

· 高等学校计算机基础教材精选 ·

C++程序设计

(第2版)

刘加海 杨 锐 主 编
张银南 吴建敏 副主编

武汉工业学院图书馆



01415163



清华大学出版社

· 高等学校计算机基础教育教材精选 ·

C++程序设计

(第2版)

刘加海 杨 锴 主 编
张银南 吴建敏 副主编



清华大学出版社
北京

内 容 简 介

本书适合作为直接讲授 C++ 程序设计课程的教学用书,不需要读者具有 C 程序设计的基础。本书把 C 程序设计的基本知识结合在类中讲解,使面向对象的思想贯穿于整个教材,能帮助读者尽快掌握面向对象的编程思想,提高面向对象的编程能力。

本书共分 14 章,内容包括 C++ 程序设计入门、C++ 程序的文件组织与基本运算符、循环程序设计、分支程序设计、函数、数组与指针、指针与函数、类与对象、运算符重载、继承与多态性、I/O 流与文件、模板和异常、可视化程序设计。

本书内容通俗易懂、言简意赅、重点突出。本书内容的安排循序渐进、深入浅出,以具体实例来分析和阐明 C++ 程序设计中面向对象的方法与应用。为加深读者对程序设计思想的理解,每章都给出了与教材同步的思考题与上机练习题。清华大学出版社还出版了与本书配套的《C++ 程序设计实验、辅导与习题解答》,书中给出了 17 个 C++ 实验、C++ 编程要点及本书所有习题的解答。

本书作者根据在浙江大学多年讲授 C++ 程序设计的教学经验,结合大量实例,系统地讲述了 C++ 程序设计的主要知识点、编程特点与编程方法。本书适合作为高等学校理工科各专业的学生学习面向对象程序设计的教材或参考书。

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

C++ 程序设计/刘加海,杨锴主编. —2 版. —北京:清华大学出版社,2013.8

高等学校计算机基础教育教材精选

ISBN 978-7-302-33025-7

I. ①C… II. ①刘… ②杨… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 145968 号

责任编辑:焦虹

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:李红英

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

出版说明

—— 高等学校计算机基础教材精选 ——

在教育部关于高等学校计算机基础教育三层次方案的指导下,我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践,全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验,取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快,目前我国的高等教育事业正面临着新的发展机遇,但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要,进一步推动我国高等学校计算机基础教育事业的发展,我们在全中国各高等学校精心挖掘和遴选了一批经过教学实践检验的优秀教学成果,编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次,包括面向各高校开设的计算机必修课、选修课,以及与各类专业相结合的计算机课程。

为了保证出版质量,同时更好地适应教学需求,本套教材将采取开放的体系和滚动出版的方式(即成熟一本、出版一本,并保持不断更新),坚持宁缺毋滥的原则,力求反映我国高等学校计算机基础教育的最新成果,使本套丛书无论在技术质量上还是出版质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作,在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度,以适应高校计算机基础教育课程层次化、多样化的趋势,从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来,把自己的教学成果与全国的同行们分享;同时也欢迎广大读者对本套教材提出宝贵意见,以便我们改进工作,为读者提供更好的服务。

我们的电子邮件地址是: jiaoh@tup.tsinghua.edu.cn。联系人: 焦虹。

清华大学出版社

第 2 版前言

C++ 程序设计(第 2 版)

C++ 程序设计语言具有面向对象的属性,有着丰富的数据结构、灵活的控制语句、清晰的程序结构、良好的可移植性,运行效率高、安全可靠,易于开发与维护大型系统程序。目前很多高校的理工科专业都将 C++ 程序设计作为首选的程序设计课程。

目前 C++ 程序设计的教材很多,虽各有自己的特点,但存在的问题也是显而易见的。存在的主要问题是:

(1) 需要首先学习 C 程序设计,然后再学习 C++ 程序设计。这种情况占用的学时数较多,很多高校都难以满足;而且先学习 C 程序设计再学习 C++ 程序设计,面向对象的思想很难建立。

(2) 目前也有一些 C++ 程序设计的教材将 C 程序设计方面的内容安排在前半部分,后半部分为面向对象的内容;从而使 C 与 C++ 的内容分离,C 程序设计的内容过于简单。学生学习了 C++ 程序设计后,只知道了大概,无法建立面向对象的思想,既没有学好 C 程序设计也不会 C++ 程序设计。

(3) 教学中把程序设计课程讲解为程序语法课程,这在国内很普遍。课程中着重论述 C++ 的基本概念,而不是从面向对象的程序设计入手,学生程序设计的能力很难提高。

笔者在十多年前就打算写一本融合 C、C++ 与可视化程序设计方面的书,虽经近几年的努力,终于完成了书稿,但完成的书稿与原设想的风格相差较远,此事就耽搁下来。近年来浙江大学不断加大教学改革的力度,一方面不断减少教学时数,另一方面又需要学生掌握 C++ 程序设计的基本知识与程序设计的能力。浙江大学计算机学院将此课程作为精品课程来建设,使得我们又一次思考如何把 C 与 C++ 的内容尽可能较好地结合在一起,使学生既掌握面向对象的思想又不削弱程序设计的能力。因而本书一开始就引入了类的概念,把 C 与 C++ 的内容融合在类中,减少了较空泛的概念讲解,增加了程序设计举例后的思考与上机调试的项目。

第 2 版在第 1 版的基础上做了以下修改:

(1) 把第 1 版中的第 10 章分成“继承”与“多态性”两章,丰富了实例讲解,逻辑性得到了加强。把第 1 版中的第 11 章调整到第 13 章,理由是继承与多态性是面向对象程序设计中的重要内容,适当提前便于学生掌握;而模板和异常处理与整个内容体系不完全一致,它是对程序编写的完善与容错的改进,因而放在较后面进行介绍。

(2) 对第 1 版中的第 1、8、9、10、11 章的内容修改较大,文字叙述逻辑性更强,将原有实例基本改写成用类对象来设计。

(3) 对第 1、2、3 章中的某些重要内容给出了脚注,并重写了第 3、4、5、6、7 章中的例题。例题的设计大多采用面向对象的方法,有些不宜改写为类设计的,在程序中都以函数调用形式出现。一方面是为了程序的可读性,另一方面是为了与类中的成员函数相对应,在学习中对函数编写具有更好的迁移性。

(4) 模板与异常、I/O 流与文件、可视化程序设计初步与第 1 版相比变化较小。

(5) 改正了书中原有的一些错误。

本书的主要特点

涵盖 C 与 C++ 程序设计的主要内容。全书以类与对象为主线,把 C 的基本概念完全融合在类中。全书简单易懂、内容广泛,将各种基础概念都融合在程序设计中,对程序分析、设计、调试、运行都给出了详细的步骤,并针对例题给出进一步的思考题与上机练习题,可以有效地帮助学生理解程序设计的思想,提高程序设计的能力。

本书的读者对象

本书可作为高等学校理工科专业的学生学习面向对象程序设计的教材或参考书。书中内容的安排循序渐进、深入浅出,以具体的实例来分析和阐明 C++ 程序设计中面向对象的方法与应用。为加深读者对程序设计思想的理解,每章都给出了与教材同步的思考题与上机练习题。

本书的主要内容

第 1 章 C++ 程序设计入门。主要介绍面向对象的特征、类的概念及类的定义、对象的概念及定义、对象对类成员的引用、C++ 程序的基本构成以及在 Visual C++ 环境下 C++ 程序的调试方法。

第 2 章 C++ 程序的文件组织与基本运算符。主要介绍 C++ 程序的基本构成、C++ 文件、函数的原型、语句与常用运算符与表达式。

第 3 章 循环程序设计。主要介绍循环的概念、用 while、do-while 及 for 构成的循环语句,着重讨论 while、for 循环的执行过程以及循环的嵌套及循环的应用。

第 4 章 分支程序设计。主要介绍 if 语句的三种形式,即 if、if-else、if-else if 语句的执行流程及它们的应用。讨论了 if 嵌套的应用、switch 语句的执行流程及在实际中的应用。

第 5 章 函数。主要介绍函数的概念、函数的参数传递、默认的函数参数、函数重载与内联函数的概念、变量的不同存储类型在函数模块中的作用范围与生存期及函数的递归调用。

第 6 章 指针与数组。主要介绍指针变量对一维数组元素的引用方法、指针在一维数组中的应用、指针变量对字符串的引用、数组指针在二维数组中的应用、运算符 new 和 delete 的使用。

第9章 运算符重载。主要讲解运算符在对象之间运算时的含义可以由成员函数或友元函数进行重新解释,着重讲解常用运算符的重载方法及其应用。

第10章 继承。主要讲述继承与派生的基本概念、多继承中派生类的构造函数的定义及多继承中的二义性问题,讲解了虚基类的应用。

第11章 多态性。主要讲述面向对象程序设计中实现多态性的方法,讲解虚函数与多态性的关系、纯虚函数与抽象类在面向对象程序设计中的应用。

第12章 I/O流与文件。讲解C++流的概念,如何使用ios类的成员函数及操作符实现格式化输入与输出,论述了文件的概念及对文本文件、二进制文件的操作。

第13章 模板和异常处理。主要讲解函数模板、类模板的基本概念及应用、异常处理的基本思想与程序设计。

第14章 可视化程序设计初步。主要讲解如何通过继承MFC来生成应用程序中的派生类,实现图形界面的程序设计。

在章节前加*的章节,读者可以暂时忽略,不影响对本书的学习。

本书中提供了大量的例题与思考题、上机练习题及课后习题。为了更好地理解各章的内容,在学习中一定要及时上机调试程序,并修改一些类与语句,思考结果并验证。希望读者能够将阅读本书和自己的实践结合起来,动手思考、设计所有的思考和实验题。清华大学出版社还出版了与本书配套的《C++程序设计实验、辅导与习题解答》,此书给出了17个实验及教材中全部习题的解答。

本书作者对C与C++程序设计有着丰富的应用研究和教学经验。本书由浙江大学城市学院、计算机学院刘加海,浙江大学城市学院杨锴、严冰,浙江大学计算机学院季江民、陈建海,杭州浙大灵通科技有限公司吴建敏,浙江科技学院张银南,浙江商业职业技术学院孔美云等教师编写。

作者对给予大力支持的浙江大学计算机学院表示衷心的感谢。

由于编者水平有限,书中难免有疏漏与不足之处,敬请各位专家和广大读者批评指正。需要教学课件、源程序等可发电子邮件到Liujh@zucc.edu.cn。

编 者

目录

C++ 程序设计(第 2 版)

第 1 章 C++ 程序设计入门	1
1.1 C++ 类	1
1.1.1 类的概念	1
1.1.2 C++ 中类的定义	3
1.1.3 类成员数据	5
1.1.4 类成员函数的定义方法	6
1.1.5 类对象的定义及对象对成员函数的引用方法	8
1.2 C++ 的标准输入输出流对象	9
1.2.1 标准输入输出流对象的基本应用	9
1.2.2 输入输出流对象的成员函数及应用	11
1.3 常量与变量	15
1.3.1 整型常量	16
1.3.2 实型常量	17
1.3.3 字符常量	17
1.3.4 变量的数据类型及其定义	17
1.3.5 整型变量	18
1.3.6 实型变量	19
1.3.7 字符变量与字符串	19
1.3.8 变量声明的位置	22
1.4 类对象初步	23
1.4.1 类对象的概念	23
1.4.2 类对象的定义	23
1.4.3 类成员函数中的构造函数与析构函数	27
1.5 Visual C++ 6.0 环境下的程序调试	29
1.5.1 Visual C++ 6.0 源程序编辑、编译、连接、运行过程	29
1.5.2 打开已存在的文件	34
1.5.3 C++ 程序的调试	35
1.6 面向对象编程方法的基本特征	40
1.6.1 抽象	40

1.6.2	封装	40
1.6.3	继承	41
1.6.4	多态性	43
习题		45
第2章	C++ 程序的文件组织与基本运算符	46
2.1	C++ 程序的多文件结构	46
2.2	C++ 中的函数	49
2.2.1	函数原型	49
2.2.2	函数体定义	50
2.2.3	函数的调用方式	50
2.3	C++ 语句	51
2.4	运算符	52
2.4.1	算术运算符	53
2.4.2	关系运算符	54
2.4.3	逻辑运算符	54
2.4.4	位运算符	56
2.4.5	引用	56
习题		58
第3章	循环程序设计	60
3.1	while 循环程序设计	60
3.2	do-while 循环程序设计	66
3.3	for 循环程序设计	68
3.3.1	for 循环结构	68
3.3.2	for 语句的几种变形	72
3.4	break 语句和 continue 语句	73
3.5	循环嵌套的应用	76
习题		81
第4章	分支程序设计	83
4.1	if 语句的应用	83
4.2	if-else 语句的应用	86
4.3	if-else if 语句的应用	92
4.4	if 嵌套语句的应用	95
4.5	switch 的应用	96
习题		102

第 5 章 函数及其应用	104
5.1 函数的基本概念	104
5.2 系统函数的应用	106
5.3 自定义函数	107
5.3.1 函数定义的形式	107
5.3.2 函数的参数	108
5.4 默认的函数参数	112
5.5 函数重载	114
5.6 内联函数	117
5.6.1 内联函数的声明方法	117
5.6.2 内联函数的特点	117
5.7 域分辨操作符::	119
* 5.8 变量存储类型与变量生存期、作用域	120
5.8.1 auto 存储类型的变量与作用范围	121
5.8.2 static 存储类型的变量与作用范围	122
5.8.3 register 存储类型的变量与作用范围	124
5.8.4 extern 存储类型的变量与作用范围	124
5.9 函数的嵌套与递归调用	125
5.9.1 函数的嵌套调用	125
5.9.2 函数递归调用	127
习题	129
 第 6 章 指针与数组	 139
6.1 一维数组	139
6.1.1 一维数组的定义	140
6.1.2 一维数组的引用、初始化与赋值	141
6.2 二维数组	152
6.2.1 二维数组的定义	152
6.2.2 二维数组的元素表示、初始化与赋值	152
6.2.3 二维数组可作为一维数组来使用	155
6.3 指针的基本概念	156
6.3.1 指针	156
6.3.2 指针间的运算	159
6.3.3 指针与 const 限定符	161
6.3.4 类与 const 限定符	162
6.4 一维数组与指针	164
6.5 字符串与字符指针变量	167
6.5.1 字符数组与字符串	167

6.5.2	指针变量与字符串	170
6.6	数组指针	171
6.7	指针数组	173
6.7.1	指针数组的性质	173
6.7.2	指针数组的初始化	174
6.8	运算符 new 和 delete 与指针	175
	习题	178
第 7 章	指针与函数	183
7.1	指针与函数参数	183
7.2	指向函数的指针	190
7.3	返回值为指针的函数	195
7.4	命令行参数	199
7.4.1	命令行参数的概念	199
7.4.2	命令行参数的表示方法	199
	习题	201
第 8 章	类与对象	203
8.1	类的构造函数	203
8.1.1	构造函数的特点	204
8.1.2	默认参数的构造函数	210
8.2	类的析构函数	211
8.2.1	析构函数的特点	211
8.2.2	构造函数、析构函数调用顺序	212
8.3	拷贝构造函数	214
8.3.1	使用已有对象初始化另一个对象	215
8.3.2	类对象作为函数的参数	218
8.3.3	类对象作为函数的返回值	219
8.4	类对象的应用	223
8.5	类静态成员	234
8.5.1	类的静态数据成员	234
8.5.2	类的静态成员函数	237
8.6	类的友元	240
8.6.1	友元函数	240
8.6.2	友元成员	244
8.6.3	友元类	247

8.7.2	常成员函数	252
8.7.3	常数据成员	253
8.7.4	常引用	254
8.8	容器类	255
* 8.9	类与结构	257
8.10	对象数组与对象指针	259
* 8.10.1	对象数组	259
* 8.10.2	指向类对象的指针	261
* 8.10.3	指向类成员的指针	263
8.10.4	this 指针	265
习题		267

第 9 章 运算符重载 277

9.1	运算符重载的基本概念	277
9.1.1	C++ 中可重载的运算符	278
9.1.2	运算符重载的定义形式	279
9.2	成员函数重载运算符	279
9.3	友元函数重载运算符	283
9.4	成员函数运算符与友元运算符函数的比较	285
9.5	单目运算符的重载	287
9.6	赋值运算符的重载	293
9.7	二元运算符的重载	295
9.8	重载运算符()	302
习题		303

第 10 章 继承 308

10.1	继承与派生	308
10.2	继承访问控制	310
10.2.1	继承	310
10.2.2	公有(public)继承	311
* 10.2.3	私有(private)继承	313
* 10.2.4	保护继承(protected)	316
10.3	派生类的构造函数的设计	317

10.4.2	多继承中派生类的构造函数与析构函数	327
10.5	多继承中的二义性问题	333
10.6	虚基类	335
	习题	339
第 11 章	多态性	347
11.1	多态性的概念	347
11.2	虚函数	350
11.2.1	虚函数的定义	350
11.2.2	虚函数的调用	351
11.2.3	虚函数和重载函数的区别	355
11.3	纯虚函数与抽象类	357
11.3.1	纯虚函数	357
11.3.2	抽象类	358
* 11.4	多态性的异质单向链	361
	习题	367
第 12 章	I/O 流与文件	374
12.1	C++ 流的概念	374
12.1.1	streambuf 类	375
12.1.2	ios 类	375
12.2	用 ios 类的成员函数实现格式化输入与输出	376
12.2.1	I/O 状态标志字	376
12.2.2	ios 类中用于控制输入输出格式的成员函数	377
12.2.3	ios 类中的其他成员函数	381
12.3	用 I/O 操纵符实现格式化输入与输出	382
12.3.1	I/O 操纵符	382
12.3.2	用户自定义操纵符	385
12.4	文件的操作	386
12.4.1	文件的操作过程	386
12.4.2	定义文件流对象	387
12.4.3	文件的打开与关闭	387
12.4.4	文件的操作方式	389
12.4.5	文本文件应用举例	390
12.4.6	二进制文件的操作	396

12.5.2 输入运算符“>>”重载	404
习题	407
第 13 章 模板和异常处理	413
13.1 模板	413
13.1.1 函数模板	414
13.1.2 类模板	420
13.2 异常处理	432
13.2.1 异常处理的基本思想	432
13.2.2 异常处理的实现	433
13.2.3 异常生命周期	437
13.2.4 异常规格说明	438
13.2.5 异常处理中的构造与析构	439
习题	440
第 14 章 可视化程序设计初步	442
14.1 Windows 程序设计基本概念	442
14.1.1 Windows 消息	442
14.1.2 消息的种类	443
14.2 Windows 程序设计举例	444
14.2.1 CWinApp 类	445
14.2.2 CFrameWnd 类	445
14.2.3 程序举例——框架编程实现	446
14.2.4 应用程序举例——消息框编程实现	448
14.2.5 应用程序举例——菜单编程实现	451
14.2.6 应用程序举例——图形、文字、图像编程实现	456
14.2.7 应用程序举例——对话框程序的实现	461
14.2.8 应用程序举例——通用对话框程序设计	463
习题	477
附录 A ASCII 表	480
附录 B 运算符及其优先级汇总表	482
附录 C C++ 语言的保留字	484
附录 D 常用库函数	485

本章要点

- 面向对象的特征;
- 类的概念及类的定义;
- 类数据成员的构成及成员函数的定义;
- 对象的概念及定义;
- 对象对类成员的引用;
- C++ 程序的基本构成;
- Visual C++ 环境下 C++ 程序的调试方法。

本章难点

- 如何理解类与对象;
- 类中的数据封装性与数据接口;
- 类中数据访问特性;
- 如何建立面向对象的编程思想。

本章导读

计算机高级语言可分为面向过程的程序设计语言与面向对象的程序设计语言。C 是面向过程的程序设计语言,而 C++、Java 等是面向对象的程序设计语言。在面向对象的程序设计中,由对象的特征抽象为类,类将数据(特征)与函数(操作或方法)封装在一起,把对象的特征和操作作为一个整体来处理。由类通过实例化后成为对象,对象通过调用它的成员函数产生其功能。在学习面向对象的程序设计时,要牢记该牢记对象的概念、类的定义,以及对象如何调用成员函数。本书 C++ 的程序的调试环境是 Visual C++ 6.0 集成环境,希望读者掌握在 Visual C++ 6.0 环境下 C++ 程序的调试方法。

1.1 C++ 类

1.1.1 类的概念

在现实世界中,类是对一组相同特征对象的抽象描述。例如,作为教师对象,有张三、李四、王五等。每个对象有不同的性别、职称和学历等特征,也有讲授不同学科课程的能

力等方法。教师类是对教师这类对象所具有的共同属性(特征)方法(操作)集合的抽象描述。教师这类对象应具有性别、职称和学历特征,应有讲授某学科本科生课程的能力;有些教师还有做某学科研究生导师的能力。

在面向对象方法中,类是具有相同操作功能和相同的数据格式或属性的对象的集合。它是针对某一类对象的特性描述,如图 1.1 所示。

例如,草、花、树、水稻可以归为植物类。它们具有植物的特性,并且草、花、树、水稻是植物类的对象。同理,马、狗、蛇、老虎属于动物类。它们具有动物的所有特性,并且马、狗、蛇、老虎是动物类的对象。也就是说运用抽象的方式描述这类对象。类通过对象的公共特征和操作抽象而成,它规定了这些对象的公共属性和方法。对象是类的一个实例。

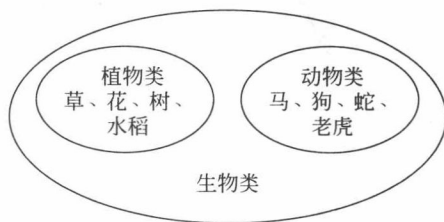


图 1.1 具有相同特征对象的抽象描述——生物类

例如要在程序中描述一种动物:猫。需要在程序中设计一种数据类型,这种数据类型能够描述猫这种动物的特性与方法。比如:猫有四只脚、一条尾巴以及猫的叫声、猫抓老鼠等方法。因而,当程序中要描述一只小花猫时,可用这种猫类(数据类型)产生一只名叫小花猫的对象。同理,苹果是一个类,而放在桌上的那个苹果则是一个对象。

类具有抽象性。对象和类的关系相当于一般的程序设计语言中变量和变量类型的关系。面向对象方法中程序设计的主体是类。类是对具有相同属性和方法的一个或多个对象的抽象描述。

类具有继承性。子类可以在继承父类所有属性和方法的基础上,再增加自己特有的属性和方法,或在某些操作中与父类有不同的方法,因此类具有继承性。类的这种特性使面向对象方法设计的模块比传统方法设计的模块具有更高的重复使用率,从而极大地提高了面向对象方法设计的软件系统的可维护性和系统升级能力。

在面向对象的程序设计中,首先定义一个类,类中包含对象的特征即类的成员数据、对象的方法即类的成员函数。类具有模块性,类定义完成后,就可以定义类的对象。

一般认为,对象是现实世界中的一个实际存在的事物,它可以是有形的也可以是无形的。对象具有自己的静态特征与动态特征。由此可见,现实世界中的对象具有如下特征:

- (1) 有一个名字用来唯一标识该对象。</

1.1.2 C++ 中类的定义

类的定义包括类的数据成员的定义和类的成员函数的定义,在 C++ 中类的定义方式为:

```
class 类名
{
    private:
        私有数据及成员函数;
    protected:
        保护数据及成员函数;
    public:
        公有数据及成员函数;
};
```

其中, class 是保留字^①,用于类定义。类名是一种标识符,类的命名规则与变量的命名规则^②相同。一对花括号内是类的说明部分,说明该类的成员,包括数据成员和成员函数两部分。数据成员是用 C++ 基本数据类型定义的变量或已定义类的对象,用于描述某个具体对象的属性;成员函数设计成函数形式,是某个具体对象与外部程序的接口。外部程序可通过向该对象发消息即调用某个成员函数,使用该对象提供的服务。从访问权限来分,类的成员又分为公有的(public)、私有的(private)和保护的保护的(protected)。

一般情况下,应该将数据成员设计成私有的,将成员函数设计成公有的。这里先讨论前两类数据,保护类的成员在以后的章节中进行讨论。

注意:

- (1) C++ 中的常用数据类型有整型(int)、实型(double)、字符型(char)。
- (2) 函数指具有独立功能的程序块。C++ 中对象的方法通常用函数描述,函数有返回值、参数、函数体等。
- (3) 变量的命名规则: 由字母、数字、下划线构成,并由字母与下划线开头。
- (4) 关键字 private 和 public 的后面都有一个冒号“:”,类定义后面有一个分号“;”。

关键字 public、private 和 protected 被称为访问权限修饰符或访问控制修饰符,用它们来说明类成员的访问权限。它们在类内与出现的先后顺序无关,并且允许多次出现。

类的公有成员用 public 来说明,公有部分往往是一些操作,即成员函数。它提供给用户接口功能,使用户通过这些成员函数来访问私有数据。这部分成员可以在程序中引用,如图 1.3 所示。

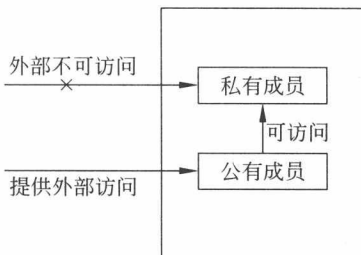


图 1.3 类成员访问示意图

① 在 C++ 中保留字有特殊含义,只能用于特定的场合。例如 private 用于数据隐藏,protected 用于继承的数据,public 用于外部数据访问