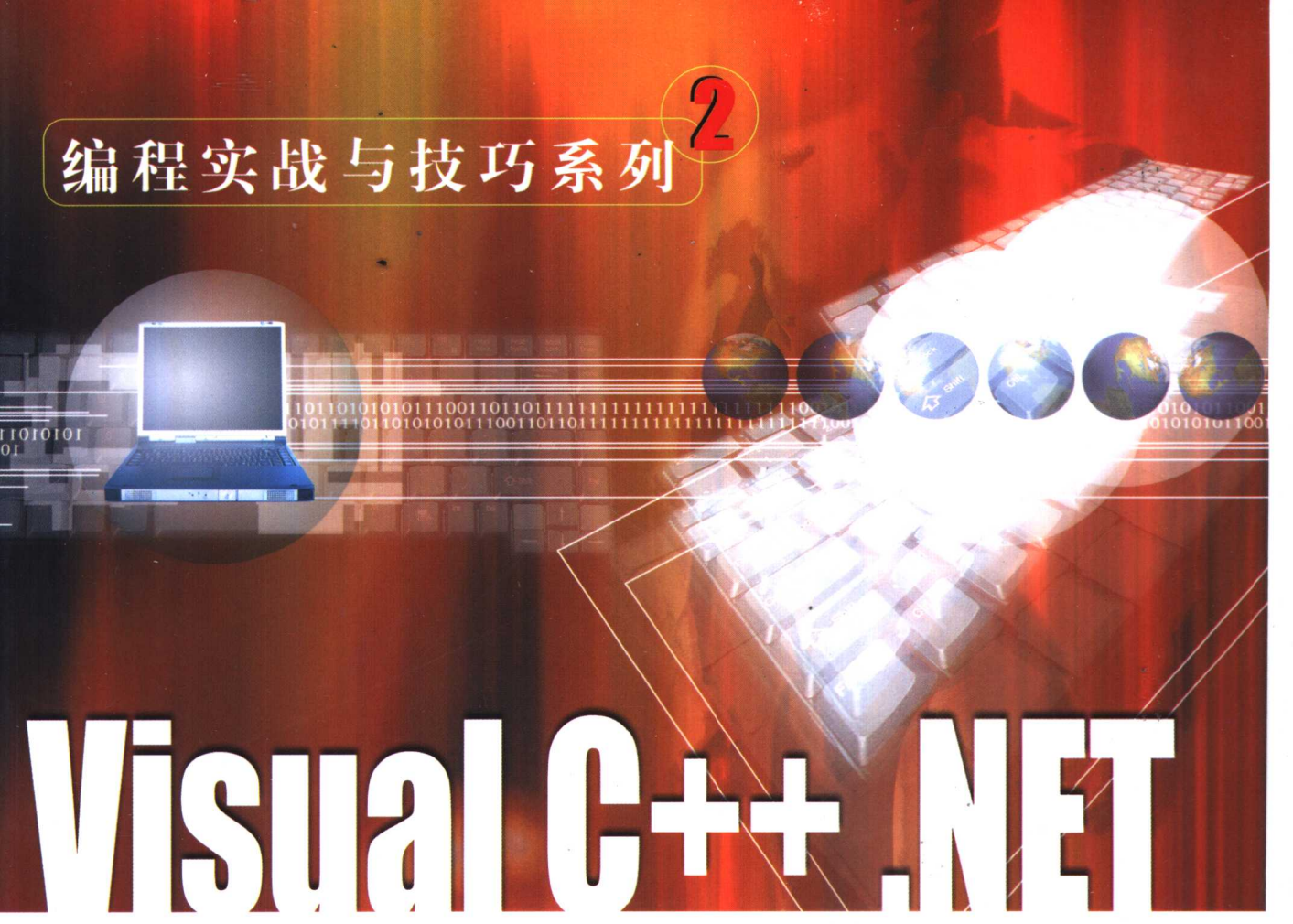

编程实战与技巧系列 2

Visual C++ .NET 精彩案例 237

北京希望电子出版社　总策划
臧桂鹏　编　写

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

编程实战与技巧系列 2

Visual C++ .NET 精彩案例 237

北京希望电子出版社 总策划
臧桂鹏 编 写

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书以237个Visual C++精彩实例全面解析了C++语言、编程的思路、方法和技巧。

该书内容涉及Visual C++编程语言，文档与视图结构，界面、对话框与控件，菜单、工具栏和状态栏，文件与系统，COM组件，数据库，多媒体，ATL与ActiveX控件，以及网络和Internet编程等各个方面。选例经典而具有代表性，分析透彻，编程逻辑思路简练，具有很强的实用性、可操作性和参考价值。

本书非常适合Visual C++初学者和中级程序员，亦可作为业余爱好者、高校计算机专业师生自学、教学用书，以及专业程序员参考书。

本版CD内容为本书237个实例的全部源代码。

盘书系列名：编程实战与技巧系列（2）

盘　书　名：Visual C++ .NET精彩案例237
总　策　划：北京希望电子出版社
文本著作者：藏桂鹏
责任编辑：但明天
出版、发行者：北京希望电子出版社
地　　址：北京市海淀区知春路甲63号卫星大厦三层 100080
网　　址：www.bhp.com.cn　E-mail: lwm@bhp.com.cn
电　　话：010-62520290,62521724,62528991,62630301,62524940,62521921,82610344（发行）
　　010-82675588-202（门市）　010-82675588-501,82675588-201（编辑部）
经　　销：各地新华书店、软件连锁店

排　　版：希望图书输出中心　马 君
CD制作者：希望多媒体开发中心
CD测试者：希望多媒体测试部
CD产生者：北京中华新联光盘有限责任公司
文本印刷者：北京媛明印刷厂
开本/规格：787毫米×1092毫米　1/16　26.5印张　611千字
版次/印次：2003年3月第1版　2003年3月第1次印刷
印　　数：1～3000册
本 版 号：ISBN 7-89498-045-5
定　　价：43.00元（本版CD）

前 言

Visual C++是 Microsoft 公司开发的 Microsoft Visual Studio 套件的一部分，它提供了一个集成的开发界面和许多有效的辅助开发工具，Visual C++.NET 是目前的最新版本。Visual C++.NET 开发平台是一个具有高度综合性的软件开发工具，在高性能的执行效率与底层控制和快速可视化开发方面均表现出色。其应用程序向导为很大一部分类型的程序提供框架代码，用户不用编写很多繁琐的程序代码。

现在的图书市场上关于 Visual C++的书籍可谓浩如烟海，然而真正能够满足读者需要的却是少之又少，如果您正在寻找一本简明实用、内容丰富的参考书，本书将是一个最好的选择。

作者从实际的应用开发中精心选取了大约 250 个典型的、非常实用的编程技术，涉及了 Visual C++程序开发的几乎各个方面，每种编程技术作为一个相对完整的知识点，由浅入深，简明透彻，从基础方法到高级编程技巧，内容丰富翔实，能够满足不同层次读者的各种需要。

每个知识点代表了一种典型的编程技术，熟练掌握它们的同时，读者还可以得到很多的编程的启发，通过大胆创新、组合扩充，读者完全可以创建自己的很有的特色的解决方案，这也正是作者选取部分知识点，借以抛砖引玉的目的。

本书内容包括 10 章，主要包括以下内容：

第 1 章是 C++语言编程基础，这是本书的语言基础，简要介绍了 C++编程的特点、方法和注意事项，有利于深入的学习 Visual C++编程。

第 2、3、4 章分别介绍文档与视图结构、对话框与控件编程以及菜单、工具栏和状态栏的编程，全面分析了 Visual C++的界面部分的编程方法。

第 5 章是文件与系统编程，主要介绍常用的文件操作以及系统编程的方法。

第 6、7 章分别介绍 COM 组件、ATL 与 ActiveX 控件编程，COM 技术应用非常广泛，是目前的主流编程技术之一。

第 8、9、10 章分别介绍多媒体技术、Internet 网络开发和数据库编程等内容。

本书由臧桂鹏执笔编写，此外，张忠东、李晓裳、范智育、王宏生、李毅、王瑾、吴浩等同志在整理材料方面给予了作者很大的帮助，在此一并致以感谢！

由于时间仓促，加之编者的水平有限，缺点和错误在所难免，恳请专家和广大读者不吝赐教，批评指正。

编 者

2002.8

目　录

第1章　C++语言编程基础

在学习 Visual C++.NET 编程之前，应该了解一下 C++语言的基础知识，尤其是面向对象的一些基本概念。本章简要介绍了 C++编程的特点、方法和注意事项。

读者应该熟练掌握本章的内容，以便为后续章节的 Visual C++.NET 编程打下良好的基础。

本章主要内容

- C++的特点
- 类说明
- 构造函数
- 析构函数
- 虚函数
- 运算符重载
- 静态变量

1　C++的封装性

在 C 语言中，编程思想是结构化的。C++比 C 有许多优点，主要有封装性、继承性和多态性，这也是 C++具有面向对象程序设计语言的特点。

封装把数据与操作结合成一体，使程序结构更加紧凑，同时避免了数据紊乱带来的调试与维护的困难。

支持数据封装，就是支持数据抽象。在 C++中，类是支持数据封装的工具，对象则是数据封装的实现。在结构化程序设计中，数据只被看作是一种静态的结构，它只有等待到调用函数来对它进行处理。在面向对象程序设计中，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，设计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。

对象被说明为具有一个给定类的变量。每个给定类的对象包含有这个类所规定的若干个私有成员、公有成员以及保护成员。

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++中的类是数据和函数的封装体，结构可作为一种特殊的类，可以包含函数，也包含私有或保护的成员。

封装简化了程序员对对象的使用，只需要知道输入什么和输出什么，而对类内部进行什么操作不必追究。封装还使得一个对象可以像一个部件一样用在各个程序中，而不用担心对象的功能受到影响。例如类 Demo：

```
#include "math.h"
#include "iostream.h"
class Demo
{
public:
    int GetValue()
    {
        SetValue();
        Value=5;
        return Value;
    };
protected:
    void SetValue()
    {
        Value=4;
    };
private:
    //数据是私有的
    int Value;
};
```

```
void main()
{
    //定义 Demo 类的对象
    Demo demo;
    int value_1;
    double value_2;
    value_1=demo.GetValue();
    //开平方根
    value_2=sqrt(value_1);
    cout<<value_2<<"\n";
}
```

输出结果是 3，对于 main()里的操作，无法更改类 Demo 的 Value 值，这样保护了数据 Value，实现了数据的隐藏。函数 SetValue()和数据 Value 结合在一起操作，实现了封装。

2　C++的继承性

继承增强了软件的可扩展性，并且为代码重用提供了强有力的手段。

在面向对象的语言中，可以从一个类派生另一个类。派生类（也称子类）继承了其基类和祖先类的数据成员和成员函数。派生类代表基类的一种改良，因为增加了新的属性或新的操作。派生类一般通过声明新的数据成员和成员函数来增加新的功能。另外，派生类在继承的成员函数不合适时，也可以弃之不用。

举例说明：水果类是一个基类，它有一些水果基本的共有特性如有营养、好吃等；桔子类、苹果类和香蕉类是水果类的派生类，它们在继承水果类的一些基本特性的基础上，增加了各自的特性，如特殊的形状、味道等。香蕉苹果和富士苹果是苹果类的派生类，它们又增加了各自的新特性。

在设计苹果类时，可以利用继承水果类的一些基本功能，再添加一些苹果的特性；同样，在苹果类的基础上，添加一些特殊的功能，就可以完成香蕉苹果类和富士苹果类的设计，从而减少了很大的工作量。

下面，首先设计水果类 Fruit：

```
Class Fruit
{
public:
    void Eat()
    {
        cout<<"Can be eaten.\n";
    }
    void Taste()
    {
        cout<<"Taste is good.\n";
```

```
    }
};
```

苹果类 Apple 需要水果类 Fruit 同样的成员函数 void Eat()和 void Taste()，另外增加了新函数 void Shape()和 void Color()。在设计时，不需要重写 void Eat()和 void Taste()函数，只需要继承水果类（public Fruit），苹果类的实现代码如下：

```
class Apple: public Fruit
{
public:
    void Shape()
    {
        cout<<"The shape is a circle.\n";
    }
    void Color()
    {
        cout<<"Diferent kinds of color.\n";
    }
};
```

其中，“calss Apple:public Fruit”表示类 Apple 公有继承类 Fruit，关键字 public 可以替换为 private，其不同的含义如表 1-1 所示。

表 1-1　访问权限的继承

派生类型	基类中的访问权限	派生类中的访问权限
pubic	public	Public
	protected	protected
	private	private
private	public	全是 private
	protected	
	private	

这样苹果类 Apple 具有 4 个成员函数：void Eat()，void Taste()，void Shape()和 void Color()函数，大大减少了代码的编写工作量。

3　C++的多态性

多态性使程序员在设计程序时可以对问题进行更好的抽象，以设计出重用性和维护性最佳的程序。多态性是一种面向对象程序设计的功能，C++可以建立具有相同成员函数名的对象，这些函数在概念上相似，但在功能上却完全不同。

例如，在各种几何形状类中，比如如点、线、正方形、矩形、圆和椭圆等，每一种类都有自己的 Draw()函数，它负责正确地画出所需要图形。调用点类的 Draw()函数，它画出的是点；调用圆类的 Draw()函数画出的是圆。简单地说，多态性就是“函数名相同，实现却不同”。

多态性的实现一般是通过在派生类中重定义基类的虚函数来实现的。

例如基类 CDrawTool 定义了 3 个虚函数 DrawBegin()，DrawTo()和 DrawEnd()，它们用于实现绘画的功能。

```
class CDrawTool
{
public:
    CDrawTool() { }
protected:
    virtual void DrawBegin(const CPoint& point) {}
    virtual void DrawTo(const CPoint& point){}
    virtual void DrawEnd() {}
};
```

类 CPenTool 继承了 CDrawTool 类，用同样的函数实现了“画点”的功能。

```
class CPenTool: public CDrawTool
{
public:
    CPenTool(): CDrawTool () {}
protected:
    virtual void DrawBegin(const CPoint& point);
    virtual void DrawTo(const CPoint& point);
    virtual void DrawEnd();
};
```

类 CFillTool 继承了 CDrawTool 类，用同样的函数实现了“填充”的功能。

```
class CFillTool: public CDrawTool
{
public:
    CFillTool(): CDrawTool() {}
protected:
    virtual void DrawBegin(const CPoint& point);
};
```

类 CBoxTool 继承了 CDrawTool 类，并扩充了虚函数 DrawObject()，用来实现“画封闭曲线”的功能。

```
class CBoxTool: public CDrawTool
{
public:
    CBoxTool(bool filled) : CDrawTool(), Filled(filled) {}
protected:
    virtual void DrawBein(const CPojnt& point);
```

```
    virtual void DrawTo(const CPoint& point);
    virtual void DrawEnd();
    virtual void DrawObject(CDC& dc) {}
protected:
    bool Filled;
    int X1;
    int X2;
    int Y1;
    int Y2;
};
```

类 CRectTooI 继承了 CBoxTool 类，具体实现了“画矩形”的功能。

```
class CRectTool: public CBoxTool
{
public:
    CRectTool(bool filled): CBoxTool(filled) {}
protected:
    virtual void DrawObject(CDC& dc);
};
```

类 CEllipSCTool 继承了 CBoxTool 类，具体实现了“画椭圆”的功能。

```
class CEllipseTool: public CBoxTool
{
public:
    CEllipseTool(bool filled): CBoxTool(filled) {}
protected:
    virtual void DrawObject(CDC& dc);
};
```

类的函数具体实现如下：

```
void CPenTool::DrawBeqin(const CPoint& point)
{
    State->DC->MoveTo(point);
}
void CPenTool::DrawTo(const CPoint& point)
{
    State->DC->LineTo(point);
}
void CPenTool::DrawEnd()
{
    State->DC->RestorePen();
```

```
}
void CFillTool::DrawBegin(const CPoint& point)
{
      ……
      //实现填充的代码
}
void CBoxTool::DrawBegin(const CPoint& point)
{
      X1＝X2= point.X;
      Y1＝Y2= point.Y;
      ……
      DrawObject( *DC );
}
void CBoxTool::DrawTo(const CPoint& point)
{
      DrawObject( *DC );
      X2 = poiot.X;
      Y2 = point.Y;
      DrawObject( *DC );
}
void CBoxTool::DrawEnd()
{
      DrawObject( *DC );
      ……
      DrawObject( *DC );
}
void CRectTool::DrawObject:(CDC& dc)
{
      //画矩形
      dc.Rectangle(x1,Y1,X2,Y2);
}
void CEllipseTool:DrawObject(CDC& dc)
{
      //画椭圆
      dc.Ellipse(X1,Y1,X2,Y2);
}
```

从上例可以看出，在不同的类中，同样的函数名可以实现不同的功能。

4　类的声明

在 C++中，类声明以关键词 class 开始，在类中所声明的内容以花括号括起来，花括号后的分号作为类说明语句的结束标志。类中定义的数据和函数称为这个类的成员，包括成员变量和成员函数。例如类名为 Hello 的声明形式如下：

```
class Hello
{
public:
    //公有的数据和函数
protected:
    //保护的数据和函数
private:
    //私有的数据和函数
};
```

C++类成员提供了 3 种可访问的权限，其中关键字分别是 public，protected 和 private。对于定义这些成员的访问权限，说明如下：

（1）public（公共部分）：只有类的成员函数、对象、派生类的成员函数及其对象可访问；

（2）protected（保护的部分）：只有类及其派生类的成员函数能访问保护的成员，对象无权访问保护的成员。

（3）private（私有部分）：只有类的成员函数能访问私有成员，对象、基类无权访问私有成员。

各个类声明的组成部分有以下原则：

（1）在一个类说明中，关键字 public，protected 和 private 出现的顺序和次数可以是任意的。例如：

```
class Hello
{
pubic:
    void init(int a,int b);
private:
    int X;
pubic:
    int GetA();
private:
    int Y;
protected:
    void SetupWindow();
};
```

以上的类说明与下面的类说明等同：

```
class Hello
{
public:
    void init(int a,int b);
    int GetA();
private:
    int X;
    int Y;
protected:
    void SetupWindow();
};
```

（2）如果没有指定类的属性，C++编译器则把成员当作保护的。

（3）成员变量一般应该放在保护的部分，避免放在公有部分，允许它们被派生类的成员函数访问。

（4）应该使用成员函数来设置和查询数据成员的值。设置操作数据成员的成员函数有助于执行合法性检查和更改其他数据成员。

（5）类可以有多个构造函数，一般放在公共部分，当只有一个时，必须放在公共部分声明。

（6）成员函数的定义一般放在一个单独的源文件中。

在类中，可以像说明普通变量一样说明类中的数据成员，数据成员可以具有任何类型，但是不能在类说明中，对数据成员使用表达式进行初始化。在类中声明的任何成员不能使用 extern，auto 和 register 关键字。一个类说明的数据成员描述了对象的内部数据结构，类中说明的成员函数用于操作对象的这些数据。除了在类中对这些成员函数进行函数声明外，还必须在程序中定义这些成员函数的功能实现。定义成员函数的一般形式如下：

```
返回类型 类名::成员函数名（参数说明）
{
    函数体
}
```

其中，“::”是作用域运算符。“类名::”用于表明其后的成员函数是属于该类的。在“函数体”中可以直接访问类中说明的成员函数和数据成员，以描述该成员函数对它们所进行的操作。

类一旦被说明后，就可以像变量 int，float 一样使用，声明它的对象。例如：

```
Hello A,B;
```

该语句声明了 A 和 B 是类 Hello 的对象。

5　初始化对象

构造函数和析构函数是在类体中说明的两种特殊的成员函数。

构造函数用于创建对象，也就是类的入口，在创建对象时，使用给定的值来将对象初始化。要调用类的成员函数，首先要通过构造函数对对象进行初始化。C++中有下列关于