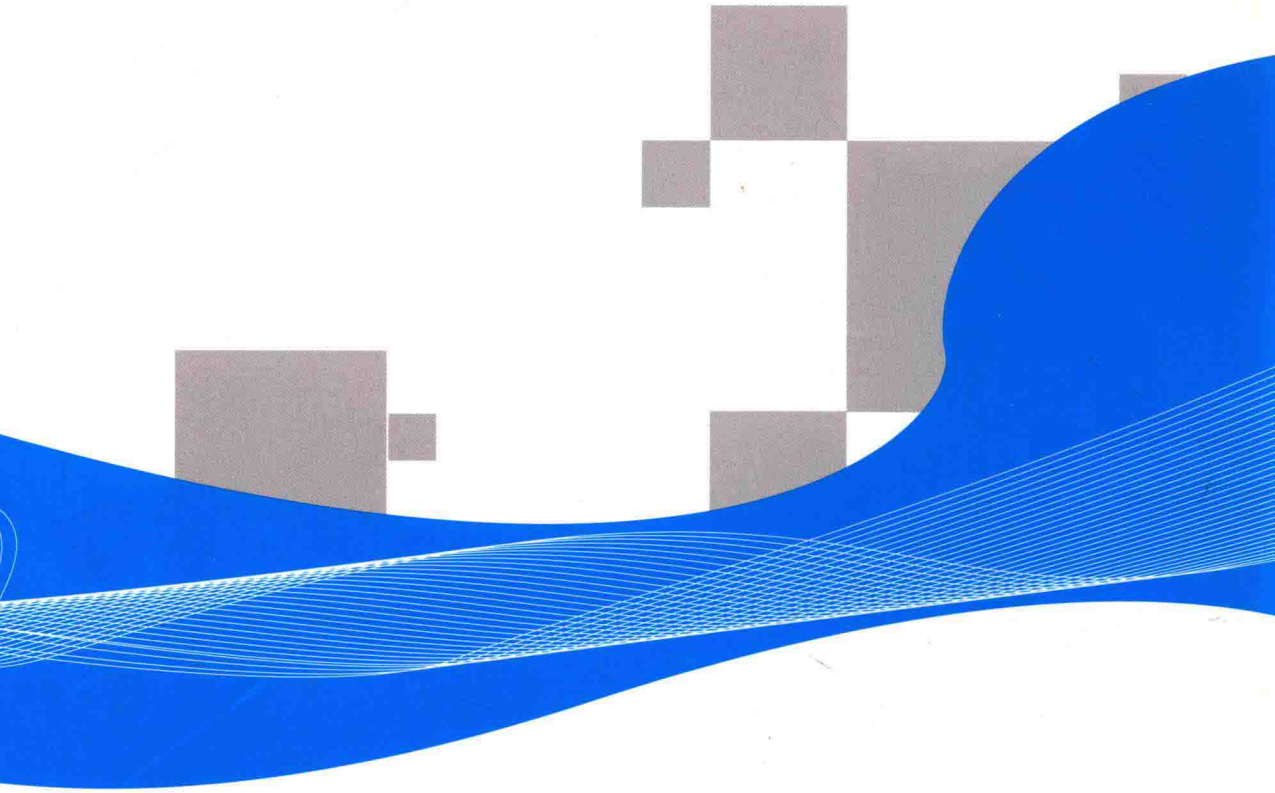
Microsoft®

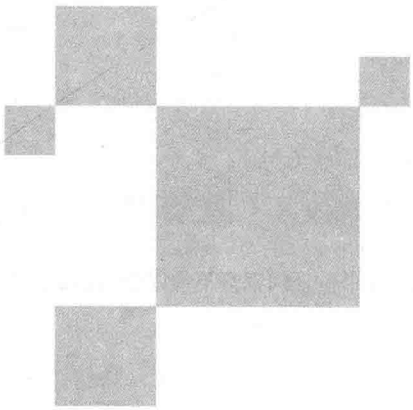


# Visual C# 2008
# 程序设计语言

微软公司 著

人民邮电出版社
POSTS & TELECOM PRESS

CD-ROM



# Visual C# 2008
## 程序设计语言

微软公司　著


人民邮电出版社
北京

**图书在版编目（CIP）数据**

Visual C# 2008程序设计语言 / 微软公司著. -- 北京 : 人民邮电出版社, 2011.2
ISBN 978-7-115-22718-8

Ⅰ. ①V… Ⅱ. ①微… Ⅲ. ①C语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2010)第060926号

## 版 权 声 明



## 内 容 提 要

本教材主要讲解 Visual C# 2008 程序语言的基本要素，包括程序的基本结构、基本数据类型，以及如何在 Visual Studio 2008 中使用 Visual C# 2008 创建 WinForm 应用程序、Web 应用程序、XML Web Service 及实现 Ajax 应用。同时，本书也讲解了面向对象的一些基础知识，包括类的抽象、封装、继承、多态等。读者可按照章节次序循序渐进地学习，并配合实验，综合掌握 Visual C# 2008 的相关知识和技能。

本教材主要面向希望学习 Visual C# 2008 语言和面向对象编程原理，并能进行代码初步编写和调试的读者。



**Visual C# 2008 程序设计语言**

◆ 著　　微软公司
责任编辑　刘　浩

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京铭成印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：16.25　　2011 年 2 月第 1 版
字数：390 千字　　2011 年 2 月北京第 1 次印刷

ISBN 978-7-115-22718-8

定价：53.00 元（附光盘）

**读者服务热线：(010)67132692　印装质量热线：(010)67129223**
**反盗版热线：(010)67171154**

**编审和组织策划** Kyle Uphoff　Foong Chee Ngiam　王　林
田本和　蒋　斌　梁　健

**技术编审** 蒋　斌　宫　丽　刘明烁　王建中　郭　健
张　梅

# 前　言

C#（读作“C sharp”）是一种编程语言，它是为生成在.NET Framework 上运行的各种应用程序而设计的。C#简单、功能强大、类型安全，而且是面向对象的。C#凭借在许多方面的创新，在保持 C 语言风格表现力和雅致特征的同时，实现了应用程序的快速开发。

本书介绍了 Visual C# 2008 程序设计的基本结构和数据类型，面向对象的基本概念，以及如何使用 Visual C# 2008 进行 Windows 窗体和 Web 应用程序开发。另外还对于 Ajax 和 XML Web Services 进行了简单的介绍。

通过本课程的学习，读者能够：

- 掌握 C#程序的基本结构和 C#内置数据类型。
- 掌握表达式和条件语句、循环语句。
- 理解面向对象思想。
- 掌握定义类、声明方法、使用构造函数和使用静态类成员。
- 掌握在对象中封装信息、创建从其他对象继承功能的对象和通过使用抽象类实现多态。
- 掌握接口使用方法、创建和调用委托和使用委托来处理事件。
- 掌握数组、字符串的创建和使用方法。
- 了解 LINQ 查询表达式的使用。
- 掌握如何创建和使用范型。
- 掌握格式化字符串的方法。
- 掌握程序生成、异常处理和调试方法。
- 掌握 Windows 窗体程序的创建方法和常用控件用法。
- 了解如何创建 Web 窗体应用程序、在 Web 窗体应用程序中访问数据。
- 了解使用 ADO.NET 连接数据库并进行简单数据库查询的方法。
- 了解使用 Ajax 进行无刷新界面编程。
- 了解如何创建和使用 XML Web Service。
- 掌握.NET Framework 的基本组成和类库组成。

## 本书适用对象

本书主要面向希望初步掌握 Visual C# 2008 编码能力的读者，帮助读者了解 C#基本程序结构、C#内置数据类型、面向对象思想，以及使用 Visual C# 2008 进行面向对象编程的基本方法。此外，本书还帮助读者掌握范型、委托的使用，以及程序的异常处理和对程序的简单调试方法。最后读者还将了解如何使用 Visual C# 2008 进行 Windows 窗体应用程序和 Web 应用程开发，以及如何进行 Ajax，XML Web Service 编程的基本方法。

## 本书结构

本书共分为 11 章，内容分为 4 大部分。

- 编程入门：

对于没有任何编程经验的读者，先要建立编程和开发的概念，并通过动手实践增加感性认识。本部分基于上述目标作如下设计。

第 1 章介绍编程开发的入门知识和概念，介绍开发工具的安装并运行第一个程序，建立感性认识。

第 2 章介绍了 Visual C#编程语言和编程工具，并接触可视化开发的概念。

- 基础语言：

对于任何编程语言，其基础元素基本一致。本部分介绍开发的基础知识，帮助读者开发最简单的程序，从而了解程序解决问题的基本方法和基本思路。

第 3 章介绍数据类型、变量、表达式和范型的使用。

第 4 章介绍分支、循环语句。

- 进阶学习：

掌握了基础知识以后，读者需要了解应用程序的开发方法，面向对象，以及一些语法知识和基本调试能力。

第 5、第 6 章主要介绍面向对象的一些基础知识，包括类的抽象、封装、继承、多态，以及 Visual C# 2008 中的新特性，包括自动属性、Lambda 表达式、对象和集合初始化器，以及使用 LINQ 查询表达式。

第 7 章　讲解调试程序的基本知识，以及如何对程序进行调试生成和排错。

第 8 章　讲解数组和字符串的相关知识。

- 知识拓展：

掌握了程序编写的技术知识以后，读者需要能够开发一些简单的应用，并对后续要学习的开发知识有所展望。这一部分的设计就是基于上述两部分目标。

第 9 章　介绍 Windows 应用程序开发。通过本章的学习读者可以开发简单的 Windows 应用程序，并能完成本书的案例。

第 10 章　初步介绍 Web 应用程序开发、ADO.NET 技术、Ajax、XML Web Service 技术。这部分帮助读者对这些技术建立感性认识和初步概念，后续课程会详细介绍。

第 11 章　介绍.NET Framework，帮助读者初步了解.NET Framework 的组成，对类库应用建立感性认识，后续课程会详细介绍。最后本书附带了一个地铁查询系统综合实例，帮助读者进一步了解如何使用 Visual Basic 2008 进行项目开发。

## 教学参考资料

获得微软授权使用该教材进行教学的教师，除了可以获得上述材料外，还可以从微软公司获得如下教学资料。

| 教参 | 章节教参与教学大纲 |
| --- | --- |
| 实验与习题答案 | 实验的具体操作步骤和习题答案 |
| 课外阅读 | 对于委托和事件的进一步学习 |
| 代码 | 每章节中的示例代码以及案例设计的源代码 |
| PowerPoint | 用于进行课堂演示 |

# 目　录

# 第1章 概 述

当今社会，软件无处不在。软件是软件开发人员根据要实现的功能，编写出的可以被计算机执行的指令，从而控制计算机完成相应的任务。

计算机可以直接识别的“语言”（指令）是二进制代码。普通开发人员直接用二进制代码编写软件是困难的。于是，高级编程语言诞生了。高级编程语言的代码近似于人类的语言，可以转换成为二进制代码执行。这样，开发软件就方便多了。

在软件开发过程中，开发人员需要编写大量的代码。同时，他们也会遇到各种导致工作效率下降的问题，例如，输入代码时出现语法错误，程序调试效率低下，等等。集成开发环境（Integrated Development Environment，IDE）的出现缓解了这些问题。它集成了从代码编写到调试、排错等各种开发工具，减轻了开发人员的工作负担，提高了开发效率。

现在的软件已经越来越复杂和庞大，开发工作一般都是在某个软件（也叫做运行环境）的基础上进行开发的。开发出来的软件通过调用下层软件的功能来实现。例如，记事本软件Notepad 是在 Windows 操作系统上开发出来的。Windows 操作系统就是 Notepad 的运行环境。Notepad 对计算机的操作都是通过调用 Windows 的功能来实现的。

作为一名开发人员，其开发工作基本上都离不开上面的 3 个要素，即：编程语言、集成开发环境和运行环境。具体到本书中就是 C#编程语言、Visual Studio 集成开发环境和.NET Framework 运行环境。在接触具体语言、开发环境和运行环境之前，先来详细了解一些基础知识。学完本章后读者可以：

- 了解最基本的计算机软、硬件知识。
- 了解使用编程方式来解决实际问题的思想和过程。
- 初步了解数据结构和算法的概念。
- 了解编程的概念和常用的编程工具。
- 掌握安装 Visual Studio 2008 的方法。
- 了解简单 C#程序的基本结构。
- 掌握 C#的代码格式。

## 1.1 计算机软件与硬件相关知识简介

### 1.1.1 计算机软件

计算机软件是指令、数据和文档的集合。它具有以下两个特征：

- 指令按照特定顺序组织，能够使计算机具有信息处理能力。
- 标志一定功能、完成一定任务或产生一定结果。

比如 Windows 中的“记事本”工具，它是由一系列程序指令组成的，可以完成文字处理

的任务，能够输入或输出文字数据，还包括一个帮助说明文档。这就是一个软件。

软件分为系统软件和应用软件。

系统软件中最典型的就是操作系统（Operating System，OS）。操作系统管理、控制各种计算机硬件资源（如内存、处理器和硬盘空间等），把这些资源分配给运行在操作系统上的应用程序，同时还对这些应用程序进行控制和管理。比如，平时玩的游戏，程序本身不具备直接操作硬件设备（如显卡和声卡）的能力，它需要调用操作系统提供的接口才能实现。

应用软件运行在操作系统上，实现各种应用功能。它们不能脱离操作系统而独立运行，甚至不能在不同的操作系统上运行，比如 Internet 浏览器、Office 软件套装和 Windows Media Player 多媒体播放软件等，这些软件都可实现不同的应用，又基于 Windows 操作系统实现各自的功能，它们都是应用软件。

在平时的使用中，最常见的两种应用软件是 Windows 应用软件和 Web 应用软件。

Windows 应用软件就是基于 Windows 平台的应用软件。这类软件通过调用 Windows 平台提供的各种接口来实现自己的功能，有很强的平台依赖性。

Web 应用软件一般是基于浏览器/服务器（Browser/Server，B/S）架构的软件，这类软件的客户端不依赖于特定的平台，只需要浏览器能正常显示 HTML 编写的代码和脚本语言（如 VBScript 和 JavaScript）编写的脚本。大部分的功能都由服务器端提供。例如，在线电子商务网站、BBS 论坛、基于企业内部网的报价单或销售报告的管理和生成工具等。

时下流行的 XML Web Service 也可以算是一种 Web 应用软件，它以 XML 和 SOAP 为标准，可以和其他应用程序相结合构建分布式系统。例如，某网站提供了天气信息查询的 XML Web Service。有一个程序员想自己编写一个桌面提醒程序，每天提醒用户注意当天的天气。要实现此功能，程序员只要实现桌面提醒的功能即可，而天气信息可以通过调用天气查询的 XML Web Service 获得，不需要自己去开发一套天气信息查询系统。

本书所介绍的 C#语言可以在.NET Framework 开发平台的基础上，构建上述所说的各种应用程序，包括 Windows 应用软件和 Web 应用软件。

### 1.1.2 计算机硬件

软件是运行在硬件上的，那么软件又是如何使用各种硬件设备的呢？为此，首先来了解一下计算机硬件的组成。

冯·诺依曼体系结构指出，计算机应该遵循如下原则：

- 计算机要执行的指令和要处理的数据都采用二进制表示。
- 把要执行的指令和要处理的数据按照顺序编成程序，存储到计算机内部让它自动执行。

冯·诺依曼体系结构是大部分现代计算机运行的理论基础。现代计算机的结构主要由以下几部分组成。

① CPU（Central Processing Unit）——中央处理器。其主要功能是进行算术和逻辑运算，内部结构大概可以分为控制单元、算术逻辑单元和存储单元等几个部分。按照其处理信息的字长可以分为 8 位微处理器、16 位微处理器、32 位微处理器以及 64 位微处理器等。

② 存储器——在计算机中用来存放信息的部件。现代计算机中的存储器一般都由内存和

外存组成。内存由半导体元件组成，可分为两类：只读存储器（Read-Only Memory，ROM）和随机存储器（Random Access Memory，RAM）。ROM 中的信息只能读出，不能写入，断电时信息不会丢失，典型应用就是基本输入/输出系统（Basic Input/Output System，BIOS）。而 RAM 中的信息可读、可写、可修改，只要计算机断电，信息就会丢失。由于 RAM 中的信息在计算机断电时会丢失，因此，应及时将 RAM 中的信息转移。外存储器主要包括软盘、硬盘和光盘。硬盘是计算机主要的存储设备，容量大、读/写速度快。光盘是一种新的存储器，容量大，相对于硬盘而言其速度较慢，在计算机中必须依赖光盘驱动器（光驱）来读取光盘上的信息。

③ 输入设备——用于向计算机软件输入指令或者要处理的数据。最常见的计算机输入设备包括鼠标和键盘。这也是标准配置。此外还可能有触摸屏、手写板、语音识别器等设备可以接受用户输入。

④ 输出设备——输出设备种类多样，标准的输出设备是显示器。此外可能还包含音箱和打印机等。

在了解了计算机的基本硬件组成以后，读者也许要问：软件是如何在这样的操作系统上执行的呢？

首先，软件需要存储在一种永久性存储介质上，可能是硬盘或光盘上。在软件要运行的时候，操作系统会把软件加载到内存中，然后由 CPU 按照顺序或者并行地执行软件指令。在此过程中，软件可能需要用户从输入设备上输入一些数据进行处理，同时以各种形式在输出设备上把提示和结果信息传递给用户，如使用音箱播放声音、在显示器上显示图像等。

## 1.2　以编程方式解决问题的一般方法

首先来看一个例子：在期终考试结束以后，计算一下本班的数学成绩平均分。

如果要以程序来解决这个问题，一般可以遵循如下流程。

① 分析。也就是定义问题。一定要明确问题的实质是什么，即分析什么是最终输出，什么是必要输入，必要输入是如何变成最终输出的。

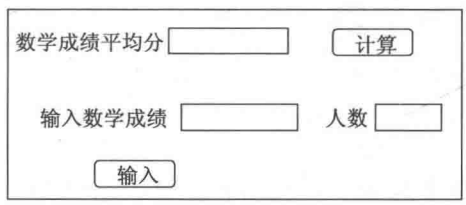
上面这个问题中，最终输出的是本班的数学成绩平均分。要获得平均分，必须获得两个数据：本班数学成绩的总和以及本班人数。

② 设计。设计出解决问题的方法。根据问题找到正确解决问题的一套逻辑方法，通常被称为算法。在这个问题中，核心算法为：

本班数学成绩平均分=本班数学成绩的总和/本班人数

③ 选择用户界面。确定以什么方式接受用户输入。

制作一个应用程序界面。用不同的文本框来输入成绩，显示人数和最终的平均分。用户可以用“输入”按钮确认输入成绩，用“计算”按钮确认成绩输入完毕，同时计算并显示平均分。



④ 代码实现。用编程语言把算法转换成程序代码的过程。本例中用 C#编写代码实现上面的例子。

⑤ 测试和调试。查找并解决程序中的任何错误。因为很难保证所编写的程序中不会有一些小的失误，那么就要在编程结束后，以各种形式对程序进行测试，并解决所发现的问题。

⑥ 完成程序的文档说明，包括对问题和程序的描述，如解决问题的方法，用户界面如何使用等。

### 1.2.1 随堂练习

编写一个“数学成绩平均分计算器”的使用说明。描述用户如何输入成绩，并且得到平均分。

### 1.2.2 数据结构和算法

计算机解决问题的实质就是高速的算术和逻辑运算。对于一个程序来说，其本质就是计算方法和数据，用术语来说就是算法和数据结构。

算法是指在有限步骤内求解某一问题所使用的一组定义明确的规则。简单地说，就是计算机解题的过程。

在实际开发过程中，首先要建立解决问题的算法，然后再用程序来实现算法。比如上题中计算数学平均成绩，首先得出计算平均成绩的算法，然后用程序来实现。

一个算法应该具有以下 5 个重要的特征：

- 有穷性：一种算法必须保证执行有限步骤之后结束。
- 确切性：算法的每一步骤必须有确切的定义。
- 输入：一种算法有 0 个或多个输入，以刻画运算对象的初始情况。
- 输出：一种算法有一个或多个输出，以反映对输入数据进行加工后的结果。
- 可行性：算法原则上能够精确地运行，而且人们用笔和纸做有限次运算后即可完成。

算法是程序的灵魂。算法的优劣直接决定了程序的执行效率和质量。例如，对于计算机下棋程序，算法好的程序，同一时间可以搜索的棋步比算法差的程序高出很多，那么它就会比后者“聪明”很多。

数据结构是计算机存储和组织数据的方式。作为一门学科，数据结构主要研究数据的各种逻辑结构和存储结构以及对数据的各种操作。数据结构用于将真实世界中的各种数据有效地呈现和存储在计算机中。这就好比是在仓库中存放货物，不同的货物有不同的包装和存放方法。包装、存放合理，取用的时候就方便，反之则会导致很多问题。

数据结构和算法是密切相关的。不同的数据结构往往需要不同的算法对其进行操作。

## 1.3 编程语言和开发工具

### 1.3.1 编程语言

编程语言通常分为 3 类：机器语言、汇编语言和高级语言。

1．机器语言

机器语言是用二进制代码表示的、计算机能直接识别和执行的机器指令的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言具有灵活、直接执行和速度快等特点。

用机器语言编写程序，编程人员首先要熟记所用计算机的全部指令代码和代码的含义。编写程序时，程序员得自己处理每条指令和每一数据的存储分配和输入/输出，还得记住编程过程中每步所使用的存储单元处于何种状态。这是一件十分繁琐的工作，编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且，编出的程序全是0和1的指令代码，直观性差，还容易出错。现在，除了计算机生产厂家的专业人员外，绝大多数程序员已经不必学习机器语言了。

2．汇编语言

为了克服机器语言难读、难编、难记和易出错的缺点，人们就用与代码指令实际含义相近的英文缩写词、字母和数字等符号来取代指令代码（如用ADD表示运算符号“+”的机器代码），于是就产生了汇编语言。所以说，汇编语言是一种采用了助记符的、仍然是面向机器的计算机语言。汇编语言采用助记符来编写程序，比用机器语言的二进制代码编程要方便些，在一定程度上简化了编程过程。汇编语言的特点是用符号代替了机器指令代码，而且助记符与指令代码一一对应，基本上保留了机器语言的灵活性。使用汇编语言能较好地发挥机器的特性，程序执行效率较高。

由于汇编语言中使用了助记符，用汇编语言编制的程序送入计算机，计算机不能像对待机器语言编写的程序那样直接识别和执行，必须通过预先放入计算机的“汇编程序”的加工和翻译，才能变成能够被计算机识别和处理的二进制代码程序。用汇编语言等非机器语言书写好的符号程序称为源程序，运行时，“汇编程序”要将源程序翻译成目标程序。目标程序是机器语言程序，它一旦被安置在内存的预定位置上，就能被计算机的CPU处理和执行。

汇编语言像机器指令一样，是硬件操作的控制信息，因而仍然是面向机器的语言，使用起来还是比较繁琐费时，通用性也差。汇编语言是低级语言。但是，使用汇编语言来编写系统软件和过程控制软件，其目标程序占用内存空间少，运行速度快，有着高级语言不可替代的优势。

3．高级语言

不论是机器语言还是汇编语言，都是面向硬件进行具体操作的。语言对机器的过分依赖，要求使用者必须对硬件结构及其工作原理都十分熟悉，这对非计算机专业人员来说是难以做到的，对于计算机的推广应用也非常不利。计算机事业的发展，促使人们去寻求一些与人类自然语言相接近且能为计算机所接受的语义确定、规则明确、自然直观、通用易学的计算机语言。这种与自然语言相近并为计算机所接受和执行的计算机语言称为高级语言。高级语言是面向用户的语言。高级语言编写的程序需要编译或者解释成为机器语言才能在计算机上运行。

需要指出的是，高级语言编写的程序转换为低级语言编写的程序有两种方法：解释或者编译。

解释：执行方式类似于日常生活中的“同声翻译”，应用程序源代码一边由相应语言的解释器“翻译”成目标代码（机器语言），一边执行，因此效率比较低，而且不能生成可独立执

行的可执行文件，应用程序不能脱离其解释器，但这种方式比较灵活，可以动态地调整、修改应用程序。

编译：编译是指在源程序执行之前，就将程序源代码“翻译”成目标代码（机器语言），因此其目标程序可以脱离其语言环境独立执行，使用比较方便、效率较高。但应用程序一旦需要修改，必须先修改源代码，再重新编译生成新的目标文件才能执行。只有目标文件而没有源代码，修改很不方便。现在大多数的编程语言都是编译型的，例如 C++、C#、Delphi 等。

使用高级语言编写的一行行程序代码被称为源代码，编译后的程序称为目标代码或者可执行程序。

C#语言编写的源程序扩展名为.cs。如果计算机上已经安装了.NET Framework，则可使用.NET 命令行窗口，运行命令 csc xxx.cs 来编译刚写好的程序。此时会在 xxx.cs 的同一目录下出现一个名为 xxx.exe 的程序。通过双击此 xxx.exe 文件名，可以执行程序。

### 1.3.2 开发工具

开发人员可以使用 Windows 自带的“记事本”来进行程序的开发。但是，如果所有的代码都要手动添加和输入，然后手动编译程序，使用命令调试程序等，那么工作效率非常低。为了解决上述问题，集成开发环境应运而生。集成开发环境集成了各种工具和功能方便程序员开发，如防止程序员犯低级错误或者笔误的自动语法检查、方便的图形化调试功能等，大大减少了编码的语法错误并降低了代价，使得程序的运行更简单。微软在推出 C#这种开发语言的同时，也推出了对应的集成开发环境 Visual Studio 系列产品。

本书仅介绍最新版本的 Visual Studio 2008 集成开发环境的使用。Visual Studio 2008 在很多方面进行了升级，并且新增了很多功能，这些方面包括：开发环境，代码编辑，项目、解决方案和项目生成、测试和部署调试器，帮助以及扩展性和自动化。

下面就先来安装 Visual Studio 2008。

### 1.3.3 安装 Visual Studio 2008

本节主要介绍如何在计算机上安装 Visual Studio 2008 开发工具包。

1．准备工作

在开始安装前需要做如下准备。

（1）硬件

- 处理器：频率在 1.6 GHz 或以上，建议采用 2.2 GHz 或更快的 CPU。
- 系统内存：最低要求 384 MB，推荐 1024 MB。
- 硬盘：如果不安装 MSDN，则安装驱动器上要有 3.8 GB 可用空间，系统驱动器上要有 1 GB 可用空间。如果安装 MSDN，则在完全安装 MSDN 的安装驱动器上要有 6 GB 的可用空间；在进行默认 MSDN 安装的安装驱动器上要有 4 GB 的可用空间。系统驱动器上要有 1.8 GB 可用空间。
- 显示器：最低要求 1024x768 增强色 16 位，推荐使用 1280x1024 显示器。