

· 高等学校计算机基础教材精选 ·

C++语言基础教程

(第3版)

吕凤翥 编著



清华大学出版社

· 高等学校计算机基础教育教材精选

C++语言基础教程

(第3版)

吕凤翥 编著

清华大学出版社
北京

内 容 简 介

书中全面、系统地讲述了 C++ 语言的基本概念、基本语法和编程方法,较详尽地介绍了 C++ 语言面向对象的重要特征:类和对象、继承和派生类、多态性、虚函数、函数模块和类模板等内容。本书具有丰富的例题,每章后面均提供了相当数量的练习题和作业题。

本书特点是:通俗易懂,由浅入深,突出重点,难点详解,偏重应用。本书不仅可作为高等学校 C++ 语言课程的教材,还可作为 C++ 语言的自学教材和参考书。

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++ 语言基础教程/吕凤翥编著. —3 版. —北京:清华大学出版社,2013

高等学校计算机基础教育教材精选

ISBN 978-7-302-32369-3

I. ①C… II. ①吕… 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 093797 号

责任编辑:焦 虹

封面设计:傅瑞学

责任校对:白 蕾

责任印制:何 芊

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

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

印 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:25 字 数:577 千字

版 次:1999 年 9 月第 1 版 2013 年 6 月第 3 版 印 次:2013 年 6 月第 1 次印刷

印 数:1~3000

定 价:39.00 元

产品编号:042536-01

出版说明

——高等学校计算机基础教育教材精选——

在教育部关于高等学校计算机基础教育的方针指导下,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各个层次,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)。我们要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,使本套教材无论在技术质量上还是文字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师资和生源水平、专业领域等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(E-mail);联系人:焦虹。

清华大学出版社

前言

—C++ 语言基础教程(第3版)—

本书是作者在多年从事 C++ 语言教学的基础上编写的。书中总结了教学中的经验和教训,并针对学生在学习过程中遇到的困难和提出的问题进行讲解。本书第 2 版出版以来,已被多所学校作为教材,印数逾 20 万册。为更好地满足教学需求,根据读者建议及教学实践,作者对本书第 2 版进行了认真的修订,删去了一些不必要的内容,增加了函数模板和类模板等新内容。本书的特点是:通俗易懂,适于自学;由浅入深,便于理解;概念明确,语言简洁;例题丰富,内容全面;重点突出,难点详解。

本书较全面、系统地讲述了 C++ 语言的基本概念和编程方法。通过学习本书,读者能够正确理解 C++ 语言中面向对象的方法,基本掌握 C++ 语言中的词法、语法,并且可以达到使用 C++ 语言编写简单程序的目的。

本书第 3 版继承了第 2 版的章节结构,共分 9 章。前 4 章讲述了 C++ 语言的基本词法和语法规则,包括字符集、语法规则、程序结构、运算符和表达式、各类语句、函数格式和调用方式、作用域及存储类等内容。在这些内容中,C++ 语言只是对 C 语言中的相应内容进行一些改进和补充,与 C 语言的内容很相近。第 5~8 章讲述了 C++ 语言所支持的面向对象的程序设计方法的内容,包括类和对象的概念及定义格式、对象的赋值和运算、继承性和派生类、多态性和虚函数等内容。这些内容使 C++ 语言成为一种面向对象的程序设计语言,这是学习 C++ 语言的重点和难点。这部分内容是 C 语言所没有的。第 9 章讲述了标准文件的读写函数和对一般文件的操作。本书每章都提供了较多例题,例题的针对性较强。全书共有 180 道例题。每章后面都备有相当数量的练习题和作业题。读者通过练习题可以检查自己对本章所学内容掌握的情况,练习题的覆盖面很广;通过作业题可以练习所讲过的主要内容,包括概念性的训练和方法技巧训练。对于初学 C++ 语言的读者,一是要弄清基本概念;二是要多看程序,从中学习方法和技巧,从而积累编程经验。本书提供了这两方面的训练。

本书的所有例题和作业题中要求分析输出结果的程序都在 Visual C++ 6.0 版本的编译系统下运行通过,在其他版本的编译系统中一般也都可以运行。

本书可作为高等学校教材,也可作为教师和学生的参考书。本书还适合作为自学 C++ 语言的教材。

作者在本书的编写过程中,查阅了许多有关外文资料和说明书,并阅读过一些翻译的

书籍,现谨对这些书的作者和译者提供的帮助表示最衷心的感谢。由于时间仓促、作者水平有限,书中难免会有不足和错误之处,恳请广大读者提出宝贵的意见。

谢谢喜欢阅读本书的读者!

作 者
于北京大学燕北园

目录

C++ 语言基础教程(第3版)

第1章 C++ 语言概述	1
1.1 面向对象程序设计的有关概念	1
1.1.1 面向对象的由来和发展	1
1.1.2 面向对象的有关概念	2
1.2 C++ 语言是一种面向对象的程序设计语言	4
1.2.1 C++ 语言对面向对象程序设计方法的支持	4
1.2.2 C++ 语言与 C 语言的关系	5
1.3 C++ 语言的词法及语法规则	6
1.3.1 C++ 语言的字符集	6
1.3.2 单词及语法规则	7
1.4 C++ 程序结构的特点	9
1.4.1 一个 C++ 语言的示范程序	9
1.4.2 C++ 程序的组成部分	9
1.4.3 C++ 程序的书写格式	12
1.5 C++ 程序的实现	13
1.5.1 C++ 程序的编辑、编译和运行	14
1.5.2 Visual C++ 6.0 版本的基本用法	15
练习题	19
作业题	19
第2章 数据类型和表达式	23
2.1 基本数据类型	23
2.2 常量和变量	25
2.2.1 常量	25
2.2.2 变量	28
2.3 数组	29
2.3.1 数组的定义	29
2.3.2 数组的赋值	30
2.3.3 字符数组	32

2.4	枚举	33
2.4.1	枚举类型和枚举变量	33
2.4.2	枚举变量的值	34
2.5	指针和引用	35
2.5.1	指针	35
2.5.2	指针和数组	38
2.5.3	字符指针	43
2.5.4	引用	45
2.6	运算符	46
2.6.1	算术运算符	46
2.6.2	关系运算符	48
2.6.3	逻辑运算符	48
2.6.4	位操作运算符	48
2.6.5	赋值运算符	49
2.6.6	其他运算符	50
2.6.7	运算符的优先级和结合性	53
2.7	表达式	53
2.7.1	表达式的种类	53
2.7.2	表达式的值和类型	54
2.7.3	表达式中的类型转换	59
2.8	类型定义	61
2.9	结构和联合	63
2.9.1	结构	64
2.9.2	联合	67
	练习题	69
	作业题	70

第3章	预处理和语句	75
3.1	预处理功能	75
3.1.1	文件包含命令	76
3.1.2	条件编译命令	76
3.1.3	宏定义命令	79
3.2	语句	85
3.2.1	表达式语句和空语句	85
3.2.2	复合语句和分程序	85
3.3	选择语句	86
3.3.1	条件语句	86
3.3.2	开关语句	88

3.4	循环语句	91
3.4.1	while 循环语句	91
3.4.2	do-while 循环语句	92
3.4.3	for 循环语句	93
3.4.4	多重循环	95
3.5	转向语句	99
3.5.1	goto 语句	99
3.5.2	break 语句	100
3.5.3	continue 语句	101
	练习题	102
	作业题	102

第4章	函数和作用域	111
4.1	函数的定义和说明	111
4.1.1	函数的定义格式	113
4.1.2	函数的说明方法	113
4.2	函数的调用	115
4.2.1	函数的值和类型	115
4.2.2	函数的传值调用	116
4.2.3	函数的引用调用	118
4.3	函数的参数	120
4.3.1	函数参数的求值顺序	120
4.3.2	设置函数参数的默认值	121
4.3.3	使用数组作为函数参数	122
4.4	内联函数	124
4.4.1	内联函数引入的原因	124
4.4.2	内联函数的定义方法	125
4.4.3	使用内联函数应注意的事项	125
4.5	函数重载	126
4.5.1	参数类型上不同的重载函数	126
4.5.2	参数个数上不同的重载函数	127
4.6	函数的嵌套调用和递归调用	128
4.6.1	函数的嵌套调用	128
4.6.2	函数的递归调用	129
4.7	作用域	131
4.7.1	标识符的作用域规则	132
4.7.2	作用域的种类	132
4.7.3	关于重新定义标识符的作用域规定	133

4.7.4	局部变量和全局变量	135
4.7.5	内部函数和外部函数	138
4.8	C++ 语言的系统函数	141
4.8.1	C++ 语言系统函数概述	141
4.8.2	字符串处理函数	143
4.9	函数模板	147
4.9.1	函数模板的概念	147
4.9.2	函数模板的使用	149
	练习题	151
	作业题	152

第 5 章	类和对象(一)	161
5.1	类的定义	161
5.1.1	什么是类	161
5.1.2	类的定义格式	162
5.1.3	注意事项	164
5.2	对象的定义	166
5.2.1	对象的定义格式	166
5.2.2	对象成员的表示方法	166
5.3	对象的初始化	169
5.3.1	构造函数和析构函数	169
5.3.2	默认构造函数和默认析构函数	171
5.3.3	复制构造函数	171
5.4	成员函数的特性	174
5.4.1	内联函数和外联函数	175
5.4.2	重载性	176
5.4.3	设置参数的默认值	177
5.5	静态成员	178
5.5.1	静态数据成员	179
5.5.2	静态成员函数	182
5.6	友元	183
5.6.1	友元函数	183
5.6.2	友元类	185
5.7	类的作用域	187
5.8	局部类和嵌套类	188
5.8.1	局部类	188
5.8.2	嵌套类	189
5.9	对象的生存期	190

练习题.....	192
作业题.....	193
第 6 章 类和对象(二)	200
6.1 对象指针和对象引用	200
6.1.1 指向类的成员的指针.....	200
6.1.2 对象指针和对象引用作为函数参数.....	203
6.1.3 this 指针	205
6.2 对象数组和对象指针数组	207
6.2.1 对象数组.....	207
6.2.2 指向数组的指针和指针数组.....	209
6.2.3 带参数的 main() 函数	214
6.3 常类型	215
6.3.1 一般常量和对象常量.....	215
6.3.2 常指针和常引用.....	216
6.3.3 常成员函数.....	219
6.3.4 常数据成员.....	221
6.4 子对象和堆对象	222
6.4.1 子对象.....	222
6.4.2 堆对象.....	224
6.5 类型转换	230
6.5.1 类型的自动隐式转换.....	230
6.5.2 构造函数具有类型转换功能.....	231
6.5.3 类型转换函数.....	232
6.6 类模板	234
6.6.1 类模板的引进.....	234
6.6.2 类模板和模板类.....	235
6.6.3 类模板应用举例.....	238
6.7 应用实例——链表	240
练习题.....	244
作业题.....	245
第 7 章 继承性和派生类	254
7.1 基类和派生类	254
7.1.1 派生类的定义.....	255
7.1.2 派生类的三种继承方式.....	255
7.1.3 基类成员在派生类中的访问权限.....	256
7.1.4 成员访问权限的控制.....	256

7.2	单继承	259
7.2.1	派生类构造函数和析构函数	260
7.2.2	子类型和赋值兼容规则	265
7.3	多继承	269
7.3.1	多继承的概念	269
7.3.2	多继承的构造函数和析构函数	269
7.3.3	多继承的二义性问题	272
7.4	虚基类	277
7.4.1	虚基类的引入和说明	277
7.4.2	含有虚基类的派生类的构造函数	279
7.5	应用实例——日期和时间	281
	练习题	283
	作业题	284
第 8 章	多态性和虚函数	294
8.1	函数重载	294
8.2	运算符重载	297
8.2.1	运算符重载的几个问题	297
8.2.2	运算符重载函数的两种形式	298
8.2.3	其他运算符的重载举例	305
8.3	静态联编和动态联编	309
8.3.1	静态联编	309
8.3.2	动态联编	310
8.4	虚函数	311
8.5	纯虚函数和抽象类	318
8.5.1	纯虚函数	318
8.5.2	抽象类	321
8.6	虚析构函数	323
8.7	程序举例	325
	练习题	331
	作业题	332
第 9 章	C++ 语言的 I/O 流库	341
9.1	屏幕输出	342
9.1.1	使用预定义的插入符	342
9.1.2	使用成员函数 put() 输出一个字符	344
9.1.3	使用成员函数 write() 输出一个字符串	345
9.2	键盘输入	346

9.2.1	使用预定义的提取符.....	346
9.2.2	使用成员函数 get() 获取一个字符	348
9.2.3	使用成员函数 read() 读取一串字符	351
9.3	插入符和提取符的重载	352
9.4	格式化输入和输出	355
9.4.1	使用成员函数设置流的格式化标志位.....	355
9.4.2	格式输出函数.....	357
9.4.3	操作子.....	359
9.5	磁盘文件的输入和输出	360
9.5.1	磁盘文件的打开和关闭操作.....	360
9.5.2	文本文件的读写操作.....	362
9.5.3	二进制文件的读写操作.....	365
9.5.4	随机访问数据文件.....	367
9.5.5	其他有关文件操作的函数.....	370
9.6	字符串流	373
9.6.1	ostream 类的构造函数	374
9.6.2	istream 类的构造函数	375
9.7	流错误的处理	376
9.7.1	状态字和状态函数.....	377
9.7.2	清除和设置流的状态位.....	378
	练习题.....	379
	作业题.....	380

附录 A	ASCII 码表	385
-------------	-----------------------	------------

C++ 语言是一种应用较广的面向对象的程序设计语言,使用它可以实现面向对象的程序设计。本书主要讲述 C++ 语言的特点和语法,介绍使用 C++ 语言编程的方法。为了使读者对 C++ 语言的特性有所了解,本章首先介绍一些有关面向对象程序设计的基本概念。因为面向对象的设计与面向过程的设计是有很大的区别的,因此面向对象的程序设计是在面向过程的程序设计基础上的一个质的飞跃。学习 C++ 语言首先要认识它面向对象的特性和实现面向对象的方法。

1.1 面向对象程序设计的有关概念

1.1.1 面向对象的由来和发展

下面回顾计算机语言的发展过程,以便了解面向对象的方法是如何产生的。

20 世纪 50 年代中期,出现了高级程序设计语言 FORTRAN,它在计算机语言的发展史上具有划时代的意义。该语言引进了许多现在仍然使用的程序设计的概念,如变量、数组、循环、分支等。但是,该语言在使用中也发现了一些不足之处。例如,不同部分的相同变量名容易发生混淆等。

20 世纪 50 年代后期,高级语言 Algol 在程序段内部对变量实施隔离。Algol 60 提出了块结构的思想,由“Begin ... End”来实现块结构,这样就不会使不同块内同名变量相互混淆,对数据实行了保护。这实际上也是一种初级的封装。

20 世纪 60 年代开发的 Simula 67,是面向对象语言的鼻祖。它将 Algol 60 中的块结构概念向前推进了一大步,提出了对象的概念。对象代表着待处理问题中的一个实体,在处理问题的过程中,一个对象可以某种形式与其他对象通信。从概念上讲,一个对象是既包含数据又包含处理这些数据操作的一个程序单元。Simula 语言中也使用了类的概念,类用来描述特性相同或相近的一组对象的结构和行为。该语言还支持类的继承。继承可将多个类组成为层次结构,进而允许共享结构和行为。

20 世纪 70 年代出现的 Ada 语言是支持数据抽象类型的最重要的语言之一。数据抽象是由数据结构及作用在数据结构上的操作组成的一个实体。它把数据结构隐藏在操作

接口的后面,通过操作接口实现外部的交流。对外部来讲,只知道做什么,而不知道如何做。再将类型扩展到数据抽象上,即将某种类型的操作汇集起来作为一个整体看待,并与该类型一起看做一个独立的单元,就构成了抽象数据类型。因此,可以说,数据抽象类型是数据抽象封装后的类型。它包含了该类型下的操作集和由操作集间接定义的数据类型的值集。Ada 语言中面向对象的抽象结构是包。它支持数据抽象类型、函数和运算符重载以及多态性等面向对象的机制。但是,Ada 语言不是全面地支持继承,因此人们常称它为一种基于对象的语言。

后来出现的 Smalltalk 语言是最有影响的面向对象的语言之一。它丰富了面向对象的概念。该语言并入了 Simula 语言的许多面向对象的特征,包括类和继承等。在该语言中,信息的隐藏更加严格,每种实体都是对象。在 Smalltalk 环境下,程序设计就是向对象发送信息。这个信息将表示为一种操作,如两个数相乘,创建一个新类的对象等。Smalltalk 语言是一种弱类型化的语言,一个程序中的同一个对象可以在不同时间内表现为不同的类型。

20 世纪 80 年代中期以后,面向对象的程序设计语言广泛应用于程序设计,并且有了许多新的发展,出现了更多的面向对象的语言。归纳起来,大致可分为如下两类。

1. 开发全新的面向对象的语言

具有代表性的全新的面向对象的语言有 Object-C。它是在 C 语言上扩展而成的,是 Smalltalk 语言的变种。Eiffel 语言除了有封装和继承外,还集成了几种强而有力的面向对象的特征,它是一种很好的面向对象的语言。Smalltalk 80 语言经历了多次修改和更新,新版本有很大改进。这类全新的面向对象的语言学习起来要从头开始。

2. 对传统语言进行面向对象的扩展

这类语言又称为混合型语言,C++ 语言就是其代表。C++ 语言是 20 世纪 80 年代早期由贝尔实验室设计的一种面向对象的语言。它在 C 语言的基础上增加了对面向对象程序设计的支持。这类语言的特点是既支持传统的面向过程的程序设计,又支持新型的面向对象的程序设计。对于一些已经较好地掌握了 C 语言的人来讲,学习 C++ 语言相对容易一些。另外,C++ 语言具有丰富的 C 语言应用基础和开发环境的支持,普及起来也相对快些。这些就是 C++ 语言当前得以广泛应用的主要原因。

1.1.2 面向对象的有关概念

什么是面向对象?简单地说,面向对象是一种软件开发方法。具体地讲,面向对象是一种运用新概念和新方法来构造系统的软件开发方法。这些新概念和新方法包括对象、类、封装、聚合、继承和多态性等。它们体现了面向对象这种方法的新特点。

下面对面向对象中使用的新概念和新方法进行简单解释。

1. 对象

对象是软件系统的基本构成单位。对象是对客观世界中实际存在的某种事物的抽象,即是描述客观事物的一个实体。对象是研究问题和分析问题的出发点,是构成程序的主要成员。

对象是一组属性和一组行为的集合。属性是用来描述对象的静态特性,它使用若干数据来表示;行为是用来描述对象的动态特性,它使用若干操作来表示。于是,可以说对象是数据与操作的集合。

对象之间传递信息是通过消息实现的。当一个操作被调用时,就有一条消息被发送到这个对象上,这个消息将带来所执行的操作的详细内容。

2. 类

简单地讲,类是一种类型,称为类类型。这种类型也是一种由用户定义的自定义类型,只不过是一种更复杂、更先进的类型。

类是对具有相同属性对象的描述。类是创建对象的样板,它包含着所创建对象的数据描述和操作的定义。类是一种具有共同属性和行为的若干对象的统一描述体。

分类是人们认识客观世界的一种常用的思维方法,将具有相同属性的事物划分为一类,得到一个抽象的概念,用来表示一类事物。例如,桌子、椅子、高楼、桥梁等都是被抽象出来的一类事物。分类的原则就是抽象,类是抽象数据类型的实现。

3. 封装

封装是面向对象方法中的重要特性。封装是把对象的属性和行为(即数据和操作)结合成为一个封装体。该封装体内包含对象的属性,它们由若干不同类型的数据组成;还包括对象的行为,它们由若干种操作组成。操作是通过函数来实现的,又称方法。

封装体还具有隐藏性。封装体内的某些数据和方法在封装体外是不可见的,即是不可访问或改变的。这些封装体外不可见的成员被隐藏在封装体内,具有安全性。

封装体与外界联系是通过称为接口的通道进行的。封装体的外部特性就是由这些接口提供的。外部操作通过接口进行,也就是要使用方法对封装体内隐藏的数据进行操作。

4. 聚合

聚合是类之间的一种包含关系。在处理一个复杂问题时,常常把复杂问题分解为若干个简单问题。通过逐个解决简单问题来解决复杂问题。具体实现方法是在一个类中可以包含另外一个类的对象。于是一个复杂类可由若干简单类的对象组成,这种方法称为聚合。例如,描述一架飞机是较为复杂的,因此可将飞机拆分为机翼、机身、机尾和发动机等若干个部位。在描述飞机的类中可以包含机翼类、机身类、机尾类和发动机类的对象。

5. 继承

继承是面向对象方法中的另一个重要特性,称为继承性。继承是创建新类的一种方法。使用继承可以解决一般类和特殊类的关系。特殊类具有一般类的全部属性和行为,并且它还有自己特殊的属性和行为,这时称特殊类是对一般类的继承。

使用继承可以简化人们对事物的描述。例如,已经有了对汽车的描述,再对轿车进行描述就简单多了。轿车继承了汽车的一般特性,只需描述轿车所特有的性能就够了。继承可以减少冗余性,提高重用性。

继承为软件开发带来了许多方便。可将已开发好的类存放到类库内。开发新系统时,便可直接使用或继承使用已有的类,这将会减少编程的工作量,并提高编程质量。

6. 多态性

多态性指的是一对多的状态。函数重载和运算符重载是多态性的体现。一个函数名

或同一个运算符对应于不同的实现或功能,这便是一种多态性。更重要的是多态性体现在动态联编上。在一般类中定义的行为或方法,被特殊类继承后,可有不同的实现或操作,并在运行中进行联编。例如,定义一个一般类“几何图形”,它具有求面积的方法,该方法可以不具有具体含义。再定义若干个特殊类“圆形”、“矩形”等,它们都继承一般类,每个特殊类中也具有求面积的方法。这些方法根据不同的几何图形求面积的公式,用各自的方法实现。在实际操作中根据运行时出现的几何图形,调用对应的求面积方法进行计算。这就是面向对象中的多态性。

综上所述,面向对象的方法可归纳为如下几点。

(1) 将客观事物中抽象出的数据和方法构成一个集合体,这就是对象,即对实体的描述。

(2) 将相同类型的对象抽象出共性,形成类。类具有封装性和隐藏性。

(3) 类是一个封装体,类中大多数数据只能通过本类的方法进行处理。这些数据在类外是不可见的,即是无法进行访问的。

(4) 类是通过外部接口与外界发生关系的,这些外部接口提供了类的行为。

(5) 对象之间是通过消息进行通信的。

1.2 C++ 语言是一种面向对象的程序设计语言

在没有具体讲述 C++ 语言的程序特点和语法规则及编程方法之前,先了解一下 C++ 语言具有面向对象程序设计语言的哪些特点,这对学习 C++ 语言是十分有益的。

1.2.1 C++ 语言对面向对象程序设计方法的支持

1. C++ 语言支持数据封装

支持数据封装就是支持数据抽象。在 C++ 语言中,类是支持数据封装的工具,对象则是数据封装的实现。

面向过程的程序设计方法与面向对象的程序设计方法在对待数据和函数关系上是不同的。在面向过程的程序设计中,数据只被看成是一种静态的结构,它只有等待调用函数来对它进行处理。在面向对象的程序设计中,将数据和对该数据进行合法操作的函数封装在一起作为一个类的定义。另外,封装还提供一种对数据访问严格控制的机制。因此,数据将被隐藏在封装体中,该封装体通过操作接口与外界交换信息。

每个类的对象都包含这个类所指定的若干成员。

在 C 语言中可以定义结构,但这种结构只包含数据,而不包含函数。C++ 语言中的类是数据和函数的封装体。

2. C++ 语言的类中包含私有、公有和保护成员

C++ 语言的类中可定义三种不同访问控制权限的成员。一种是私有(private)成员,只有在类中说明的函数才能访问该类的私有成员,而在该类外的函数不可以访问私有成