

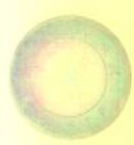
计算机程序设计与软件开发系列丛书



Visual C++实用编程技术

——从 C++、面向对象到窗口程序设计

位元文化 编著



华中理工大学出版社

HUAZHONG UNIVERSITY OF SCIENCE
AND TECHNOLOGY PRESS

图书在版编目(CIP)数据

Visual C++实用编程技术

——从C++、面向对象到窗口程序设计/位元文化编著

武汉:华中理工大学出版社, 1999年9月

ISBN 7-5609-2060-8

I. V...

I. 位...

II. Visual C++-面向对象-程序设计

N. TP311

本书封面贴有华中理工大学出版社激光防伪标志,无标志者不得销售。

版权所有 盗印必究

Visual C++实用编程技术

——从C++、面向对象到窗口程序设计

位元文化编著

责任编辑:谢燕群

封面设计:梁书亭

责任校对:郭有林

监 印:张正林

出版发行:华中理工大学出版社

武昌喻家山 邮编:430074 电话:(027)87542624

经销:新华书店湖北发行所

录排:华中理工大学惠友科技文印中心

印刷:荆州市今印集团有限责任公司

开本:787×1092 1/16

印张:31.75

字数:709 000

版次:1999年9月第1版

印次:1999年9月第1次印刷

印数:1—3 000

ISBN 7-5609-2060-8/TP·344

定价:48.00元

(本书若有印装质量问题,请向出版社发行科调换)

内 容 简 介

本书涵盖了与 C++ 及窗口程序设计的重要概念。从面向对象的角度出发,详细讲解了 C++ 实现面向对象的实践过程、面向对象与窗口程序设计的关系、Visual C++ 程序设计技巧;以图解的方式讲解了指针的概念以及指针与字符串、数组与函数间的关系;以模板概念为起点,介绍了标准模板程序库;以程序范例的实际讨论为基础,介绍了面向对象在程序代码再利用与易于扩充性方面的作用;通过运用面向对象概念建立图书管理系统的程序实例,进一步讲解了面向对象的运作方式;以简洁的窗口程序范例以及连贯的窗口程序开发过程,为读者打开一扇扇窗口程序设计的大门。

本书是一本概念与语法并陈、范例与说明并重、入门与提高兼顾的最佳 C++ 学习教材。

目 录

第 1 章 面向对象的基本概念	(1)
1.1 小心错误的概念	(1)
1.2 面向对象概念与 C++	(1)
1.2.1 思维程序设计	(1)
1.2.2 面向对象思维	(2)
1.3 面向对象概念入门	(2)
1.3.1 对象	(2)
1.3.2 类	(3)
1.3.3 信息	(3)
1.3.4 继承	(3)
1.4 程序的基本概念	(4)
第 2 章 Hello C++!	(5)
2.1 Hello C++ 程序	(5)
2.2 程序注解的使用	(6)
2.2.1 什么是程序注解?	(6)
2.2.2 区段注解	(6)
2.2.3 单行注解	(6)
2.3 载入头文件	(7)
2.4 类的建立	(7)
2.4.1 定义类	(7)
2.4.2 定义属性	(8)
2.4.3 定义方法	(8)
2.5 数据的输出——cout 对象	(9)
2.6 main() 主程序	(9)
2.7 建立对象	(9)
2.8 信息——调用对象的方法	(9)
第 3 章 变量、数据类型与常量	(11)
3.1 什么是变量	(11)
3.2 数据类型	(11)
3.2.1 整数	(12)
3.2.2 浮点数	(12)
3.2.3 字符	(12)
3.3 给变量一个好名字——匈牙利命名法	(13)

3.4 变量的输入与起始值设定	(14)
3.4.1 变量的输入	(14)
3.4.2 起始值的设定	(15)
3.5 变量的有效范围	(15)
3.6 自定义常量	(15)
3.6.1 常量	(15)
3.6.2 常量的自定义	(16)
3.6.3 const 常量	(17)
第 4 章 运算符	(18)
4.1 名词解释	(18)
4.2 指派运算符“=”的迷惑	(18)
4.3 算术运算符	(19)
4.4 比较运算符	(19)
4.5 逻辑运算符	(20)
4.6 ++、--与 sizeof 运算符	(23)
4.7 运算符的优先顺序	(24)
4.8 运算式中的类型转换	(25)
4.8.1 自动类型转换的规则	(25)
4.8.2 强制类型转换	(26)
第 5 章 数据输出/输入与格式控制	(27)
5.1 数据的输入——cin 对象	(27)
5.2 数据输出格式控制	(28)
第 6 章 流程控制	(30)
6.1 判断式与循环	(30)
6.1.1 判断式	(30)
6.1.2 循环	(30)
6.2 if...elseif...else 判断式	(31)
6.2.1 if 判断式	(31)
6.2.2 if...else 判断式	(32)
6.2.3 if...else if...else 判断式	(33)
6.3 switch...case 判断式	(36)
6.4 for 循环	(38)
6.5 while 循环	(42)
6.6 do...while 循环	(45)
6.7 break、continue、goto 语句	(48)
6.7.1 break 语句	(48)

6.7.2	continue 语句	(49)
6.7.3	goto 语句	(50)
第 7 章	数组与字符串	(53)
7.1	什么是数组	(53)
7.2	一维数组的使用	(53)
7.2.1	定义数组	(53)
7.2.2	设定起始值	(54)
7.2.3	数组的储存	(54)
7.2.4	数组元素的存取	(54)
7.2.5	小心出错	(56)
7.3	二维数组的使用	(56)
7.3.1	定义二维数组	(56)
7.3.2	二维数组元素的存取	(57)
7.3.3	二维数组的起始值设定	(57)
7.4	多维数组	(59)
7.5	字符串变量	(60)
第 8 章	指 针	(62)
8.1	变量的家	(62)
8.1.1	变量住在	(62)
8.1.2	&运算符	(62)
8.2	指针的定义与使用	(64)
8.2.1	指针的定义	(64)
8.2.2	指针的使用	(64)
8.2.3	起始值的设定	(65)
8.2.4	将数据放进指针指向的地址里	(66)
8.2.5	指针为何要定义数据类型	(67)
8.3	指针的指针	(67)
8.4	用 new 运算符设定指针起始值	(69)
第 9 章	函 数	(71)
9.1	函数与程序有什么关系?	(71)
9.2	函数的建立	(72)
9.2.1	函数的执行	(72)
9.2.2	函数的建立	(73)
9.2.3	函数的调用	(73)
9.2.4	函数的原型定义与参数值的默认	(73)
9.3	参数的传递	(76)

9.3.1 传值调用(call by value)	(77)
9.3.2 传址调用(call by address)	(78)
9.3.3 传引用调用(call by reference)	(80)
9.3.4 为何要有传地址与传引用	(82)
9.3.5 到底哪个好	(82)
第 10 章 指针与数组、字符串、函数的关系	(83)
10.1 指针与数组	(83)
10.1.1 VS 指针	(83)
10.1.2 指向数组的指针	(88)
10.2 字符串数组	(89)
10.2.1 用指针定义字符串变量	(89)
10.2.2 储存字符串的数组	(91)
10.3 指针、数组与函数	(93)
10.4 指针、字符串与函数	(95)
第 11 章 动态内存的配置	(97)
11.1 利用 new 动态定义数组的大小	(97)
11.1.1 数组的遗憾	(97)
11.1.2 动态定义数组大小	(97)
11.1.3 释放内存	(98)
11.2 动态二维数组与指针的指针	(99)
第 12 章 软件革命——面向对象	(104)
12.1 电脑的虚拟世界	(104)
12.2 面向对象思维如何模拟世界	(105)
12.2.1 面向对象的抽象化	(105)
12.2.2 模拟世界的动态与静态	(105)
12.3 面向对象系统的运作机制	(106)
第 13 章 类与对象	(107)
13.1 数据隐藏	(107)
13.2 建立类与对象	(107)
13.2.1 类的建立	(107)
13.2.2 对象的建立与使用	(109)
13.3 存取权限的控制	(111)
13.3.1 类成员的分级	(111)
13.3.2 const 的使用	(113)
13.4 对象的生命期	(116)

13.4.1	对象的产生——构造函数	(117)
13.4.2	对象状态的改变	(124)
13.4.3	对象的消失——析构造函数	(126)
13.5	类的成员	(129)
13.5.1	对象成员 VS 类的成员	(129)
13.5.2	静态属性	(130)
13.5.3	静态成员函数	(130)
13.6	指针与对象	(132)
13.6.1	对象的指针	(132)
13.6.2	动态配置对象内存	(134)
13.7	对象参数的传递	(136)
13.7.1	对象参数的传入与返回	(136)
13.7.2	this 指针的使用	(138)
13.8	类的运算	(141)
13.8.1	运算的多元性	(141)
13.8.2	重载 “=” 运算符	(142)
13.8.3	重载 “+” 运算符	(146)
13.8.4	重载 “[]” 运算符	(149)
13.8.5	类型转换与类	(149)
13.9	友元类与友元函数	(153)
13.9.1	友元类	(153)
13.9.2	友元函数	(155)
第 14 章	类继承与组合——程序代码的再利用	(157)
14.1	继承的意义	(157)
14.1.1	名词解释	(157)
14.1.2	分类的角度	(158)
14.1.3	遗传的角度	(158)
14.1.4	继承对系统发展的意义	(158)
14.2	C++ 的继承机制	(159)
14.2.1	C++ 是怎样做到的	(159)
14.2.2	继承的三种方式——public、protected、private	(160)
14.2.3	属性与继承	(165)
14.2.4	继承与成员函数	(167)
14.2.5	继承与对象的初始值设定	(170)
14.2.6	类成员的继承	(176)
14.2.7	friend 的继承	(178)
14.2.8	基础类与衍生类的类型转换	(179)
14.3	多重继承	(181)

14.3.1 多重类的继承	(181)
14.3.2 类的虚拟继承	(185)
14.4 组合	(189)
14.4.1 组合的意义与完成机制	(189)
14.4.2 以对象变量建立组合关系	(193)
14.4.3 以对象引用(指针)建立组合关系	(196)
第 15 章 信息—对象间的交谈	(201)
15.1 信息的多态性(Polymorphism)	(201)
15.1.1 信息在面向对象系统中的角色	(201)
15.1.2 信息多态性的建立机制	(202)
15.2 重载与再定义——静态的多态性	(204)
15.2.1 函数的重载	(204)
15.2.2 函数的再定义	(207)
15.3 动态链接的使用——动态的多态性	(207)
15.3.1 动态链接与继承	(207)
15.3.2 C++的动态链接机制——虚拟函数	(208)
15.3.3 虚拟析构函数	(214)
15.3.4 对象指针数组	(215)
第 16 章 模板 (TEMPLATE)	(219)
16.1 模板的概念	(219)
16.1.1 链接列表	(219)
16.1.2 模板类	(223)
16.2 模板函数	(227)
16.3 多参数模板	(228)
第 17 章 数据流与文件的输出/输入	(231)
17.1 简介数据流	(231)
17.1.1 什么是数据流	(231)
17.1.2 数据流类	(232)
17.2 数据流的控制	(232)
17.2.1 数据流的格式控制	(232)
17.2.2 格式标志	(233)
17.2.3 运算符	(235)
17.2.4 函数	(236)
17.2.5 常用的输出/输入函数	(237)
17.2.6 自定义类的输出/输入	(239)
17.2.7 数据流的错误侦测	(240)

17.3	文件的输出/输入	(242)
17.3.1	fstream 类	(242)
17.3.2	格式化文字模式的文件 I/O	(245)
17.3.3	二进制模式的文件 I/O	(247)
17.3.4	文件指针的使用	(249)
17.3.5	利用文件储存对象	(254)
第 18 章	ANSI/ISO C++标准的新增语法	(257)
18.1	namespace	(257)
18.1.1	为何要有 namespace	(257)
18.1.2	namespace 的定义与使用	(257)
18.2	新的头文件载入方式	(259)
18.3	执行时间类型信息——RTTI(Run-Time Type Information).....	(260)
18.4	类型转换	(261)
18.5	bool、mutable 与 explicit 构造函数	(267)
18.6	标准字符串类	(269)
18.6.1	简介标准字符串类	(269)
18.6.2	字符串对象的操作	(269)
18.7	例外处理	(273)
18.7.1	简介例外处理	(273)
18.7.2	例外处理的机制	(274)
18.7.3	函数的例外处理	(275)
18.7.4	例外对象	(278)
18.7.5	多重例外处理	(280)
18.7.6	标准例外处理类	(281)
第 19 章	标准模板程序库(STANDARD TEMPLATE LIBRARY)	(283)
19.1	认识 STL	(283)
19.1.1	什么是 STL	(283)
19.2	容器与指位器	(284)
19.2.1	容器	(284)
19.2.2	指位器	(285)
19.3	序列容器	(287)
19.3.1	vector	(287)
19.3.2	deque	(290)
19.3.3	list	(293)
19.4	关联容器	(296)
19.4.1	set 与 multiset	(296)
19.4.2	map 与 multimap	(299)

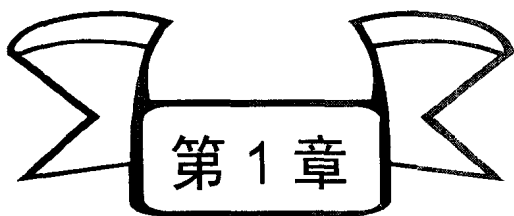
19.5	算法.....	(302)
19.5.1	算法的介绍.....	(302)
19.5.2	数据编辑算法.....	(303)
19.5.3	搜寻算法.....	(305)
19.5.4	比较算法.....	(306)
19.5.5	排序相关算法.....	(308)
19.5.6	计算算法.....	(309)
19.6	函数对象.....	(311)
19.6.1	函数对象的应用.....	(311)
19.6.2	STL 提供的函数对象.....	(312)
19.6.3	自定义函数对象.....	(314)
19.7	适配器.....	(316)
19.7.1	简介适配器.....	(316)
19.7.2	stack 容器.....	(317)
19.7.3	queue 容器.....	(318)
19.7.4	priority_queue 容器.....	(320)
19.7.5	倒转指位器.....	(322)
19.7.6	插入指位器.....	(324)
第 20 章	多文件程序的建立与图书管理系统范例.....	(326)
20.1	建立多文件程序.....	(326)
20.1.1	为何要建立多文件程序.....	(326)
20.1.2	如何建立一个多文件程序.....	(326)
20.2	图书管理系统的开发.....	(328)
20.2.1	系统的开发过程.....	(328)
20.2.2	图书管理系统的分析与设计.....	(329)
20.3	图书管理系统的建立.....	(335)
20.3.1	系统的结构.....	(335)
20.3.2	主程序、Object 与 LibraryObject 类.....	(335)
20.3.3	Librarian 类.....	(337)
20.3.4	Book 类.....	(341)
20.3.5	Reader 类.....	(342)
20.3.6	Database 类.....	(344)
第 21 章	窗口程序的基本概念.....	(348)
21.1	窗口程序的运作机制.....	(348)
21.1.1	事件、信息与窗口运作.....	(348)
21.1.2	窗口的基本构造.....	(349)
21.1.3	资源的概念.....	(349)

21.2	如何编写窗口程序	(349)
21.2.1	什么是 Application Frameworks.....	(349)
21.2.2	AF 与面向对象	(350)
21.2.3	MFC 的类继承层次	(350)
21.3	Hello MFC!——第一个窗口程序	(351)
21.3.1	建立窗口程序的基本概念	(351)
21.3.2	Hello MFC 程序范例.....	(351)
21.3.3	CWndApp 类与程序进入点	(352)
21.3.4	窗口框架对象	(352)
21.4	建立自定义窗口	(355)
21.4.1	自定义窗口框架对象	(355)
21.4.2	MyFrame 程序范例.....	(356)
21.4.3	自定义窗口框架与资源文件的建立	(357)
21.4.4	菜单的操作.....	(358)
21.5	窗口信息处理与绘图.....	(358)
21.5.1	窗口信息的传递与处理.....	(358)
21.5.2	建立信息映射表与设备上下文的程序范例.....	(360)
21.5.3	信息映射表与回应函数的建立	(362)
21.5.4	利用鼠标绘图	(362)
21.5.5	信息方框的使用与窗口的破坏	(363)
21.6	Document/View 的基本结构.....	(365)
21.6.1	什么是 Document/View 结构.....	(365)
21.6.2	建立以 Document/View 为框架的窗口应用程序	(366)
21.6.3	Document/View 的框架与建立过程	(367)
21.6.4	文件模板类的应用	(369)
21.6.5	CView 类的使用.....	(370)
21.6.6	CDocument 类的使用.....	(370)
21.7	Document/View 框架的应用.....	(370)
21.7.1	窗口的重绘.....	(370)
21.7.2	repaint 程序范例.....	(371)
21.7.3	Document/View 的运作机制	(374)
21.7.4	窗口的重绘.....	(375)
第 22 章	菜单、工具栏、状态栏、加速键、图标.....	(376)
22.1	菜单的建立.....	(376)
22.1.1	简介菜单	(376)
22.1.2	menu 程序范例.....	(377)
22.1.3	菜单的建立与设定	(382)
22.1.4	菜单的切换.....	(382)

22.1.6	修改系统菜单与建立弹出式菜单	(388)
22.2	UPDATE_COMMAND_UI 信息	(388)
22.2.1	什么是 UPDATE_COMMAND_UI 信息	(388)
22.2.2	COMMAND_UI 程序范例	(389)
22.2.3	UPDATE_COMMAND_UI 信息的接收与回应	(391)
22.2.4	UI 元件的修改	(391)
22.3	快捷键的建立	(392)
22.4	工具栏、状态行与字符串表	(393)
22.4.1	工具栏、状态行与字符串表的使用	(393)
22.4.2	controlbar 程序范例	(393)
22.4.3	工具栏的建立	(396)
22.4.4	状态行的建立	(400)
22.4.5	字符串表的建立	(403)
22.5	使用图标、位图与光标资源	(403)
22.5.1	图标、位图与光标资源的用途	(403)
22.5.2	resource 程序范例	(404)
22.5.3	图标资源的使用	(406)
22.5.4	位图资源的使用	(407)
22.5.5	鼠标光标资源的使用	(407)
22.6	建立 painter 的窗口界面	(408)
22.6.1	painter 的窗口界面	(408)
22.6.2	painter1 程序内容	(409)
22.6.3	painter1 的程序结构	(413)
22.6.4	painter1 的自定义类说明	(414)
22.6.5	控制工具栏的隐藏与显示	(416)
第 23 章	绘图的基本概念	(417)
23.1	窗口的基本图形原理	(417)
23.1.1	图形设备接口(Graphics Device Interface)	(417)
23.1.2	设备上下文(Device Context)	(417)
23.1.3	颜色的定义	(418)
23.1.4	MFC 的 GDI 类	(418)
23.1.5	建立画笔对象	(419)
23.1.6	建立画刷对象	(419)
23.2	CDC 类的介绍	(420)
23.2.1	CDC 类的衍生类	(420)
23.2.2	CDC 类提供的形状绘制函数	(420)
23.3	画笔、画刷与绘图模式的控制	(422)
23.3.1	画笔与画刷的选取	(422)

23.3.2	绘图模式的控制	(423)
23.4	painter2 程序范例	(424)
23.4.1	painter2 的绘图功能	(424)
23.4.2	painter2 程序内容	(425)
23.4.3	painter2 的程序结构	(431)
23.4.4	painter2 的自定义类说明	(431)
23.4.5	MyView 类所处理的信息	(437)
23.4.6	动态链接的应用	(441)
第 24 章	DOCUMENT/VIEW 结构的应用	(442)
24.1	图形对象的重绘	(442)
24.1.1	记录图形对象与窗口重绘的机制	(442)
24.1.2	painter3 程序范例	(443)
24.1.3	窗口重绘的机制	(446)
24.1.4	painter3 的自定义类	(446)
24.1.5	形状对象的重绘机制	(448)
24.2	应用程序的文件储存	(452)
24.2.1	MFC 的文件储存机制	(452)
24.2.2	painter4 程序范例	(453)
24.2.3	painter4 的自定义类	(464)
24.2.4	painter4 的 Serialize 机制	(464)
24.2.5	painter4 的文件操作功能	(466)
第 25 章	对话框的使用	(468)
25.1	对话框的种类	(468)
25.2	对话框的建立	(468)
25.2.1	对话框组成	(468)
25.2.2	对话框资源与对话框对象	(468)
25.2.3	简介控制项	(469)
25.3	Modal 对话框的使用	(469)
25.3.1	painter5 程序范例	(469)
25.3.2	painter5 的自定义类	(472)
25.3.3	painter5 的结构	(473)
25.3.4	SWidthDlg 对话框类	(473)
25.3.5	对话框对象的建立	(476)
25.4	对话框的数据交换与检查机制	(477)
25.4.1	DDX 与 DDV	(477)
25.4.2	painter6 程序范例	(478)
25.4.3	数据交换与检查机制	(479)

25.4.4	DDX 与 DDV 机制的建立	(479)
25.4.5	DDX 与 DDV 函数的说明	(480)
25.5	Modelless 对话框	(481)
25.5.1	painter7 程序范例	(481)
25.5.2	painter7 的自定义类	(486)
25.5.3	数据交换机制	(487)
25.5.4	Modelless 对话框的建立	(487)
25.5.5	其他更改的部分	(490)



面向对象的基本概念

在这一章里，将简单地利用真实世界的各种个体间的关系来说明面向对象的对象、类、继承、信息这四个基本概念。让您对面向对象概念有基本的认识。

1.1 小心错误的概念

在介绍“面向对象”之前，首先要澄清两个重要概念：

1. 面向对象，它只是支持面向对象机制的程序语言
2. 面向对象是一种程序设计的思维方式

为何要在本书的开始就立即提出这两个概念呢？主要是因为太多人把面向对象跟 C++ 混为一谈，更有甚者认为 C++ 就是面向对象。探究造成这种情形的原因，是因为 C++ 的风行抢走了面向对象的锋头。

其实，C++ 只是一个基于 C 语言（C 是一种程序语言），然后加上支持面向对象概念机制而成为支持面向对象概念的程序语言。因此在先天上，C++ 就不是一个纯的面向对象语言，但目前兴起的另一种程序语言——Java 才是一个纯面向对象语言，因为它的里里外外都是以面向对象概念设计的。

1.2 面向对象概念与 C++

C++ 与面向对象概念间有什么关系呢？

1.2.1 思维程序设计

思维这两个字一直是让人很难懂的，因为它实在太抽象了，感觉上它多少带了一点艺术成分，这是惯用理性的人所不容易接近的世界。但是在真正了解面向对象后，人们比较喜欢称它为一种思维。若用思考方式，感觉上比较生硬、缺乏变动的感觉。但是在前面一节里，我却用了思考模式，目的只是希望降低思维二字所带来的不明确感，往后的章节里，本书将以思维代替思考方式。

也许有些读者想问，为什么程序设计会扯上思维呢？原因在于所有程序设计师设计出的电脑系统，都是在利用电脑模拟真实世界的某些行为。在模拟的过程中，思维充当了一个重要的角色。举个例子来说吧！请各位回想一下到图书馆借书的情形。现在绝大部分的

图书馆的还书、借书操作，都是由电脑完成的；在电脑尚未普及的时候，图书馆的借/还书操作则完全由人工来完成。换句话说，现在图书馆的图书管理程序，就是在做当年没有电脑时的人工借/还书操作，即电脑里的程序所做的是，就是模拟真实世界里的借/还书操作。思维的用途，就在于如何思考，并设计出一个图书管理程序来模拟真实世界的人工借/还书操作，以完成电脑化的图书管理操作。因此，思维就是将现实世界转化为程序模拟的方法。

1.2.2 面向对象思维

上节说明了设计思维与电脑程序的关系。用了设计思维完成电脑系统（程序）的设计与分析，相当于建筑师画好了设计图，接下来的问题就是要如何施工了。

当用了面向对象思维设计系统后，接下来当然是用具备面向对象概念的程序来建构系统！因为面向对象程序语言是符合面向对象思维的程序开发工具。虽然用面向对象概念设计程序并不一定要用面向对象程序语言编写系统，但是若不用面向对象程序语言建构系统，将有许多面向对象概念不易完成，因为这些语言根本不支持这种机制。至此澄清了面向对象概念与面向对象程序语言的关系，一个是设计的概念，一个是完成设计的工具。

C++则是面向对象程序语言中被广为接受的一种，但是 C++并不是一个纯的面向对象语言。在用面向对象概念完成现实世界的分析、设计出电脑系统的蓝图后，接下来就要用 C++这类面向对象语言来实现设计蓝图。

1.3 面向对象概念入门

前面一直在说明面向对象概念与程序设计及面向对象程序之间的关系，在这一节里，则开始说明面向对象概念。

面向对象概念其实是一个很直觉的思维模式，因为它以分析真实世界为出发点，较其他设计概念更接近人类直觉观点。它的本质在于用面向对象概念寻找、分析出现实世界的对象与对象间的关系，然后运用程序语言模拟这些现实世界的对象，最后完成电脑系统的建构。面向对象概念建构在四种基本概念之上，对象、类、信息与继承。

在下面的几个小节里，将把面向对象与现实世界作一比较，分析对象、类、信息、继承这四个基本概念与真实世界间的关系。

1.3.1 对象

对象就相当于现实世界的生物，比如：老虎、猫、狗……这些动物，在面向对象概念里，都是对象。再扩展一点来说，其实对象就是东西，也就是周围的一切有形或者无形的东西。

那面向对象概念是如何描述、模拟这些现实世界的物体的呢？回答这个问题之前，来看看现实世界的物体有啥特性。首先，物体的静态特性大概有名字、颜色、长宽……等，对面向对象概念来说，这些静态特性就是对象的属性（Attribute），所以猫是该种生物的名称，也就是该种生物的属性之一。东西如果是活的，它就有动作，而这些动作就是物体的动态特性，在面向对象里称为对象的方法（Method）或者操作（Operation），比如：猫叫就是猫这种生物的“方法”。