高等学校计算机语言应用教程

Visual C++.NET 应用教程

唐大仕　刘　光　编著

本书配光盘

清华大学出版社 · 北京交通大学出版社

高等学校计算机语言应用教程

Visual C++ .NET 应用教程

唐大仕　刘　光　编著

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

C++语言是一门面向对象的程序设计语言，是高校广泛使用的程序设计教学语言之一，而 Visual C++ .NET 则是 C++语言中最新且最为流行的编程环境。本书对 C++语言（特别是托管的 C++语言）及 Visual C++ .NET 应用编程进行了较为全面的阐述。全书分为四个部分：第一部分介绍了 C++语言基础，包括数据类型、表达式、语句、数组、指针、函数；第二部分介绍了 C++面向对象特点，包括类、对象、继承、多态性、异常处理；第三部分是 Visual C++ .NET 的基本应用，包括 Windows 窗体、控件、图形图像处理、流与文件；第四部分是 C#的高级应用，包括 ADO .NET 数据库应用、多线程、网络通信、远程调用、XML Web Service，以及与非托管代码的互操作等。

本书内容详尽，由浅入深，既介绍语法，又讲解语言机制，还注重 C++在 Windows 及 Web 中的应用。本书提供了大量典型实例，配套光盘中附有全部源程序。

本书内容和组织方式立足高等学校的教学教材，也可作为计算机技术的培训教材，还可作为学习 Visual C++ .NET 的自学用书。

图书在版编目(CIP)数据

Visual C++ .NET 应用教程/唐大仕，刘光编著．—北京：清华大学出版社；北京交通大学出版社，2006.5
（高等学校计算机语言应用教程）
ISBN 7-81082-698-0

Ⅰ．V…　Ⅱ．①唐…　②刘…　Ⅲ．C语言-程序设计-高等学校-教材　Ⅳ．TP312

中国版本图书馆 CIP 数据核字（2006）第 013431 号

责任编辑：谭文芳
出 版 者：清华大学出版社　邮编：100084　电话：010-62776969　http://www.tup.com.cn
　　北京交通大学出版社　邮编：100044　电话：010-51686414　http://press.bjtu.edu.cn
印 刷 者：北京东光印刷厂
发 行 者：新华书店总店北京发行所
开　本：185×260　印张：23　字数：589 千字　附光盘 1 张
版　次：2006 年 5 月第 1 版　2006 年 5 月第 1 次印刷
书　号：ISBN 7-81082-698-0/TP·264
印　数：1～5000 册　定价：39.00 元（含光盘）

前　言

C++语言是一门面向对象的程序设计语言，是高校广泛使用的程序设计教学语言之一，如何学好这门语言是编者一直在思考的问题。

学习程序设计语言一般可通过两条途径：一是扎扎实实地理解其基本思想（如面向对象）及基本概念（如封装、继承、多态等），如果没有对这些基本概念的真正理解，学习不可能深入，不可能有后劲；二是在应用中学习，这当然要求对基本类库（如界面元素、IO、流、数据库处理、多媒体播放等）较熟悉，并在有条件的情况下结合一些具体的项目开发来进行学习。在高等学校的程序设计课程中，最好能将以上两条途径结合起来，说得更简明一点，就是要将语言与应用结合起来。绝不是“为了学语法而学语言”，而应该“为了实际应用而学语言”。基于此，学习C++就要选择一个比较好的应用环境来进行学习。

C++语言诞生已经有二十多年了，得到了广泛的应用，而托管的C++则是C++发展的最新结果，Visual C++ .NET则是C++语言中最为流行的编程环境之一。托管的C++语言不仅继承了传统的C++的优点，并且具有托管语言的优点（如内存的自动管理），而且更为重要的是，借助于Visual C++ .NET这个开发工具，C++语言可以利用.NET作为其强大的平台，而在Windows图形用户界面、ASP .NET Web应用、XML Web Service及ADO .NET数据库等方面有广泛的应用。

本书对C++语言（特别是托管的C++语言）及Visual C++ .NET应用编程进行了较为全面的阐述。本书在内容安排上，大致可以划分为四个部分：第一部分介绍C++语言基础，包括数据类型、表达式、语句、数组、指针、函数；第二部分介绍C++面向对象特点，包括类、对象、继承、多态性、异常处理；第三部分是Visual C++ .NET的基本应用，包括Windows窗体、控件、图形图像处理、流与文件；第四部分是C#的高级应用，包括ADO .NET数据库应用、多线程、网络通信、远程调用、XML Web Service及与非托管代码的互操作等。

本书便于学习，每章前面有知识的内容提要，后面有学习要点的小结。本书提供了大量的典型实例，并且配备了适量的习题。配套光盘中附有全部示例的源程序。

本书内容和组织方式立足高等学校的教学教材，也可作为计算机技术的培训教材，还可作为Visual C++ .NET的自学用书。

虽然做了不少努力，书中仍然存在着许多缺点和不足，恳请读者批评指正。也欢迎读者与编者联系（http://www.dstang.com及dstang2000@263.net）。

编　者

2006年3月

目　录

第 1 章　C++ 与 Visual C++ .NET

C++ 是一种有长期发展历史的语言，本章介绍 C++ 及托管 C++ 的特点，以及如何使用 Visual C++ .NET 编写简单的 C++ 程序，以便能对 C++ 有一个概略性的认识。

本章要点：

- ☑ C++ 语言的发展史
- ☑ 面向对象的基本特点
- ☑ Microsoft .NET
- ☑ 托管 C++
- ☑ 使用 Visual C++ .NET 开发应用程序

1.1　C++ 语言与面向对象程序设计

在开始学习 C++ 之前，首先要了解 C++ 的发展历程，并认识到它所代表的设计思想，特别是它面向对象的程序设计思想。

1.1.1　C++ 语言及其发展

1. C++ 语言的诞生

C++ 起源于 C 语言，C++ 是经过扩展和增强的 C 语言，C++ 是 C 的一个超集（所有 C++ 编译器也能够编译 C 程序），同时，C++ 包含了面向对象程序设计的思想。

C 语言是计算机语言不断发展的结果，这个发展过程开始于 Martin Richards 所发明的 BCPL（Base Combined Programming Langurage）语言，随后 Ken Thompson 从 BCPL 语言中发展出了 B 语言。在 1970 年 Dennis Ritchie 将 B 语言发展为 C 语言，并应用在 UNIX 操作系统中。

虽然 C 是被广泛使用的程序设计语言之一，但为了能够管理日益复杂的程序，C++ 应运而生。1979 年，Bjarne Stroustrup 在贝尔实验室中发明了 C++，他最初将这种新的语言命名为“带有类的 C”，1983 年，更名为 C++。

在 C++ 中增加的许多功能都是用来支持面向对象程序设计的。从本质上来说，C++ 就是一种面向对象的 C。由于 C++ 建立在 C 的基础之上，C 程序员可以平滑地过渡到 C++。

如今，C++ 具有广泛的用途。在编译器、编辑器、帮助工具、Windows 程序、游戏及网络程序中都使用到了 C++。由于 C++ 也拥有 C 语言的高效性，因此可以用来编写许多高性能的软件。

2. C++ 的不同版本

自从 C++ 被发明以来，它经历了三次主要的修订，每一次修订都为 C++ 增加了新的特

征并作了一些修改。第一次修订是在 1985 年，第二次修订是在 1990 年，而第三次修订发生在 C++ 的标准化过程中。在 20 世纪 90 年代早期，人们开始为 C++ 建立一个标准，并成立了一个 ANSI（American National Standards Institute，美国国家标准学会）和 ISO（International Standards Organization，国际标准化组织）的联合标准化委员会。该委员会在 1994 年 1 月提出了第一个标准化草案。这个草案在保持 Stroustrup 最初定义的所有特征的同时，还增加了一些新的特征以反映当时 C++ 的发展情况。

在完成 C++ 标准化的第一个草案后不久，Alexander Stepanov 创建了标准模板库（Standard Template Library，STL）。STL 是一组通用的例程，可以使用这些例程来处理各种类型的数据。STL 使得 C++ 标准被极大地扩展了。

1997 年 11 月通过了该标准的最终草案，它比 Stroustrup 最初定义的 C++ 要复杂得多。1998 年，C++ 的 ANSI/ISO 标准被投入使用。通常，这个版本的 C++ 被认为是标准 C++。所有的主流 C++ 编译器都支持标准 C++，包括微软的 Visual C++ 和 Borland 公司的 C++ Builder。

本书要讨论的 C++ 是“托管 C++”（Managed C++），它是在标准 C++ 的基础上进行了一些扩展，它比标准的 C++ 更容易使用。托管 C++ 需要 Microsoft .NET 环境，这个环境将在 1.2 节进行介绍。

1.1.2 面向对象的程序设计

面向对象是 C++ 的重要特征，所以需要对面向对象程序设计的相关概念进行一些解释。在面向对象程序设计中包含了结构化程序设计的优点及其他一些功能强大的概念，从而可以更有效地组织程序。以面向对象的风格来进行程序设计，可以将一个问题分解为几个小的组成部分，每个组成部分都能够成为自我容纳的对象，并且包含与这个对象相关的数据和指令。通过这种方法，可以有效地降低程序的复杂性，从而能够管理规模更大的程序。

所有的面向对象程序设计语言都有 3 个共同点：封装、多态和继承。本书后面将会对这些概念进行详细的讨论，在此首先对它们作一个简单的介绍。

1. 封装

所有的程序都是由两种基本的元素组成：程序语句（代码）和数据。代码是程序执行动作的部分，而数据则是这些动作所影响的信息。封装是一种程序设计机制，它将代码及代码所处理的数据捆绑在一起，并将这些代码和数据与外界隔离以保证它们的安全性。当代码和数据以这种方式连接在一起时就得到了一个对象。换句话说，对象是支持封装的元素。

在对象中，代码或者数据都可以成为对象的私有成员或者公有成员。只有对象中的代码才能知道及访问对象的私有代码或者私有数据。也就是说，程序中对象之外的任何代码都不能访问对象的私有代码或者私有数据。通常使用对象的公有成员来提供一个接口以控制对象私有成员。

2. 多态

多态（来源于希腊语，意思是“多种形式”）是指一个接口可以被用来执行某一类通用行为的特征，而具体的行为根据特定情况的不同而不同。多态的一个简单示例是汽车的方向盘。无论使用何种驾驶系统，其方向盘（可以类比为接口）的作用都相同。也就是说，无论

汽车是手动驾驶、自动驾驶或者其他的驾驶方式，方向盘的行为都是一样的。因此，只要知道如何操作方向盘，就可以驾驶任意类型的汽车。同样的道理也可以适用于程序设计。多态的概念经常表示“一种接口，多种方法”。这意味着可以为一组相关的行为设计一个通用的接口。多态通过将同样的接口应用于某一类通用的行为来降低程序的复杂性。编译器将自动根据接口的实际使用情况来选择具体的行为（方法），而无需程序员手工来做这种选择。程序员只需要知道如何使用这个通用接口。

3. 继承

继承是指一个对象获得其他对象的属性的过程。继承是很重要的，因为它支持按层次结构分类的概念。许多知识都是通过按层次结构（从上到下的层次）分类才变得易于管理的。例如，红苹果是苹果分类中的一种，而苹果又是水果分类的一种，水果又是更大的分类——食物分类中的一种。也就是说，食物分类中的一些属性（例如，可食用、有营养）在逻辑上也适用于食物的子分类水果。除了这些属性之外，水果还有一些特定的属性（例如，多汁、有甜味），这些特定的属性将水果与其他类型的食物区别开来。而苹果类又定义了一些苹果所特有的属性（长在树上，非热带水果等）。红苹果将继承前面所有分类的这些属性，并且还可以定义一些红苹果所独有的属性。

如果没有层次结构，那么每个对象都将明确地定义它的所有属性。然而，在使用了继承之后，对象只需要定义那些可以将它从它所在的分类区别出来的特征就可以了。对象可以从它的父类中继承所有的公有属性。也就是说，继承机制可以使对象成为一个通用类的特殊实例。

随着对本书的深入学习，将会看到，C++ 中的许多特征都是用来支持封装、多态及继承的。当然，C++ 可以编写任意类型的程序，包括非面向对象的程序。

1.2　Microsoft .NET 及托管 C++

1.2.1　什么是 Microsoft .NET 框架

尽管 C++ 是一种可以单独学习的计算机编程语言，但是要真正地学好它还必须与它的应用相结合。本书所介绍的 C++ 语言与它的运行期环境——Microsoft .NET 框架仍然有密切联系，C++ 使用的函数库是 .NET 框架定义的函数库中的一部分。由于以上原因，对 .NET 框架有一个基本了解是非常必要的。事实上，本书不仅介绍 C++ 语言本身，更重要的是要介绍 C++ 语言在 .NET 环境中的应用。

1. 什么是 Microsoft .NET 框架

Microsoft .NET 是一个综合性的术语，它描述了微软公司最近才发布的许多技术。总的说来，Microsoft .NET 包括的技术领域有：.NET 框架（ .NET Framework）、.NET 企业版服务器和 .NET 语言和语言工具。

其中 .NET 框架定义了一种支持开发和执行与平台无关的应用程序的环境。它使得不同的计算机语言能协同工作，并提供了程序的安全性、可移植性，以及对 Windows 平台的统一编程模式。.NET 框架包含了两个重要的部分：一个是公共语言运行环境（Common Language Runtime，CLR），它管理程序的执行，使程序具有可移植性、支持混合语言编程，并且提供安全机制；另一个是 .NET 类库，它给程序提供访问运行期环境的能力。这些类

库提供了包括输入输出、图形用户界面、网络功能、数据库访问等多方面的功能。

2. 公共语言运行环境

公共语言运行环境（CLR）管理 .NET 代码的执行。它的工作原理是：当编译 C++ 程序时，除了可以按传统方式生成可执行代码外，还以一种可托管的方式进行编译，在后一种方式中，编译器并不输出可执行代码，而是一种特殊的伪代码。这些伪码被称为微软中间语言（Microsoft Intermediate Language，简称 MSIL）。MSIL 定义了一系列与 CPU 类型无关的可移植指令集。从本质上说，MSIL 是一种可移植的汇编语言。

当程序运行时，CLR 负责把中间代码翻译（解释）成可执行代码。任何被编译成 MSIL 的程序都能运行在已实现 CLR 的环境下。这就是为什么 .NET 框架具有可移植性的原因。

除了 MSIL，当编译托管 C++ 程序时输出的另一部分是元数据（Metadata）。元数据用来描述在程序中使用的数据，它使得程序可以与其他的代码交互，并与 MSIL 位于同一个文件中。元数据可以由编译程序自动生成，并与编译后的 MSIL 指令共同包含在二进制代码文件中。元数据携带了源代码中类型信息的描述，程序使用的类型描述与其自身绑定在一起。在系统运行程序时，系统通过读取并解析元数据来获得应用程序中的类型信息，并且保证类型的安全性。

在实际的编程中，没有必要对 CLR、MSIL 或元数据进行更多的了解，因为大部分工作是由 Microsoft .NET 框架来完成的。

3. 开发工具

.NET 为开发人员提供了功能强大的管理与开发工具。其中 Visual Studio .NET 是 .NET 的重要的集成开发工具，另外，还有一些相关的工具，如编译、连接、反汇编、证书管理、注册工具、XML 工具等。

Visual Studio .NET 中可以使用多种编程语言，其中 Visual C++ .NET 是可以用 C++ 语言来进行编程的。

1.2.2 托管 C++

用于开发 .NET Framework 的语言有 Visual C#、VB .NET 和 C++ 托管扩展（Managed Extensions for C++）。其中C++ 托管扩展是在 C++ 基础上建立起来的，为 Visual C++ 程序员开发 .NET 框架应用程序而设计。为叙述方便，我们将 C++ 托管扩展称之为“托管 C++”。

1. 什么是托管

托管（managed，又译为“受控”）是 .NET 的一个专门概念，它是融于 CLR 中的一种新的编程理念，可以简单地把“托管”视为“ .NET”。使用托管 C++ 意味着编写的代码可以被 CLR 所管理，并能开发出具有最新特性（如垃圾自动收集、程序间相互访问等）的 .NET 框架应用程序。

托管的 C++ 应用程序包括托管代码、托管数据和托管类 3 个组成部分。

(1) 托管代码：.Net 环境提供了许多核心的运行（Runtime）服务，比如异常处理和安全策略。为了能使用这些服务，必须要给运行环境提供一些信息代码（元数据），这种代码就是托管代码。使用 Visual C++ .NET 时，应选择“建立托管的 C++”项目。

(2) 托管数据：托管数据是由公共语言运行的垃圾回收器进行分配和释放的数据，与托管代码密切相关。Visual C++ .NET 数据在默认情况下是非托管数据，不过，通过使用特殊的关键字，C++ 数据可以被标记为非托管数据。

(3) 托管类：尽管 Visual C++ .NET 数据在默认情况下是非托管数据，但是在使用 C++ 的托管扩展时，可以使用“__gc”关键字将类标记为托管类。就像该名称所显示的那样，它表示类实例的内存由垃圾回收器管理（garbage collection）。

2. 托管 C++ 与标准 C++ 的主要区别

托管 C++ 是从标准 C++ 建立而来的，但它与标准 C++ 还是有一定的区别，这主要体现在以下几个方面。

(1) 广泛采用“名称空间”（namespace）

名称空间是类型的一种逻辑命名方案，.NET 使用该命名方案将类型按相关功能的逻辑类别进行分组，利用名称空间可以使开发人员更容易地在代码中浏览和引用类型。

(2) 基本数据类型的变化

托管 C++ 的数据类型更加丰富，不仅包含了标准 C++ 中的数据类型，而且新增了 __int64（64 位整型）、Decimal（96 位十进制数）、String *（字符串类型）和 Object *（对象类型）等类型。

(3) 新增__gc 关键字

一个__gc 类或结构意味着该类或结构的生命周期是由 .NET 开发平台自动管理及垃圾自动收集，用户不必自己去调用 delete 来删除，可以有效地避免诸如内存泄露等问题。

(4) 简化属性操作

在标准 C++ 中分别通过 get_和 put_成员函数来设置或获取相关属性的值，在托管 C++ 中的属性使用__property 关键字，对属性的调用就好比是对一个属性变量进行操作，简化了调用过程。

(5) 托管 C++ 的委托

在 C/C++ 中，一个函数指针就是内存地址。这个地址不会带有任何其他附加信息，如函数的参数个数、参数类型、函数的返回值类型等，因而不具备类型安全性。而 .NET 框架在函数指针的基础增加了提供类型安全的机制，称为委托（delegate）。

3. 使用托管 C++ 的好处

简单地说，托管 C++ 是一组语言扩展，它帮助 Microsoft Visual C++ 开发人员为微软 .NET 编写应用程序。托管代码由于具有垃圾回收和托管类库等功能，通常为开发人员提供了更高的工作效率。

托管 C++ 为开发人员使用 .NET Framework 提供了很大的灵活性。托管 C++ 代码可以与传统的非托管的 C++ 混合在一个应用程序中，用托管 C++ 编写的应用程序可以利用两种代码的优点。

1.3　简单的 VC++ .NET 程序

1.3.1　VC++ .NET 集成开发环境

Visual Studio .NET 或者其中的一个组成部分 Visual C++ .NET（简称 VC++ .NET）