

高等学校计算机软件技术基础课程系列教材

Software Development
Technology and Its Applications

计算机软件开发技术与应用

丛培盛 龚沛曾 主编

高 枚 王睿智 编



高等教育出版社
HIGHER EDUCATION PRESS

内容提要

本书是在原“计算机软件技术基础”课程教学内容的基础上，为适应社会发展和软件开发教学的需要，经过三年实际教学实践，逐步调整、编排而成的。全书分为基础篇、软件工程与设计篇、开发实例与实验篇3个部分，主要内容包括C# NET 程序设计基础、数据结构、软件工程、数据库技术、软件开发实例与实验等。

本书层次清晰、由浅到深、环环相扣，在内容的选择和编排上，既考虑了对当今软件业中应用比较广泛的概念如UML建模、多层构架的介绍，又考虑了各部分之间的相互贯通及学生的基础和接受能力。

本书可作为高等学校非计算机类专业本科生的“计算机软件开发技术与应用”课程的教材，也可作为软件应用工程师的入门参考书。

图书在版编目(CIP)数据

计算机软件开发技术与应用/丛培盛，龚沛曾主编；高枚，王睿智编. —北京：高等教育出版社，2012.2

ISBN 978-7-04-034163-8

I. ①计… II. ①丛… ②龚… ③高… ④王… III. ①软件开发-高等学校-教材 IV. ①TP311.52

中国版本图书馆CIP数据核字(2011)第279872号

策划编辑	饶卉萍	责任编辑	饶卉萍	封面设计	于文燕	版式设计	杜微言
插图绘制	尹莉	责任校对	刘莉	责任印制	张福涛		

出版发行	高等教育出版社	咨询电话	400-810-0598
社 址	北京市西城区德外大街4号	网 址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印 刷	北京市鑫霸印务有限公司	网上订购	http://www.landraco.com
开 本	787mm × 1092mm 1/16		http://www.landraco.com.cn
印 张	21 25	版 次	2012年2月第1版
字 数	470千字	印 次	2012年2月第1次印刷
购书热线	010-58581118	定 价	29.00元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 34163-00

前言

“计算机软件开发技术与应用”课程由“计算机软件技术基础”课程逐步演化而来,是“C/C++程序设计语言”课程的后续课程,学习对象为非计算机专业的学生。已有的“计算机软件技术基础”课程的相关教材在内容的设置上雷同,基本以 Visual C++6.0 为教学开发环境并由传统的 5 大部分如软件工程、数据结构、数据库、操作系统等组成,现代化的软件设计概念却比较单薄,不够实用。

Visual C++6.0 作为教学语言是合适的,但随着新一代开发环境如 Visual Studio. NET、IBM WSAD 的兴起,Visual C++6.0 已不是当前应用软件的主流开发环境。因此,迫切需要一本能把握当前主流技术并强调实用的教材以满足目前的教学需要。

为此,在了解目前软件建设中企业采用的主流技术,包括分析设计方法、软件构架、规范、开发环境等后,在参考其他同类教材并在强调实用性的基础上,同济大学“计算机软件开发技术与应用”课程教学团队重新编排了课程内容,并将课程命名为“计算机软件开发技术与应用”。希望通过对本课程的学习,使学习者能把握较前沿的技术并在此基础上掌握软件设计与实现的技巧。

通过新的安排,课程内容编排如下。

(1) C#. NET 程序设计基础:这部分内容着重介绍面向对象的基本知识、C# 的基本语法、Windows 简单程序的编写及控件的使用。

(2) 数据结构:用 C# 实现常见的数据结构,并增加对 C#. NET 中一些实用类知识的介绍。

(3) 软件工程:采用面向对象的分析设计方法,讲解软件设计的方法学。这部分内容的编排,尽量避免过多地讲解 UML 的相关概念和理论,仅抽取其扼要的用例及用例中的类分析、类划分等内容进行讲解,辅以具体的小例子,并在后续的实例中采用相应的方法进行分析设计,使读者容易理解。

(4) 数据库:主要讲解关系模式设计,与软件工程用例的实体类分析形成对照,为后续的开发实例做准备。

(5) 开发实例与实验:把面向对象分析设计中的一些概念进行具体的应用,将软件逻辑多层



构架技术融入到实例中,结合数据库设计、数据结构知识,形成知识的综合应用实验,提高学生软件制作的水平。

本书共5章,第1章由龚沛曾编写,第2章由高枚、丛培盛编写,第3章由丛培盛编写,第4章由王睿智、丛培盛编写,第5章、第6章由丛培盛、龚沛曾编写。在教材编写过程中,得到了同济大学各位领导和同事的悉心指导与帮助,在此表示衷心的感谢。

虽然本书的编写花费了很长的时间,且在教学试用过程中进行了反复的整理和修改,但仍难免有错,恳请广大读者批评指正。

编 者

2011年9月

目录

第1部分 基础篇

第1章 C#.NET 程序设计基础	3	1.5.3 异常处理	48
1.1 C#.NET 概述	4	思考题	50
1.1.1 C#和.NET 简介	4	第2章 数据结构	51
1.1.2 Visual Studio .NET 集成开发 环境	6	2.1 数据结构概述	52
1.1.3 C#Windows 应用程序开发 过程	10	2.1.1 数据结构的概念	52
1.2 C#可视化界面	12	2.1.2 数据的逻辑结构	53
1.2.1 控件基本概念	12	2.1.3 数据的物理结构	53
1.2.2 窗体	15	2.1.4 数据结构的运算	55
1.2.3 常用控件	16	2.2 线性表	56
1.3 C#基本语法	20	2.2.1 线性表基本概念	56
1.3.1 C#数据类型、表达式	20	2.2.2 顺序表	57
1.3.2 控制结构	23	2.2.3 链表	67
1.3.3 数组	26	2.3 堆栈和队列	79
1.4 C#面向对象程序设计	27	2.3.1 堆栈	80
1.4.1 面向对象编程基本概念	27	2.3.2 队列	85
1.4.2 类定义	28	2.4 树型结构	91
1.4.3 继承	36	2.4.1 树的定义和基本概念	92
1.4.4 文本文件读写	40	2.4.2 二叉树	93
1.5 程序调试	44	2.5 图	102
1.5.1 错误类型	44	2.5.1 图的定义和基本概念	102
1.5.2 程序的跟踪调试	45	2.5.2 图的存储结构	102
		2.5.3 图的遍历	103
		2.6 查找	108



2.6.1 顺序查找	108	2.7.3 归并排序	121
2.6.2 对半查找法	109	2.8 .NET 中 C#实用类	124
2.6.3 二叉排序树及其查找	110	2.8.1 C#常见数据结构	124
2.7 排序	113	2.8.2 实用类 ArrayList 及 List	125
2.7.1 选择排序	114	2.8.3 哈希表 Hashtable	127
2.7.2 交换排序	117	思考题	129

第 2 部分 软件工程与设计篇

第 3 章 软件工程	133	思考题	165
3.1 概述	134	第 4 章 数据库技术	166
3.1.1 软件工程的形成与发展	134	4.1 概念数据建模	167
3.1.2 软件工程定义	134	4.1.1 概念数据建模过程	167
3.2 软件项目管理概述	136	4.1.2 UML 数据建模	167
3.2.1 软件项目管理的内容	136	4.1.3 E-R 数据建模	171
3.2.2 软件项目管理过程	137	4.2 关系数据模型	172
3.2.3 影响软件项目成功的因素	138	4.2.1 基本概念	173
3.3 软件工程范型	139	4.2.2 关系的规范化	175
3.3.1 瀑布模型	140	4.2.3 概念数据模型到关系模型的 转化	178
3.3.2 快速原型模型	141	4.3 物理数据库设计	180
3.3.3 螺旋模型	142	4.3.1 数据类型	180
3.3.4 快速应用开发模型	143	4.3.2 数据的完整性	183
3.4 系统分析	144	4.3.3 管理索引	186
3.4.1 需求的确定	144	4.3.4 数据库实施	186
3.4.2 需求的组织	144	4.4 结构化查询语言	195
3.4.3 分析类	146	4.4.1 结构化查询语言基础	196
3.5 系统设计	149	4.4.2 SQL 数据检索语句	197
3.5.1 类及构件设计	150	4.4.3 SQL 数据更新语句	203
3.5.2 体系结构设计	152	4.4.4 SQL 的定义语句	205
3.5.3 人机界面设计	155	4.5 数据库访问	206
3.6 详细设计	155	4.5.1 ADO.NET 核心组件	207
3.6.1 详细设计的任务	156	4.5.2 数据库联接与管理	210
3.6.2 详细设计的描述工具	156	4.5.3 数据库访问	211
3.7 编码	162	思考题	220
3.7.1 命名规范	162		
3.7.2 编码风格	163		



第3部分 开发实例与实验篇

第5章 软件开发实例	225	5.3.6 学生选课	273
5.1 系统的需求分析	226	5.4 程序最终部署	281
5.1.1 系统的需求简述	226	思考题	282
5.1.2 系统的用例图	226	第6章 实验	283
5.1.3 数据分析	228	实验1 窗体设计	283
5.1.4 关系数据库设计	232	实验2 面向对象的程序设计及 调试	285
5.2 系统设计	235	实验3 顺序表及链表	290
5.2.1 模块的划分及主窗体	235	实验4 堆栈的操作	296
5.2.2 项目目录管理	236	实验5 队列	297
5.2.3 命名规则	237	实验6 二叉树	299
5.2.4 软件的层次构架	238	实验7 数据库操作	300
5.2.5 数据准备	243	实验8 SQL语句操作	307
5.3 编码实现	246	实验9 数据库联接测试	312
5.3.1 学生输入	246	实验10 读XML文件	317
5.3.2 学生查询	254	实验11 代码复用	320
5.3.3 课程查询	260	实验12 登录及身份认证	325
5.3.4 用户登录及身份认证	264		
5.3.5 开课查询	270		
参考文献	329		

第1部分

基础篇

C#. NET 程序设计基础

第 1 章

自从计算机高级语言成为大学教学的必备内容后,我们已经经历了 Basic、FORTRAN、Cobol、Pascal、C/C++、Java、C#等多种语言。20 世纪 80 年代开始是软件开发工具大发展的时期,如面向桌面数据库的 FoxPro 曾流行一时,随着数据库服务器的完善,出现了大量的可视化的、基于各种语言的软件开发工具,如 Visual Basic(VB)、PowerBuilder、Delphi、C++Builder、JBuilder 等,由于开发效率大大提高,这些工具获得了快速应用开发(Rapid Application Development, RAD)的美誉。20 世纪后期,面向对象的分析设计工具逐步在软件开发中取得了统治地位,更加促使了面向对象软件开发工具的发展。到目前为止,Visual Studio. NET 平台是熟练使用 VB、C++、C#等语言开发者最普遍使用的高效开发平台,Eclipse、IntelliJ IDEA 则是 Java 语言开发者经常使用的开发环境。当然,目前非常流行的智能手机操作系统如 Android、Windows phone 7 等开发平台及应用软件,也是发展的一个重要方向。

本书中后续的数据结构和应用开发章节使用的开发环境是 .NET 平台,采用的语言是 C#,本章简要介绍 C#. NET 环境、常用控件和基本语法,为后续内容打好基础。另外,由于 C#是一种完全面向对象的程序设计语言,即使是编写一个最简单的程序,也需要从一个类开始,所以本章中也简单介绍了类、对象的基本概念。已经熟悉 C#编程的读者,可以跳过本章。



1.1 C#.NET 概述

1.1.1 C#和.NET 简介

1. C#语言简介

C#语言是 Microsoft 公司推出 .NET 平台时开发的一种面向对象的新语言,由 Turbo Pascal、Delphi、Visual J++ 的首席设计师 Anders Hejlsberg 用 3 年时间设计而成,语法结构特征与 C++、Java 非常类似,所有的语言元素都是真正的对象,同时结合了 VB 的可视化编程特征,可以说该语言集 C++ 的计算效率、Java 的安全性和 VB 可视化为一身。

C#源于 C/C++,但消除了 C/C++ 中一些复杂的特性,如多重继承、指针,使语法更加简洁,是 .NET 平台上使用最普遍的开发工具之一,广泛应用于 Internet Web 及 Windows 应用程序的开发上。

2. .NET 开发平台

(1) .NET 开发平台的组成

.NET 开发平台主要由 .NET 框架(.NET Framework)、.NET 开发技术及开发工具等组成,如图 1.1.1 所示。.NET 框架是开发平台的基础,包括公共语言运行时库(Common Language Runtime,CLR)和基础类库;.NET 开发技术则包含数据库访问组件 ADO.NET 和 XML,使得操作数据非常方便,也使在 Internet 上交换数据变得简单易行;.NET 开发工具中,Microsoft 公司的 Visual Studio.NET 是使用最广泛的,它集成了 VB、Visual C++、Visual C#和 Visual J#语言。

(2) .NET 开发平台的特点

① 支持多语言开发。程序员可以使用自己熟悉的语言进行开发,也可以在一个应用程序中使用多种语言,不同语言编写的模块也容易实现整合。

② 开发多种应用程序。在 Visual Studio .NET 的支持下,程序员可以使用任何一种语言开发多种应用程序,主要包括 Windows 应用程序和 Web 应用程序(ASP.NET)。

③ 各种语言工具使用同一个类库。传统的开发环境中,不同的语言使用不同的函数库且调用方式各异,它们是不能通用的。而 .NET 开发环境中,不管使用何种语言,都使用同一个基础类库。

④ 公共语言运行时库。公共语言运行时库提供了执行程序服务,与 CPU 特性无关。.NET 程序需要经过两次编译才能在 CPU 上运行,第一次是被编译成与 CPU 无关的中间语言(MSIL),在 CLR 的支

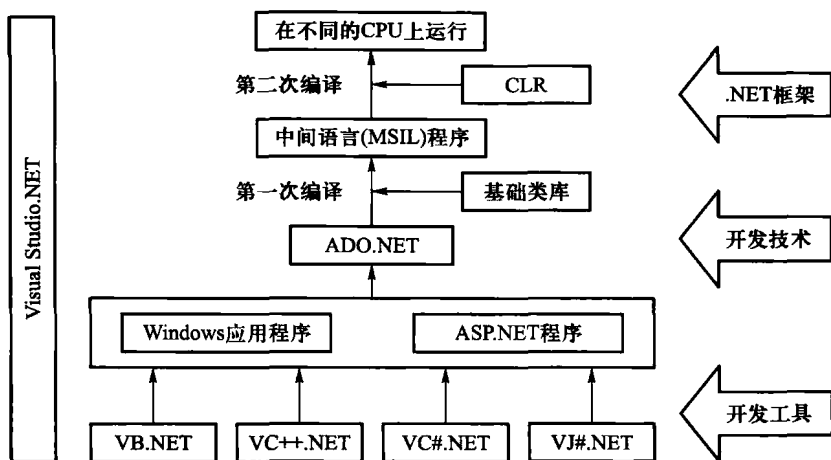


图 1.1.1 .NET 开发平台结构

持下,中间语言被编译成由本地 CPU 指令集组成的程序,实现了.NET 跨硬件平台的目标。

如图 1.1.2 所示,Visual Studio.NET 依赖于.NET 框架提供的服务,包括 Microsoft 公司第三方提供的编译器等。

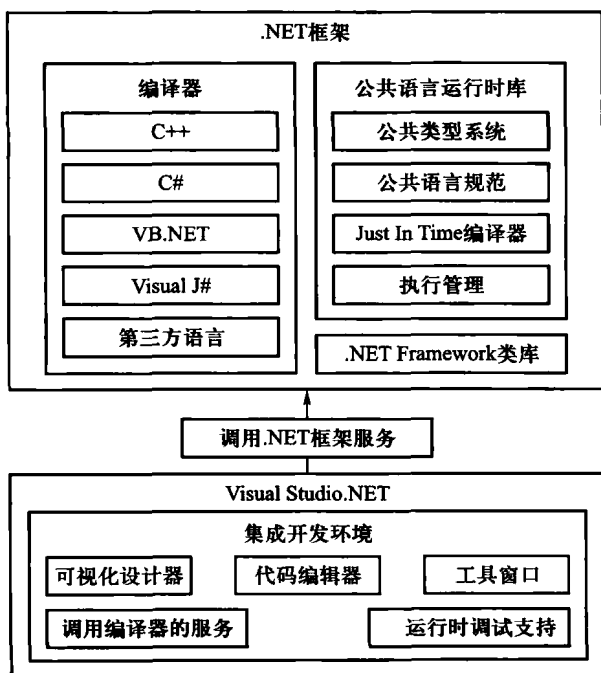


图 1.1.2 Visual Studio.NET 和.NET 框架关系



(3) Visual Studio .NET 和 .NET 框架

.NET 应用程序在运行时都必须有 .NET 框架的支持。实际上, Visual Studio .NET 生成的可执行应用程序文件(.exe 类型的文件)都是由 MSIL 码组成的,它在不同的计算机上运行时依赖于本地计算机的 CPU 指令集,而这个任务,是由 .NET 框架的 CLR 完成的。目前,较新的 Windows 操作系统,如 Vista、Windows 7 等,都已经自动配置了 .NET 框架,而早期的操作系统,如 Windows XP 等,则在用户安装 .NET 框架后才能支持 .NET 程序的运行。

1.1.2 Visual Studio .NET 集成开发环境

集成开发环境(Integrated Development Environment, IDE)是指集程序的设计、编辑、运行、调试以及部署等功能于一体的软件开发工具,可以大大提高程序开发的效率。下面以 Visual Studio .NET 2005 为例,简单介绍一下 Visual Studio .NET 开发环境。

1. 进入 Visual Studio .NET 2005

Visual Studio .NET 以项目为单位进行应用程序的开发。启动 Visual Studio .NET 后,进入起始页,用户可以打开以前创建的项目,也可以创建新项目。这里以创建新项目为例,选择“文件”→“新建”→“项目”命令,打开“新建项目”对话框,如图 1.1.3 所示。

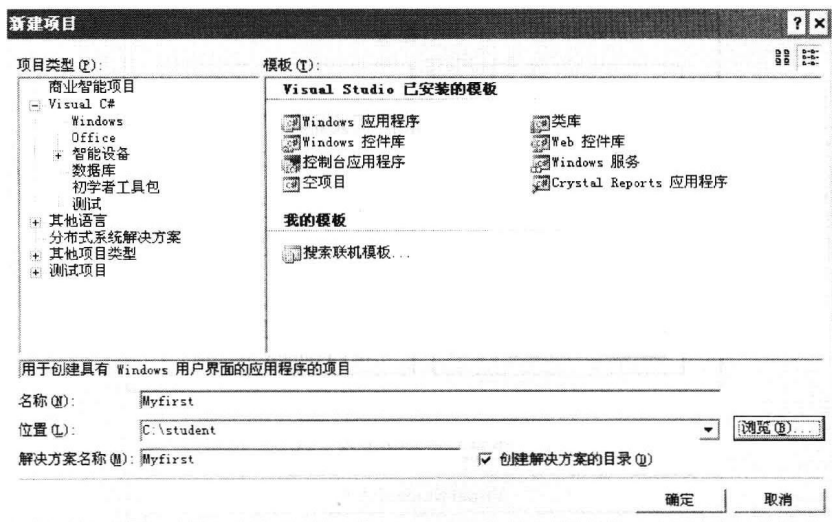


图 1.1.3 “新建项目”对话框

在“项目类型”列表框中选择“Visual C#”;在“模板”列表中选择“Windows 应用程序”;在“名称”文本框中输入项目名称;在“位置”下拉列表中选择项目存放的文件夹。本例中,项目命名为“MyFirst”,位置存放在“C:\student”。“解决方案名称”文本框会被系统自动命名为与“名



称”文本框同样的内容,如果感觉不妥可以进行修改。

在以上信息确定后,单击“确定”按钮进入 C#.NET 程序设计环境,如图 1.1.4 所示。

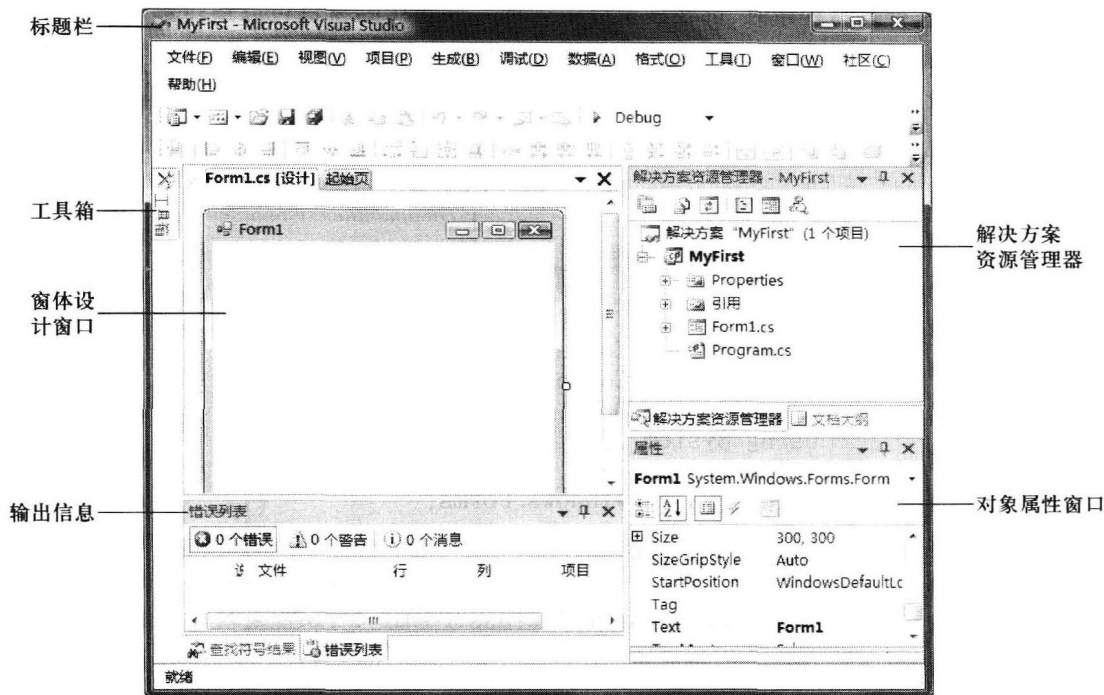


图 1.1.4 .NET C#开发集成环境

.NET C#集成开发环境由多窗体组成,这些窗口被分成两大类,一类是位置相对固定的,如窗体设计窗口和代码窗口;另一类是可以浮动、隐藏、停靠的其他窗口,如对象属性窗口等,这类窗体的位置可以调整,用户也可以通过“视图”菜单决定它们是否显示。

2. 主要窗口简介

(1) 主窗口

主窗口位于集成开发环境的顶部,图 1.1.4 中标题栏为“MyFirst-Microsoft Visual Studio”的窗口就是主窗体。主窗体关闭后,就退出了集成开发环境。

(2) 窗体设计窗口

窗体设计窗口是 C#Windows 应用程序运行时显示的窗口,也是操作者和程序交互的接口。在设计的过程中,设计者一般会从工具箱中选择一些控件放置在设计窗体上,通过这些控件查看程序运行的结果或接受用户的输入。一个应用程序可以由多个窗体组成,通过选择“项目”→“添加 Windows 窗体”命令可以为应用程序添加新窗体。



(3) 代码设计窗口

代码设计窗口是设计者实现程序逻辑的地方,如图 1.1.5 所示。设计者在项目中设计类、针对窗体控件事件进行编写程序时,都在该窗口中进行。打开代码窗口可以通过解决方案资源管理器完成,方法是选择一个后缀是 cs 的文件,然后单击解决方案资源管理器工具栏上的“查看代码”按钮。针对设计窗体,也可以通过单击窗体上的控件的相应事件来打开代码窗口,例如窗体上的按钮控件等。

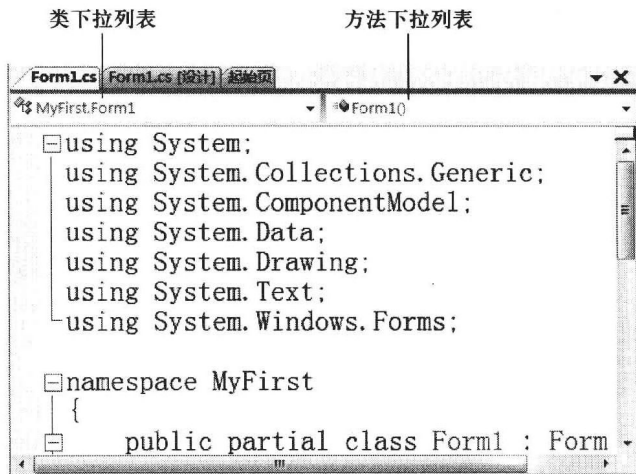


图 1.1.5 代码设计窗口

代码窗口有如下内容。

① 类下拉列表。该下拉列表只显示当前设计者操作的类。

② 方法下拉列表。显示设计者操作类的当前操作的方法,通过它选择类的方法,可以快速地进入指定方法的代码进行查看或编辑。

代码窗口中的代码会被系统标记为不同的颜色,蓝色为关键字,黑色为常规代码,绿色为注释。如果代码中有错误,错误处会以红色的波浪线标识,当设计者将鼠标移动到这些地方时,系统会自动提示错误的原因,以便设计者在编译代码前就能及时发现错误。

在新建一个 C#Windows 应用程序后,系统会自动生成一个默认名为 Form1 的窗体,在 Form1.cs 文件中可以看到如下代码:

```
public partial class Form1 : Form
```

从语法定义中可以知道,C#把窗体定义成一个类,它继承自 Form 类,Form 类在这里为缩写,它实质上是 System.Windows.Forms.Form 类。

在代码窗口中,设计者可以看到很多“+”“-”号,通过它们,程序员可以将代码进行折叠或展开。



(4) 对象属性窗口

属性窗口用于帮助程序员设置可视化窗体中控件对象的属性。如图 1.1.6 所示,它由以下 3 部分组成。

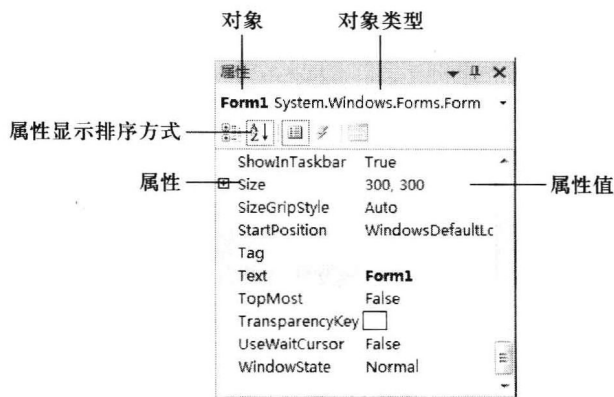


图 1.1.6 对象属性窗口

① 对象下拉列表:单击其右边的下拉按钮,可以看到当前选定的设计窗体中的所有对象以及这些对象的类型。

② 属性显示排序方式:有两种排序方式,即按字母顺序排序和按分类顺序排序。

③ 属性列表:列出所选对象在设计模式下可编辑的属性及其默认值。

(5) 解决方案资源管理器

一个 .NET 应用程序由多个文件共同组成,这些文件共同组成了项目的解决方案,并以树状结构图显示,如图 1.1.7 所示。一个解决方案主要包含以下类型的文件。

① 代码文件(.cs 文件):.cs 文件是程序源码文件。

② 项目文件(.csproj 文件):每个 C# 项目对应一个项目文件。本例中的项目名为 MyFirst, 其对应的项目文件名为 MyFirst.csproj。项目通常由引用文件和代码模块文件组成。

③ 引用文件:每个 C# 项目都会自动引用系统定义类库文件,如 System.Data。程序员也可以定义自己的类库,生成 DLL 文件后,在新的项目中加以引用,或引用成熟的第三方的类库。

④ 解决方案文件(.sln 文件):在建立一个项目时,系统定义解决方案文件名与项目文件名相同,仅扩展名不同。

(6) 工具箱窗口

工具箱窗口包括 C# 项目开发时需要经常使用的工具条目,以目录树方式分类组织,如图 1.1.8 所示。工具箱窗口由 9 个目录组成,常用的目录如下。

① 所有 Windows 窗体:存放了所有的窗体设计时所需的控件,是下面所有分支的总和。

② 公共控件:包含了所有运行时可视化的控件。



③ 容器:可视化的容器控件作为容器,可以在其内放置其他的控件,如 GroupBox、Panel 控件。

④ 菜单和工具栏:窗体中的主菜单、工具栏和弹出式菜单。

⑤ 数据:与数据库相关的控件。NET 2005 中,与数据库相关的控件不是默认显示的,在程序员需要的情况下,可以通过选择“工具”→“选择工具箱项”命令将其选择到工具箱中。

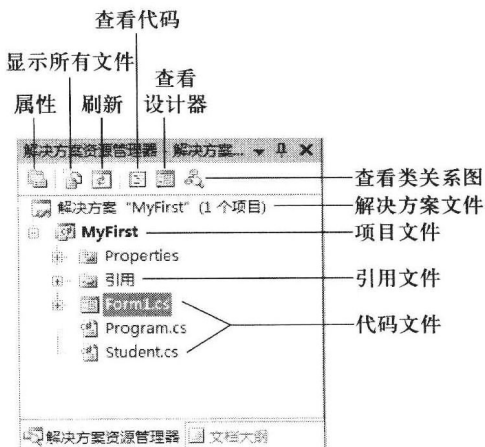


图 1.1.7 解决方案资源管理器

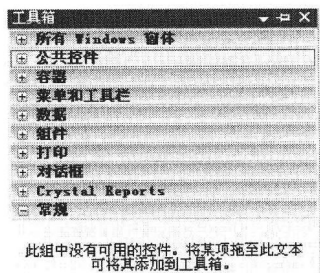


图 1.1.8 工具箱

1.1.3 C#Windows 应用程序开发过程

下面以一个简单的 C#Windows 小程序为例,向读者介绍程序开发过程,以便读者能快速地了解 C#的编程环境。建立和运行一个 C#Windows 应用程序的步骤如下。

- ① 建立一个新项目。
- ② 设计程序运行界面,向界面中添加控件对象。
- ③ 设置对象属性。
- ④ 针对对象事件进行编程。
- ⑤ 保存、调试、运行程序。

例 1.1 设计一个如图 1.1.9 所示的程序界面。在程序运行启动后,单击“显示”按钮,显示“Welcome to c#”,如图 1.1.10 所示。

1. 建立一个新项目

启动 Visual Studio .NET 后,通过选择“文件”→“新建”→“项目”命令,建立一个新项目,项目命名为“HelloCS”。