

高职高专计算机系列规划教材

# C++

武马群 主编

繆春池 吕峻闽 谢志龙 编著

北京工业大学出版社

## 内 容 提 要

本书主要讲述 C++ 语言的基本语法、结构化程序设计方法和面向对象程序设计方法。前 8 章介绍了 C++ 语言的基础知识,包括基本数据类型、运算符与表达式、结构控制语句、数组、函数、指针、构造型数据结构等知识;后 4 章介绍了 C++ 面向对象程序设计的基本概念和方法,包括类、对象、继承、派生、多态以及输入输出流等知识。书中文字流畅、通俗易懂、概念清楚、深入浅出、例题丰富,每章都附有习题和上机实验,以加深和巩固学生所学知识。

## C++ 语言程序设计

武马群 主编

缪春池 吕峻闽 谢志龙 编著

北京工业大学出版社出版发行

邮编:100022 电话:(010) 67392308

各地新华书店总经销

徐水宏远印刷厂印刷

2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷

787 mm×1 092 mm 16 开本 印张 18.75 字数 480 千字

印数:1~5 000 册

ISBN 7-5639-1530-3/T·264

定价：26.00 元

C 语言是一种应用十分广泛的编程语言，具有功能丰富、语句简洁、使用方便、语法灵活、数据结构多样、能对硬件进行操作、高移植性和通用性等诸多优点，它既有高级语言的特点，又有汇编语言等低级语言的特点，因此被称为“中级语言”，成为编制系统软件和应用软件的首选语言。

随着计算机技术的发展、用户软件需求的不断增长以及软件规模的不断扩大，C 语言已经显得比较吃力了。C++语言正是在这种情况之下应运而生的，和 C 语言相比，C++保留了原来 C 语言的所有特点和优点，即向下兼容所有 C 语言的程序，而在 C 语言的基础上增加了面向对象程序设计的方法，增加了“类”和“对象”的概念。

C++语言是我国各大学都开设的重要课程，在高职院校的计算机课程中，C++语言也是学习程序设计语言的必修课程。在编写本书过程中，作者结合自己多年从事 C++语言教学的经验，理论联系实际，知识讲解力求通俗易懂。本书在体系结构安排上尽可能将概念、知识点与例题结合起来，每章结束后都对本章内容进行小结。在例题的选择上具有针对性强的特点，通过一些典型程序将前后的一些知识点联系起来，使读者在对比中理解各种程序实现方式的特点和异同，能够融会贯通，举一反三。每章还附有必要的习题，让学生加深和巩固所学知识，提高学生的编程能力，培养良好的编程风格和习惯。

全书共 12 章，分别是：C++语言概述、基本数据类型、运算符与表达式、结构控制语句、数组、函数、指针、构造型数据结构、类与对象、派生与继承、多态与虚函数、I/O 流与文件流。从 C++语言的基本数据元素、基本语句和结构控制语句、构造数据类型的定义和使用、函数的定义和调用，到指针的灵活运用等主要方面进行了由浅入深的讲解。本书的特点是结构合理、层次分明、例题丰富、通俗易懂、实用性强，适合于初学者使用。

由于作者水平有限，加上时间仓促，书中的缺点和错误在所难免，恳请读者批评指正。

编者

2005 年 6 月

# 目 录

第 1 章 C++语言概述.....	1
1.1 C++语言概述.....	1
1.1.1 C 语言的发展历史.....	1
1.1.2 C 语言的特点.....	1
1.1.3 C++的诞生.....	2
1.2 面向对象的程序设计.....	3
1.2.1 对象 ( Object ) .....	3
1.2.2 类 ( Class ) .....	4
1.2.3 属性 ( Property ) .....	4
1.2.4 事件 ( Event ) .....	4
1.2.5 方法 ( Method ) .....	5
1.2.6 消息 ( Message ) .....	5
1.3 VC++ 6.0 的程序开发环境.....	5
1.3.1 VC++ 6.0 的安装.....	5
1.3.2 启动 VC++ 6.0 开发环境.....	7
1.3.3 退出 VC++ 6.0 开发环境.....	8
1.3.4 VC++ 6.0 程序开发环境介绍.....	9
1.3.5 第一个 C++程序.....	9
1.3.6 C++语言程序的编制运行过程.....	12
1.4 简单的 C++程序.....	13
1.4.1 C++程序的结构.....	13
1.4.2 C++语言程序设计基础.....	15
【本章小结】.....	23
【习题】.....	23
【实验】.....	24
第 2 章 基本数据类型.....	25
2.1 C++语言的数据类型.....	25
2.2 常量与变量.....	26
2.2.1 常量.....	26
2.2.2 变量.....	28
2.3 整型数据.....	29

2.3.1	整型常量	29
2.3.2	整型变量	30
2.4	实型数据	32
2.4.1	实型常量	32
2.4.2	实型变量	33
2.5	字符型数据	34
2.5.1	字符型常量	34
2.5.2	字符型变量	35
2.5.3	字符串常量	37
2.6	布尔型数据	38
	【本章小结】	38
	【习题】	39
	【实验】	40
第3章	运算符与表达式	42
3.1	运算符与表达式概述	42
3.1.1	表达式的组成	42
3.1.2	运算符的分类	42
3.2	算术运算符与表达式	43
3.3	关系运算符与表达式	45
3.4	逻辑运算符与表达式	46
3.5	位运算符与表达式	48
3.5.1	按位与	48
3.5.2	按位或	48
3.5.3	按位异或	49
3.5.4	按位取反	49
3.5.5	按位左移	50
3.5.6	按位右移	50
3.5.7	位运算复合赋值运算符	50
3.6	其他运算符与表达式	51
3.6.1	赋值运算符与表达式	51
3.6.2	条件运算符与表达式	52
3.6.3	逗号运算符与表达式	53
3.6.4	强制类型转换运算符与表达式	53
3.7	运算符的优先级	54
	【本章小结】	55
	【习题】	56
	【实验】	57

第 4 章 结构控制语句 .....	59
4.1 程序的基本控制结构 .....	59
4.1.1 顺序结构 .....	59
4.1.2 选择结构 .....	60
4.1.3 循环结构 .....	61
4.2 条件选择语句 .....	62
4.2.1 if 语句 .....	62
4.2.2 if 语句的嵌套 .....	63
4.3 多路分支语句 .....	66
4.4 循环语句 .....	69
4.4.1 while 型循环语句 .....	69
4.4.2 do...While 型循环语句 .....	71
4.4.3 for 型循环语句 .....	73
4.4.4 goto 型循环语句 .....	75
4.4.5 continue 语句和 break 语句 .....	76
4.4.6 循环嵌套 .....	78
【本章小结】 .....	81
【习题】 .....	81
【实验】 .....	85
第 5 章 数组 .....	86
5.1 一维数组 .....	86
5.1.1 一维数组的定义 .....	86
5.1.2 一维数组的初始化 .....	87
5.1.3 一维数组元素的使用 .....	87
5.2 字符数组 .....	92
5.2.1 字符数组的定义 .....	92
5.2.2 字符数组的初始化 .....	92
5.2.3 字符数组的使用 .....	93
5.2.4 字符串处理函数 .....	94
5.2.5 字符串类 string .....	98
5.3 二维数组 .....	100
5.3.1 二维数组的定义 .....	100
5.3.2 二维数组的初始化 .....	101
5.3.3 二维数组的使用 .....	102
【本章小结】 .....	105
【习题】 .....	105

【实验】 .....	108
第 6 章 函数 .....	110
6.1 用户自定义函数 .....	110
6.1.1 函数的定义 .....	110
6.1.2 函数的调用 .....	111
6.1.3 函数调用中的数据传递 .....	115
6.2 函数的嵌套调用和递归调用 .....	120
6.2.1 函数的嵌套调用 .....	120
6.2.2 函数的递归调用 .....	121
6.3 变量的作用域和生存期 .....	123
6.3.1 变量的作用域 .....	123
6.3.2 变量的生存期 .....	127
6.4 函数的作用范围 .....	129
6.4.1 内部函数 .....	129
6.4.2 外部函数 .....	130
6.5 内联函数 .....	131
6.6 带默认参数值的函数 .....	132
6.7 带命令行参数的 main 函数 .....	133
【本章小结】 .....	134
【习题】 .....	135
【实验】 .....	139
第 7 章 指针 .....	140
7.1 指针与指针变量 .....	140
7.1.1 地址、指针与指针变量 .....	140
7.1.2 指针变量的使用 .....	142
7.2 指针与数组 .....	145
7.2.1 指向一维数组的指针 .....	145
7.2.2 指向二维数组的指针 .....	148
7.2.3 指向字符串的指针 .....	152
7.3 指向特殊类型数据的指针变量 .....	154
7.3.1 指针数组 .....	154
7.3.2 指向指针的指针 .....	155
7.4 指针与函数 .....	157
7.4.1 指针变量作为函数参数传递 .....	157
7.4.2 返回指针的函数 .....	159

7.4.3 指向函数的指针 .....	160
7.5  引用 .....	163
7.5.1 引用的定义及简单使用 .....	163
7.5.2  引用作为参数 .....	164
7.6  动态内存管理 .....	167
7.6.1  分配内存 new 运算符 .....	167
7.6.2  释放内存 delete 运算符 .....	167
【本章小结】 .....	167
【习题】 .....	168
【实验】 .....	171
第 8 章  构造型数据结构 .....	172
8.1  结构体 .....	172
8.1.1  结构体的定义 .....	172
8.1.2  结构体变量的使用 .....	174
8.1.3  结构体数组 .....	175
8.1.4  指向结构体的指针 .....	176
8.1.5  结构体变量作为参数在函数之间进行传递 .....	178
8.2  指针与链表 .....	178
8.2.1  链表的定义 .....	178
8.2.2  链表的操作 .....	180
8.3  共用体 .....	185
8.3.1  共用体的定义 .....	185
8.3.2  共用体变量的使用 .....	186
8.4  枚举类型 .....	188
8.4.1  枚举类型的定义 .....	188
8.4.2  枚举类型变量的使用 .....	189
8.5  自定义类型 .....	190
【本章小结】 .....	192
【习题】 .....	192
【实验】 .....	196
第 9 章  类与对象 .....	197
9.1  面向对象程序设计方法的相关概念 .....	197
9.1.1  对象 ( Object ) .....	197
9.1.2  抽象 ( Abstraction ) .....	199
9.1.3  封装 ( Encapsulation ) .....	199

9.1.4	继承 ( Inheritance )	200
9.1.5	多态性 ( Polymorphism )	201
9.2	类	202
9.2.1	类的定义	202
9.2.2	类的数据成员	203
9.2.3	类的成员函数	205
9.3	对象	209
9.3.1	对象的定义	209
9.3.2	对象成员的引用	210
9.3.3	构造函数	211
9.3.4	析构函数	216
9.4	静态成员	217
9.4.1	静态数据成员	218
9.4.2	静态成员函数	220
9.5	对象指针和 this 指针	221
9.5.1	对象指针	221
9.5.2	this 指针	222
9.6	友元	224
9.7	对象数组	225
	【本章小结】	226
	【习题】	227
	【实验】	228
第 10 章	派生与继承	229
10.1	派生类	229
10.1.1	派生类的定义	230
10.1.2	派生类成员的访问属性	231
10.2	派生类的构造函数和析构函数	238
10.2.1	构造函数	238
10.2.2	析构函数	241
10.3	多继承	241
10.3.1	多继承的定义	241
10.3.2	多继承的构造函数	243
10.4	虚基类	245
10.4.1	虚基类的定义	245
10.4.2	虚基类的构造函数	245
10.4.3	虚基类的应用实例	246

【本章小结】 .....	248
【习题】 .....	249
【实验】 .....	249
第 11 章 多态性与虚函数 .....	251
11.1 多态性 .....	251
11.1.1 多态性的概念 .....	251
11.1.2 简单的多态性实例 .....	251
11.2 虚函数 .....	252
11.2.1 虚函数的定义 .....	252
11.2.2 纯虚函数 .....	255
11.2.3 抽象类 .....	257
11.3 运算符重载 .....	261
11.3.1 运算符重载的定义 .....	261
11.3.2 重载运算符的规则 .....	265
【本章小结】 .....	266
【习题】 .....	266
【实验】 .....	267
第 12 章 I/O 流与文件流 .....	268
12.1 输入输出流的简介 .....	268
12.2 输出流 .....	269
12.2.1 标准输出流对象 cout .....	269
12.2.2 标准错误输出流对象 cerr .....	270
12.2.3 自定义格式输出 .....	271
12.3 输入流 .....	275
12.3.1 标准输入流 cin .....	275
12.3.2 cin 的成员函数 get() .....	276
12.3.3 cin 的成员函数 getline() .....	277
12.4 文件流 .....	278
12.4.1 文件的打开与关闭 .....	278
12.4.2 ASCII 文件的读写 .....	280
12.4.3 二进制文件的读写 .....	282
【本章小结】 .....	285
【习题】 .....	285
【实验】 .....	286

附录一 常用的 ASCII 码表 .....287

附录二 C++语言的关键字.....288

# 1 C++

C 语言是一种面向过程的结构化和模块化的语言，自出现至今经历了 30 多年，是使用最广泛的程序设计语言之一。随着计算机技术的发展 3001 用户软件需求的不断增长以及软件规模的不断扩大，C 语言已经显得比较吃力了。C++ 语言正是在这种情况之下应运而生的。C++ 语言具有对 C 语言的兼容性，并提供了更加安全和严格的语法，是一种面向对象的程序设计方法。

本章首先将介绍 C++ 语言的发展历史及主要特点，然后介绍 C++ 语言程序的开发环境和开发步骤，使读者在简单的 C++ 程序的编制过程中，初步掌握 C++ 程序的构成、特点、书写格式和编制规范。

## 1.1 C++ 语言概述

### 1.1.1 C 语言的发展历史

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的，并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL ( Basic Combined Programming Language ) 发展演变而来。

在 1970 年，AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言，并用 B 语言写了第一个 UNIX 操作系统。1972 年，贝尔实验室的 Ritchie 在 B 语言的基础上设计出了 C 语言，取的是 BCPL 语言的第二个字母。到了 1973 年，两人又用 C 语言一起改写了 UNIX 操作系统 90% 以上的代码，也就是 UNIX 的第 5 个版本。1978 年 Ritchie 和 Brian Kernighan 编写了《The C Programming Language》一书，并于 1988 年做了修订，该书就是 C 语言版本的基础，被称为标准 C。

随着计算机的日益普及，出现了许多 C 语言版本，正是 C 语言的易移植性的特点，使得 UNIX 系统在 AT&T、VAX 等计算机系统上能够迅速实现。但是由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准化协会( ANSI ) 为 C 语言制定了一套 ANSI 标准并成为现行的 C 语言标准，包括了 1983 年和 1987 年两个版本。到了 1990 年，国际标准化组织( ISO ,International Standard Organization ) 接受了 87 ANSI C 作为 ISO C 的标准，即 ISO 9899-1990。目前比较流行的 C 语言的编译系统都是以此作为基础，如 Turbo C、Borland C、Microsoft C 等。

### 1.1.2 C 语言的特点

C 语言发展如此迅速，而且成为最受欢迎的语言之一，主要因为它具有强大的功能。C 语言有如下的特点。

(1) C 语言提供的语句简洁, 使用方便, 格式紧凑, 语法灵活。C 语言一共有 32 个关键字, 9 种控制语句, 语句简练, 书写自由。以下关键字是由系统定义的, 不能用做其他定义。

与数据类型相关的关键字:

```
char      int      short      long      signed      unsigned      float
double    enum     struct    union    typedef     void
```

与存储类型相关的关键字:

```
auto      register    static      extern
```

与控制语句相关的关键字:

```
if          else      switch     case      default    do
while      for       break      continue  goto       return
```

其他的关键字:

```
const      sizeof     volatile
```

下面是 C 语言中的 9 种控制语句:

```
if... else ...      switch... case      for...      while ...
do... while      continue      break      goto      return
```

(2) C 语言的运算符十分丰富, 一共有 34 种运算符, 有算术、关系、逻辑、位、赋值、指针、条件、逗号、下标、类型转换运算符等多种类型。

(3) C 语言的数据结构多样, 有整型、实型、字符型、枚举类型等基本类型, 有数组、结构体、共用体等构造类型以及指针类型, 还为用户提供了自定义数据类型, 特别是引入了指针概念, 能够实现复杂的数据结构。

(4) C 语言的控制语句形式多样、使用方便。有两路分支、多路分支和循环结构几种控制语句, 便于结构化模块的实现和控制, 也便于程序的编制和维护。

(5) C 语言是一种模块化的程序设计语言, 采用自顶向下、逐步求精的结构化程序设计方法, 各模块功能独立, 以函数形式编制, 通过函数之间的相互调用和数据传递, 实现系统整体的功能要求。像这样把大型系统的实现化整为零便于分工合作以及共享。

(6) C 语言可以直接访问地址、进行位运算, 从而能对硬件进行操作, 因此 C 语言既具有高级语言的编写简单方便、便于理解的优点, 又具有低级语言的与硬件结合紧密的优点。因此 C 语言被称为介于高级语言和低级语言之间的中级语言。

(7) C 语言具有很强的移植性, 由 C 语言编写的程序基本不用太多的修改就可以用于不同型号的计算机上, 程序和硬件的匹配由 C 语言的编译程序完成, 同时也可以多种操作系统下使用。

(8) C 语言具有很好的通用性, 既可以用于编写应用软件, 也适合编写系统软件。例如 UNIX 操作系统的源代码就是用 C 语言编制的。

### 1.1.3 C++的诞生

C 语言、Pascal 语言等面向过程的结构化程序设计语言适合处理较小规模的程序。随着计算机技术的发展、用户软件需求的不断增长以及软件规模的不断扩大, 这类语言对于处理软件规模较大、用户需求较复杂的问题就力不从心了。20 世纪 80 年代, 计算机科学家提出了面向对象的程序设计方法 (OOP, Object Oriented Programming), 在这种背景下, 仍然是

来自 AT&T 贝尔实验室的 Dr.Bjarnne Stroustrup 在 C 语言的基础上开发出了 C++ 语言。

和 C 语言相比, C++ 保留了原来 C 语言所有特点和优点, 即向下兼容所有 C 语言的程序, 并在 C 语言的基础上增加了面向对象程序设计的方法, 增加了“类”和“对象”的概念。

结构化程序设计的精髓就是“自顶向下、逐步求精”, 具体来说是将一个程序划分成若干个能够完成独立功能的模块, 在模块中采用顺序、选择和循环这 3 种基本结构实现。在面向对象程序设计中, 首先分析具有相同类型的事物, 将其中共同的属性和特点抽象成一种新的数据结构——类, 然后定义类的实例——对象, 在对象中封装了属性和方法, 通过接口与程序进行数据的交换等操作。

C++ 新增加了以下关键字:

new	delete	public	private	protected
class	template	this	inline	virtual
overload	friend	catch		

## 1.2 面向对象的程序设计

尽管类与对象的概念相当简单, 但是功能却很强大, 简单来讲, 面向对象程序设计语言的特点就是抽象、封装、继承和多态。

(1) “抽象”是从具体事物中提取共同的性质, 包括对数据的抽象和对行为的抽象。数据抽象就是对事物属性的描述, 比如圆所共有的性质有圆心、半径等; 行为抽象就是对事物所进行操作的抽象, 比如圆共有的操作有计算周长、计算面积等。

(2) “封装”是把对象相关的数据和操作结合起来, 将对象的属性以及操作实现的代码隐藏起来, 通过对外提供的接口进行控制。如用户不需要知道如何计算圆周长, 只要将半径作为参数就可以得到结果。

(3) “继承”是在原有类(父类或者基类)的基础上建立一个新的类(子类或者派生类), 通过继承, 派生类得到了基类的属性和操作, 因此在派生类中只需要定义基类没有的属性和操作即可。

(4) “多态”是指当不同对象在收到同一消息时会产生不同的操作。C++ 支持两种多态性, 一种是编译时通过函数重载和操作符重载来实现的, 称之为“静态多态性”; 另外一种是在运行时通过虚函数来实现的, 称之为“动态多态性”。

在本节中, 只是介绍面向对象程序设计的基本概念, 更详细的内容将在后面的章节中进一步讲解。

### 1.2.1 对象 (Object)

面向对象的程序设计, 顾名思义, 程序设计的中心是对象, 对象也就是程序可以控制的运行实体, 程序依靠对象的特定动作来完成特定的任务。利用面向对象的程序设计方法, 把程序和数据封装起来作为一个对象, 并为每个对象赋予属性。

现实生活中各种不同的物体都可以看做是不同的对象, 选择其中的哪些进行处理, 根据

不同人的需要来定。例如，学校管理者的管理对象主要是学生，建筑师描述的对象主要是建筑体，工厂质检员的工作对象主要是生产出来的产品等。

而选择对象的目的是为了对所选对象进行控制和处理，那么这个控制和处理过程又是如何进行的呢？比如学生对象具有各种不同方面的特征，在此称为属性，包括姓名、学号、所学专业、学习成绩、兴趣爱好、基本生活需求等。那么，学校要完成对学生的有序管理，就要从学生的一些普遍共性和不同个性出发，从学习、生活中的不同方面入手实施管理：针对学生的学习，主要由教务处来安排其学习课程的内容、进度和时间；针对学生的基本生活需求，主要由后勤部门安排其住宿、就餐、洗衣等事项；针对学生的阅读需要，由图书馆进行图书的管理并制定借阅规则供学生阅读图书。

由此可见，针对对象的处理是根据对象的特征制定相应的控制过程来达到目的的，这样的原理同样适用于编程过程中的对象设计。在面向对象的程序设计中，对象的特征被称为属性，针对对象的控制过程和相关操作被称为事件和方法，这 3 者称为描述对象的 3 个基本特征。建立一个对象后，针对对象的操作可以通过与该对象有关的属性、事件和方法来描述，再把对象和描述对象的属性值以及描述对象行为方式的代码结合成为一个整体，如此动静结合，完成对象功能的描述，进而使多个对象相互配合，从而实现程序的功能。

## 1.2.2 类 (Class)

对象可以是某个甚至某些具体的实体，比如某一个学生或者某个学校的所有学生。但如果把学生这一大群体抽象出来，就被称为“类”。可以说“类”是对同一种对象的抽象，是对既具有共性又具有个性的同种对象中共性的提取。例如，不同大小、不同使用范围的球（足球、排球、网球、乒乓球等）可以被列为“球体”这一类，其共性是球表面每一点到球心的距离均相等；不同区域、不同国家和地区、不同规模和名称的城市，也可以列入城市这一类，城市的基本特征、基本功能、基本管理内容和手段都是相同的。

通过引入类的概念，可以对同一类对象共同或相似的属性、事件和方法进行统一的描述，这样在建立和使用某个对象时，就只需关注其具有个性方面的属性、事件和方法，省去大量重复的描述和操作，提高代码的可重复使用性，也便于程序的维护。

## 1.2.3 属性 (Property)

属性是指一个对象的某个方面的特征，通过数据来描述，改变对象的属性即改变对象的特征。往往一个对象具有多个方面的属性，比如，学生具有姓名、学号、年龄、籍贯、所学专业、成绩、兴趣爱好等方面的属性；一个球体具有半径、颜色、质量、质地、弹性、用途等方面的属性；对话框中的按钮则具有名称、左边界、上边界、高度、宽度、颜色、字体、字形等方面的属性。

## 1.2.4 事件 (Event)

程序员设置了对象的属性后，只是针对对象的静态特征进行了定义，但是程序总是在动

态中执行从而实现其功能的，即用户通过鼠标和键盘针对程序界面进行动态的操作。因此，用户对界面中的哪些对象进行何种操作，并且引发什么样的执行结果，则是程序员接下来要设计的工作，而这项工作是通过事件的定义来完成的。

事件是指能被对象识别的动作以及由该动作引发的结果。引发的结果需要编制代码来设定并执行，我们把这段代码称为事件过程。与事件密切相关的3个要素分别如下：

(1) 发生事件的对象。并不是每个对象都有事件发生，其中哪些对象需要定义事件过程，程序员要根据程序的功能要求进行分析和设计。

(2) 引发事件的动作。选择好事件对象后，到底是什么操作去触发事件过程的执行（是单击鼠标还是敲击键盘），是第二步需要确定的事情。

(3) 引发事件后执行的事件过程代码。触发特定对象后，将会引发怎样的结果，需要通过编制事件过程代码来描述，到此进入到根据功能要求进行程序编制的阶段。

### 1.2.5 方法 (Method)

方法是指对象要执行的某些特定动作，可以把方法理解成对象的内置函数，程序员可以直接调用而不需提供其他的命令，这样可以省去程序员为实现一些公共的、常用的功能而频繁编程的麻烦，减轻了程序员的编程工作量。方法可以由系统提供，同时程序员也可以根据自己的需要定义。

### 1.2.6 消息 (Message)

消息就是调用函数，通过消息，在对象之间可以进行交互和通讯。通过向对象发送消息来实现对象的操纵，对象根据消息的内容调用相应的方法，通常类的公有部分描述了可以发送给对象的消息，并给出了相应的方法。

## 1.3 VC++ 6.0 的程序开发环境

我们都知道，文字处理常用的编辑器有记事本、Word、写字板等工具，编写程序也需要一个编辑的环境，常用的C++编辑工具有Turbo C++、MS C++、Borland C++ Builder，在本节中，主要介绍微软的Visual C++，简称VC。源程序编辑完成后，需要利用编译器进行编译，将源码转换成计算机能够识别以及可以执行的程序，刚才所提到的程序编辑环境都具有编译的功能。

### 1.3.1 VC++ 6.0 的安装

VC++ 6.0 是 Microsoft 公司 Visual Studio 6.0 的一个组件，将安装光盘插入光驱，系统会自动运行并弹出如图 1-1 所示的 Visual Studio 6.0 企业版安装向导对话框，单击 Next（下一步）按钮，弹出最终用户许可协议对话框，如图 1-2 所示。