

高等学校应用型本科创新人才培养计划指定教材
高等学校计算机类专业“十三五”课改规划教材



WinForm

■ 程序设计及实践

济宁学院
青岛英谷教育科技有限公司 编著



西安电子科技大学出版社
<http://www.xduph.com>

高等学校应用型本科创新人才培养计划指定教材

高等学校计算机类专业“十三五”课改规划教材

WinForm 程序设计及实践

济宁学院

编著

青岛英谷教育科技有限公司

西安电子科技大学出版社

内 容 简 介

本书分为两大部分：理论篇和实践篇。理论篇从最基本的概念出发，深入地讲解了 C# 的基础知识以及新特性，具体包括 C# 概述、C# 语言基础、窗体和常用控件、界面设计、面向对象程序设计、ADO.NET 数据库访问、数据绑定和操作、文件处理、多线程应用程序以及 .NET4.0 的新特性。实践篇介绍了基于 Visual Studio 2010 环境开发 Windows 窗体应用程序的方法，具体包括窗体常用控件的使用、ADO.NET 数据库的连接及访问、数据绑定控件在界面中的数据绑定及操作等内容。

本书重点突出，偏重应用，结合实例和案例的讲解、剖析及实现，使读者能迅速理解和掌握相关知识，全面提高动手能力。

本书适应面广，可作为本科计算机科学与技术、软件工程、网络工程、计算机软件、计算机信息管理、电子商务和经济管理等专业的程序设计课程的教材。

图书在版编目(CIP)数据

WinForm 程序设计及实践/济宁学院，青岛英谷教育科技有限公司编著。

—西安：西安电子科技大学出版社，2015.8

高等学校计算机类专业“十三五”课改规划教材

ISBN 978-7-5606-3786-0

I. ① W… II. ① 济… ② 青… III. ① Windows 操作系统—高等学校—教材 IV. ① TP316.7

中国版本图书馆 CIP 数据核字(2015)第 173039 号

策 划 毛红兵

责任编辑 毛红兵 刘炳桢

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com

电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2015 年 8 月第 1 版 2015 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 21

字 数 494 千字

印 数 1~3000 册

定 价 53.00 元

ISBN 978-7-5606-3786-0/TP

XDUP 4078001-1

如有印装问题可调换

高等学校计算机类专业 “十三五”课改规划教材编委会

主编：吴海峰

编委：王 燕	王成端	薛庆文	孔繁之
李 丽	张 伟	李树金	高仲合
吴自庠	陈龙猛	张 磊	郭长友
王海峰	刘 斌	禹继国	王玉锋

◆◆◆ 前 言 ◆◆◆

本科教育是我国高等教育的基础，而应用型本科教育是高等教育由精英教育向大众化教育转变的必然产物，是社会经济发展的要求，也是今后我国高等教育规模扩张的重点。应用型创新人才培养的重点在于训练学生将所学理论知识应用于解决实际问题，这主要依靠课程的优化设计以及教学内容和方法的革新。

另外，随着我国计算机技术的迅猛发展，社会对具备计算机基本能力的人才需求急剧增加，“全面贴近企业需求，无缝打造专业实用人才”是目前高校计算机专业教育的革新方向。为了适应高等教育体制改革的新形势，积极探索适应 21 世纪人才培养的教学模式，我们组织编写了高等院校软件专业系列课改教材。

该系列教材面向高校软件专业应用型本科人才的培养，强调产学研结合，内容经过了充分的调研和论证，并参照了多所高校一线专家的意见，具有系统性、实用性等特点，旨在使读者在系统掌握软件开发知识的同时，提高其综合应用能力和解决问题的能力。

该系列教材具有如下几个特色。

1. 以培养应用型人才为目标

本系列教材以应用型软件人才为培养目标，在原有体制教育的基础上对课程进行了改革，强化“应用型”技术的学习，使读者在经过系统、完整的学习后能够掌握如下技能：

- ✧ 掌握软件开发所需的理论和技术体系以及软件开发过程规范体系；
- ✧ 能够熟练地进行设计和编码工作，并具备良好的自学能力；
- ✧ 具备一定的项目经验，能够进行代码的调试、文档编写、软件测试等；
- ✧ 达到软件企业的用人标准，做到学校学习与企业需求能力的无缝对接。

2. 以新颖的教材架构来引导学习

本系列教材采用的教材架构打破了传统的以知识为标准编写教材的方法，采用理论篇与实践篇相结合的组织模式，引导读者在学习理论知识的同时，加强实践动手能力的训练。

- ✧ 理论篇：学习内容的选取遵循“二八原则”，即重点内容由企业中常用的 20% 的技术组成。每章设有本章目标，以明确本章学习重点和难点，章节内容结合示例代码，引导学生循序渐进地理解和掌握这些知识和技能，培养学生的逻辑思维能力，掌握软件开发的必备知识和技巧。
- ✧ 实践篇：多点集于一线，以任务驱动，以完整的具体案例贯穿始终，力求使学生在动手实践的过程中，加深对课程内容的理解，培养学生独立分析和解决问题的能力，并配备相关的知识拓展和拓展练习，拓宽学生的知识面。

另外，本系列教材借鉴了软件开发中“低耦合，高内聚”的设计理念，组织结构上遵循软件开发中的 MVC 理念，即在保证最小教学集的前提下可以根据自身的实际情况对整

个课程体系进行横向或纵向裁剪。

3. 提供全面的教辅产品来辅助教学实施

为充分体现“实境耦合”的教学模式，方便教学实施，该系列教材配备了可配套使用的项目实训教材和全套教辅产品。

- ✧ 实训教材：集多线于一面，以辅助教材的形式，提供适应当前课程(及先行课程)的综合项目，按照软件开发过程进行讲解、分析、设计、指导，注重工作过程的系统性，培养学生解决实际问题的能力，是实施“实境”教学的关键环节。
- ✧ 立体配套：为适应教学模式和教学方法的改革，本系列教材提供完备的教辅产品，主要包括教学指导、实验指导、电子课件、习题集、实践案例等内容，并配以相应的网络教学资源。教学实施方面，提供全方位的解决方案(课程体系解决方案、实训解决方案、教师培训解决方案和就业指导解决方案等)，以适应软件开发教学过程的特殊性。

本书由济宁学院、青岛英谷教育科技股份有限公司编写，参与本书编写的有吴海峰、王燕、宁维巍、朱仁成、宋国强、何莉娟、杨敬熹、田波、侯方超、刘江林、方惠、莫太民、邵作伟、王千等。本书在编写期间得到了各合作院校专家及一线教师的大力支持与协作，在此衷心感谢每一位老师与同事为本书出版所付出的努力。

由于水平有限，书中难免有不足之处，欢迎大家批评指正！读者在阅读过程中发现问题，可以通过邮箱(yujin@tech-yj.com)发给我们，以帮助我们进一步完善。

本书编委会

2015年3月

目 录

理 论 篇

第 1 章 C# 概述	3
1.1 .NET 框架	4
1.1.1 .NET 框架结构	4
1.1.2 .NET 框架的优点	5
1.1.3 .NET 的术语	6
1.1.4 C# 与 .NET 框架	6
1.1.5 .NET 框架应用程序种类	7
1.2 第一个 C# 程序	8
本章小结	9
本章练习	9
第 2 章 C# 语言基础	11
2.1 数据类型	12
2.2 变量和常量	12
2.2.1 变量	12
2.2.2 常量	13
2.3 运算符	13
2.3.1 算术运算符	13
2.3.2 比较运算符	14
2.3.3 逻辑运算符	15
2.4 流程控制语句	16
2.4.1 分支语句	16
2.4.2 循环语句	20
2.5 数组	22
2.5.1 声明数组	22
2.5.2 数组初始化	22
2.5.3 访问数组元素	22
2.5.4 二维数组	23
本章小结	24
本章练习	24
第 3 章 窗体和常用控件	25
3.1 控件概述	26

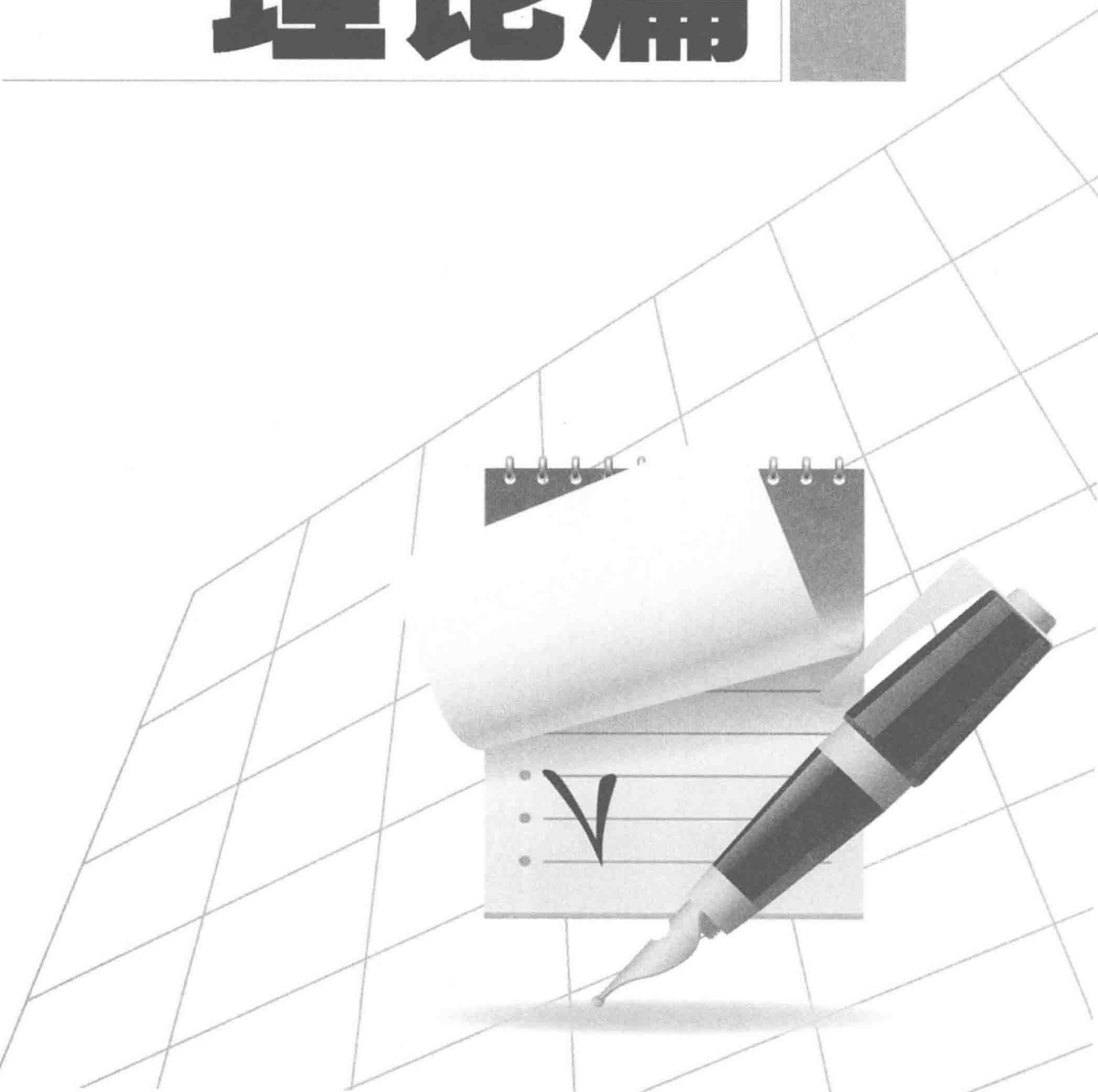
3.1.1 控件的基本属性	27
3.1.2 控件的基本事件	29
3.2 窗体	30
3.3 常用控件	35
3.3.1 按钮(Button)控件	35
3.3.2 标签(Label)控件	36
3.3.3 文本控件	36
3.3.4 选择控件	39
3.3.5 图片框(PictureBox)控件	48
3.3.6 容器控件	51
本章小结	53
本章练习	54
第 4 章 界面设计	55
4.1 界面设计概述	56
4.2 菜单	57
4.2.1 主菜单	57
4.2.2 上下文菜单	60
4.3 工具栏	62
4.4 状态栏	64
4.5 对话框	65
4.6 MDI 界面设计	68
本章小结	69
本章练习	70
第 5 章 面向对象程序设计	71
5.1 C# 中的面向对象	72
5.2 类和对象	73
5.2.1 类	73
5.2.2 对象	77
5.3 继承	80
5.4 多态	81
5.4.1 重载	81

5.4.2 重写	82	8.2.3 Path	139
5.5 this 和 base 关键字	83	8.3 文件流操作	141
5.5.1 this 关键字	83	8.3.1 FileStream	141
5.5.2 base 关键字	84	8.3.2 StreamReader 类和 StreamWriter 类	142
本章小结	85	8.3.3 BinaryReader 类和 BinaryWriter 类	146
本章练习	85	本章小结	148
第 6 章 ADO.NET 数据库访问	87	本章练习	148
6.1 ADO.NET 简介	88	第 9 章 多线程应用程序	149
6.2 ADO.NET 结构	89	9.1 线程概述	150
6.2.1 ADO.NET 中的命名空间和类	89	9.1.1 进程、线程和应用程序域	150
6.2.2 ADO.NET 结构原理	90	9.1.2 线程限制	151
6.3 SQL Server 2008	91	9.1.3 C# 对多线程的支持	151
6.4 ADO.NET 的核心对象	94	9.2 C# 中多线程的实现	151
6.4.1 Connection	95	9.2.1 线程的创建	151
6.4.2 Command	97	9.2.2 线程的状态	154
6.4.3 DataReader	98	9.2.3 线程的优先级	155
6.4.4 DataAdapter 和 DataSet	101	9.2.4 线程池	155
本章小结	107	9.2.5 线程组件	156
本章练习	107	本章小结	159
第 7 章 数据绑定和操作	109	本章练习	159
7.1 数据控件	110	第 10 章 .NET4.0 的新特性	161
7.1.1 DataGridView	110	10.1 推断类型	162
7.1.2 配置 DataGridView 控件	112	10.2 扩展方法	163
7.2 数据操作	116	10.3 对象初始化器	166
7.2.1 数据查询过滤	118	10.4 匿名类	167
7.2.2 添加数据	119	10.5 Lambda 表达式	168
7.2.3 修改数据	123	10.6 LINQ 查询	169
7.2.4 删除数据	124	10.6.1 LINQ 简介	169
本章小结	128	10.6.2 LINQ 查询步骤	170
本章练习	128	10.6.3 LINQ 查询关键字	171
第 8 章 文件处理	129	10.7 dynamic 新关键词	174
8.1 文件概述	130	10.8 可选或默认参数	174
8.1.1 文件类型	130	10.9 命名参数	175
8.1.2 文件访问方式	130	本章小结	176
8.2 System.IO 模型	131	本章练习	176
8.2.1 Directory	131		
8.2.2 File	136		

实 践 篇

实践 1 C# 概述.....	179	实践 5.3	233
实践指导	179	实践 5.4	236
实践 1.1	179	知识拓展	237
实践 1.2	182	拓展练习	248
实践 1.3	184	实践 6 ADO.NET 数据库访问	249
知识拓展	186	实践指导	249
拓展练习	190	实践 6.1	249
实践 2 C# 语言基础	191	实践 6.2	256
实践指导	191	实践 6.3	258
实践 2.1	191	实践 6.4	263
实践 2.2	192	知识拓展	266
知识拓展	193	拓展练习	275
拓展练习	198	实践 7 数据绑定和操作	276
实践 3 窗体和常用控件	199	实践指导	276
实践指导	199	实践 7.1	276
实践 3.1	200	实践 7.2	279
实践 3.2	202	实践 7.3	295
实践 3.3	204	实践 7.4	301
知识拓展	206	知识拓展	307
拓展练习	210	拓展练习	309
实践 4 界面设计	211	实践 8 文件处理	310
实践指导	211	实践指导	310
实践 4.1	211	知识拓展	311
实践 4.2	215	拓展练习	313
实践 4.3	217	实践 9 .NET4.0 的新特性	314
知识拓展	219	实践指导	314
拓展练习	226	实践 9.1	314
实践 5 面向对象程序设计	227	实践 9.2	319
实践指导	227	知识拓展	322
实践 5.1	227	拓展练习	325
实践 5.2	230		

理论篇





第1章 C# 概述

本章目标

- 了解 .NET 框架的结构
- 理解 CLR、IL、CLS、JIT 概念
- 了解 C# 语言的特点以及与 .NET 框架的联系
- 掌握 C# 程序的结构



1.1 .NET 框架

2000 年, 微软公司推出的“Microsoft .NET 下一代互联网软件和服务战略”, 引起了 IT 行业的广泛关注。2002 年, 微软公司发布 Visual Studio .NET(内部版本号为 7.0), 在这个版本中, 引入了建立在 .NET 框架上的托管代码机制和一门新的编程语言 C#, 之后微软公司先后发布了 Visual Studio .NET 2003、Visual Studio 2005、Visual Studio 2008、Visual Studio 2010、Visual Studio 2012、Visual Studio 2013 和 Visual Studio 2015, 对应的 .NET 框架经历了 .NET Framework 1.0、.NET Framework 2.0、.NET Framework 3.5、.NET Framework 4.0 和 .NET Framework 4.5 版本。

目前 .NET 平台已经非常成熟, 除了支持传统的 Windows 应用程序和 Web 应用程序开发, 还可以支持跨平台移动开发和云开发等。Visual Studio 是非常流行的 Windows 平台应用程序集成开发环境, 其发展历史如表 1-1 所示。

表 1-1 Visual Studio 的发展历史

名 称	内部版本	发布日期	支持 .NET Framework 版本
Visual Studio .NET 2002	7.0	2002-02-13	1.0
Visual Studio .NET 2003	7.1	2003-04-24	1.1
Visual Studio 2005	8.0	2005-11-07	2.0
Visual Studio 2008	9.0	2007-11-19	2.0、3.0、3.5
Visual Studio 2010	10.0	2010-04-12	2.0、3.0、3.5、4.0
Visual Studio 2012	11.0	2012-08-25	2.0、3.0、3.5、4.0、4.5
Visual Studio 2013	12.0	2013-10-17	2.0、3.0、3.5、4.0、4.5、 4.5.1、4.5.2
Visual Studio 2015	14.0	2014-11-10	2.0、3.0、3.5、4.0、4.5、 4.5.1、4.5.3、4.5.5

1.1.1 .NET 框架结构

微软 .NET 平台的核心是一系列新技术的集合, 统称为 .NET Framework, 它是一个可以作为平台支持下一代 Internet 的可编程集合环境。.NET 框架集提供了一个可快速开发各种应用程序的平台, 其目的就是让用户在任何地方、任何时间, 以及利用任何设备都能访问他们所需要的信息、文件和程序。例如使用 .NET 框架可以开发 Web 应用程序、Windows 窗体应用程序以及类库等。

.NET 框架体系结构由以下四个主要部分组成:

- ✧ 公共语言运行时(Common Language Runtime, CLR)。
- ✧ 基础类库(Base Class Library)。
- ✧ ADO.NET 数据库访问。



◇ 活动服务器页面(C#)。

其层次结构如图 1-1 所示。

1. 公共语言运行时(CLR)

公共语言运行时是 .NET 框架应用程序的执行引擎。在程序运行过程中, CLR 负责管理内存分配、启动或删除线程和进程、实施安全性策略,同时满足当前组件对其他组件的需求。在开发阶段,公共语言运行时实现了高度的自动化,使开发人员的工作变得非常轻松,尤其是映射功能显著减少了开发人员将业务逻辑转化成可复用组件的代码编写量。

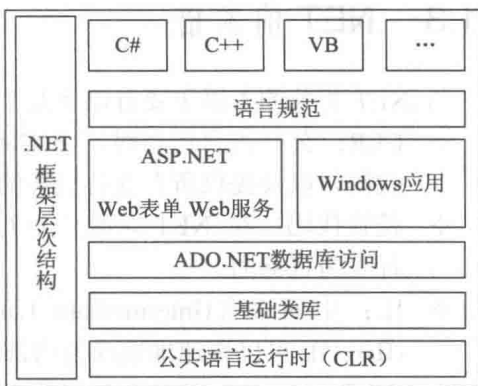


图 1-1 .NET 框架体系结构

2. 基础类库

.NET 框架为开发人员提供了一个统一的、面向对象的、层次化的、可扩展的类库集(API)。在 .NET 平台支持的各种语言中都可以使用这个类库。

3. ADO.NET 数据库访问

ADO.NET 起源于 ADO(ActiveX Data Objects),它是一组用于和数据源交互的面向对象的类库。ADO.NET 实现了 ADO 无法满足的三个重要需求:① 提供了断开的数据库访问模型,这对 Web 环境至关重要;② 提供了与 XML 的紧密集成;③ 提供了与 .NET 框架的无缝集成。

4. 活动服务器页面(C#)

C# 提供了 Web 应用程序模型,该模型由一组控件和一个基本结构组成。C# 使 Web 应用程序的构建变得非常容易,开发人员可以直接使用 C# 控件集;C# 还提供了一些基本结构服务(诸如会话状态管理和进程重启服务),这些服务大大减少了开发人员要编写的代码量,并使应用程序的可靠性得到大幅度提高。

1.1.2 .NET 框架的优点

.NET Framework 具有以下几个优点:

- ◇ 统一的程序设计模式: .NET Framework 提供了跨语言的面向对象的统一程序设计模式,这种模式可用于读写文件、数据库访问等,例如 ADO.NET。
- ◇ 跨平台应用: .NET 应用程序可以运行在任意被 CLR 所支持的系统中。
- ◇ 多语言集成: .NET 允许多种语言进行集成,例如可以在 C# 中使用一个用 VB 实现的类,即对象之间能够进行相互作用而不考虑开发这些对象的语言。
- ◇ 自动资源管理: CLR 会对应用程序所使用的资源进行自动检测,释放不使用的资源,无须程序员干预,减轻了程序员的负担。
- ◇ 轻松部署: .NET Framework 提供了安装部署项目,可以对应用程序进行部署,轻松形成安装文件,便于在目标计算机中进行安装部署。



1.1.3 .NET 的术语

与 .NET 相关的术语主要有以下几个：

- ✧ **CLR**：公共语言运行时，它实际管理代码，可以处理加载程序、运行程序的代码，以及提供所有支持服务的代码。
- ✧ **托管代码**：在 .NET 环境中运行的任何代码都称为托管代码，它们都是以运行库为目标的。
- ✧ **IL**：中间语言(Intermediate Language)，编译器将源代码编译成中间语言(IL)，IL 可以快速地编译为内部机器代码。
- ✧ **.NET 基类**：这是一个扩展的类库，它包含预先写好的代码，执行 Windows 上的各种任务，例如显示窗口和窗体、访问 Windows 基本服务、读写文件、访问网络和访问数据源。
- ✧ **CLS**：公共语言规范(Common Language Specification)，这是确保代码可以在任何语言中访问的最小标准集合，所有用于 .NET 的编译器都应支持 CLS。CLS 构成了可以在 .NET 和 IL 中使用的功能子集，代码也可以使用 CLS 外部的功能。
- ✧ **JIT**：Just-In-Time 编译，此术语用于表示执行编译过程的最后阶段，即从中间语言转换为内部机器代码。其名称的来源是部分代码按需要即时编译的。

应用程序在 .Net Framework 中的执行顺序如图 1-2 所示。

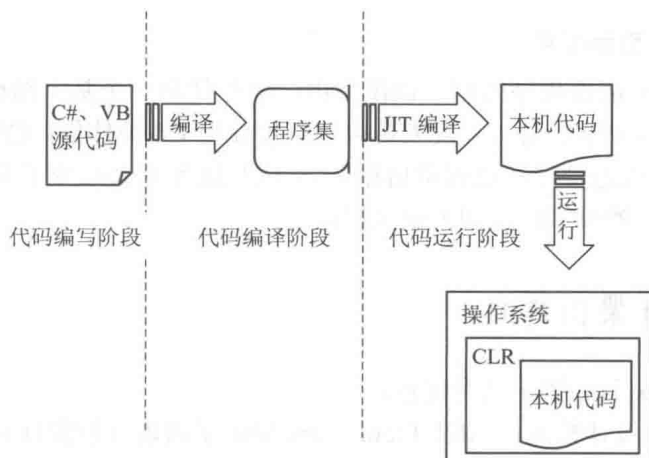


图 1-2 应用程序在框架中的执行顺序

1.1.4 C# 与 .NET 框架

.NET 框架支持多种开发语言，在生成中间语言之前，可以使用框架支持的任何一种语言进行开发，生成中间语言之后，各个语言可以对中间语言进行调用，从而实现框架内的代码重用。

.NET 框架支持 C#、Visual Basic、C++、J# (已从 Visual Studio 2008 中去除)、F# 几种



语言。

图 1-3 显示了 C# 在 .NET 框架中的具体位置。

C# 是一种为了迎合 .NET 创建分布式应用程序的目标而产生的程序设计语言。C# 中所引进的一些关键特征如下：

- ✧ 继承。
- ✧ 构造函数和析构函数。
- ✧ 重载。
- ✧ 覆盖。
- ✧ 结构化异常处理。
- ✧ 多线程。

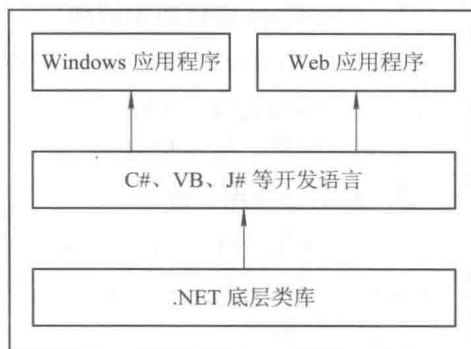


图 1-3 C# 在 .NET 框架中的具体位置



注意

继承不仅仅是 C# 的特征，更是 .NET Framework 的特征。在 .NET Framework 中可以使用任意一种语言创建一个基类，而在另一种语言中创建继承该基类的子类。这为多语言代码之间的重用提供了可能性。

1.1.5 .NET 框架应用程序种类

.NET 框架是一个综合的开发框架，开发人员可以利用它创建不同类型的应用程序。

运行 Visual Studio 2010(以下简称 VS2010)后，单击“文件”→“新建”→“项目”，将打开如图 1-4 所示的“新建项目”窗口。在窗口左侧“已安装模板”中可以看到，VS2010 提供了 Windows、Web 等各种类型应用程序的模板。

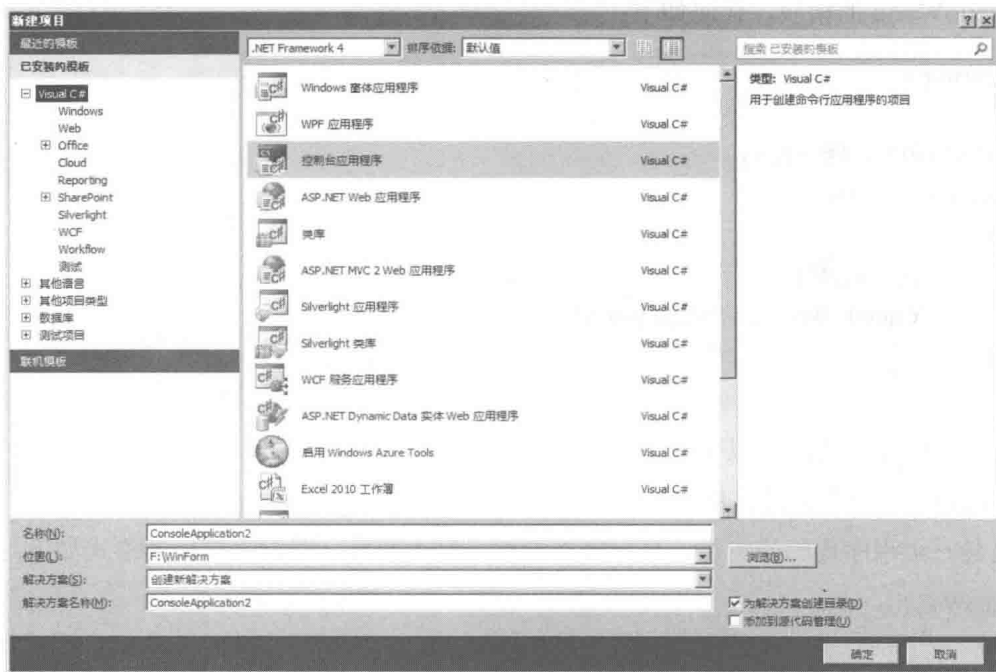


图 1-4 新建项目



1. Windows 窗体应用程序

Windows 窗体应用程序是指运行在 Windows 操作系统上的窗口式应用程序。.NET 框架封装了 Win32 API, 提供一个高效的、面向对象的、可扩展的类库, 使 Windows 应用程序开发更加简单, 效率更高。

2. 控制台应用程序

控制台应用程序是指通过命令行运行的控制台应用, 此种应用程序通过 DOS 环境下的命令行与用户进行交互。在开发极少或根本不需要用户交互的实用工具程序时, 可以使用控制台应用程序。

3. 类库

类库模板用于创建可重用的组件, 类库所形成的 .dll 文件可以应用于多个项目, 在不同项目中只需引入 .dll 就可以使用此类库所提供的功能。

**注意**

本书在讲解 C# 的基础语法时使用控制台应用程序, 讲解窗体界面时则使用 Windows 窗体应用程序。

1.2 第一个 C# 程序

【示例 1.1】 使用 C# 编写 HelloWorld 控制台应用程序。

运行 VS2010, 创建一个项目名称为 ch01 的控制台应用程序, 并在该项目中创建一个名为 HelloWorld 的模块, 代码如下:

```
class HelloWorld
{
    //Main 函数, 程序的入口
    static void Main()
    {
        //控制台输出
        Console.WriteLine("Hello World! ");
    }
}
```

上述代码中需要注意以下几点内容:

(1) class 是定义类的关键字。

类是一组程序代码的组合, 其出现必须有一对大括号, 即定义模块的格式如下:

```
class HelloWorld
{
    //方法体
}
```