



新手学编程系列

打开C# 程序设计大门的金钥匙

新手学 C#

明杰 等 编著

10

小时多媒体视频讲解

- ◎ 由浅入深：从基本概念开始讲解，逐步深入到实际开发
- ◎ 实例丰富：讲解知识点时穿插了240个实例，有较强的实用性
- ◎ 面向就业：提供了常见面试题，帮助读者了解入职面试的相关知识
- ◎ 案例典型：提供了2个取材于实际项目的案例，提高读者开发水平
- ◎ 视频教学：提供了10小时多媒体教学视频，学习起来更加直观



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



新手学编程系列

新手学

C#

明杰 等 编著

10

小时多媒体视频讲解

CD-ROM



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

内 容 简 介

C#是目前最流行的开发语言之一, 本手册详细讲解了从 C#的基础语法到程序整体设计的方法, 旨在帮助开发者快速掌握使用 C#语言编写程序的方法。

全书包括.NET Framework 环境以及 Visual Studio 开发工具、C#语言的基本语法、C#与面向对象、常用的字符串操作、线程操作、C#中的集合知识、C#中的网络通信、C#中文件与流的操作、程序的本地化、获取计算机信息、反射、消息队列、异常处理措施和 C#3.0 的新特性、数据的访问操作、ASP.NET 程序的开发、Windows 窗体程序的开发。为了方便于读者学习, 本书最后两章分别分析了 Web 程序和 Windows 窗体程序的开发过程。同时还介绍了 ASP.NET 下如何按照三层架构开发 Web 程序, 以及客户端/服务器端程序的开发流程。

本手册适合广大 C#语言的初学者, C++程序员学习 C#和大专院校学生阅读。

本光盘内容为实例源代码、语音视频教学及电子教案(PPT)。

本光盘及配套手册由北京希望电子出版社独家发行, 未经出版者书面许可, 任何单位和个人不得擅自摘抄、复制光盘和本手册的部分或全部内容, 并以任何方式进行传播。

需要本书或技术支持的读者, 请与北京清河 6 号信箱(邮编: 100085)发行部联系, 电话: 010-62978181(总机)转发行部、010-82702675(邮购), 传真: 010-82702698, E-mail: tbd@bhp.com.cn。

新手学 C# / 明杰等编著. —北京希望电子出版社, 2010
(新手学编程系列)

ISBN 978-7-89498-991-8

I. 新… II. 明… III. C#—程序设计

责任编辑: 刘 芯 / 责任校对: 马 君

责任印刷: 双 青 / 封面设计: 一品设计

北京希望电子出版社 出版

北京市海淀区上地三街 9 号嘉华大厦 C 座 611

邮政编码: 100085

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

北京市四季青双青印刷厂印刷

北京希望电子出版社发行 各地新华书店经销

*

2010 年 1 月第 1 版

2010 年 1 月第 1 次印刷

印数: 1-3 000

开本: 787mm×1092mm 1/16

印张: 30.5

字数: 707 千字

定价: 49.80 元(配 1 张光盘)

前言

随着信息信息技术的发展，要求项目开发周期越来越短，功能越来越强大，程序的稳定性越来越高。对于开发者来说，虽然 C 语言与 C++ 语言都具有强大的功能，但是它们太复杂。C 语言与 C++ 语言的很多功能，使用概率小却又不能不掌握，因此在开发时会潜藏很多不能发现的错误。这时微软公司开发了 C# 语言。C# 语言牺牲了 C 语言与 C++ 语言的部分功能，但是减少了开发者的负担，使开发者能够快速高效地开发出稳定的程序。

为了使开发者掌握如何利用 C# 语言编写程序，本手册全面介绍了 C# 语言的基本知识点，同时还通过一系列的实例，介绍了编程过程中需要经常使用的各种操作。

为了让读者更方便地学习，光盘提供全程多媒体教学视频及配套手册，包括知识讲解和操作演示。读者可以通过光盘很轻松地掌握书中各项内容。

本手册的特点

1. 循序渐进，由浅入深

为了使读者更好地理解 C# 语言的特性，本手册从编译环境开始，由最基本的语法知识介绍到程序整体设计，读者可以按照本手册的编写章节进行学习。阅读完本手册后，读者可以单独编写功能模块。

2. 内容充实，知识点与实际编程紧密相关

本手册不但包含了 C# 语言的知识，还包含了面向对象程序设计的知识。手册中详细地介绍了面向对象的编程，还给出了实际编程时常用的设计方法。对于 C# 语言的知识点，本手册从实际出发，将项目开发过程中需要使用的知识点一一介绍。

3. 实例丰富，介绍透彻

本手册包含丰富的代码实例，做到了每一个重要的知识点都有代码来演示如何使用，每个实例都经过编译器调试通过。同时每一个实例包含丰富的注释，读者可以非常容易地理解代码的编写思路。读者可以将每个实例都输入到编译器中，从而逐渐掌握 C# 程序的编写思路。

4. 案例精讲，使读者能对 C# 编程的整个过程有所了解

本手册最后给出了两个案例，一个是使用 C# 编写 Web 程序，另一个是使用 C# 编写 Windows 窗体程序。通过两个程序的介绍，用户可以了解到如何使用 C# 进行 Web 程序以及 Windows 窗体程序的编写。

本手册的内容

第 1 篇：介绍了 C# 的运行环境、基本语法，编写 C# 程序的 Visual Studio 2008 编译器的安装，并介绍了如何创建不同类型的 C# 程序。

第 2 篇：介绍了面向对象的基本知识，面向对象程序设计中的继承机制，面向对象程序

设计中接口的使用，面向对象程序设计时常用的设计方法。

第3篇：介绍了如何进行字符串的操作，线程、存储数据集合、反射和消息队列的相关知识，C#中网络通信功能的实现，如何进行文件的操作，如何实现程序在不同的区域实现不同的内容，计算机的组成以及如何获取计算机的基本信息，C#3.0的一些新特性以及C#的异常捕获机制。

第4篇：介绍了如何利用C#继续数据库相关的数据操作，ASP.NET设计架构以及一些常用控件的使用，Windows窗体的基本知识以及一些常用的控件。

第5篇：介绍了一个基于Web的ASP.NET程序和一个基于Windows窗体的程序。

源代码

在使用本手册中的例子时，可以手工输入所有代码，也可以直接使用光盘里的源代码文件。有关代码方面的问题可与 martt0656@163.com 联系。

适合的读者

- ☐ C++程序员希望学习C#编程
- ☐ 希望学习C#语言的初学者
- ☐ 有C#语言基础，但对C#中某些操作不熟悉的人员
- ☐ 大中专院校的学生
- ☐ 社会培训班学生

本手册作者

本手册主要由明杰编著，其他参与编著的人员有卜庆玲、冯曼菲、匡妍娜、雷成健、李小波、刘浩然、刘会神、马震、齐志华、舒军、孙大林、王辉、王沛、王石、王晓悦、熊英、张杰、袁福庆、赵显琼、韩延峰、李刚、张佳楠、张金霞、孔鹏、张昆、陈刚、姚志鹏等。在此一并表示感谢！

编著者

目 录

第 1 篇 C#基本语法

第 1 章 C#与.NET 概述	2
1.1 C#语言简介	2
1.1.1 C#语言的特点	2
1.1.2 C#与其他面向对象语言的比较	3
1.2 .NET Framework 环境	3
1.2.1 C#与.NET Framework 的关系	3
1.2.2 .NET Framework 的组成及 发展历史	3
1.3 C#创建.NET 应用程序	6
1.3.1 编译器 VS 安装的介绍	6
1.3.2 创建 ASP.NET 的 Web 程序	9
1.3.3 创建 Windows 程序	10
1.3.4 创建类库	13
1.4 常见面试题	14
1.5 小结	14
1.6 习题	15
第 2 章 C#基本语法	16
2.1 变量和常量	16
2.1.1 变量的命名与作用域	16
2.1.2 命名规范	17
2.1.3 常量	18
2.2 数据类型	18
2.2.1 常用的值类型	19
2.2.2 引用类型	20
2.3 运算符和表达式	20
2.3.1 运算符	21
2.3.2 表达式	23
2.4 流程控制	23
2.4.1 条件语句	23
2.4.2 循环语句	27
2.4.3 跳转语句	29
2.5 数组和枚举	30
2.5.1 数组	30

2.5.2 枚举	32
2.6 常用的预处理器指令	34
2.6.1 #region 与#endregion	34
2.6.2 #if,#else 和#endif	34
2.7 常见面试题	35
2.8 小结	35
2.9 习题	36

第 2 篇 面向对象的程序设计

第 3 章 类和结构	38
3.1 面向对象	38
3.1.1 面向对象的由来	38
3.1.2 面向对象举例	39
3.2 类定义	40
3.2.1 类的数据成员	40
3.2.2 类的成员函数	41
3.2.3 类的静态成员	42
3.2.4 类成员的保护机制	43
3.2.5 object 类	43
3.3 类的初始化、赋值和析构	44
3.3.1 类的初始化	44
3.3.2 类的构造函数	45
3.3.3 类的析构函数	47
3.3.4 按成员赋值	47
3.4 类的方法	48
3.4.1 方法的声明方式	48
3.4.2 方法设计的一般准则	49
3.4.3 方法中的参数	50
3.4.4 静态方法与非静态方法	53
3.4.5 方法的返回值	54
3.4.6 方法的重载	55
3.4.7 操作符的重载	56
3.5 类的属性	57
3.5.1 属性的声明	57
3.5.2 属性的访问	58

3.6 结构.....	59	5.4 常见面试题.....	86
3.6.1 定义结构.....	59	5.5 小结.....	89
3.6.2 使用结构.....	60	5.6 习题.....	89
3.6.3 结构间的转换.....	61	第6章 类与设计模式.....	90
3.6.4 结构与类的比较.....	63	6.1 类实现简单工厂模式.....	90
3.7 常见面试题.....	64	6.1.1 简单工厂模式的应用 场合及组成.....	90
3.8 小结.....	65	6.1.2 构造简单工厂.....	91
3.9 习题.....	65	6.2 类实现工厂模式.....	93
第4章 继承.....	66	6.2.1 工厂模式的应用场合及组成.....	93
4.1 继承机制的简介.....	66	6.2.2 构造工厂.....	93
4.1.1 继承是什么.....	66	6.3 类实现抽象工厂模式.....	96
4.1.2 基本概念.....	67	6.3.1 抽象工厂模式的应用 场合及组成.....	96
4.1.3 何时使用继承.....	67	6.3.2 构造抽象工厂.....	96
4.2 多态性.....	69	6.4 类实现单件模式.....	99
4.2.1 多态性是什么.....	69	6.4.1 单件模式的应用场合及 使用方式.....	99
4.2.2 虚函数.....	70	6.4.2 创建一个单件的全局访问点.....	99
4.2.3 派生类中虚函数的重载.....	70	6.5 类实现生成器模式.....	100
4.3 继承的类型.....	71	6.5.1 生成器模式的应用场合及组成.....	101
4.3.1 公有继承.....	71	6.5.2 生成器模式的效果.....	101
4.3.2 受保护的继承.....	72	6.6 类实现组合模式.....	104
4.3.3 私有继承.....	73	6.6.1 组合模式的应用场合及组成.....	104
4.4 抽象与密封.....	73	6.6.2 常见的组合实例.....	105
4.4.1 抽象与密封的概念.....	74	6.7 类实现外观模式.....	108
4.4.2 抽象类与抽象方法.....	74	6.7.1 外观模式的应用场合及组成.....	108
4.4.3 密封类和方法.....	76	6.7.2 外观模式的效果.....	109
4.5 常见面试题.....	76	6.8 常见面试题.....	111
4.6 小结.....	78	6.9 小结.....	114
4.7 习题.....	78	6.10 习题.....	114
第5章 接口.....	79	第3篇 C#实践应用	
5.1 接口概述.....	79	第7章 字符串与数字的操作.....	116
5.1.1 接口是什么.....	79	7.1 字符串基本知识.....	116
5.1.2 基本概念.....	79	7.1.1 字符串的表示.....	116
5.1.3 接口的组成.....	80	7.1.2 String 类.....	117
5.2 接口的定义.....	80	7.1.3 StringBuilder 类.....	117
5.2.1 接口的声明方式.....	80	7.2 字符串的转换操作.....	118
5.2.2 接口的继承方式.....	81	7.2.1 字符串的分割.....	118
5.3 接口的实现.....	82	7.2.2 子串的获取.....	119
5.3.1 类对接口实现.....	82		
5.3.2 多接口的继承.....	83		
5.3.3 显式的实现接口.....	85		
5.3.4 抽象类与接口的区别.....	86		

7.2.3 字符串的比较.....	119	9.2.3 Stack 集合.....	158
7.2.4 字符串的合并.....	120	9.2.4 HashTable 集合.....	159
7.2.5 字符串的格式.....	122	9.3 泛型集合的使用.....	160
7.2.6 字符串的替换、查找与删除....	123	9.3.1 Queue 与 Stack 形式的 泛型集合.....	161
7.2.7 字符串的其他操作.....	123	9.3.2 List 形式的泛型集合.....	162
7.3 数字的转换操作.....	124	9.4 常见面试题.....	163
7.3.1 显示的数字转换.....	124	9.5 小结.....	164
7.3.2 数字与字符串和其他类型 数字类型的转换.....	125	9.6 习题.....	164
7.4 常见面试题.....	127	第 10 章 网络通信操作.....	165
7.5 小结.....	127	10.1 网络通信的协议.....	165
7.6 习题.....	127	10.1.1 什么是网络通信.....	165
第 8 章 线程操作.....	128	10.1.2 网络模型.....	165
8.1 线程.....	128	10.1.3 TCP/UDP 协议.....	167
8.1.1 线程声明.....	128	10.1.4 什么是 Socket.....	168
8.1.2 线程的启动.....	129	10.2 Socket 类的使用.....	168
8.1.3 线程的悬挂或暂停.....	130	10.2.1 如何创建 Socket 类.....	168
8.1.4 线程的终止.....	131	10.2.2 利用 Socket 监听端口.....	169
8.1.5 线程如何调用资源.....	133	10.2.3 利用 Socket 建立连接.....	169
8.2 委托与事件.....	134	10.2.4 利用 Socket 接收和发送消息....	170
8.2.1 使用委托的意义和使用方式....	134	10.2.5 利用 Socket 关闭连接.....	171
8.2.2 简单的委托示例.....	137	10.2.6 同步接收方的套接字.....	171
8.2.3 委托与事件的关系.....	139	10.2.7 同步发送方的套接字.....	172
8.3 多线程的处理.....	142	10.2.8 异步接收方的套接字.....	173
8.3.1 多线程的工作方式.....	142	10.2.9 异步发送方的套接字.....	176
8.3.2 线程池.....	143	10.3 一些相关的通信类.....	179
8.3.3 线程的优先级.....	144	10.3.1 利用 TcpClient 建立 客户端请求.....	179
8.3.4 线程的同步.....	145	10.3.2 利用 TcpListener 开始和 关闭监听.....	180
8.3.5 使用共享的资源.....	150	10.3.3 利用 TcpListener 接收连接.....	181
8.4 常见面试题.....	152	10.3.4 利用 TcpClient 发送和 接收消息.....	182
8.5 小结.....	153	10.3.5 利用 UdpClient 建立 客户端连接.....	184
8.6 习题.....	153	10.3.6 利用 UdpClient 发送和 接收消息.....	184
第 9 章 集合.....	154	10.4 常见面试题.....	186
9.1 集合的基本知识.....	154	10.5 小结.....	187
9.1.1 集合的使用场合.....	154		
9.1.2 使用集合的优点.....	154		
9.2 非泛型集合的使用.....	155		
9.2.1 ArrayList 集合.....	155		
9.2.2 Queue 集合.....	157		

10.6 习题	187	12.3.2 运行时读取本地化资源	216
第 11 章 文件与流	188	12.4 常见面试题	219
11.1 目录操作	188	12.5 小结	219
11.1.1 如何创建目录	188	12.6 习题	220
11.1.2 如何删除目录及子目录	189	第 13 章 获取计算机信息	221
11.1.3 如何获取目录下文件信息	190	13.1 计算机组成简介	221
11.1.4 如何获取目录信息	190	13.1.1 计算机硬件系统	221
11.2 文件操作	191	13.1.2 计算机软件系统	222
11.2.1 创建文件	191	13.2 获取本机的信息	222
11.2.2 复制文件和删除文件	192	13.2.1 获取本机的软件信息	222
11.2.3 加密与解密文件	193	13.2.2 获取本机的硬件信息	225
11.2.4 读取和修改文件内容	194	13.2.3 获取本机的网络信息	228
11.3 流操作	194	13.3 常见面试题	229
11.3.1 流的概念	195	13.4 小结	229
11.3.2 使用流读取文件	195	13.5 习题	230
11.3.3 使用流写入文件	197	第 14 章 反射	231
11.3.4 二进制文件的读取和写入	198	14.1 反射机制	231
11.4 FTP 中的文件操作	199	14.1.1 反射的运行机制	231
11.4.1 FTP 简介	199	14.1.2 C# 中反射应用的类	232
11.4.2 利用 FTP 创建连接	199	14.2 反射的应用	235
11.4.3 利用 FTP 在服务器上 进行目录操作	200	14.2.1 枚举中反射的应用	236
11.4.4 利用 FTP 上传文件	202	14.2.2 类中反射的应用	239
11.4.5 利用 FTP 下载文件	204	14.3 常见面试题	242
11.4.6 利用 FTP 删除文件	205	14.4 小结	243
11.4.7 利用 FTP 关闭连接	207	14.5 习题	243
11.5 常见面试题	208	第 15 章 消息队列	244
11.6 小结	209	15.1 消息队列的基本知识	244
11.7 习题	209	15.1.1 消息队列的概念	244
第 12 章 程序的本地化	210	15.1.2 消息队列的类型	245
12.1 本地化概述	210	15.1.3 消息队列的优缺点	245
12.1.1 本地化的体系	210	15.2 消息的处理	246
12.1.2 本地化的优点	211	15.2.1 消息的安全性	246
12.2 本地化的方式	211	15.2.2 创建消息队列	246
12.2.1 使用 C# 中的本地化类	211	15.2.3 发送消息到消息队列	248
12.2.2 创建资源文件	214	15.2.4 接收消息队列的消息	250
12.2.3 使用资源文件	214	15.2.5 查询消息队列内信息	252
12.3 本地化的实现示例	215	15.2.6 消息队列示例	255
12.3.1 设计本地化资源	215	15.3 常见面试题	260
		15.4 小结	260

15.5 习题	260	17.6 XML 介绍	292
第 16 章 其他常用操作	261	17.6.1 XML 的介绍	292
16.1 程序异常的捕获	261	17.6.2 XML 结构	292
16.1.1 异常类	261	17.6.3 读取 XML 文件	293
16.1.2 捕获异常	262	17.6.4 写入 XML 文件	298
16.2 C# 3.0 的新特性	265	17.6.5 利用 XML 填充数据	300
16.2.1 隐含类型的局部变量	266	17.6.6 利用 XML 存储数据	302
16.2.2 扩展方法	266	17.7 常见面试题	304
16.2.3 匿名类型	269	17.8 小结	305
16.2.4 对象与集合初始化器	270	17.9 习题	305
16.3 常见面试题	272	第 18 章 ASP.NET 的 Web 程序开发	306
16.4 小结	272	18.1 ASP.NET 介绍	306
16.5 习题	272	18.1.1 什么是 ASP.NET	306
第 4 篇 C# 开发方向		18.1.2 ASP.NET 的工作方式	307
第 17 章 数据访问	274	18.2 .NET 环境下 Web 页面基本控件使用	307
17.1 常用的数据库	274	18.2.1 使用 Label 控件	308
17.1.1 Oracle 数据库简介	274	18.2.2 使用 TextBox 控件	308
17.1.2 Access 数据库简介	275	18.2.3 使用 Button 控件	310
17.1.3 SQL Server 数据库简介	275	18.2.4 使用 ListBox 控件	312
17.2 .NET 下的数据库连接方式	276	18.2.5 使用 DropDownList 控件	314
17.2.1 通过字符串连接数据库	276	18.2.6 使用 CheckBoxList 控件	316
17.2.2 通过控件连接数据库	277	18.2.7 使用 FileUpload 控件	319
17.3 ADO.NET 概述	278	18.2.8 使用 GridView 控件	322
17.3.1 ADO.NET 的设计目标	278	18.3 网站部署的基本步骤	328
17.3.2 ADO.NET 的结构	279	18.3.1 部署网站的环境要求	328
17.3.3 ADO.NET 与 ADO 的区别	279	18.3.2 部署网站的步骤	329
17.4 SQL Server 数据库处理	280	18.4 常见面试题	330
17.4.1 利用 ADO.NET 连接 SQL Server 数据库	280	18.5 小结	331
17.4.2 利用 ADO.NET 执行 SQL Server 数据库的处理命令	281	18.6 习题	331
17.4.3 SQL Server 数据库处理示例	282	第 19 章 Windows 窗体的开发	332
17.5 利用 DataSet 类管理读取的数据	284	19.1 Windows 窗体开发知识简介	332
17.5.1 DataSet 类中的表	284	19.1.1 什么是 Windows 窗体开发	332
17.5.2 DataSet 的表关系	286	19.1.2 Windows 窗体程序的 工作机制	333
17.5.3 如何在 DataSet 中添加表	287	19.2 .NET 环境下 WinForm 基本 控件使用	333
17.5.4 填充 DataSet	288	19.2.1 Label 控件的使用	334
17.5.5 获取 DataSet 中的数据	289	19.2.2 TextBox、RichTextBox 与 Button 控件的使用	335
17.5.6 利用 DataSet 更新数据	291	19.2.3 TreeView 控件的使用	337
		19.2.4 ProgressBar 控件的使用	340

19.2.5	WebBrowser 控件的使用	341
19.2.6	TabControl 控件的使用	343
19.2.7	MenuStrip 与 ToolStrip 控件的使用	347
19.2.8	OpenFileDialog 控件的使用	351
19.2.9	SaveFileDialog 控件的使用	354
19.2.10	DataGridView 控件的使用	356
19.3	窗体	361
19.3.1	Form 类	361
19.3.2	多文档界面	363
19.3.3	自定义控件	367
19.4	常见面试题	370
19.5	小结	370
19.6	习题	370

第 5 篇 C#开发案例

第 20 章	基于 Web 的信息管理系统	372
20.1	系统分析	372
20.1.1	功能简介	372
20.1.2	系统的流程	372
20.1.3	系统的结构	374
20.2	系统详解	375
20.2.1	数据库设计	375
20.2.2	母版页的设计	377
20.2.3	通用类的设计	379
20.2.4	用户登录	385
20.2.5	用户注册	387
20.2.6	用户管理	391
20.2.7	信息的提交	395
20.2.8	计算机信息的管理	398
20.2.9	查询已提交的信息	402
20.3	主要技术	410
20.3.1	母版页的使用	410
20.3.2	SiteMap 的使用	411
20.3.3	Web 页面存储用户状态	413
20.4	小结	418
第 21 章	Windows 下的聊天室程序	419

21.1	聊天室程序分析	419
21.1.1	功能简介	419
21.1.2	工作流程	420
21.1.3	系统结构	420
21.2	聊天室程序公共类设计	421
21.2.1	发送消息的基类	421
21.2.2	请求连接的消息类与查询 在线用户消息类的设计	422
21.2.3	聊天消息与更新 消息的类设计	424
21.2.4	在线人员类的设计	425
21.2.5	消息转化方法类的设计	425
21.2.6	发送消息的方法类	431
21.3	聊天室程序服务器端设计	432
21.3.1	服务器端的界面设计	433
21.3.2	服务器端 DataGridView 控件数据源的创建	435
21.3.3	服务器端消息的接收	436
21.3.4	服务器端消息的分析	437
21.3.5	服务器端消息的发送	440
21.3.6	服务器端刷新在线用户并 断开指定的连接	441
21.4	聊天室程序客户端设计	443
21.4.1	客户端注册界面设计	443
21.4.2	客户端注册	446
21.4.3	客户端消息发送界面设计	449
21.4.4	客户端接收消息	453
21.4.5	客户端分析消息	454
21.4.6	客户端的查询请求	456
21.4.7	客户端之间的消息发送	458
21.4.8	客户端保存消息内容	459
21.4.9	客户端目录的快捷操作	460
21.5	聊天室程序的主要设计技术	461
21.5.1	多线程的使用	462
21.5.2	类的设计	466
21.5.3	跨线程使用资源	476
21.6	小结	477

第 1 篇 C#基本语法

第 1 章 C#与.NET 概述

第 2 章 C#基本语法

第 1 章 C#与.NET 概述

C#是微软公司开发的一种面向对象的设计语言。微软提供了团队管理器，从而能够快速地进行高效的团队开发工作。C#也具有完美的网络开发能力，遵循微软的网站开发框架，能够快速开发出功能强大的网络程序。本章将粗略地介绍 C#语言，内容包括：

- C#语言特点；
- .NET Framework 环境；
- 如何建立.NET 应用程序。

通过本章的学习，读者可以了解到 C#程序的运行环境。同时还能安装.NET 的编译环境，并能创建基本的.NET 应用程序。

1.1 C#语言简介

C#是从 C 和 C++演化而来的一门语言，是一门彻底地面向对象的语言。由于 C#不需要进行内存的管理，因此也是一门安全的语言。由于 C#的语法和主流的开发语言比较接近，因此学习起来相对简单。本节主要介绍 C#语言的特点及与其他面向对象语言的比较。

1.1.1 C#语言的特点

C#作为一种新的面向对象的语言，它语法简单，完全符合面向对象的思想，能够快速开发出功能较强的应用程序。它必须运行在微软的应用平台上，主要具有以下两个特点：

- 语法非常简单：C#属于 C 系语言，所以语法方面也继承了绝大部分 C 和 C++语言的特点。同时为了安全性，C#隐藏了 C++中的指针操作符，避免了直接操作内存。对于 C++中用于对面成员访问的操作符 (::、->、.)，C#也只保留了 “.” 操作符来访问对象的成员。C#还明确了值的类型，使 C++中的模糊比较操作不再出现。由于 C#规定了是强类型的，因此变量之间转化时，必须指明转换成的数据类型。
- 面向对象的设计思想：C#具有面向对象语言所应有的一切特性，封装、继承与多态。C#提供装箱与拆箱的操作来封装和获取对象，从而减轻程序员的负担。C#语言只允许单继承，所以一个类只有一个基类。如果用户想实现多继承，可以将某些功能设计为接口，然后由继承的类来实现接口功能。C#没有全局变量与全局函数，一切都必须封装在类中，使得代码的可读性增强，并降低了发生命名冲突的机会。

C#主要有三个版本，分别是 C#1.2、C#2.0 以及 C#3.0。每个版本还有一些补丁，完善当时版本的功能。其中 C#1.2 是最早的版本。

1.1.2 C#与其他面向对象语言的比较

- C#与 C++的区别: C#与 C++一样都是面向对象的语言。C#不能自己操作内存,由统一的垃圾管理机制来管理内存的产生和释放。但 C++可以操作内存,由程序员来管理内存的产生和释放。C#是完全面向对象的,不存在一个全局变量和函数;C++虽然也是面向对象的,但同时它仍然保留面向过程的特征。C#摒弃了 C++的类可以继承多个类的特征,采用接口来实现类的多重继承,减轻了开发者开发时管理多个类初始化的负担。
- C#与 Java 的区别: C#与 Java 从语言设计上来说基本上没有区别,它们都需要运行在“虚拟平台”上。但 C#提供了一个委托,用于封装命名方法或匿名方法,而 Java 却没有相应的方法来实现相似的功能。

▣ 技巧: 学习一门语言时,可以思考另一门语言对应的语法。通过类比学习,能够提高语言的掌握速度,同时了解语言的特点。

1.2 .NET Framework 环境

.NET Framework 是一个框架,这个框架负责管理 C#开发的程序。.NET Framework 同时还提供很多封装好的类来完成常用的功能。本节介绍.NET Framework 与 C#的关系,.NET Framework 的组成及版本历史。

1.2.1 C#与.NET Framework 的关系

C#是一种开发语言,它是专门为.NET Framework 而设计的一门语言。C#所使用的很多类都是由.NET Framework 封装。当程序员编写完代码后,编译器会将代码转换成.NET Framework 能够识别的中间代码,然后将代码托管给.NET Framework 管理。

但需要注意的是 C#是一门语言,而.NET Framework 是一个环境。虽然 C#是专门针对.NET Framework 开发的,但是两者不能理解为一个东西。例如,在 Windows 环境中,Windows 负责管理所有安装的应用程序,负责分配资源和回收资源给这些应用程序。应用程序不能直接去操作各种资源,只能通过 Windows 提供的一些功能接口来访问系统的资源。

本手册为了介绍方便,会将.NET Framework 提供的类库和相关功能,统一称为 C#提供的功能。但是两者还是有区别的。

1.2.2 .NET Framework 的组成及发展历史

.NET Framework 的公共运行库主要进行程序的执行管理,包括代码执行、内存管理、线程执行以及代码安全性验证等功能。.NET Framework 由公共运行库和类库组成。

对象产生时内存的分配和释放都由.NET Framework 的公共运行库来管理,因此不会出现当需要使用某个对象时却无法调用或者调用了错误的信。也不会出现当使用完某个对象后,却忘记释放这个对象占用的资源,造成后面的程序需要占用新的内存空间时,却没有内存可用的情况。

.NET Framework 的公共运行库还管理着托管的代码。程序员编写的代码首先被转化为中间代码，然后被转化为.NET Framework 可识别的专用代码。由于在公共运行库中，每种语言编写的代码转化为的专用代码后都是一样的，因此编译后的代码可以互相使用。也就是说，在.NET Framework 上用各种语言编写的类可以互相通信。例如，用 VB.NET 编写的类，可以在 C#编写的方法中作为参数使用。

.NET Framework 的类库是指由微软提供的、封装好的、进行各种特殊处理的类，如字符串类、线程类。这些类库都位于不同的命名空间中，在每一个命名空间中可能存在同一个名称的类，但是由于存在不同的命名空间因此也是唯一的。例如，同一款家具数量很多，但是由于买家具的人不同，因此家具的归属权也不同。

.NET Framework 的类库可以完成许多原先需要调用 Windows API 才能完成的任务。它将类的属性设置的简单易懂，例如转化为字符串的方法称为 ToString，得到变量类型的方法称为 GetType，变量名称的属性为 Name。通俗易懂的类的名称和属性大大减少了学习的难度，加快了开发的进度。在本手册的介绍中，为了方便，会将.NET Framework 的类库统称为 C#的类库。

基于.NET Framework 进行开发时，可以将代码委托给.NET Framework 管理，并且能够非常方便地调用.NET Framework 的类库来完成许多功能。基于.NET Framework 开发时，编译器会将代码编译为中间代码，然后将中间代码传递给.NET Framework，.NET Framework 根据中间代码分配适当内存调用相关的类库，最后生成操作系统能识别的 01 指令，具体过程如图 1-1 所示。

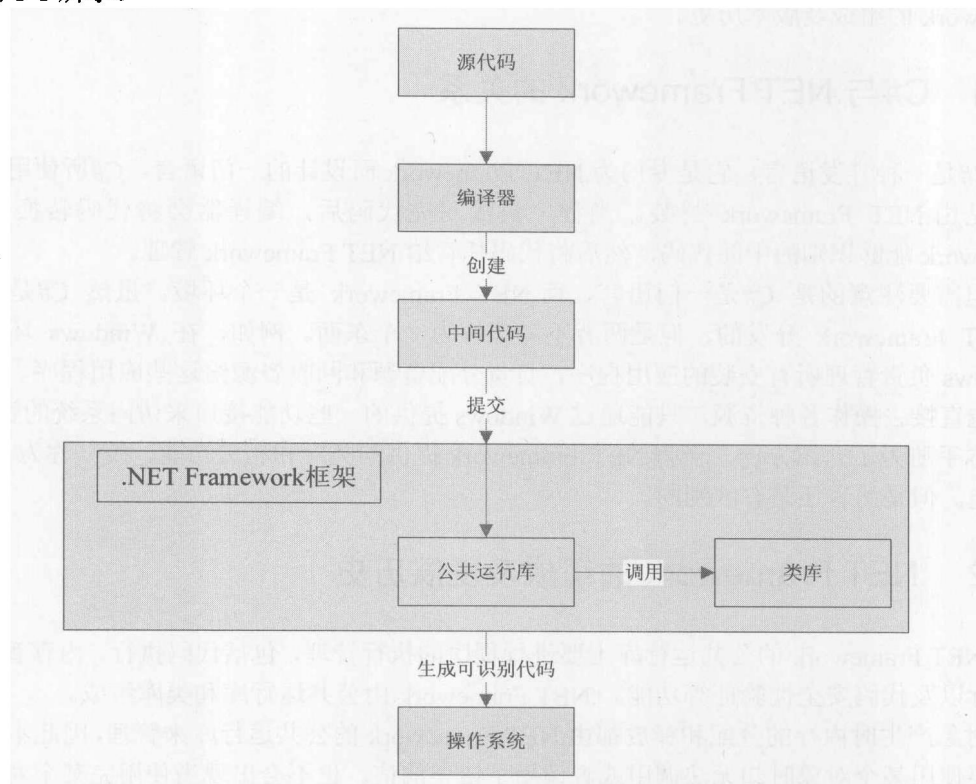


图 1-1 基于.NET Framework 的流程

说明：可以将.NET Framework 框架理解为一个管理器。它提供各种接口，作为机器和人的中间体。例如，现实世界中的钱，它是人和物品的中间转换体。

.NET Framework 的版本主要有四个版本，分别是.NET Framework1.1、.NET Framework2.0、.NET Framework3.0 以及.NET Framework3.5。其中每个版本都提供了相应的补丁程序来修补原先设计不够完善的地方。本手册基于.NET Framework3.5 版本，但是大部分代码示例也能在.NET Framework2.0 或者.NET Framework3.0 下运行。

对于没有安装.NET 框架的用户，首先需要在 <http://www.microsoft.com/downloads/details.aspx?familyid=333325fd-ae52-4e35-b531-508d977d32a6&displaylang=zh-cn> 地址中下载.NET Framework3.5 框架。下载完成后，按照以下步骤安装。

(1) 双击安装包，弹出如图 1-2 所示的界面。这个界面提示用户正在初始化系统。

(2) 初始化系统完成后，弹出如图 1-3 所示的界面。在这个界面中有提示用户安装需要遵守的约定。

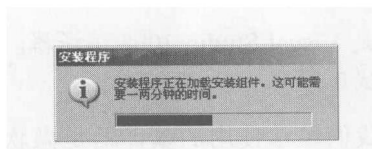


图 1-2 初始化窗口

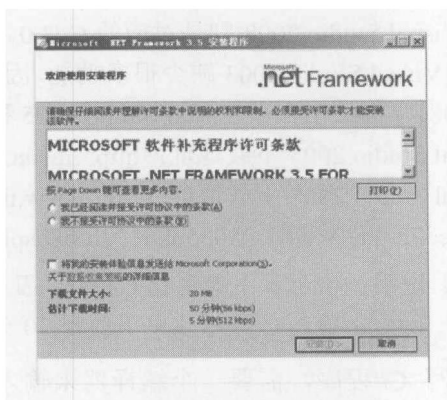


图 1-3 协议提示界面

(3) 接收许可后，单击【安装】按钮，弹出如图 1-4 所示的界面。这个界面是开始正式安装.NET Framework3.5 的框架。

(4) 安装完成后，弹出如图 1-5 所示的界面。在这个界面中提示用户安装完成。

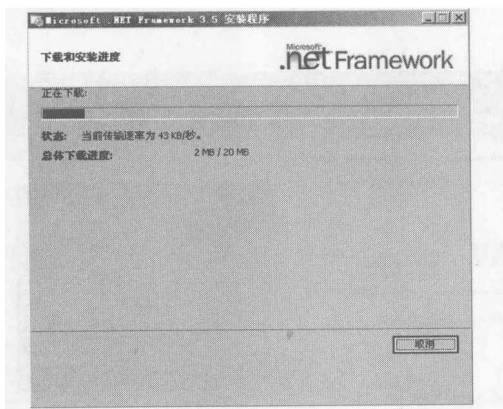


图 1-4 .NET Framework3.5 框架安装界面

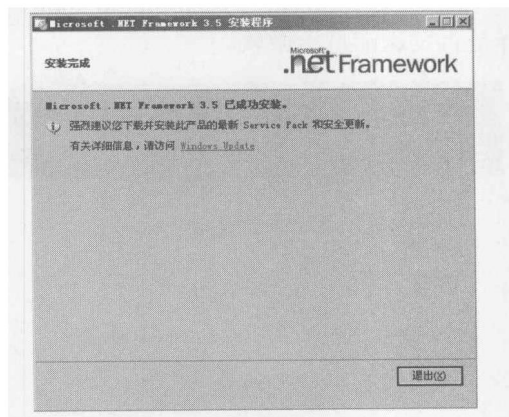


图 1-5 安装完成界面

安装完.NET Framework3.5 框架后，便可以运行 C#编写的程序。

1.3 C#创建.NET 应用程序

C#创建.NET 应用程序指 C#的开发方向。一般可以创建三种类型的程序：一种是 ASP.NET 的程序，另一种是 Windows 的程序，还有一种是类库程序。其中类库中包含类的定义和实现，可以被其他的程序调用，但不能作为单一的程序运行。本节介绍创建基于 C#的三种类型的程序。

1.3.1 编译器 VS 安装的介绍

微软 VS 编辑器主要有 Visual Studio 2003、Visual Studio 2005 和 Visual Studio 2008 等三个版本。其中 Visual Studio 2003 对应.NET Framework1.1，Visual Studio 2005 对应.NET Framework2.0，Visual Studio 2008 对应.NET Framework3.5。

从 C#语言方面来说，Visual Studio 2003 对应的是 C# 1.2，Visual Studio 2005 对应的是 C# 2.0，Visual Studio 2008 则是对应的 C#3.0。

由于 Visual Studio 2003 缺少很多功能，因此一般开发采用 Visual Studio 2005 或者 Visual Studio 2008。下面给出下载 Visual Studio 2005 和 Visual Studio 2008 中文试用版编辑器的地址：

Visual Studio 2005 下载地址：<http://msdn.microsoft.com/zh-cn/vstudio/aa718667.aspx>。

Visual Studio 2008 下载地址：<http://www.microsoft.com/downloads/details.aspx?FamilyID=83c3a1ec-ed72-4a79-8961-25635db0192b&DisplayLang=zh-cn>。

说明：由于本手册讲述 C#3.0，因此采用的是 Visual Studio 2008 编译器。用户可以在上面给出的地址中下载到 90 天的中文试用版。

要编写 C#程序，需要一个编译器来输入代码。我们经常使用的编译器是微软开发的 Visual Studio。写代码时如果需要查询类的使用方式，则需要安装 MSDN。下面将讲述如何安装 VS2008 编译器以及 MSDN。首先是安装 VS2008 编译器，安装步骤如下：

(1) 双击安装盘，弹出如图 1-6 所示的初始界面。在这个界面中提示用户选择需要安装的软件。

(2) 选择 Install Visual Studio 2008，然后弹出如图 1-7 所示的页面。该界面是让用户选择是否发送消息给微软。



图 1-6 安装软件选择界面

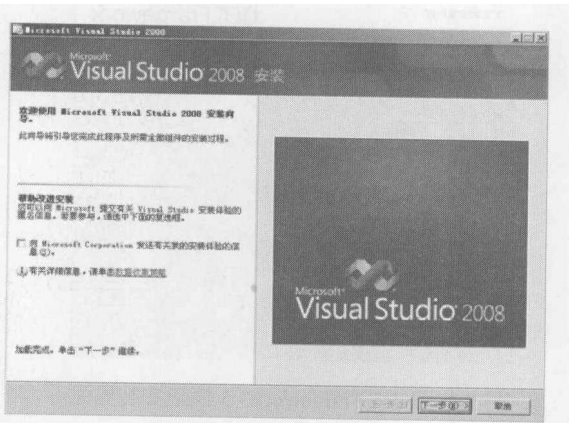


图 1-7 是否发送消息界面