

● Practical...

C# 程序设计 实用教程

Programming

黄德志 汤庆恩 编著

CR02: CPS3PLN

535812

106838/ A/0

5000

3

冶金工业出版社

0 4 0 5 0 6

0629

C# 程序设计 实用教程

Programming

清华大学出版社



C#程序设计实用教程

黄德志 汤庆恩 编著

北 京

冶金工业出版社

2002

内容简介

C#是一种面向对象的编程语言,它能够快速而容易的为微软.NET平台开发解决方案。本书全面地介绍了C#语言。首先介绍C#的产生的背景及特点、数据类型、运算符等基础知识。接着介绍了主要语句,类,数组,接口,代表、结构和枚举,以及继承和重载等。最后还介绍了输入和输出、继承、重载、异常处理、编译预处理和调试技术、不安全代码及代码属性等。

本书内容全面,实例丰富,并配有适量的练习。学好一门程序设计语言最关键的就是多上机编写程序。本书中丰富、精炼、富有代表性的实例将是读者迅速学会C#的好帮手。看完本书并上机编译本书的实例,一定会发现,您已经掌握了C#,并可以用C#编写各种程序了。

本书适合C#语言的初、中级读者使用,对于学过C、C++或者Java的读者,本书将是一本从原来的程序设计语言过渡到C#的理想教材。本书可作为本、专科学生学习计算机编程语言的教科书,也可以作为广大编程爱好者学习和提高的参考书。

图书在版编目(CIP)数据

C#程序设计实用教程 / 黄德志等编著. —北京:
冶金工业出版社, 2002.11
ISBN 7-5024-3116-0

I. C... II. 黄... III. C语言—程序设计—教材
IV. TP312

中国版本图书馆CIP数据核字(2002)第075169号

出版人 曹胜利(北京沙滩嵩祝院北巷39号,邮编100009)

责任编辑 程志宏

中山市新华印刷厂有限公司印刷;冶金工业出版社发行;各地新华书店经销

2002年11月第1版,2002年11月第1次印刷

787mm×1092mm 1/16; 19印张; 436千字; 294页; 1-2500册

25.00元

冶金工业出版社发行部 电话:(010)64044283 传真:(010)64027893

冶金书店 地址:北京东四西大街46号(100711) 电话:(010)65289081

(本社图书如有印装质量问题,本社发行部负责退换)

前 言

1. 关于 C#

作为一名开发人员，如何面临新的挑战？如何开发下一代的互联网应用？C#将给你完美的答案。C#是 Microsoft 最新主推的编程工具，它是一门全新的、面向对象的程序语言，具有高级的语法结构，优秀的编程开发环境和高效率的编译工具。我们可以利用它来编写基于通用网络协议的 Internet 服务软件，也可以编写 Windows 图形用户界面程序，还可以编写各种数据库，网络服务应用程序。可以预言，它将是未来几年最热门，最实用的软件开发手段。因此，认识 C#，掌握 C#这门语言，对于一名开发人员而言，将是非常有意义的。

2. 本书特点

本书以循序渐进，深入浅出的方式引导读者步入 C#程序设计的大门。为了方便没有编程基础的读者的学习，本书从基本的数据类型与语句开始讲解，逐步过渡到类、接口、数组、继承、重载等复杂机制，最后还介绍了提高程序可靠性的异常处理机制，以及编译预处理和调试技术，不安全代码及代码属性等提高内容。本书提供了丰富、典型的例题，并且每一章均附有精选的练习。

本书内容丰富，结构清晰，文笔流畅，语言通俗易懂，并兼顾了深度和广度。本书中所使用的实例均在计算机上编译通过，读者必须安装 C#编译器和.NET 运行阶段环境，以便进一步上机体会 C#程序设计技巧和面向对象程序设计的精髓。

3. 本书的内容结构

本书共分 16 章，各章内容安排如下：

第 1 章：C#的产生及其特点。主要介绍了 C#的产生背景，C#的特点，C#与 C++、Java 的比较，以及 C#的编程环境。

第 2 章：C#程序入门。主要介绍了用 C#编写经典的 Hello World 程序、C#程序的基本组成部分、C#程序的输入和输出。

第 3 章：C#的数据类型。主要介绍了 C#数据类型的分类、变量和常量、数值类型、引用类型、装箱和拆箱。

第 4 章：运算符。主要介绍了运算符的类型、标点符号、赋值运算符、数学/算术运算符、关系运算符、位运算符、逻辑运算符、条件运算符等。

第 5 章：C#的主要语句。主要介绍了 C#程序的基本结构、选择语句、循环语句、转移语句、异常处理语句等。

第 6 章：类。主要介绍了什么是面向对象编程、类的声明、类的成员、类的方法、属性、索引、域、命名空间等。

第 7 章：数组。主要介绍了数组的声明、数组元素的使用、数组的初始化、AOA 数组等。

第 8 章：接口。主要介绍了接口的声明、接口的使用、接口的命名空间等。

第 9 章：代表、结构和枚举。主要对代表、结构和枚举进行了介绍。

第 10 章：输入和输出。主要介绍了控制台的输入、控制台的输出、字符串、格式化输出等。

第 11 章：继承。主要介绍了继承的基本知识、多态、终极基类等。

第 12 章：重载。主要介绍了什么是重载、方法重载、构造函数重载、运算符重载等。

第 13 章：异常处理。主要介绍了异常处理的概念、常见的异常、异常处理的使用、checked 语句和 unchecked 语句的用法、自定义异常类等。

第 14 章：编译预处理和调试技术。主要介绍了编译预处理和 C# 调试技术。

第 15 章：不安全代码及代码属性。主要对不安全代码和代码属性进行了介绍。

第 16 章：应用实例。本章综合前面所学的知识，介绍了 C# 在实际工作中的应用，通过两个综合实例，希望读者能从中得到启发和提高。

此外，为了使读者巩固所学的知识，本书还附有练习题，并给出了参考答案，以方便读者对照学习。

4. 本书的适用对象

本书适合 C# 语言的初、中级读者使用，对于 C、C++ 或者 Java 有一定基础的读者，本书将是以往的程序设计语言过渡到 C# 语言的理想教材。本书可作为本、专科学生学习计算机编程语言的教科书，也可作为广大编程爱好者学习和提高的参考书。

编写本书，是在 C# 程序设计教学上从繁到简的一次尝试，虽力图做好，但由于编者水平有限，不足之处在所难免，望广大同仁和读者不吝赐教，以便使本书质量进一步得到提高。如果读者在学习的过程中遇到疑难问题或不妥之处，可以登录 [HTTP://www.TianDing.net](http://www.TianDing.net)，在该网站的相关论坛进行探讨。

编 者

2002 年 10 月

目 录

第 1 章 C#的产生及其特点	1	3.1 C#数据类型的分类	12
1.1 C#的产生背景	1	3.1.1 数值类型、引用类型	12
1.2 C#的特点	2	3.1.2 两种基本数据类型最根本的区别	12
1.2.1 遵循新的 Web 设计标准	2	3.2 变量和常量	14
1.2.2 消除重要的编程错误	2	3.2.1 变量	14
1.2.3 降低开发成本	2	3.2.2 常量	15
1.2.4 广泛的协同工作能力	2	3.3 数值类型	15
1.3 C#与 C++、Java 的比较	3	3.3.1 bool 类型	17
1.4 C#的编程环境	3	3.3.2 sbyte 类型	17
小结	5	3.3.3 byte 类型	18
习题一	5	3.3.4 short 类型	19
一、选择题	5	3.3.5 ushort 类型	20
二、简答题	6	3.3.6 int 类型	20
三、上机题	6	3.3.7 uint 类型	21
第 2 章 C#程序入门	7	3.3.8 long 类型	21
2.1 用 C#编写经典的 Hello World 程序	7	3.3.9 ulong 类型	22
2.1.1 用 C#编写 Hello World 程序	7	3.3.10 char 类型	24
2.1.2 C#与 C++编写程序的比较	8	3.3.11 float 类型	24
2.1.3 C#与 Java 编写程序的比较	8	3.3.12 double 类型	25
2.2 C#程序的基本组成部分	8	3.3.13 decimal 类型	25
2.2.1 空白	8	3.3.14 C#的系统类型	27
2.2.2 注释	8	3.4 引用类型	27
2.2.3 关键字	9	3.4.1 类类型	28
2.2.4 字面值	9	3.4.2 接口类型	30
2.2.5 标识符	9	3.4.3 数组类型	32
2.3 C#程序的输入和输出	9	3.4.4 代表类型	34
2.3.1 C#程序的输入	9	3.4.5 object 类型	35
2.3.2 C#程序的输出	9	3.4.6 string 类型	36
小结	10	3.5 装箱和拆箱	38
习题二	10	3.5.1 装箱操作	38
一、选择题	10	3.5.2 拆箱操作	39
二、简答题	10	小结	40
三、上机题	11	习题三	40
第 3 章 C#的数据类型	12	一、选择题	40

二、简答题	41	第 5 章 C#的主要语句	56
三、上机题	41	5.1 C#程序的基本结构	56
第 4 章 运算符	42	5.1.1 顺序结构	56
4.1 运算符的类型	42	5.1.2 选择结构	56
4.1.1 运算符的分类	42	5.1.3 循环结构	57
4.1.2 运算符的结构	42	5.2 选择语句	57
4.2 标点符号	42	5.2.1 if 语句	57
4.3 赋值运算符	43	5.2.2 switch 语句	58
4.4 数学/算术运算符	43	5.3 循环语句	59
4.4.1 加法和减法运算符	43	5.3.1 while 语句	59
4.4.2 乘法和除法运算符	43	5.3.2 do...while 语句	60
4.4.3 正负运算符	44	5.3.3 for 语句	63
4.4.4 求模运算符	44	5.3.4 foreach 语句	64
4.4.5 复合运算符	44	5.4 转移语句	65
4.5 关系运算符	44	5.4.1 return 语句	65
4.5.1 关系运算符的使用	44	5.4.2 break 语句	65
4.5.2 is 运算符	45	5.4.3 continue 语句	66
4.6 位运算符	46	5.4.4 goto 语句	66
4.6.1 二进制基础	46	5.5 异常处理语句	68
4.6.2 位运算符的使用	46	5.6 其他语句	69
4.7 逻辑运算符	47	5.6.1 固定语句 fixed	69
4.7.1 逻辑与运算符	47	5.6.2 非保护语句 unsafe	71
4.7.2 逻辑或运算符	47	5.6.3 锁定语句 lock	72
4.7.3 逻辑非运算符	47	小结	74
4.7.4 逻辑异或运算符	47	习题五	74
4.8 条件运算符	48	一、选择题	74
4.8.1 ?:运算符	48	二、简答题	75
4.8.2 as 运算符	48	三、上机题	75
4.9 其他运算符	49	第 6 章 类	76
4.9.1 new 运算符	49	6.1 什么是面向对象编程	76
4.9.2 sizeof 运算符	51	6.1.1 OOP 技术	76
4.9.3 typeof 运算符	51	6.1.2 OOP 技术的主要特征	76
4.10 运算符的优先级	52	6.2 类的声明	77
小结	53	6.3 类的成员	78
习题四	53	6.3.1 数据成员	78
一、选择题	53	6.3.2 函数成员	78
二、简答题	54	6.4 类的方法	79
三、上机题	55	6.4.1 方法的声明	79

6.4.2 方法的调用	80	三、上机题	110
6.4.3 方法的输入参数	80	第8章 接口	111
6.4.4 最常见的方法——Main 方法	82	8.1 接口的声明	111
6.5 属性	83	8.1.1 类和接口的比较	111
6.6 索引	89	8.1.2 如何声明接口	111
6.6.1 this 关键字	89	8.2 接口的使用	112
6.6.2 索引器的使用	91	8.2.1 接口的使用方法	112
6.7 域	93	8.2.2 使用多重接口	115
6.7.1 域的声明	93	8.3 接口的命名空间	116
6.7.2 公有域和私有域	94	小结	117
6.8 命名空间	96	习题八	117
6.8.1 命名空间的声明	96	一、选择题	117
6.8.2 命名空间的使用	97	二、简答题	118
小结	99	三、上机题	118
习题六	99	第9章 代表、结构和枚举	119
一、选择题	99	9.1 代表	119
二、简答题	100	9.1.1 代表的声明	119
三、上机题	100	9.1.2 代表的使用	119
第7章 数组	101	9.1.3 事件	121
7.1 数组的声明	101	9.2 结构	125
7.1.1 一维数组	101	9.2.1 结构的声明	125
7.1.2 二维数组	101	9.2.2 结构的使用	126
7.1.3 多维数组	101	9.2.3 结构和类的区别	128
7.2 数组元素的使用	102	9.3 枚举	130
7.2.1 一维数组元素的使用	102	9.3.1 枚举的声明	130
7.2.2 二维数组元素的使用	102	9.3.2 枚举的使用	131
7.3 数组的初始化	103	小结	132
7.3.1 一维数组的初始化	103	习题九	132
7.3.2 二维数组的初始化	105	一、选择题	132
7.4 AOA 数组	106	二、简答题	134
7.4.1 AOA 数组的声明	107	三、上机题	134
7.4.2 AOA 数组元素的使用	107	第10章 输入和输出	135
7.4.3 AOA 数组元素的初始化	107	10.1 控制台的输入	135
7.4.4 AOA 数组应用举例	108	10.2 控制台的输出	136
小结	109	10.3 字符串	136
习题七	109	10.4 格式化输出	137
一、选择题	109	10.4.1 数字的格式化	137
二、简答题	110		

10.4.2 日期和事件的格式化	140	13.3.1 throw 语句(抛出异常)	165
小结	143	13.3.2 try-catch 语句	166
习题十	143	13.3.3 finally 语句	171
一、选择题	143	13.4 checked 语句和 unchecked 语句 的用法	173
二、简答题	143	13.5 自定义异常类	174
三、上机题	143	13.5.1 System.Exception 类	174
第 11 章 继承	144	13.5.2 System 命名空间的异常类	177
11.1 继承的基本知识	144	13.5.3 自定义异常类的使用	178
11.1.1 派生 Manager 类	144	小结	182
11.1.2 base 关键字	146	习题十三	182
11.1.3 继承的访问	147	一、选择题	182
11.1.4 成员访问级别	147	二、简答题	183
11.2 多态	147	三、上机题	183
11.3 终极基类——object 类	149	第 14 章 编译预处理和调试技术	184
小结	149	14.1 编译预处理	184
习题十一	150	14.1.1 预处理的声明	184
一、选择题	150	14.1.2 #if、#elif、#else、 #endif	185
二、简答题	150	14.1.3 #error、#warning	188
三、上机题	150	14.1.4 修改行号	189
第 12 章 重载	151	14.1.5 区域简介	190
12.1 什么是重载	151	14.2 C#调试技术	192
12.2 方法重载	151	14.2.1 什么是调试	192
12.3 构造函数重载	154	14.2.2 运用调试器调试程序	192
12.4 运算符重载	156	小结	196
12.4.1 什么是运算符重载	156	习题十四	196
12.4.2 重载+/-运算符	157	一、选择题	196
12.4.3 重载++/--运算符	157	二、简答题	196
12.4.4 重载关系运算符	159	三、上机题	196
小结	161	第 15 章 不安全代码及代码属性	198
习题十二	161	15.1 不安全代码	198
一、选择题	161	15.1.1 概述	198
二、简答题	162	15.1.2 不安全代码模块	199
三、上机题	162	15.1.3 指针的使用	203
第 13 章 异常处理	163	15.2 代码属性	209
13.1 异常处理的概念	163	15.2.1 代码属性类的声明	209
13.2 常见的异常	163		
13.3 异常处理的使用	165		

15.2.2 条件代码属性	209	16.2.5 程序演示	276
15.2.3 作废代码属性	211	小结	278
15.2.4 外部方法和 DllImport 代码属性	213	附录 A C#关键字	279
15.2.5 System.AttributeUsage 类	214	A.1 C#关键字名	279
15.2.6 System.AttributeTargets 枚举值	215	A.2 C#关键字说明	279
15.2.7 自定义代码属性类	216	附录 B .NET 的安装	282
小结	219	参考答案	284
习题十五	219	第 1 章	284
一、选择题	219	第 2 章	284
二、简答题	220	第 3 章	285
三、上机题	220	第 4 章	286
第 16 章 应用实例	221	第 5 章	287
16.1 记事本编辑器	221	第 6 章	289
16.1.1 系统简介	221	第 7 章	289
16.1.2 编程思路	221	第 8 章	290
16.1.3 C# 的控件	221	第 9 章	290
16.1.4 编程步骤及程序演示	224	第 10 章	291
16.2 银行存储系统	271	第 11 章	291
16.2.1 编程目的	271	第 12 章	291
16.2.2 系统简介	271	第 13 章	293
16.2.3 编程思路	271	第 14 章	294
16.2.4 程序代码	272	第 15 章	294

第 1 章 C#的产生及其特点

本章介绍了 C#的产生背景, C#的特点, C#与 Java、C++、C 的不同之处, 以及 C#的编程环境。

本章重点包括:

- 1) C#的特点。
- 2) C#与 C++、Java 的比较。
- 3) C#的编程环境。

1.1 C#的产生背景

C#是一种现代的, 类型安全的, 面向对象的编程语言, 它使得程序员能够快速而容易的为微软.NET 平台开发解决方案。在过去的 20 年中, C 和 C++已经是开发商开发企业软件时使用最广泛的编程语言之一。这两种语言为开发者提供了大量细致灵活的控制, 这种灵活性是以生产成本作为代价的。就拿 MS-VB 来与之比较, C 和 C++应用程序相对来说需要较长的开发时间。

由于复杂性和长周期, 许多 C 和 C++程序员已经在寻找一种能在功能和生产力之间提供更好均衡的编程语言。

有几种编程语言是通过牺牲 C 和 C++程序员常需要的灵活性来提高生产力。这样的解决方案对开发者的约束太多(例如: 通过省略低级代码控制)并且通用性很差。

它们与先前存在的系统很难相互操作, 并且它们与当前的 Web 设计方法不能很好的吻合。

对 C 和 C++程序员来说, 理想的解决方法应该是快速的开发与访问所有潜在平台的能力相结合。

开发环境应该完全与新的 Web 标准同步并容易与现存的应用系统集成, 同时, C 和 C++程序员还希望能够在必要的时候在底层编写代码。

微软为了解决上述问题, 提出了一种叫 C#的语言(英文中读作: C Sharp)。C#是一种现代的, 面向对象的语言, 它使开发人员能够在微软新的.NET 平台上快速建立广泛的应用, 其提供的工具和服务能充分发掘系统的计算和通讯能力。

因为其优良的面向对象设计, 在构建从高级业务对象到系统级应用的各种不同组件时, C#是一个首要的选择。

使用简易的 C#语言构造, 组件可以被转换为 Web 服务, 从而允许从运行在任何操作系统上的任何语言中跨越 Internet 调用它们。

不仅如此, C#的设计为 C++程序员带来了快速的开发能力, 而不用牺牲 C++已有的功能和控制能力。

通过这种继承, C#高度地保持了与 C 和 C++的一致。开发者只要熟悉 C 和 C++语言就可以快速地掌握 C#并写出更多的 C#应用程序。

1.2 C#的特点

1.2.1 遵循新的 Web 设计标准

新的应用程序开发模型意味着越来越多的解决方案需要使用新的 Web 标准, 例如: HTML, XML 和 SOAP (Simple Object Access Protocol)。现存的开发工具都是在 Internet 之前或之初开发的。因此, 它们总是不能很好地适应新的 Web 技术。

C#程序员能用一个扩展的框架来构建微软.NET 平台上的应用。C#包括内建的支持使任何组件转换为一个能在 Internet 上运行任何平台上的任何应用中被调用的服务。

对程序员来说, 更出色的是这个 Web 服务框架能使现存的 Web 服务看起来像本地的 C#对象, 因此允许开发者权衡利用它们已有的面向对象的编程技能与在现存的 Web 服务。

有更多的特点使得 C#成为主要的 Internet 开发工具。例如: XML 是新出现的在 Internet 中传递结构化数据的方法。这种数据集通常很小。为了提高性能, C#允许 XML 数据被直接映射到一个结构数据类型而不是一个类。在数据量不大时, 这是一种更有效的处理方式。

1.2.2 消除重要的编程错误

即便是专家级的 C++程序员都避免不了最简单的错误 (如: 忘记初始化一个变量), 这些简单的错误常导致不可预见的问题, 它们长时间隐蔽, 不易发现。一旦程序投入生产运行后, 要排除哪怕是最简单的编程错误, 都要付出昂贵的代价。C#现代的设计排除了大多数普通的 C++编程错误。

例如:

- 1) 垃圾清理减轻了程序员自己管理内存的负担。
- 2) 在 C#中的变量是自动被环境初始化的。
- 3) 变量类型是安全的。
- 4) 最终使一种语言更加容易编写和维护解决复杂业务问题的程序。

1.2.3 降低开发成本

更新软件组件是一个容易出错的任务。修订代码版本无意中可能会改动程序的语义。

为了帮助程序员解决这个问题, C#语言中包含了转换支持。例如: 与 C++和 Java 不同, 方法重载必须显式进行; 这有助于防止代码错误和保留转换的灵活性。有关的特点还有本地的接口和接口继承支持。这些特点进化并发展了复杂的框架。

总之, 这些特点使一个工程的后继版本的开发方法更加健壮, 同时为成功的应用降低了整体的开发成本。

1.2.4 广泛的协同工作能力

可管理、类型安全环境对大多数企业应用是适合的。但是现实世界中的经验告诉, 有些应用延续需要“本地”代码, 其原因要么是因为性能问题, 要么是因为与现存的应用程序接口问题。这样的情况强迫开发者使用 C++来开发, 即使用生产性更好的开发环境。

C#通过下面的方式解决了这个问题：

1) 包含组件对象模型（COM）的本地支持和基于 Windows API 的支持。

2) 允许有限制的使用本地指针在 C#中，每一个对象自动地成为一个 COM 对象。开发者不再需要显式地实现 IUnknown 和其他 COM 借口，而是将它们内置了。同样，C#编程能在本地使用现存的 COM 对象，与它们用什么语言创建的无关。

对于需要这种特性的开发者来说，C#包含了一个特别的特性：那就是一个程序可以调用任何本地 API。在一个特别标记的代码块中，允许开发者使用指针和传统的 C/C++特点，如自己管理内存和指针算法。

相对于其他开发环境，这是一个巨大的优点。它意味着 C#程序员能建立自己现存的大量 C 和 C++代码，而不用丢弃它们。

3) COM 支持和本地 API 访问支持，目的是向开发者提供了重要的能力和控制，不用离开 C#环境。

1.3 C#与 C++、Java 的比较

C#的语言规范由 Microsoft 的 AndersHejlsberg 与 ScottWiltamuth 编写。对于程序员而言，对 C#和 C++、Java 作一番比较总是很有必要的。

如果早就知道 C#更接近 Java 而不是 C++，事情也不值得大惊小怪；如果是刚刚加入 C#阵营的读者，如表 1-1 让自己作出判断。

表 1-1 比较 C#、C++和 Java 最重要的功能

功能	C#	C++	Java
继承	允许继承单个类，允许实现多个接口	允许从多个类继承	允许继承单个类，允许实现多个接口
接口实现	通过“interface”关键词	通过抽象类	通过“interface”关键词
内存管理	由运行时环境管理，使用垃圾收集器	需要手工管理	由运行时环境管理，使用垃圾收集器
指针	支持，但只在很少使用的非安全模式下才支持。通常以引用取代指针	支持，一种很常用的功能	完全不支持。代之以引用
源代码编译后的形式	.NET 中间语言（IL）	可执行代码	字节码
单一的公共基类	是	否	是
异常处理	是	返回错误	是

显然，结论应该是：Java 和 C#虽然不是孪生子，但 C#最主要的特色却更接近 Java 而不是 C++。

1.4 C#的编程环境

本书中所用的 C#的编程环境是 Microsoft Visual Studio.NET。安装完之后启动程序，

可看到启动后界面如图 1-1 所示。

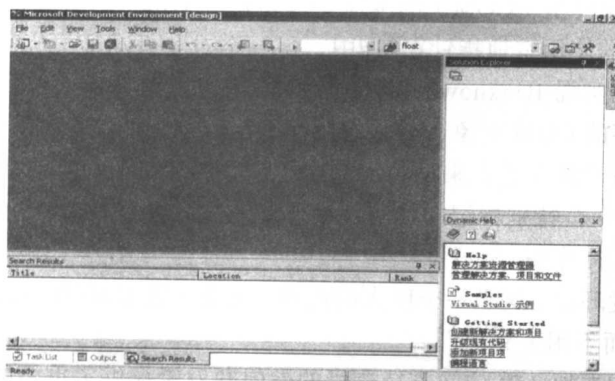


图 1-1

1) 选择 File/New/Project 新项目窗体显示, 可看到界面如图 1-2 所示。然后在项目类型窗口中选择 Visual C#项目, 在模板窗口中选择 EmptyProject。

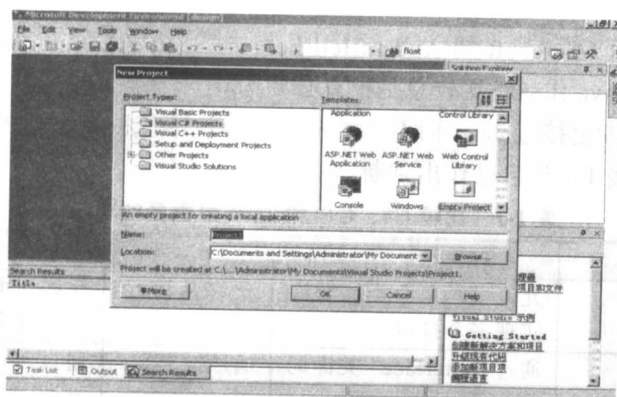


图 1-2

2) 在刚才创建的项目中创建该文件的所使用的资源文件, 首先选择 File/AddNewItem, 出现 Add New Item 窗体, 如图 1-3 所示。

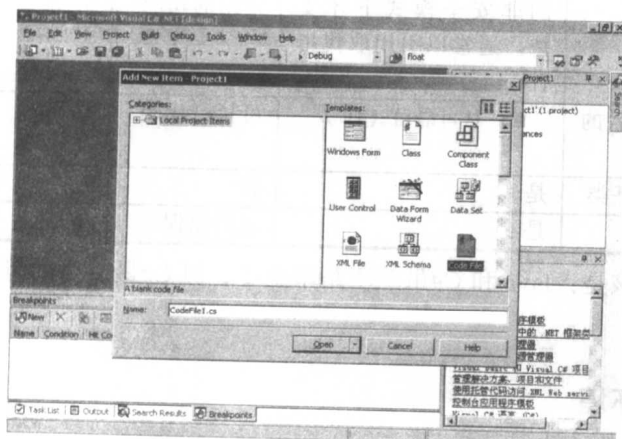


图 1-3

3) 在类别窗口中, 单击 Local Project Items, 在模板窗口中, 选择 Code File, 此时便可以在 Code File 里面输入代码进行编译了, 如图 1-4 所示 (具体编译例子看后面的章节)。

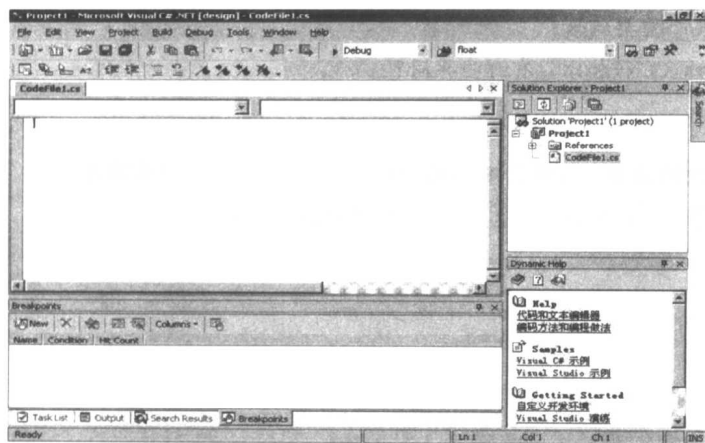


图 1-4

小结

本章首先介绍了 C# 的产生背景、C# 的特点、C# 与 Java、C++ 的不同之处, 以及 C# 的编程环境, 使读者对 C# 有感性的认识。为下一章对 C# 入门程序的学习打好基础。

习题一

一、选择题

1. C# 是 () 公司开发的 () 语言。
A. Microsoft B. Sun
C. Boland D. 面向对象程序设计
E. 面向用户程序设计 F. 面向过程程序设计语言
2. 和 C# 最接近的语言是 (), 它是由 () 公司开发的。
A. C B. C++
C. Java D. Delphi
E. Boland F. Sun
3. 在继承方面, C# ()。
A. 允许继承单个类, 允许实现多个接口
B. 允许从多个类继承
C. 允许继承单个类, 允许实现多个接口
D. 不允许继承
4. 以下 () 语言不支持指针。
A. C B. C++ C. C# D. Java
5. 以下 () 语言不支持异常处理。

A. C

B. C++

C. C#

D. Java

二、简答题

1. C#有哪几个重要特性?
2. C#与Java、C++相比,有哪些异同?

三、上机题

1. 根据后面的附录B,安装完Visual.Net,为以后的上机学习作准备.
2. 根据1.4节C#的编程环境,进入code代码编辑界面.