

微软.NET程序员系列

Microsoft Press

- ◆ 欧美读者评价 ★★★★★
- ◆ 微软官方权威文档
- ◆ 辞典风格语言精炼
- ◆ 深入阐述C#编程语言
- ◆ 程序员必备之参考手册

MICROSOFT®

VISUAL C# .NET

语言参考手册

[美] 微软公司 著
熊盛新 许志庆 李 钦 译
Visual Studio .NET产品组 审校

Microsoft
.net



清华大学出版

微软.NET 程序员系列

TP312C
W4405

Visual C# .NET 语言参考手册

[美] 微软公司 著
熊盛新 许志庆 李钦 译
Visual Studio .NET 产品组 审校

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书主要介绍 C# 编程语言, 着重点是编程而非语言规范。本书分两大部分, 共 26 章。主要介绍了 C# 关键字和 C# 特性, 具体内容包括: 关键字, 运算符, 属性(Attribute), 声明, 成员, 预处理器指令, 数组, 构造函数和析构函数, 索引器, 属性(Property), 签名和重载, 范围, 参数传递, Main 方法, 条件方法, 委托, 事件和 XML 文档等。

本书语言精炼, 结合丰富的程序代码示例, 对 C# 语言的编程语法进行了详细讲解。本书适用于有一定经验的编程人员。

Microsoft Visual C# .NET Language Reference

Microsoft Press

Copyright © 2002 by Microsoft Corporation

Original English language edition published by Microsoft Press, a Division of Microsoft Corporation

All rights reserved.

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China only.

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行, 未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

图书在版编目(CIP)数据

Visual C#.NET 语言参考手册/美国微软公司编;熊盛新译. —北京:清华大学出版社,2002

(微软.NET 程序员系列)

书名原文: Microsoft Visual C#.NET Language Reference

ISBN 7-302-05776-1

I. V… II. ①美…②熊… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 058567 号

北京市版权局著作权合同登记号: 图字 01-2002-3160 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 李朋朋

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 24 彩插页: 4 字数: 550 千字

版 次: 2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

书 号: ISBN 7-302-05776-1/TP·3414

印 数: 0001~4000

定 价: 49.00 元

《微软.NET 程序员系列》序

自 2000 年 6 月微软宣布自己的 .NET 战略以来,在不到两年的时间里,.NET 已经从战略变成现实。.NET 带来了全新的、快速而敏捷的企业计算能力,也给软件开发商和软件开发人员提供了支持未来计算的高效 Web 服务开发工具。作为微软 .NET 战略的重要组成部分——Visual Studio .NET (中文版)已经于 2002 年 3 月 22 日正式在中国推出。

Visual Studio .NET 是一个功能强大、高效并且可扩展的编程环境。它充分展现了应用程序开发的潜能,并提供了生成应用程序所需的工具和技术。这些应用程序将给当今的企业、机构提供强大的支持,并推动下一代基于 XML Web 服务软件的发展。

有了 Visual Studio .NET,那些对全世界数百万的专业和业余程序员来说曾一度极端复杂、费时费力,甚至让人望而生畏的编程任务现在已不再神秘。更重要的是,Visual Studio .NET 使开发人员能运用既有的技能和知识来迎接新的编程挑战。

在 10 年前,Visual Basic 1.0 成为数以百万计的开发人员的革命性的应用程序开发语言。现在,Visual Studio .NET 为未来的 10 年做好了准备。

微软出版社为了配合 Visual Studio .NET 的推广以及 .NET 技术的普及,邀请 Visual Studio .NET 项目开发组的核心开发人员和计算机图书专业作家精心编写了英文版《微软.NET 程序员系列》丛书;该丛书自面市以来,在美国图书销量排行榜上一直高居前列,颇受好评,成为程序开发人员和网络开发人员了解 .NET 技术的权威工具书。尤其是《Microsoft .NET Framework 应用程序设计》一书,长期占据美国及欧洲此类书籍的排行榜冠军位置,程序开发人员不可不读此书。

清华大学出版社为了满足中国广大程序开发人员、网络开发人员学习最新技术的渴望,在微软出版社的配合下,从《微软.NET 程序员系列》这套丛书中精选了 40 本翻译成中文,以满足国内广大读者的需要。这套丛书阵容庞大,几乎涵盖了 .NET 技术及其应用的各个方面;也正因为如此,翻译和编辑加工的工作量也大得惊人。但为了保持国外优秀技术图书的魅力,同时使读者领会新技术的真谛,本丛书的翻译和编辑都是经过严格筛选的、具有很高的翻译水平或丰富编辑经验的技术人员;另外,我们还聘请微软公司 Visual Studio .NET 产品组的技术专家审读每一本书,确保在技术上准确无误。

相信这套丛书定会帮助程序开发人员、网络开发人员以及那些具有一定编程基础的中、高级读者,快速、全面掌握 .NET 技术,协助他们为技术生涯的下一个 10 年做好准备,为培养新一代软件人才,并推动中国软件产业的快速发展起到积极的作用!

这套丛书中的《C#技术内幕》和《C#编程技术》已于去年与读者见面,并得到读者的广泛好评。目前,本丛书中在编的共有 17 本,将从 6 月份起陆续和读者见面。这些书目包括:

- **《Microsoft .NET Framework 程序设计》**

全面、详细地介绍了 Microsoft .NET Framework，可以帮助开发人员和设计人员轻松、高效地创建高性能且安全可靠的 .NET 应用程序。
- **《应用程序升级——Visual Basic 6.0 到 Visual Basic .NET》**

升级 Visual Basic 6.0 程序代码的最佳指导。涵盖了 Visual Basic .NET 的全部新功能和各种方案，这些方案能够以最小的中断进行代码的移植并维护混合环境。
- **《ASP .NET Web 应用程序开发新思维》**

介绍了最新 Web 应用程序构建技术，带给您 Web 应用程序开发的新思想。帮助开发人员充分利用 Microsoft .NET，开发高效、安全的 Web 应用程序。
- **《构建 Web 解决方案——应用 ASP.NET 和 ADO.NET》**

介绍 ADO.NET 与 ASP.NET 的综合应用，使开发人员可以将 Web 开发技术与数据库开发技术完美地结合起来，构建功能更加强大的 Web 应用程序和服务。
- **《Microsoft Windows 网络编程（第 2 版）》**

主要讲述 Winsock 网络编程技术。详细介绍了如何编写高性能、可扩展的 Winsock 应用程序，还讲述了如何使用 C# 开发 Winsock 程序。
- **《构建 XML Web 服务——基于 Microsoft .NET 平台》**

详细深入地讲解了用于创建 XML Web 服务的主要协议和工具，并深入探讨了如何利用它们构建功能强大、高效的 Web 商业解决方案。还介绍了 Microsoft .NET My Services 以及其他基于 XML 的微软新技术。
- **《Microsoft .NET My Services 精解》**

该书由 My Services 开发组的专家共同编写。详细解释了各项 .NET 服务，并介绍了 .NET My Services 消息界面模型、数据操纵语言、安全授权模型、系统文档结构以及管理模型，帮助您创建出充分利用 .NET My Services 优点的 Web 服务。
- **《移动设备 .NET 应用程序设计》**

介绍了如何使用 ASP.NET、Visual Studio .NET 和移动通信 Internet 工具箱创建 Web 应用程序，在各种移动设备上以适当格式动态显示相同页面。有助于学习如何使用微软的信息服务器，以及如何为移动设备提供 E-mail 访问。
- **《SQL Server 2000 与 Visual Basic .NET 编程》**

介绍如何使用 SQL Server 2000、Visual Basic .NET、ADO.NET、ASP.NET、XML Web 服务和其他数据工具，开发数据库访问应用程序。
- **《Microsoft .NET Server 企业解决方案》**

介绍如何计划、开发和部署企业电子商务解决方案。将理论与实际紧密结合，详细讲解了如何解决电子商务集成时遇到的问题，并进一步探讨了使用 Microsoft .NET Server 的各种工具和技术解决具体的问题。

为了使读者的学习目标更为明确，在本丛书中，又划分出了两个子系列，现有 7 本书：

● Core Reference 系列

目前共有 4 本：《Visual Basic .NET 核心编程》、《Visual C#核心编程》、《ADO.NET 核心编程》和《Microsoft .NET 核心编程》。已经列入我们的出版计划的还有另外 3 本：

《Visual C++ .NET 核心编程》、《Visual J# .NET 核心编程》和《Microsoft .NET Framework 核心编程》。

这些书分别介绍相应编程语言和.NET 框架类库的指令、示例代码、最佳编程惯例和基于案例的解决方案，内容详尽，讲解深入，集指导性与实用性于一体，最适合相关的开发人员用作专业读本。

● Language Reference 系列

目前共有 3 本：《Visual Basic .NET 语言参考手册》、《Visual C# .NET 语言参考手册》和《Visual C++ .NET 语言参考手册》。以简洁、易于浏览和使用的形式，从 A 到 Z 列出了这 3 种编程语言的技术参考。

随着技术的发展，我们将根据读者的需要，不断增加新的书目。

丛书版式特色

本丛书在风格上力求文字精炼。并采用小 5 号字编排，内容紧凑，版面清晰美观，易于阅读。此外，书中还安排了一些特色段落，提供正文之外的一些细节知识：



注意：提醒阅读和操作过程中应注意的事项，避免出现错误或问题。



提示：指点一些操作捷径或实用技巧，使您少走弯路，阅读和操作更为高效。



要点：总结关键知识点或操作细节，帮助您适时掌握要领。



注：提示首次出现的编程元素，以及书中涉及到该元素的其他位置以供参考。



警告：对可能导致不良后果的操作提出警告。

尽管我们倾心相注，精心而为，总有疏忽纰漏，恳请广大读者不吝赐教与指正，我们定会全力改进，以期在后续工作中得以完善。

本丛书在创作过程中得到了微软(中国)公司的大力支持。本丛书能够顺利出版，更是倾注了无数幕后人员的汗水和心力。在此，对他们的辛勤劳动一并表示衷心感谢！

编者

2002 年 6 月

引 言

C#程序设计

Visual C#是 Visual Studio .NET 中引入的一门程序设计语言。本书提供了关于 C#编程的参考信息，主要侧重于程序设计而非 C#语言的规范。

本书分为以下两个部分：

- 第 I 部分，“C#关键字” 描述每个关键字的用法、可能出现的环境、示例代码及其输出，并且介绍了最佳编程方法。
- 第 II 部分，“C#特性” 讨论了 C#编程的各种元素，包括运算符、属性、预处理器指令、数组、构造函数和析构函数、索引器、参数传递、main 方法、条件方法、委托、事件和 XML 的使用。本部分提供了这些元素的定义，并设计了示例代码以满足具体的需要，以及代码的运行和输出，还介绍了最佳编程惯例。



注意 本书中很多较长的示例代码可以从 Visual Studio .NET 的联机文档中下载。

表 1 列出了本书版式上的一些约定。

表 1 版式约定

约 定	描 述	示 例
等宽字体(Courier New)	表示源代码，示例代码，命令行的输入，程序的输出，嵌入在文本中的代码，变量和代码元素	<code>using System</code>
斜体	表示占位符，通常为方法或函数的参数和HTML的占位符，这些占位符表示必须由实现过程提供或者用户提供的信息	<i>identifier</i> ，接口名称
加粗	需要键入的内容。对于命令行命令，该字体表示参数值	

C#程序的通用结构

C#程序可以由一个或多个文件组成。每个文件都可以包含一个或多个命名空间。除了可以包含一

般命名空间外，命名空间还可以包括诸如类、结构、接口、枚举和代表等类型。下面是一个包括所有这些元素的 C# 程序框架。

```
//C#程序框架
using System;
namespace MyNamespacel
{
    Class MyClass1
    {
    }
    struct MyStruct
    {
    }
    interface IMyInterface
    {
    }
    delegate int MyDelegate()
    eum MyEum
    {
    }
    namespace MyNamespace2
    {
    }
    class MyClass2
    {
        public static void Main(string[] args)
        {
        }
    }
}
```


目 录

引言	xi	5.2 其他修饰符关键字	31
第 I 部分 C#关键字		5.2.1 abstract 关键字	31
第 1 章 访问关键字	3	5.2.2 const 关键字	33
1.1 base 关键字	3	5.2.3 event 关键字	35
1.2 this 关键字	5	5.2.4 extern 关键字	41
第 2 章 转换关键字	8	5.2.5 override 关键字	42
2.1 explicit 关键字	8	5.2.6 readonly 关键字	44
2.2 implicit 关键字	9	5.2.7 sealed 关键字	46
2.3 operator 关键字	11	5.2.8 static 关键字	46
第 3 章 文字关键字	14	5.2.9 unsafe 关键字	49
3.1 null 关键字	14	5.2.10 virtual 关键字	49
3.2 true 关键字	14	5.2.11 volatile 关键字	52
3.2.1 true 运算符	14	第 6 章 命名空间关键字	54
3.2.2 true 文字值	15	6.1 namespace 关键字	54
3.3 false 关键字	15	6.2 using 关键字	55
3.3.1 false 运算符	15	6.2.1 using 指令	56
3.3.2 false 文字值	16	6.2.2 using 语句	58
第 4 章 方法参数关键字	17	第 7 章 运算符关键字	60
4.1 params 关键字	17	7.1 as 关键字	60
4.2 ref 关键字	18	7.2 is 关键字	62
4.3 out 关键字	19	7.3 new 关键字	63
7.3.1 new 运算符		7.3.2 new 修饰符	64
第 5 章 修饰符关键字	21	7.4 sizeof 关键字	68
5.1 访问修饰符	22	7.5 typeof 关键字	69
5.1.1 可访问性级别	22	7.6 true 关键字	71
5.1.2 可访问域	23	7.6.1 true 运算符	71
5.1.3 对使用可访问性级别		7.6.2 true 文字值	71
的限制	25	7.7 false 关键字	72
5.1.4 访问修饰符关键字	27		

7.7.1	false 运算符.....	72	9.1.6	double 关键字.....	124
7.7.2	false 文字值.....	72	9.1.7	enum 关键字.....	126
7.8	stackalloc 运算符.....	73	9.1.8	float 关键字.....	128
第 8 章	语句关键字	75	9.1.9	int 关键字.....	129
8.1	选择语句.....	75	9.1.10	long 关键字.....	130
8.1.1	if-else 结构.....	75	9.1.11	sbyte 关键字.....	131
8.1.2	switch 关键字.....	78	9.1.12	short 关键字.....	132
8.2	迭代语句.....	81	9.1.13	struct 关键字.....	133
8.2.1	do 关键字.....	81	9.1.14	uint 关键字.....	137
8.2.2	for 关键字.....	82	9.1.15	ulong 关键字.....	138
8.2.3	foreach 和 in 关键字.....	83	9.1.16	ushort 关键字.....	139
8.2.4	while 关键字.....	92	9.2	装箱和取消装箱.....	140
8.3	跳转语句.....	93	9.2.1	装箱转换.....	140
8.3.1	break 关键字.....	93	9.2.2	取消装箱转换.....	141
8.3.2	continue 关键字.....	95	9.3	引用类型.....	143
8.3.3	goto 关键字.....	96	9.3.1	class 关键字.....	143
8.3.4	return 关键字.....	98	9.3.2	delegate 关键字.....	146
8.4	异常处理语句.....	99	9.3.3	interface 关键字.....	148
8.4.1	throw 关键字.....	99	9.3.4	object 关键字.....	151
8.4.2	try-catch 结构.....	100	9.3.5	string 关键字.....	152
8.4.3	try-finally 结构.....	103	9.4	类型引用表.....	154
8.4.4	try-catch-finally 结构.....	105	9.4.1	内置类型表.....	154
8.5	checked 和 unchecked 关键字.....	106	9.4.2	整型表.....	155
8.5.1	checked 关键字.....	106	9.4.3	浮点型表.....	156
8.5.2	unchecked 关键字.....	109	9.4.4	默认值表.....	156
8.6	fixed 语句.....	112	9.4.5	值类型表.....	157
8.7	lock 语句.....	114	9.4.6	隐式数值转换表.....	158
第 9 章	类型关键字	117	9.4.7	显式数值转换表.....	159
9.1	值类型.....	117	9.4.8	格式化数值结果表.....	160
9.1.1	值类型简介.....	117	第 II 部分 C#特性		
9.1.2	bool 关键字.....	119	第 10 章	运算符	165
9.1.3	byte 关键字.....	120	10.1	运算符概述.....	165
9.1.4	char 关键字.....	122	10.2	可重载的运算符.....	166
9.1.5	decimal 关键字.....	122	10.2.1	[]运算符.....	167

10.2.2	() 运算符.....	168	10.2.37	>>= 运算符.....	196
10.2.3	. 运算符.....	169	10.2.38	-> 运算符.....	196
10.2.4	+ 运算符.....	170	第 11 章 属性(Attribute)..... 198		
10.2.5	- 运算符.....	171	11.1	属性介绍.....	198
10.2.6	* 运算符.....	172	11.1.1	使用属性.....	198
10.2.7	/ 运算符.....	172	11.1.2	属性目标.....	199
10.2.8	% 运算符.....	173	11.1.3	全局属性.....	201
10.2.9	& 运算符.....	174	11.1.4	创建自定义属性.....	202
10.2.10	运算符.....	174	11.1.5	检索属性信息.....	203
10.2.11	^ 运算符.....	175	11.2	AttributeUsage 属性.....	205
10.2.12	! 运算符.....	176	11.3	Conditional 属性.....	205
10.2.13	~ 运算符.....	176	11.4	Obsolete 属性.....	207
10.2.14	= 运算符.....	177	11.5	属性教程.....	208
10.2.15	< 运算符.....	178	第 12 章 声明..... 215		
10.2.16	> 运算符.....	178	第 13 章 成员..... 218		
10.2.17	?: 运算符.....	179	13.1	成员简介.....	218
10.2.18	++ 运算符.....	180	13.1.1	命名空间成员.....	218
10.2.19	-- 运算符.....	181	13.1.2	结构成员.....	218
10.2.20	&& 运算符.....	182	13.1.3	类成员.....	219
10.2.21	运算符.....	183	13.1.4	其他成员.....	219
10.2.22	<< 运算符.....	184	13.2	成员访问.....	220
10.2.23	>> 运算符.....	185	13.2.1	已声明可访问性.....	220
10.2.24	== 运算符.....	186	13.2.2	可访问域.....	221
10.2.25	!= 运算符.....	187	13.2.3	实例成员的受保护访问.....	223
10.2.26	<= 运算符.....	188	13.2.4	可访问性约束.....	224
10.2.27	>= 运算符.....	189	第 14 章 预处理器指令..... 226		
10.2.28	+= 运算符.....	189	14.1	#if 指令.....	226
10.2.29	-= 运算符.....	190	14.2	#else 指令.....	227
10.2.30	*= 运算符.....	191	14.3	#elif 指令.....	228
10.2.31	/= 运算符.....	191	14.4	#endif 指令.....	228
10.2.32	%= 运算符.....	192	14.5	#define 指令.....	229
10.2.33	&= 赋值运算符.....	193	14.6	#undef 指令.....	229
10.2.34	= 运算符.....	194	14.7	#warning 指令.....	230
10.2.35	^= 运算符.....	194			
10.2.36	<<= 运算符.....	195			

14.8	#error 指令	231	18.2.2	set 访问器	273
14.9	#line 指令	231	18.3	接口属性	278
14.10	#region 指令	232	18.4	属性教程	281
14.11	#endregion 指令	233	第 19 章	签名和重载	288
第 15 章	数组	234	第 20 章	范围	290
15.1	三种数组	234	20.1	范围的通用规则	290
15.1.1	一维数组	234	20.2	名称隐藏	292
15.1.2	多维数组	236	20.2.1	通过嵌套隐藏	292
15.1.3	交错数组	237	20.2.2	通过继承隐藏	293
15.2	使用 ref 和 out 传递数组	240	第 21 章	参数传递	296
15.3	数组教程	242	21.1	传递值类型参数	296
15.3.1	数组概述	242	21.2	传递引用类型参数	299
15.3.2	声明数组	243	第 22 章	Main 方法	302
15.3.3	初始化数组	244	22.1	返回值	302
15.3.4	访问数组成员	245	22.2	命令行参数	303
15.3.5	数组是对象	246	第 23 章	条件方法	306
15.3.6	对数组使用 foreach 语句	246	第 24 章	委托	310
第 16 章	构造函数和析构函数	247	第 25 章	事件	317
16.1	构造函数	247	第 26 章	XML 文档	325
16.1.1	类构造函数	247	26.1	文档注释标记	325
16.1.2	结构构造函数	254	26.1.1	<c>标记	326
16.2	析构函数	255	26.1.2	<code>标记	326
第 17 章	索引器	259	26.1.3	<example>标记	327
17.1	索引器声明	259	26.1.4	<exception>标记	328
17.1.1	get 访问器	260	26.1.5	<include>标记	328
17.1.2	set 访问器	260	26.1.6	<list>标记	330
17.2	属性与索引器之间的比较	262	26.1.7	<para>标记	331
17.3	接口索引器	262	26.1.8	<param>标记	332
17.4	索引器教程	265	26.1.9	<paramref>标记	332
第 18 章	属性(Property)	270	26.1.10	<permission>标记	333
18.1	属性声明	270	26.1.11	<remarks>标记	334
18.2	访问器	272	26.1.12	<returns>标记	334
18.2.1	get 访问器	273			

26.1.13	<see>标记.....	335
26.1.14	<seealso>标记	335
26.1.15	<summary>标记	336
26.1.16	<value>标记	336
26.2	处理 XML 文件.....	337
附录	编译器选项	341

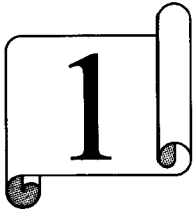
第 I 部分

C#关键字

关键字是对编译器具有特殊意义的预定义保留标识符。它们不能在程序中用作标识符，除非它们有一个@前缀。

本部分介绍的关键字包括：访问关键字(base, this)，转换关键字(explicit, implicit, operator)，文字关键字(null, true, false)，方法参数关键字(params, ref, out)，修饰符关键字(abstract, const, event, extern, override, readonly, sealed, static, unsafe, virtual, volatile)，命名空间关键字(namespace, using)，运算符关键字(as, is, new, sizeof, typeof, true, false, stackalloc)，语句关键字和类型关键字等。





访问关键字

本章介绍访问关键字(access keyword), 包括:

- `base` 访问基类的成员
- `this` 引用当前对象(为该对象调用了方法)

1.1 `base` 关键字

`base` 关键字用于从派生类中访问基类的成员, 包括:

- 调用基类上已被其他方法重写的方法
- 指定创建派生类实例时应调用的基类构造函数

对基类的访问只能在构造函数、实例方法或实例属性访问器中进行。

从静态方法中使用 `base` 关键字是错误的。

示例 1

在本例中, 基类 `Person` 和派生类 `Employee` 都有一个名为 `Getinfo` 的方法。通过使用 `base` 关键字, 可以从派生类中调用基类上的 `Getinfo` 方法。

```
// keywords_base.cs
// 访问基类成员
using System;
public class Person
{
    protected string ssn = "444-55-6666";
    protected string name = "John L. Malgraine";

    public virtual void GetInfo()
    {
        Console.WriteLine("Name: {0}", name);
    }
}
```



```
        Console.WriteLine("SSN: {0}", ssn);
    }
}
class Employee: Person
{
    public string id = "ABC567EFG";
    public override void GetInfo()
    {
        // 调用基类上的 Getinfo 方法:
        base.GetInfo();
        Console.WriteLine("Employee ID: {0}", id);
    }
}

class TestClass {
    public static void Main()
    {
        Employee E = new Employee();
        E.GetInfo();
    }
}
```

输出:

```
Name: John L. Malgraine
SSN: 444-55-6666
Employee ID: ABC567EFG
```

示例 2

本例演示如何指定在创建派生类实例时调用的基类构造函数。

```
// keywords_base2.cs
using System;
public class MyBase
{
    int num;
    public MyBase() {
        Console.WriteLine("in MyBase()");
    }
    public MyBase(int i )
    {
        num = i;
        Console.WriteLine("in MyBase(int i)");
    }

    public int GetNum()
    {
```