C#语言程序设计基础

（第3版）

郑宇军 石海鹤 王卫红 编著

清华大学出版社

TP312C
1412-3

C#语言程序设计基础

（第3版）

郑宇军 石海鹤 王卫红 编著

清华大学出版社

内 容 简 介

作为.NET平台上的核心开发语言,C#将对象技术和泛型思想有机地融合在一起,能够从根本上提高软件的开发和运行效率。本书是关于C#语言程序设计的基础教科书,深入讲解了C#的语法和类型系统、面向对象程序设计(包括对象、接口、继承和多态性)、泛型程序设计(包括泛型类型、可空类型、泛型方法和匿名方法)以及商业应用开发的基础知识(包括文件操作、数据库访问、多线程和ASP.NET应用开发)。

本书可作为高等院校计算机及相关专业的程序设计语言教材,也可供专业软件开发人员学习参考。本书另配有实验指导书供教学选用。

图书在版编目(CIP)数据

C#语言程序设计基础/郑宇军等编著. —3版. —北京:清华大学出版社,2014
21世纪高等学校规划教材·计算机科学与技术
ISBN 978-7-302-36047-6

Ⅰ. ①C… Ⅱ. ①郑… Ⅲ. ①C语言—程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2014)第065934号

责任编辑:魏江江　王冰飞
封面设计:傅瑞学
责任校对:白　蕾
责任印制:李红英

出版发行:清华大学出版社
网　　址:http://www.tup.com.cn, http://www.wqbook.com
地　　址:北京清华大学学研大厦A座　　邮　　编:100084
社 总 机:010-62770175　　邮　　购:010-62786544
投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn
质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn
课 件 下 载:http://www.tup.com.cn,010-62795954

印 装 者:北京密云胶印厂
经　　销:全国新华书店
开　　本:185mm×260mm　　**印　　张**:22.75　　**字　　数**:551千字
版　　次:2008年3月第1版　2014年6月第3版　　**印　　次**:2014年6月第1次印刷
印　　数:1～2000
定　　价:39.50元

产品编号:054798-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程”（简称“质量工程”），通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展，顺应并符合 21 世纪教学发展的规律，代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻性。教材由个人申报或各校推荐（通过所在高校的“编委会”成员推荐），经“编委会”认真评审，最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail:weijj@tup.tsinghua.edu.cn

第3版前言

本书的前身是2005年1月出版的《C# 2.0程序设计教程》,2008年3月改版为《C#语言程序设计基础》,并在2011年10月再版,分别增加了C#语言3.0和4.0的主要特性。承蒙广大读者厚爱,本书多次重印,并被众多高校选为计算机程序设计教材。此次再版,一是新增C# 4.5和C# 5.0版本的有关功能,二是根据读者使用过程中的反馈意见进行改进完善。修订的内容主要包括以下几个方面:

(1) 将第2版中的第8章"常用类型"压缩为一节(新的6.8节)。

(2) 鉴于WPF技术的发展,删去了第2版中的第9章"Windows窗体和控件",其中部分知识点分散到了前7章。新增的第8章为"WPF应用程序基础",对新一代Windows界面的开发技术进行了详细介绍。

(3) 将第2版中的第11章"元组和可空类型"也压缩为一节(新的9.6节)。

(4) 新增了第14章"数据库访问",重点介绍了ADO.NET联机和脱机数据访问技术。

(5) 删去了第2版中的最后一章"代码组织和管理"。

总的说来,这次修订在保持原书系统、易学的特点和基本框架的基础上,侧重增加了商业应用开发的内容,以帮助读者提高解决实际问题的能力。

本书配套的实验指导书、教学课件、案例程序代码也都进行了相应的升级,有关内容也会在清华大学出版社网站上进行更新。

恳请读者对我们提出更多的批评、指正和建议。我们的E-mail地址是:bookzheng@yeah.net(邮件主题请注明"CSharp程序设计"),同时也欢迎读者在CSDN社区(http://bbs.csdn.net)的".NET技术/C#"板块上与我们进行讨论和交流。

编　者

2014年2月

于浙江工业大学

目录

第 1 章　绪论 …… 1

1.1　Microsoft.NET 技术 …… 1
1.2　.NET Framework …… 2
1.3　C#语言简介 …… 3
1.4　第一个 C#应用程序 …… 5
1.5　C#程序的基本结构 …… 6
1.5.1　类型和方法 …… 6
1.5.2　命名空间 …… 7
1.5.3　程序注释 …… 8
1.5.4　程序集 …… 8
1.6　与用户进行交互 …… 10
1.6.1　给程序传递参数 …… 10
1.6.2　控制台交互 …… 11
1.6.3　Windows 窗体和 WPF 应用程序 …… 13
本章小结 …… 15
习题 1 …… 15

第 2 章　C#数据类型 …… 16

2.1　值类型 …… 16
2.1.1　整数类型 …… 16
2.1.2　字符类型 …… 17
2.1.3　实数类型 …… 18
2.1.4　布尔类型 …… 19
2.1.5　结构 …… 19
2.1.6　枚举 …… 21
2.2　引用类型 …… 22
2.2.1　类 …… 22
2.2.2　数组 …… 25
2.2.3　字符串类型 …… 27
2.2.4　接口 …… 32
2.3　类型转换 …… 35
2.3.1　数值转换 …… 35

2.3.2 枚举转换 …… 36
2.3.3 引用转换 …… 37
2.3.4 装箱和拆箱转换 …… 39
本章小结 …… 41
习题 2 …… 41

第 3 章 值和方法 …… 42

3.1 常量和变量 …… 42
3.1.1 常量 …… 42
3.1.2 变量 …… 42
3.2 字段 …… 45
3.2.1 实例字段 …… 45
3.2.2 静态字段 …… 45
3.2.3 常数和只读字段 …… 47
3.3 方法 …… 49
3.3.1 方法的定义和调用 …… 49
3.3.2 参数类型 …… 52
3.3.3 方法的标识与重载 …… 56
3.3.4 可选参数和命名参数 …… 57
3.3.5 实例方法和静态方法 …… 59
3.4 委托与方法调用 …… 60
3.5 成员访问限制 …… 62
本章小结 …… 65
习题 3 …… 65

第 4 章 操作符和表达式 …… 66

4.1 操作符 …… 66
4.2 算术表达式 …… 67
4.2.1 基本算术运算 …… 67
4.2.2 字符串相加 …… 69
4.2.3 委托加减 …… 69
4.3 自增和自减表达式 …… 70
4.4 位运算表达式 …… 71
4.4.1 取补运算 …… 71
4.4.2 与、或、异或运算 …… 72
4.4.3 移位运算 …… 73
4.5 赋值表达式 …… 74
4.6 关系表达式 …… 74
4.6.1 比较运算 …… 74

4.6.2 类型判断 …… 76
4.7 条件逻辑表达式 …… 77
4.8 其他特殊表达式 …… 79
4.8.1 一元加减表达式 …… 79
4.8.2 条件表达式 …… 79
4.8.3 类型转换表达式 …… 80
4.8.4 创建表达式 …… 81
本章小结 …… 82
习题 4 …… 82

第 5 章 流程控制 …… 83

5.1 选择结构 …… 83
5.1.1 if 语句 …… 83
5.1.2 switch 语句 …… 85
5.2 循环结构 …… 88
5.2.1 while 循环语句 …… 88
5.2.2 do…while 循环语句 …… 89
5.2.3 for 循环语句 …… 91
5.2.4 foreach 循环语句 …… 94
5.3 跳转结构 …… 96
5.3.1 break 语句 …… 96
5.3.2 continue 语句 …… 97
5.3.3 return 语句 …… 98
5.3.4 goto 语句 …… 99
本章小结 …… 100
习题 5 …… 100

第 6 章 深入理解类 …… 102

6.1 面向对象的方法学 …… 102
6.2 构造函数和析构函数 …… 103
6.2.1 构造函数 …… 103
6.2.2 析构函数 …… 106
6.2.3 静态构造函数 …… 106
6.3 属性 …… 107
6.4 索引函数 …… 112
6.5 事件 …… 115
6.5.1 事件发布与订阅 …… 115
6.5.2 使用 EventHandler 类型 …… 119
6.5.3 Windows 控件事件 …… 120

6.6 操作符重载 …… 121
6.7 this 关键字 …… 124
6.8 几个常用类 …… 127
6.8.1 Math 类 …… 127
6.8.2 StringBuilder 类 …… 128
6.8.3 ArrayList 类 …… 129
本章小结 …… 130
习题 6 …… 131

第 7 章 继承和多态性 …… 132

7.1 继承 …… 132
7.1.1 基类和派生类 …… 132
7.1.2 隐藏基类成员 …… 135
7.1.3 base 关键字 …… 136
7.1.4 对象生命周期 …… 136
7.2 多态性 …… 138
7.2.1 成员的虚拟和重载 …… 138
7.2.2 抽象类和抽象方法 …… 142
7.2.3 密封类和密封方法 …… 145
7.3 接口与继承 …… 146
7.3.1 接口和抽象类 …… 146
7.3.2 接口方法的实现 …… 146
7.3.3 基于接口的多继承 …… 150
7.4 扩展方法 …… 154
本章小结 …… 156
习题 7 …… 156

第 8 章 WPF 应用程序基础 …… 157

8.1 XAML 基础 …… 157
8.1.1 XAML 元素和对象 …… 157
8.1.2 绑定表达式 …… 160
8.1.3 资源 …… 162
8.1.4 样式 …… 163
8.2 窗体布局和控件 …… 166
8.2.1 窗体布局 …… 166
8.2.2 控件内容模型 …… 168
8.3 图形功能 …… 174
8.3.1 颜色和画刷 …… 174
8.3.2 图形绘制 …… 176

8.3.3 几何变换 …… 178
8.4 动画 …… 179
8.4.1 基于定时器的动画 …… 179
8.4.2 基于属性的动画 …… 181
8.4.3 基于路径的动画 …… 183
本章小结 …… 185
习题 8 …… 185

第 9 章 泛型基础 …… 186

9.1 为何使用泛型 …… 186
9.2 泛型的定义 …… 188
9.2.1 泛型二叉树示例 …… 188
9.2.2 成员与类型参数 …… 190
9.2.3 泛型的静态成员 …… 192
9.3 多参数泛型 …… 193
9.3.1 使用多个类型参数 …… 193
9.3.2 类型参数与标识 …… 195
9.4 类型限制 …… 196
9.4.1 主要限制 …… 197
9.4.2 次要限制 …… 197
9.4.3 构造函数限制 …… 198
9.5 泛型与继承 …… 200
9.5.1 泛型继承规则 …… 200
9.5.2 泛型二叉树排序示例 …… 202
9.6 .NET 泛型——元组和可空类型 …… 204
9.6.1 元组 …… 204
9.6.2 可空类型 …… 206
本章小结 …… 211
习题 9 …… 211

第 10 章 深入泛型技术 …… 212

10.1 泛型方法 …… 212
10.1.1 泛型方法的定义和调用 …… 212
10.1.2 泛型方法的标识和重载 …… 214
10.1.3 泛型扩展方法 …… 215
10.2 泛型接口 …… 217
10.2.1 泛型接口的定义与实现 …… 217
10.2.2 继承中的唯一性 …… 218
10.3 泛型集合与循环遍历 …… 222

10.3.1 泛型集合 …… 222
10.3.2 可遍历类型 …… 226
10.3.3 遍历器 …… 230
10.3.4 自我遍历 …… 234
本章小结 …… 237
习题 10 …… 238

第 11 章 匿名方法和 Lambda 表达式 …… 239

11.1 命名方法和匿名方法 …… 239
11.2 使用 Lambda 表达式 …… 241
11.3 泛型委托与匿名方法 …… 244
11.3.1 使用泛型委托对象 …… 244
11.3.2 .NET 类库中的泛型委托 …… 244
11.4 外部变量 …… 246
11.5 匿名方法应用示例——计算器程序 …… 248
本章小结 …… 251
习题 11 …… 251

第 12 章 异常处理 …… 252

12.1 异常和异常处理 …… 252
12.2 C# 中的异常处理结构 …… 254
12.2.1 try…catch 结构 …… 254
12.2.2 try…catch…finally 结构 …… 255
12.2.3 try…finally 结构 …… 257
12.3 异常的传播与处理 …… 258
12.3.1 异常传播 …… 258
12.3.2 异常类型 …… 259
12.4 主动引发异常 …… 262
12.4.1 throw 语句 …… 262
12.4.2 自定义异常 …… 264
12.5 使用异常的原则和技巧 …… 268
本章小结 …… 269
习题 12 …… 269

第 13 章 文件 IO 操作 …… 270

13.1 文件系统概述 …… 270
13.2 驱动器、目录和文件 …… 271
13.2.1 与 IO 操作相关的枚举 …… 271
13.2.2 驱动器 …… 271

13.2.3 目录 …… 274
13.2.4 文件 …… 276
13.3 文件流和数据流 …… 278
13.3.1 抽象类 Stream …… 279
13.3.2 文件流 FileStream …… 280
13.3.3 流的文本读写器 …… 281
13.3.4 流的二进制读写器 …… 282
13.3.5 常用的其他流对象 …… 283
13.4 文件对话框 …… 285
13.5 对象数据存取 …… 287
本章小结 …… 290
习题 13 …… 290

第 14 章 数据库访问 …… 291

14.1 关系数据库和 SQL 概述 …… 291
14.2 ADO.NET 联机数据访问 …… 293
14.2.1 连接数据库 …… 293
14.2.2 执行数据命令 …… 294
14.2.3 使用数据阅读器 …… 295
14.2.4 对象数据存取 …… 297
14.3 ADO.NET 脱机数据访问 …… 304
14.3.1 数据表和数据集 …… 304
14.3.2 数据载入和绑定 …… 305
14.3.3 数据适配器 …… 306
本章小结 …… 306
习题 14 …… 307

第 15 章 进程和线程 …… 308

15.1 进程 …… 308
15.2 线程 …… 311
15.2.1 使用多线程 …… 311
15.2.2 线程对象及其状态 …… 313
15.2.3 线程同步 …… 314
15.2.4 访问 UI 线程 …… 318
15.2.5 异步操作 …… 321
本章小结 …… 322
习题 15 …… 322

第 16 章 ASP.NET 应用开发 …… 323

16.1 在 Visual Studio 中创建 ASP.NET 应用程序 …… 323
16.2 Web 程序基本对象 …… 326
16.2.1 HTTP 请求和响应对象 …… 326
16.2.2 HTTP 服务器对象 …… 330
16.2.3 HTTP 应用程序及状态 …… 331
16.2.4 会话、视图、缓存和 Cookies …… 332
16.3 Web 服务器控件 …… 336
16.3.1 WebControl 和 Control 类 …… 336
16.3.2 文本、文本框和按钮 …… 336
16.3.3 单选框和复选框 …… 337
16.3.4 下拉框、列表框、列表项和表格 …… 338
16.3.5 文件上传控件 …… 340
16.4 Web 应用程序示例——列车时刻表查询 …… 342
本章小结 …… 347
习题 16 …… 347

第1章 绪 论

本章首先简要介绍.NET 技术的由来和发展，.NET Framework 及其核心编程语言 C#的基本信息，然后从一个简单的例子开始带领读者走进 C#的编程世界。

1.1 Microsoft .NET 技术

Internet 技术在 20 世纪 60 年代末期才开始起步，却成为人类历史上意义最为深刻的变革之一。随着网络应用需求的飞速增长，以资源共享和协同工作为主要特征的网络分布计算已经成为新一代计算和应用的主流。随着主机计算向网络分布计算的过渡，软件系统的规模和复杂度呈几何级数增加，传统的软件开发方法面临前所未有的挑战。

为突破时间、地域及平台的限制，提高软件系统的重用和集成能力，组件化软件开发技术应运而生。它将对象技术和组件技术有机地结合在一起，强调组件的可重用性和互操作性，力图通过组件集成来构建高效的分布式软件系统。20 世纪 90 年代以来出现了 3 种典型的组件技术：OMG(对象管理组)的 CORBA、Microsoft 的 COM/DCOM 以及 Sun 公司的 JavaBeans。它们各有优劣，都面临着不断改进和发展的要求，谁也没有一统天下的实力。

2000 年 6 月，Microsoft 正式推出了面向第三代 Internet 的计算计划——Microsoft .NET，其目标之一就是希望.NET 能够取代 COM，进而成为 Windows 应用和 Web 应用的主流开发模型。这可以说是 Microsoft 继使用 Windows 取代 DOS 操作系统之后又一项战略性的举措。从技术角度理解，.NET 是一个全新的计算平台，其主要特点如下：

(1) 面向异构网络、硬件平台和操作系统，为软件提供最大限度的可扩展性、互操作性及可重用性。例如，它能够将 PC 上的软件方便地移植到手机、PDA 等终端上。

(2) 实现软件系统之间的智能交互和协同工作，提高整个网络的效率和利用率，特别是实现企业级的系统集成和资源优化，给开放型企业的生产力水平带来质的飞跃。其中的一个关键组件就是以 XML 和 SOAP 协议为基础的 Web 服务。

(3) 提供一个标准化的、安全的、一致的模型和环境，简化分布式应用程序的开发难度，从而大幅提高软件系统的质量和生产率。例如，.NET 支持在不同编程语言开发的组件之间进行无缝的交互和集成。

.NET 在理念中包含了对操作系统和计算机网络设计思想的延伸，即将 Internet 本身作为构建新一代操作系统的基础。Microsoft 的宏伟目标是让.NET 彻底改变软件的开发

方式、发行方式和使用方式，构建第三代 Internet 平台，解决各种协同合作问题，实现信息的高效沟通和分享，让整个 Internet 为人们提供最全面的服务。

Microsoft 为 .NET 技术制定了一套完整的规范，并将它提交到 ECMA 等标准化组织，以促进 .NET 的广泛应用。这套规范就是公共语言架构(Common Language Infrastructure，CLI)。它包括如下组成部分。

(1) 通用类型系统(Common Type System，CTS)：定义了一套类型系统的框架，规定了数据类型的声明、使用和管理方法。.NET 中的类型都必须遵守其中的约定。

(2) 公共语言规范(Common Language Specification，CLS)：一组语言规则的集合。如果某种编程语言符合其中的所有规则，它就是标准的 .NET 编程语言，可以实现与其他 .NET 编程语言的跨语言集成；如果某个组件中的代码使用了其中规定的功能，它就是标准的 .NET 组件，可以实现与其他 .NET 组件的交互。

(3) 通用中间语言(Common Intermediate Language，CIL)：一种中性的、与处理器无关的指令语言。任何 .NET 程序代码都可被编译成 CIL 代码，而 CIL 代码又可针对不同的系统平台翻译为不同格式的机器指令(二进制代码)。

(4) 其他相关的标准化文档、协议、规范等。

1.2 .NET Framework

公共语言架构是 .NET 的技术规范，Microsoft 在 Windows 平台上针对该规范进行了完整的实现，这就是 .NET Framework，它包括 .NET 类库和公共语言运行时(Common Language Runtime，CLR)两大部分。其整体环境架构如图 1.1 所示。

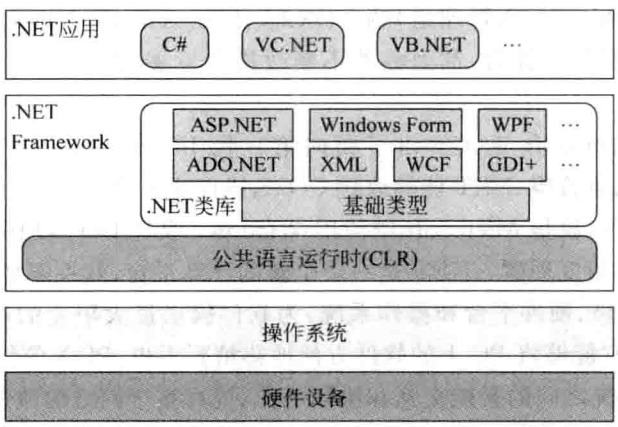

图 1.1 .NET Framework 环境结构

其中，.NET 类库是一个面向对象的类型集合，它们提供了丰富的功能以支持 Windows、Web 等各种应用的快速开发。其主要内容包括：

- 基础类型，如整数、实数、字符串等类型定义。
- 数据结构封装，如集合、链表、队列、堆栈等数据类型。
- Windows 和 Web 等界面要素，主要是按钮、标签、文本框、菜单等可视化控件。

- Web Service 要素，用于 Web 服务的定义、描述、配置、解析等。
- XML 文档处理，如 XML 文件、属性、元素、结点、读写器、解析器等类型。
- 文件输入输出，如驱动器、目录、文件、流、读写器等类型。
- 数据访问，如数据连接、数据命令、数据集、数据表、记录等类型。
- 网络通信，如主机、管道、套接字、消息等类型。
- 异常处理，用于处理系统和应用程序所引发的各种异常。
- 类型反射，用于获取程序集、对象、方法、属性、字段等目标的元数据信息类型。
- 用于应用程序管理、操作系统功能封装、安全性控制等其他方面的类型。

无论是.NET 类库中预定义的类型，还是开发人员新建的类型，它们都运行在 CLR 这个支撑平台上。从技术角度上看，CLR 就是一个虚拟机，它为.NET 应用程序提供了一个抽象于底层操作系统和硬件的运行时环境。CLR 的主要功能体现在以下 3 个方面：

(1) 管理代码的执行。各类.NET 应用程序的代码被编译为通用中间语言；在程序执行时，CLR 将通用中间语言翻译为具体的机器指令，负责加载所需的元数据类型及其他各种资源，并在执行过程中提供安全性管理、错误处理、垃圾回收等服务。

(2) 提供通用类型系统。CLR 定义了一套完整的、符合通用类型系统约定的元数据类型集合，并在此基础上对程序进行代码验证、交互控制和版本管理，以保证程序运行的可靠性。

(3) 提供系统服务。应用程序运行在 CLR 这个虚拟机上，不需要和底层操作系统以及硬件设备打交道。在需要时，由 CLR 为程序提供内存分配、资源请求、设备控制等操作系统服务。

由于 CLR 对程序执行细节进行了封装，开发人员就可以专注于程序的业务逻辑和功能流程，这就大大降低了开发难度，提高了开发效率。在 CLR 的基础上，采用不同编程语言开发的不同程序能够以一致的方式进行交互。例如，在 C# 程序中可以方便地调用 VB.NET 程序中的对象和操作。因此，CLR 是在.NET 平台上实现程序高度的互操作性、集成性、可重用性和可移植性的关键所在。

满足公共语言规范要求，基于 CLR 编译和运行的程序代码称为托管代码(Managed Code)，这种程序代码的整个运行过程都在 CLR 的管控当中，因而能够享受 CLR 所提供的各种服务功能。包括 C# 在内，目前已有超过 20 种用于编写托管代码的.NET 编程语言。

不过，.NET 也允许在程序中使用非托管代码，这些程序代码在 CLR 环境外部运行，需要自行实现垃圾回收、类型检查、安全性控制等服务机制。.NET 支持托管代码和非托管代码之间的相互调用。

1.3 C# 语言简介

自 20 世纪 80 年代以来，C/C++一直是使用最为广泛的商业化开发语言。它们提供了许多复杂的底层控制能力，但代价是相对较长的学习周期和较低的开发效率，同时也给程序的安全性带来了潜在的威胁。C++语言过度的功能扩张也破坏了面向对象的设计理念。软件行业迫切地需要一种全新的现代程序设计语言，它能够在控制能力与生产效率之间达到良好的平衡，特别是将高端应用开发与对底层平台的访问紧密结合在一起，并与 Web 标准