

Visual C# .NET

开发技术原理 与实践教程

夏敏捷 等编著
康祥顺 审校

- 快速入门。通过与C++语言对比，使读者快速掌握C#的基础语法。
- 实例丰富。在原理篇介绍基本概念时，能结合大量的实例，使读者在C#学习中，达到事半功倍的效果。
- 实用性强。书中综合应用篇的五个实用经典案例都是实践经验的结晶，使读者提高自身的编程技能，成为更高效的软件开发人员。
- 突出设计技巧。原理篇中提供了大量的实例而且讲解了相关的C#编程技巧。
- 强调面向对象的程序设计思想。综合应用篇的经典案例中涉及大量的类，真正采用面向对象的程序设计思想进行软件开发。



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

Visual C#.NET开发技术 原理与实践教程

夏敏捷 等编著

康祥顺 审校

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书首先以实例的形式向读者展示了Visual C#.NET的编程精髓,将Visual C#.NET编程中的语法、技巧等方面的知识融在其中,由浅入深,从易到难。书中内容实例丰富、讲解清晰、努力避免代码复杂冗长。简短的实例特别有助于初学者仿效理解,把握问题的精髓;能够帮助读者快速对Visual C#.NET有整体认识。接着通过5个实用的经典案例,配以详细的代码说明和图片解释,使读者在掌握前面讲解的技术或技巧后,尝试实现更为复杂的功能,并应用到更多方面,在实践中掌握和领会Visual C#.NET编程的思想精髓。

本书适用于具有一定编程经验的Visual C#.NET用户,也可供广大计算机工作者和软件开发参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Visual C#.NET开发技术原理与实践教程/夏敏捷等编著. —北京:电子工业出版社, 2008.1

ISBN 978-7-121-05064-0

I. V… II. 夏… III. 语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2007)第145803号

责任编辑:徐云鹏

特约编辑:卢国俊

印 刷:北京天竺颖华印刷厂

装 订:三河市金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

开 本:787×1092 1/16 印张:32.25 字数:820千字

印 次:2008年1月第1次印刷

定 价:48.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至zltz@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010) 88258888。

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路173信箱

电子工业出版社总编办公室

邮 编：100036

欢迎与我们联系

为了方便与我们联系，我们已开通了网站 (www.medias.com.cn)。您可以在本网站上了解我们的新书介绍，并可通过读者留言板直接与我们沟通，欢迎您向我们提出您的想法和建议。也可以通过电话与我们联系：

电话号码：(010) 68252397。

邮件地址：webmaster@medias.com.cn

前 言

Visual C#.NET（简称**C#**语言）是微软公司针对**.NET**平台推出来的一门新语言，它作为**.NET**平台的第一语言，也是微软公司推出的下一代主程序开发语言。**Visual C#.NET**几乎集中了所有关于软件开发和软件工程研究的最新成果，如面向对象、类型安全、组件技术、自动内存管理、跨平台异常处理、版本控制、代码安全管理等。它在设计、开发程序界面的时候和以前的某些程序开发语言有所不同。它既有**Visual Basic**快速开发的特点，又不乏**C++**语言强大的功能。所以，**C#**很可能成为将来最主要的软件开发语言。

本书的目的就是让读者学会在**Visual Studio.NET**环境下，开发出功能强大的**Windows**应用程序和**Web**应用程序。

本书内容共包括两篇15章。

第一篇：**Visual C#.NET**开发技术原理

第1章主要介绍了面向对象的基本概念，包括类和对象以及需要重点掌握的面向对象的封装性、继承性、多态性三大特性思想和具体体现。并且与**C++**进行对比，便于学过**C++**的人较快地掌握**C#**语言。同时介绍**Visual Studio.NET**集成开发环境以及如何创建**C#**三种应用程序等内容。

第2章主要介绍常用控件，同时向大家展示用**Windows**窗体来编写程序的特点以及技巧。

第3章主要介绍了**Visual C#.NET**语言提供的用于文件操作的类，以及如何利用它们实现对文件的存储管理、读写等各种操作。

第4章介绍了在**Visual C#.NET**语言中，利用**.NET**框架提供的一整套相当丰富的类库绘制各种图形、处理位图图像和视频、音频，从而建立多媒体程序。

第5章主要介绍利用**.NET**框架类库中提供的应用层类**TcpClient**、**TcpListener**和**UdpClient**类来实现**Socket**编程的知识。最后通过应用层类的开发实例来说明如何利用**Visual C#.NET**进行网络编程。

第6章主要介绍组件的相关概念，以及在**.NET**框架环境下如何编写组件，向组件添加域、属性以及事件；制作自己的组件和控件的一些方法和技巧。

第7章在**ADO.NET**模型的基础上介绍如何操作数据库，使读者可以熟练掌握**ADO.NET**中各种对象的操作方法以及常用**SQL**语句，并能够读、写、检索**SQL Server**数据库，还能够利用控件通过数据绑定技术显示数据库中的数据。

第8章主要介绍了开发**Web**应用程序的**ASP.NET**工作原理和**ASP.NET**常用控件，以及在**Web**应用程序中访问数据库等。

第9章讨论了**Web**服务原理，如何创建**Web**服务和**Web**服务调用的方法和过程。

第10章介绍.NET框架中与XML相关的命名空间和其中的重要类,并通过实例使读者更进一步了解XML文件的C#读写操作。

第二篇: Visual C#.NET综合应用

通过第11章的“类QQ游戏大厅的网络五子棋”、第12章的“图像处理程序实例”、第13章的“教育局学籍管理系统”、第14章的“图书馆管理系统”以及第15章的“B/S型网上考试系统设计”的案例详细介绍了系统的开发过程,并按照软件开发方法与流程,从需求分析到编码实施,让读者真正体会到软件项目开发思想,同时又对前篇的知识进行了巩固与提高。

学习编程是一个实践的过程,而不仅仅是看书、看资料的过程,亲自动手编写、调试程序才是至关重要的。通过实际的编程以及积极的思考,读者可以很快地掌握很多编程技术,而且,在编程中读者会积累许多宝贵的经验。在当前的软件开发环境下,这种编程经验对开发者尤其显得不可或缺。

本书共15章,第2章到第5章由夏敏捷编写,第1章由李志强编写,第6章由鲁丽萍编写,第7章和第8章由李娟和丁汉清编写,第10章和第11章由贾晓辉和王佩雪编写,第9章和第13章由底恒编写,第12章由郭清宇编写,第14章由赵冬编写,第15章由高亮编写。全书最终由康祥顺审校。在本书的编写过程中,为确保内容的正确性,参阅了很多资料,并且得到了众多网站和资深Web程序员的支持。在此谨向他们表示衷心的感谢。

由于编者水平有限,书中难免有错,敬请广大读者批评指正。

为了方便读者阅读,本书配套资料请登录“华信教育资源网”(http://www.hxedu.com.cn),在“资源下载”频道的“图书资源”栏目下载。

目 录

第一篇 Visual C#.NET开发技术原理

第1章 Visual C#.NET基础知识	1
1.1 从C++到C#	1
1.2 C#面向对象编程技术	9
1.3 Visual C#.NET三种应用程序结构	30
1.4 命名空间	36
1.5 Visual C#.NET常用类	38
第2章 Visual C#.NET控件及其应用	44
2.1 特殊功能文本框	44
2.2 单选按钮应用——模拟单项选择题测试	47
2.3 复选框应用——模拟多项选择题测试	49
2.4 列表框应用——小学生做加减法的算术练习程序	52
2.5 组合框应用——国家名选择	56
2.6 Timer控件用法——飘动窗体	59
2.7 图片框应用——图片自动浏览器	60
2.8 利用滚动条控件调配颜色	64
2.9 TreeView控件和ListView控件——学校系部分层列表	66
2.10 菜单使用	70
2.11 对话框控件应用——自己的记事本编辑器程序	72
2.12 多窗体设计——输入对话框的设计	82
2.13 实现控件数组的功能——计算器设计	84
第3章 文件处理和系统操作	88
3.1 文件夹浏览器设计	88
3.2 Visual C#.NET调用Windows API函数	101
3.3 注册表编程	102
3.4 注册系统热键	107
3.5 处理鼠标和键盘事件	109
第4章 图形图像和多媒体编程	112
4.1 GDI+编程	112

4.2	截图操作	122
4.3	图像显示与保存	126
4.4	动画设计——跳舞的小女孩	128
4.5	人物拼图游戏设计	129
4.6	渐显效果	134
4.7	屏幕捕获程序	135
4.8	定时抓取当前程序窗口	141
4.9	分解Gif图像和图片格式转换	143
4.10	特殊形状的窗体界面	144
4.11	生成数字字符验证码图片	149
4.12	播放声音文件	152
第5章	网络程序开发	157
5.1	网络通信编程基础	157
5.2	使用TcpClient类和TcpListener类实现TCP协议通信	163
5.3	使用UdpClient类实现UDP协议编程	172
5.4	基于UDP的P2P两人对战网络军棋	174
第6章	组件设计	197
6.1	组件的相关概念与开发	197
6.2	汉字验证码图片控件	199
6.3	渐变背景椭圆按钮控件	206
第7章	数据库编程	210
7.1	ADO.NET数据库访问技术	210
7.2	常用SQL Server语句	215
7.3	简单数据表操作方法	224
7.4	使用DataReader顺序读取记录	230
7.5	使用DataSet和DataGrid进行复杂操作	233
7.6	数据库中的图像存取	242
第8章	Web应用程序开发	246
8.1	Web窗体与ASP.NET内置对象	246
8.2	ASP.NET常用控件	250
8.3	网页间数据的传递	256
8.4	Web表单验证控件应用	257
8.5	ADO.NET对数据库的操作——读取、修改表信息	261
8.6	Web数据显示控件应用——显示表信息	263

8.7	在ASP.NET中调用存储过程	272
8.8	上传文件	275
8.9	发布即时信息	276
第9章	Web Service开发	279
9.1	Web服务基础	279
9.2	创建并公布Web服务	280
9.3	使用Web服务	282
9.4	通过Web服务上传数据实例	283
第10章	XML技术	286
10.1	XML的概念	286
10.2	使用.NET框架中的类进行XML文件的创建与读写	296
10.3	使用ADO.NET中的DataSet进行XML文件的创建与读写	305
10.4	基于XML的Web留言板设计	307
10.5	图像和XML格式文件相互转换	312
 第二篇 Visual C#.NET综合应用 		
第11章	类QQ游戏大厅的网络五子棋	315
11.1	功能需求分析	315
11.2	系统设计	315
11.3	客户端功能模块实现	318
11.4	服务器端功能模块的实现	344
第12章	图像处理程序实例	354
12.1	图像处理基础	354
12.2	图像处理基本技术	360
12.3	图像处理程序实例	375
第13章	教育局学籍管理系统	389
13.1	功能需求分析	389
13.2	数据库的设计	391
13.3	系统功能的设计	402
13.4	具体模块实现	404
13.5	系统的改进与扩展	422
第14章	图书馆管理系统	430
14.1	需求分析	430

14.2	数据库分析与设计	432
14.3	系统分析与设计	437
第15章	B/S型网上考试系统设计	462
15.1	功能需求分析	462
15.2	系统设计	462
15.3	系统实现	467



Visual C#.NET

开发技术原理

第1章

Visual C#.NET基础知识

.NET是微软推出的一种面向网络、支持各种用户终端的开发平台。Visual Studio.NET环境是Microsoft为实现其.NET技术的开发工具，在构建基于Windows平台的各种程序时，它为程序员提供了非常出色的开发环境。利用Visual Studio.NET，用户可以非常轻松地创建具有自动伸缩能力的可靠的应用程序和组件。

Visual Studio.NET集成开发环境包含Visual Basic.NET、Visual C++.NET、Visual C#.NET和Visual J#.NET四种编程语言，并允许它们共享工具，这有助于创建混合语言解决方案。

Visual C#.NET是微软公司针对.NET平台推出来的一门新语言，作为.NET平台的第一语言，也是微软公司推出的下一代主程序开发语言。Visual C#.NET（简称C#语言）几乎集中了所有关于软件开发和软件工程研究的最新成果，如面向对象、类型安全、组件技术、自动内存管理、跨平台异常处理、版本控制、代码安全管理等。它在设计、开发程序界面的时候和以前的某些程序开发语言有所不同。它既有Visual Basic快速开发的特点，又不乏C++语言强大的功能。所以，C#很可能成为将来最主要的软件开发语言。

1.1 从C++到C#

C#语言从C/C++演变而来。它是现代、简单、完全面向对象和类型安全的。作为一个C++程序员，学习C#其实很简单，需要接受的新知识不多，相对其他语言来说，应该可以更加平和地过渡到C#的开发中。这里只讲解和C++有区别的地方，相同的部分就一带而过，这样的对比学习可以让我们在已有知识的前提下快速掌握C#。

1.1.1 程序结构

首先以Hello World为例将C#程序和C++程序做一个比较：

C++:

```
#include <iostream>
using namespace std;
void main()
{
```

```
//C++程序
/* C++程序*/
cout<<"Hello World!"<<endl;
}
```

C#程序:

```
using System;
namespace HelloWorld
{
    class Class1
    {
        static void Main()// C#强制的, 是程序的真正开始!
        {
            Console.WriteLine ("Hello World!");
        }
    }
}
```

首先对于程序的进入点, 最大的区别就是Main函数的开头必须要大写。对于主函数必须要有有限定词static表明Main函数是静态的, 在内存中只能有一个副本。

C#是一个完全面向对象的程序语言, 所以它的所有代码都必须定义在一个类中, Main函数也不例外, 同时, 因为.NET程序在编译运行时都是先转为中间语言, 然后中间语言再编译为机器语言, 这样有两个好处:

(1) 如同Java一样, 写好的程序可以在不同的系统中运行, 而不需要改变程序。

(2) 使用不同的语言编写的程序, 因为要转化为相同的中间语言, 所以在程序开发中可以使用不同的程序语言编写, 而相互调用。

在写C#程序时, 必须把程序写在一个命名空间(或叫名称空间)内。

第一行中的using System和C++中使用using namespace std的含义相似。使用using关键字来简化对命名空间内包含的名称的访问。System命名空间是.NET应用程序的根命名空间, 包含了控制台应用程序所需要的所有基本功能, 就如同C++中的头文件包含在std这个命名空间中一样。

Console.WriteLine("Hello World!");中的Console是System命名空间中的一个类, 其有一个WriteLine方法, 它的作用和cout一样, 输出一行字符串。

1.1.2 数据类型

C#中的数据类型和C++是类似的。在C#中数据类型分为两种: 值类型和引用类型。值类型包括一些简单类型(例如, char、int和float)、枚举类型和结构类型。引用类型包括类、接口、委托(delegate)和数组类型。在C#中, 内置数据类型除了字符串(string)类型与对象(object)类型外, 其余均为值类型。在计算机中, 值类型的数据存储使用堆栈, 其数据长度是不变的, 值类型变量直接包含了数据, 而引用类型的数据存储使用堆, 其数据长度是可变的。

1. 浮点类型

浮点类型中添加了一个精度更高的decimal类型。对于金融方面的程序开发, 此种数据类型用来定义钱币。

2. Bool类型

Bool类型的变量只能赋值为：**false**和**true**，虽然它们的含义仍然是0和非0，但是在使用中不能再给它们赋值成整数值。

```
bool x = 1; // 错误，不存在这种写法
bool y = 1; // 错误，不存在这种写法
bool x = true; // 正确，可以被执行
```

3. 字符类型

char类型在C#中是16位的Unicode字符集表示形式，用它来表示世界上大多数语言。其取值范围为'\u0000'到'\uFFFF'，即0到65535。对于一个字符类型变量，我们可以采用如下方式对它进行赋值：

```
char c = 'H'; // 字符H
char c = '\x0048'; // 字符H，十六进制转义符（前缀为\x）
char c = '\u0048'; // 字符H，Unicode表示形式（前缀为\u）
char c = '\r'; // 回车，转义字符（用于在程序中指代特殊的控制字符）
```

常见转义字符如表1-1所示。

表1-1 转义字符

转义字符	字符名
'\'	单引号
'\"'	双引号
'\\'	反斜杠
'\n'	换行
'\r'	回车
'\t'	水平Tab
'\v'	垂直Tab

注意，它也不能接收一个整数值。

```
char c=13; // 错误，不存在这种写法
```

4. 整数类型

整数类型中添加了**byte**（8位无符号整数）、**sbyte**（8位有符号整数）、**short**（16位有符号整数），**long**变成了真正的64位有符号整数，它可以用在64位机器的编程中。

顾名思义，**uint**、**ushort**、**ulong**是没有符号的整数，它和C++中的**unsigned int**是一样的，只不过名字换了一下而已。

5. 字符串类型

string类型是字符串类型，它是引用类型，它的使用方法和C++中**string**的使用相似，**string**类型其实就是**System.String**类的别名。在程序中使用**string**或**System.String**作用是一样的。C#将字符串封装成一个类，这样有一些方法和属性可以为我们使用。例如：**ToCharArray()**把字符串放入一个字符数组中；**Length**属性返回字符串的长度，可以进行+运算等。

比如:

```
string str1 = "How "+"are you ?" ; // "+" 运算符用于连接字符串
char c = "Hello World!"[2]; // "[" 运算符可以访问string中的个别字符, c = 'e'
string str2 = "Nice to meet you !";
bool b = (str1 == str2); // "=" 运算符用于两个字符串比较, b =false
```

6. 类型的转换

C#是一个强类型的语言, 它的数值类型有一些可以进行隐式转换, 其他的必须显式转换。隐式转换的类型只能是长度短的类型转换成长的类型, 例如, int可以转换成long、float、double、decimal; 反之必须显式地转换。例如:

```
int a=7;
float b=a; //隐式转换
a=(int)c; //显式转换
```

上面的显式转换不能用在bool和string类型上, 如果希望string或者bool类型和整数类型之间进行转换, 可以使用方法Convert.To数据类型(val)。例如:

```
int a=123;
string str=Convert.ToString(a);
bool m_bool=Convert.ToBoolean(a);
```

7. 枚举类型

C++和C#的枚举类型, 定义相同, 使用也相同, 只要注意C#中定义枚举类型时只能放在执行代码外面。

例1-1 使用枚举类型。

```
using System;
namespace ConsoleApplication1
{
    class Class1
    {
        enum week {monday,tuesday,wednesday,thursday,friday,saturday,sunday};
        static void Main()
        {
            week day=week.friday ;
            int a=(int)day;
            int b=(int)week.saturday ;
            Console.WriteLine("a={0},b={1}", a, b);
            Console.ReadLine();
        }
    }
}
```

枚举元素作为整型是有值的, 编译系统按照其在枚举类型定义中的位置顺序对它们进行赋值, 默认情况下它们的值依次为0、1、2等。

所以运行结果为:

```
a=4,b=5
```

8. 指针类型

C++中奉为经典的指针类型, 在C#中已经被取消了, 不过在易用性方面来说应该是一个

进步。在C#中也可以包含不安全代码，这些代码就是使用了指针的代码。

9. 结构类型

C#中的结构类型看上去和C++没有什么区别，定义也相似，但还是有很大的区别的。最大的区别就是C#中的结构已经和类相似了，其定义中可以包括构造函数、析构函数、成员变量、成员函数、成员属性等，同C#中的类基本相同。

结构和类的区别就是结构不能继承，而且类是引用类型，结构是值类型，结构是在栈中创建的空间，所以结构最好是对少量的数据进行操作。

10. 数组类型

数组的定义和C++有区别，定义语法为：

<类型>[<变量名>

例如：

```
int[] num;
```

这样就定义了一个int类型的数组，但是切记，它可以和C++不同，[]里面可以不写内容。确定数组的大小有两个办法：

(1) 在定义时指定初始数据。例：

```
int[] num={5,3,7,3};
```

(2) 使用关键字new。例：

```
int[] num=new int[4];
```

当然两者也可以合起来，例：

```
int [] num=new int[4] {5,3,7,3};
```

注意，前面定义了4个数据，后面花括号里面就必须有4个数据，不然就是错误的。例：

```
int [] num=new int[4]{4,3} //错误;
```

对数据的使用和C++一样，但对多维数组的定义和C++是不同的。

C++中定义为：

```
int num[3][4]={1,2,3,4,5,3,2,3,4,2,3,4};
```

C#中定义为：

```
int[,] num=new int[3,4]{{1,2,3,4},{5,3,2,3},{4,2,3,4}};
```

C#在数据列表中必须使用{}将数据分组，而在C++中可以不使用{}将数据分组。

对元素的使用也相应地变为：

```
num[2,1]=3;
```

C#在数组中最富革命性的改变是添加了锯齿形数组，例如它可以添加一组{{1,2,3,4},{2,3},{2,3,1}}长度不一样的数据，在C++中只能创建一个[3][3]的数组，在C#中它能够产生这样一个锯齿形数组，第一组中有4个数据，第二组中有2个数组，第三组中有3个，这样可以不浪费内存。

锯齿数组的定义和前面的定义也有区别，它更像是一个数组中包含了一个数组，

```
int[][] num=new int[3][];  
num[0]=new int[4]{1,2,3,4};  
num[1]=new int[2]{2,3};  
num[2]=new int[3]{2,3,1};
```

下面是上面的定义的一种简洁写法：

```
int[][] sum={new int[]{1,2,3,4},new int[]{2,3},new int[]{2,3,1}};
```

1.1.3 变量、常量和运算符

变量的使用和作用域与C++类似，但是C#程序都是包含在类当中了，也就是说，定义的变量都变成了类的私有成员。

常量的定义使用const关键字，和C++类似。

C#中的运算符其优先级和C++一样，但是需要注意这四个运算符：*、->、&、sizeof。

上面的四个运算符在C#的不安全代码中可以使用，但在一般的C#代码中使用是错误的，C#取消了指针，当然和指针有关的操作符都不能用了。

1.1.4 流程控制和函数

1. if/if...else 语句

这个语句和C++没有区别，这里需要注意一个问题，看下面的程序：

```
int a;  
if (a==Getnum()) //假设Getnum返回一个整数值  
{  
    //a为非0  
}  
else  
{  
    //a为0  
}
```

在C++中，上面的代码是正确的，但是在C#中是错误的。if的条件表达式必须为bool型，这样做可以增强代码的安全性。

正确用法应该是：

```
int a=Getnum();  
if (a!=0) //假设Getnum返回一个整数值  
{  
    //a为非0  
}  
else  
{  
    //a为0  
}
```

2. switch

选择分支语句和C++用法也相似，但是C#中对它的格式要求更加严格了，每个case语句后面都必须跟上break，不然就是错误的语句。当然，如果有多个条件执行相同语句的话，可

以省略写成下面的方法:

```
switch (char c)
{
    case 'a':
    case 's':
    case 'e' : string language="English";break;
}
```

3. 循环语句

C++中的3种循环语句: for、while、do...while和C#中是相同的。

但是C#中添加了一种循环语句foreach, 它在对数组的输出方面非常灵活。

foreach的语法格式:

```
foreach (<变量类型> <变量名> in <数组名>)
{
    循环体;
}
```

不管是一维还是多维数组, 操作都一样方便, 它会自动知道数组的大小对其操作, 我们不需要关心它是否会溢出。注意, foreach语句只能对数据进行输出, 而不能改变任何数组元素的值。

使用foreach的例子:

```
int[,] a1=new int[2,2]{{1,2},{3,4}};
foreach (int b1 in a1)
{
    Console.WriteLine ("{0}",b1);
    b1=3; //错误
}
```

4. 跳转语句

C#中的跳转语句和C++中一样, 有: goto、return、break、continue。

5. 异常处理语句

(1) try语句

用于捕捉在程序执行期间发生的异常, 指定在try块内发生异常时执行的代码。常用格式:

```
try
{
    语句组;           //程序中需要执行的语句
}
catch(异常1)          //发生了“异常1”指定的异常
{
    语句组1;          //执行该异常处理
}
.....
catch(异常n)          //发生了“异常n”指定的异常
{
    语句组n;          //执行该异常处理
}
finally               //不论是否产生异常, finally块总是被执行
```