语句 .....	34
2.2.2 switch 分支语句 .....	43
2.3 循环结构 .....	48
2.3.1 while 循环结构 .....	48
2.3.2 do-while 循环结构 .....	51
2.3.3 for 循环结构 .....	54
2.3.4 嵌套循环 .....	57
2.3.5 控制循环结构 .....	60
2.4 综合例子 .....	65
本章小结 .....	77
思考题 .....	79

练习题 .....	79
<b>第3章 函数</b> .....	<b>80</b>
3.1 函数概述 .....	80
3.2 函数的定义 .....	83
3.2.1 函数定义的形式 .....	83
3.2.2 定义与声明 .....	84
3.3 返回语句 .....	85
3.4 函数参数 .....	85
3.5 函数的调用 .....	86
3.5.1 函数的调用方式 .....	87
3.5.2 嵌套调用 .....	89
3.5.3 递归调用 .....	89
3.6 局部变量和全局变量 .....	92
3.6.1 局部变量 .....	92
3.6.2 全局变量 .....	93
3.7 内部函数和外部函数 .....	96
3.7.1 内部函数 .....	96
3.7.2 外部函数 .....	97
3.8 综合例子 .....	98
本章小结 .....	104
思考题 .....	104
练习题 .....	105
<b>第4章 数组</b> .....	<b>106</b>
4.1 一维数组 .....	106
4.1.1 一维数组的定义 .....	106
4.1.2 一维数组的初始化 .....	107
4.1.3 一维数组元素的使用 .....	108
4.1.4 数组元素的排序处理 .....	109
4.2 二维数组 .....	115
4.2.1 二维数组的定义与使用 .....	115
4.2.2 二维数组的初始化 .....	116
4.2.3 二维数组元素的使用 .....	116
4.3 字符数组与字符串 .....	118
4.3.1 字符数组的定义及初始化 .....	118
4.3.2 字符串 .....	119
4.3.3 字符串的输入与输出 .....	120
4.3.4 常用字符串处理函数 .....	121

4.3.5 string 字符串类型 .....	123
4.4 数组名作为函数参数 .....	127
本章小结 .....	129
思考题 .....	130
练习题 .....	130
<b>第 5 章 构造数据类型</b> .....	<b>132</b>
5.1 结构体数据类型 .....	132
5.1.1 结构体概念 .....	132
5.1.2 结构体声明 .....	132
5.1.3 结构体类型变量的定义方法 .....	134
5.1.4 结构体变量的初始化 .....	135
5.1.5 结构体变量的使用 .....	136
5.1.6 结构体数组 .....	138
5.2 共用体数据类型 .....	140
5.2.1 共用体类型的声明 .....	140
5.2.2 共用体类型的举例 .....	140
5.3 枚举数据类型 .....	142
5.3.1 枚举概念 .....	142
5.3.2 枚举类型的声明 .....	142
5.3.3 枚举举例 .....	142
5.4 typedef 声明新的类型名 .....	144
本章小结 .....	145
思考题 .....	145
练习题 .....	145
<b>第 6 章 指针和引用</b> .....	<b>147</b>
6.1 指针概念和指针变量 .....	147
6.1.1 指针概念 .....	147
6.1.2 指针变量 .....	149
6.2 使用指针作函数参数 .....	151
6.3 引用 .....	155
6.3.1 什么是变量的“引用” .....	155
6.3.2 “引用”作为函数参数 .....	155
6.4 数组与指针 .....	156
6.4.1 指向数组元素的指针 .....	156
6.4.2 用指针变量作为函数参数 .....	157
6.4.3 字符串与指针 .....	159
6.5 指针函数 .....	160

6.6	指针数组及指向指针的指针 .....	161
6.6.1	指针数组 .....	161
6.6.2	指向指针的指针 .....	162
6.7	指向函数的指针 .....	163
6.8	用结构体类型的指针变量和引用作为函数参数 .....	164
6.8.1	指向结构体变量的指针 .....	164
6.8.2	应用举例 .....	165
6.9	链表结构 .....	165
6.9.1	链表的基本概念 .....	165
6.9.2	使用 new 和 delete 运算符进行动态分配和撤销存储空间 .....	166
6.9.3	链表举例 .....	167
	本章小结 .....	175
	思考题 .....	175
	练习题 .....	176

## 第 2 部分 面向对象程序设计

<b>第 7 章</b>	<b>类和对象 .....</b>	<b>179</b>
7.1	从面向过程到面向对象 .....	179
7.2	什么是类,什么是对象,对象和类之间的关系 .....	180
7.3	类的声明和对象的创建(定义) .....	184
7.4	类的成员变量和成员函数 .....	185
7.5	类的信息隐藏机制及面向对象设计中的几个名词 .....	187
7.6	成员选择符和对象的动态创建(定义) .....	189
7.7	类 class 和结构体 struct 区别 .....	191
7.8	通过引用来传递和返回类对象 .....	193
7.9	构造函数 .....	196
7.10	参数初始化表 .....	200
7.11	析构函数 .....	200
7.12	常量指针 this .....	203
7.13	类与 new 和 delete 操作符 .....	204
7.14	类与 static 关键字 .....	206
7.15	友元函数和友元类 .....	208
	本章小结 .....	213
	思考题 .....	214
	练习题 .....	214
<b>第 8 章</b>	<b>继承和组合 .....</b>	<b>216</b>
8.1	继承的概念及语法 .....	216

8.2	继承方式 .....	218
8.3	改变基类成员在派生类中的访问属性 .....	220
8.4	名字隐藏 .....	221
8.5	间接继承 .....	222
8.6	继承机制下的构造函数 .....	224
8.7	派生类构造函数调用规则 .....	228
8.8	继承机制下的析构函数 .....	230
8.9	多继承 .....	231
8.10	虚基类 .....	234
8.11	组合 .....	236
	本章小结 .....	239
	思考题 .....	240
	练习题 .....	240
<b>第 9 章</b>	<b>多态和虚函数 .....</b>	<b>242</b>
9.1	基类指针 .....	243
9.2	多态前提条件 .....	243
9.3	虚成员函数 .....	246
9.4	虚成员函数表 vtable .....	249
9.5	虚析构函数 .....	250
9.6	静态成员函数不能声明为虚函数 .....	253
9.7	重载、覆盖和遮蔽 .....	253
9.8	抽象基类和纯虚成员函数 .....	259
9.9	应用实例 .....	261
	本章小结 .....	269
	思考题 .....	269
	练习题 .....	269
<b>第 10 章</b>	<b>运算符重载 .....</b>	<b>273</b>
10.1	运算符重载方法 .....	273
10.2	运算符重载的规则 .....	275
10.3	运算符函数 .....	277
10.3.1	重载为类的成员函数 .....	277
10.3.2	重载为类的友员函数 .....	278
10.4	双目运算符的重载 .....	282
10.5	单目运算符的重载 .....	286
10.6	重载流插入运算符和流提取运算符 .....	288
10.6.1	重载流插入运算符“<<” .....	289
10.6.2	重载流提取运算符“>>” .....	291

10.7	不同类型数据间的转换 .....	293
10.7.1	C++标准类型数据间的转换 .....	293
10.7.2	转换构造函数(由数所转换成类的对象) .....	293
10.7.3	类型转换函数(类的对象转换为一个其他类型的数据) .....	295
	本章小结 .....	298
	思考题 .....	299
	练习题 .....	299
<b>第 11 章</b>	<b>输入/输出流</b> .....	<b>300</b>
11.1	文件、流及 C++ 的流类库 .....	300
11.1.1	流类库的优点 .....	300
11.1.2	文件与流的概念 .....	301
11.1.3	输入/输出的含义 .....	301
11.1.4	C++ 的流类库 .....	302
11.2	C++ 输入/输出的格式控制 .....	303
11.2.1	输入/输出格式控制符 .....	303
11.2.2	流操作子 .....	306
11.2.3	C++ 标准设备的输入/输出(cin、cout、cerr、clog、>>和<<) .....	308
11.3	磁盘文件的输入和输出 .....	312
11.3.1	文件的打开与关闭 .....	312
11.3.2	使用插入与提取算符对磁盘文件进行读写 .....	314
11.3.3	使用类成员函数对文件流(类对象)进行操作 .....	316
11.4	text 文件与 binary 文件 .....	320
11.4.1	按用户设置的文件形式进行读写 .....	320
11.4.2	对数据文件进行随机访问 .....	323
11.5	字符串流 .....	325
	本章小结 .....	328
	思考题 .....	328
	练习题 .....	328
<b>第 12 章</b>	<b>C++ 的异常处理</b> .....	<b>330</b>
12.1	异常处理方法 .....	330
12.1.1	异常处理机制 .....	330
12.1.2	编程规则 .....	333
12.2	异常处理的接口声明 .....	337
12.3	异常处理需注意问题 .....	337
12.4	C++ 异常传递的方法 .....	338
12.4.1	传值 .....	338
12.4.2	传指针 .....	339

12.4.3  引用传递 .....	340
本章小结 .....	340
思考题 .....	341
练习题 .....	341

### 第3部分 附 录

附录 A 编码风格和准则及算法表示方法 .....	345
A.1 编码风格和准则 .....	345
A.2 算法表示方法及解题步骤 .....	345
A.3 C++程序中全局变量的声明及函数声明与调用 .....	349
A.4 一个C++程序由多个文件组成 .....	350
附录 B C++常用库函数 .....	352
B.1 常用数学函数表 .....	352
B.2 常用字符串处理函数表 .....	352
B.3 其他常用函数表 .....	353
B.4 输入/输出的成员函数表 .....	354
附录 C 编程环境安装和调试 .....	355
C.1 Visual C++ 2010 Express 安装 .....	355
C.2 Visual C++ 2010 Express 注册 .....	357
C.3 Visual C++ 2010 Express 使用和程序调试 .....	359
附录 D 运算符优先级表及 ASCII 码表 .....	365
D.1 运算符优先级表 .....	365
D.2 ASCII 码表 .....	366
参考文献 .....	370

## 第1部分

# 面向过程程序设计



# 第 1 章

## C++语言基础

软件运行离不开数据。日常生活中我们将数字、文字等数据输入计算机中,然后用各种计算机程序处理这些数据。数据有常量和变量之分,常量是指在程序运行过程中值保持不变的量;变量是指在程序运行过程中,其值可以随时改变的量。常量和变量都属于某种数据类型。此外,处理数据需要进行运算,C++语言提供了丰富的运算符,例如进行算术计算、比较值的大小、修改变量、逻辑地组合关系等。本章将介绍这些 C++ 语言编程的基础组成成分,包括数据类型、运算符和表达式等。

### 1.1 引言

20 世纪发明的计算机到 21 世纪有了飞跃性的发展,各种形式的智能设备紧密联系,为企业生产和人民生活带来了便利。在不同场合使用的计算机(或智能设备),都由硬件和软件组成。如果没有软件,那只不过是一个空壳子。软件也就是计算机程序是指一组指示计算机每一步动作的指令,通常用某种程序设计语言编写,运行于某种目标体系结构的计算机上。打个比方,一个程序就像一个用汉语(程序设计语言)写下的红烧肉菜谱(程序),用于指导懂汉语和烹饪手法的人来做这个菜。计算机程序是通过在计算机内存中开辟一块存储空间,并用一个语句序列不断修改这块存储空间上的内容,最终得到问题的解答来解决实际问题的。

人和计算机交流,也需要解决语言问题,需要创造一种计算机和人都识别的语言,这就是计算机语言。通过使用跟人类自然语言接近的高级程序设计语言在程序中表达我们的旨意,程序则负责调度各种计算机资源完成我们下达的旨意。计算机语言通过定义变量的形式给出了申请内存的方式,并通过表达式和赋值语句给出了对内存中的数据进行运算和修改的方法,通过分支和循环语句提供了用不同方式安排语句序列的能力。大部分计算机语言还提供了基础函数库完成一些常用的计算和数据处理的功能。本书将带领大家系统地学习 C++ 语言。

### 1.2 编写 C++ 程序

#### 1. 一个简单的 C++ 程序

我们经常使用计算机来进行各种生活或工作中的数学运算,下面我们就来展示一个用

C++语言编写的简单计算机程序。

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      float a,b;
6      char op;
7      cout <<"依次输入第一个运算数、运算符(+, -, *, /)、第二个运算数" << endl;
8      cin >> a >> op >> b;
9      switch(op)
10     {
11         case '+':
12             cout << a + b << endl;
13             break;
14         case '-':
15             cout << a - b << endl;
16             break;
17         case '*':
18             cout << a * b << endl;
19             break;
20         case '/':
21             if(b == 0)
22             {
23                 cout <<"第二个数不能是 0" << endl;
24                 return 1;
25             }
26             cout << a/b << endl;
27             break;
28         default:
29             cout <<"您的输入有误!";
30     }
31     return 0;
32 }

```

大家刚开始学习看不懂这段程序没关系,我们大致地从上往下分析一下这段代码的含义。首先是第一行 `#include <iostream>`,意味着我们在写这段程序时,包含了另外一个已经写好的 `iostream` 文件,这个文件可以让我们和计算机进行输入输出。第 2 行代码可以让我们使用名为 `std` 的 `namespace` 中的标识符,如程序中的 `cin`、`cout`。第 3 行代码是程序的主函数,它是程序执行的入口。第 5 行和第 6 行定义了三个变量,分别表示第一个运算数、第二个运算数和运算符。第 7 行是向屏幕输出一行文字,进行用户提示的。第 8 行是接收用户输入的数字和符号,并赋值给相应的变量。第 9 行是一个开关逻辑,根据你输入运算符的不同,决定接下来执行哪个运算,也就是执行下面的哪一个 `case` 语句。第 11 行到第 29 行就是分情况进行相应的具体计算了,稍微特殊的是除法/的情况,会判断第二个运算数是否为 0。第 31 行程序执行到这里就正常结束了。

## 2. 用 C++ 进行输入和输出

计算机语言比如 C++ 可以实现让我们人类与机器进行交互,交互最基本方式体现为我们可以通过键盘向其输入数据,计算机在进行处理运算之后可以通过屏幕输出结果。C++ 语言已经实现了输入输出功能,我们要做的就是正确地调用。

输入和输出是将数据从计算机外部设备传送到计算机内存或反向传送的过程,数据如流水一样从一处流向另一处。C++ 中形象地将此过程称为流(Stream)。C++ 提供了用于输入输出的 iostream 类库,iostream 这个单词由 3 个部分组成,即 i-o-stream,意为 input 输入-output 输出-stream 流。

类库 iostream 包含了对输入输出流进行操作所需的基本信息,因此大多数 C++ 程序都包括 iostream。在 iostream 文件中定义了我们常用的两个流对象,cin 对应来自键盘的标准输入流,cout 对应发往屏幕的标准输出流。cin 通常与 >> 符号结合在一起,用于将键盘输入的数据分配给相应的变量;而 cout 通常与 << 符号结合在一起,用于将数据向屏幕进行输出。有一个简单而形象的方法理解运算符 << 和 >> 的作用:它们指出了数据移动的方向。

例如 >> a 箭头方向表示把数据放入 a 中。而 << a 箭头方向表示从 a 中拿出数据。大家可以参考一下先前的例子。

## 1.3 常量

常量是指在程序运行过程中,其值不能被改变的量,如数据 2, -56, 3.14, 0.22, 3.16E7 和 'hello' 都是常量。常量可分为整型常量、浮点型常量、布尔常量、字符常量、字符串常量和符号常量。

### 1. 整型常量

例如整数 6, -45 和 978 都是整型常量。C++ 把不含小数点和指数的数当作整数,因此 7.0 和 0.314159e1 则不是整型常量。大多数整型常量看作 int 类型。如果整数特别大,则有不同处理。整型常量分为八进制整型常量、十进制整型常量和十六进制整型常量三种表示形式。其表示方法如表 1.1 所示。

表 1.1 整型常量的表示方法

整型常量	表示方式	正确例子	错误例子
十进制整型	由数字 0~9 组成的数字串	0、1、364、28、-34	045(不能有前导 0)、45D(含有非十进制数码)
八进制整型	由数字 0~7 组成的数字串,并且一定要以数字 0 开头	0111(十进制 73)、011(十进制 9)、0123(十进制 83)	256(无前缀 0)、03A2(包含了非八进制数码)、-0127(出现了负号)
十六进制整型	由 0~9、a~f 或 A~F 中的数字或英文字母组成的数字串,并且一定要以 0x 或 0X 开头	0x11(十进制 17)、0Xa5(十进制 165)、0x5a(十进制 90)	6B(无前缀 0X)、0X2H(含有非十六进制数码)