



第 1 章 C#语言概述

本章学习目标

- (1) .NET 概念及组成。
- (2) Visual Studio 2022 集成开发环境。
- (3) C#程序结构及特点。
- (4) Visual Studio 2022 的安装。
- (5) C#程序创建的一般方法。

1.1 .NET 简介

1. .NET 战略的起因与发展

微软公司启动一个计划，开发一种独立于特定语言和平台的环境，在 2000 年 6 月正式推出了 .NET 战略。2002 年发布了 Microsoft Visual Studio .NET 软件包代替了原来的 Microsoft Visual Studio，该软件包的核心是 .NET Framework 1.0（简称 .NET 框架 1.0）。.NET 版本不断更新，较为流行的版本有 Microsoft Visual Studio 2008、Microsoft Visual Studio 2012、Microsoft Visual Studio 2015、Microsoft Visual Studio 2019，最新版 Microsoft Visual Studio 2022（简称 VS 2022）在北京时间 2021 年 11 月 9 日凌晨正式发布。

2. .NET 的定义

.NET 是一种用于构建多种应用的免费开源开发平台，可以使用多种语言，编辑器和库开发 Web 应用、Web API 和微服务、云中的无服务器函数、云原生应用、移动应用、桌面应用、Windows WPF、Windows 窗体、通用 Windows 平台（UWP）、游戏、物联网（IoT）、机器学习、控制台应用、Windows 服务。.NET 类库在不同应用和应用类型中共享功能，无论构建哪种类型的应用，代码和项目文件看起来都一样，可以访问每个应用的相同运行时、API 和语言功能。

.NET 战略是微软公司推出的一个全新概念，“它代表了一个集合、一个环境和一个可以作为平台支持下一代 Internet 的可编程结构”。.NET 的目的就是将互联网作为新一代操作系统的基础，对互联网的设计思想进行扩展。.NET 的最终目标就是让用户在任何地方、任何时间，利用任何设备都能访问所需的信息、文件和程序。

3. .NET 开发平台

.NET 开发平台包括 .NET 框架和 .NET 开发工具等组成部分。.NET 框架是整个开发平台的基础，包括公共语言运行库（common language runtime, CLR）和框架类库。.NET 开发工具包括 Visual Studio .NET 集成开发环境和 .NET 编程语言。Visual Studio .NET 集成

开发环境用来开发和测试应用程序。 .NET 编程语言包括 Visual Basic、 Visual C++和 Visual C#等， 这些语言用来创建运行在公共语言运行库上的应用程序。 .NET 开发平台如图 1-1 所示。

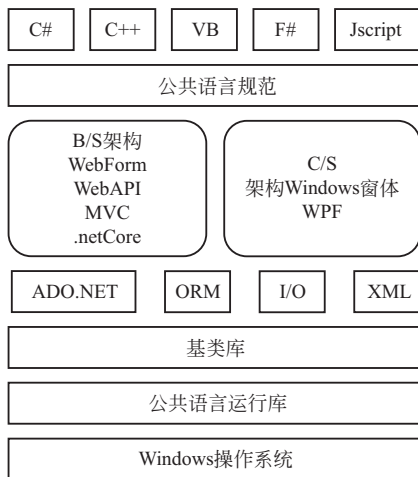


图 1-1 .NET 开发平台

4. .NET 框架

.NET 框架 (.NET framework) 是由微软开发， 致力于敏捷软件开发 (agile software development)、 快速应用开发 (rapid application development)、 平台无关性和网络透明化的软件开发平台。 .NET 是微软为下一个十年对服务器和桌面型软件工程迈出的第一步。 .NET 包含许多有助于互联网和内部网应用迅捷开发的技术。

.NET 框架是一个多语言组件开发和执行环境， 它提供了一个跨语言的统一编程环境。 .NET 框架的目的是便于开发人员更容易地建立 Web 应用程序和 Web 服务， 使得 Internet 上的各应用程序之间， 可以使用 Web 服务进行沟通。 从层次结构来看， .NET 框架又包括三个主要组成部分： 公共语言运行

库、 服务框架 (services framework) 和上层的两类应用模板——传统的 Windows 应用程序模板 (win forms) 和基于 ASP.NET 的面向 Web 的网络应用程序模板 (web forms 和 web services)。

5. Visual Studio 2022 新功能

Visual Studio 2022 已正式发布， 是 Windows 上的 64 位应用程序， 主要新增以下新功能。

- (1) 默认启用文件中的索引查找， 将代码搜索时间缩短至 1 s 左右。
- (2) 引入自动保存文件的新功能， 都会尝试保存 IDE 中的包括项目代码、 解决方案及其他杂项文件。
- (3) 智能感知与智能编码。 微软融入了大量的人工智能因素， 系统可以根据用户前面代码的内容、 关键字等因素， 自动提示下面的编码内容， 用户只需要按一下 Tab 键， 一长串代码就自动填充在用户的方法中了。
- (4) .NET 6.0。 VS 2022 全面支持 .NET 6.0， 这个框架为网站应用、 客户端应用和移动应用提供了统一的技术支持， 同时支持 Windows 和 MacOS 平台； 这个框架还支持 .NET 多平台的界面开发 (multi-platform APP UI， 也称为 .NET MAUI)。 这个跨平台的开发利器， 为开发人员编写基于多种平台的应用 (Windows、 Android、 MacOS、 iOS) 提供便捷的途径。 对于需求客户端程序的开发， 如 Web 应用、 桌面程序或移动应用， 使用 .NET Hot Reload， 不需要重新启动程序， 就可以将修改的代码应用到程序中， 从而避免在调试修改过程中丢失状态。
- (5) 改良方案过滤器。 方案过滤器可以筛选加载的项目， 如可以选择加载单个项目， 或者加载带有整个依赖关系树的项目。
- (6) 调试和诊断功能增强， 安装和更新优化。

1.2 C#语言简介

1.2.1 C#语言背景

1995 年，Sun 公司正式推出面向对象的开发语言 Java，并提出跨平台、跨语言的概念，之后，Java 就逐渐成为企业级应用系统开发的首选工具。微软开发出基于 Java 语言的编译器 Visual J++，并在很短的时间里由 1.1 版本升级到 6.0 版本，其中 Visual J++ 6.0 版集成在 Visual Studio 6.0 中。.NET 计划是微软发展的战略核心，其内容非常庞大。它的技术开发平台是 Visual Studio .NET，而 C#作为 Visual J++的替代语言也集成在该平台中。

C#（英文名为 C Sharp）是微软开发的一种面向对象的编程语言。微软公司对 C#的定义是：“C#是一种类型安全的、现代的、简单的、由 C 和 C++衍生出来的面向对象的编程语言，它是牢牢根植于 C 和 C++语言之上的，并可立即被 C 和 C++开发人员所熟悉。C#的目的就是综合 Visual Basic 的高生产率和 C++的行动力。”提到 C#不得不介绍其创始人 Anders，他可谓是编程语言的奇才。他在开发 C#语言之前曾开发了大家熟知的 Delphi 语言。微软在研发 C#语言之初是高薪聘请了这位奇才来主持开发的。C#语言是一种安全的、稳定的、简单的、面向对象的编程语言，它不仅去掉了 C++和 Java 语言中的一些复杂特性，还提供了可视化工具，能够高效地编写程序。

1.2.2 C#语言特点

(1) 简单安全。在 C++和 C 语言中程序员最头疼的问题就是指针问题，在 C#语言中已经不再使用指针，而且不允许直接读取内存等不安全的操作。它比 C、C++、Java 提供了更多的数据类型，并且每个数据类型都是固定大小的。此外，C#还提供了命名空间来管理 C#文件，命名空间相当于一个文件夹，在创建程序时，允许在一个命名空间中创建一个或多个类，方便调用和重用。

(2) 面向对象。C#语言与其他面向对象语言一样，也具有面向对象语言的基本特征，即封装、继承、多态。封装：就是将代码看作一个整体，如使用类、方法、接口等。在使用定义好的类、方法、接口等对象时不必考虑其细节，只需要知道其对象名及所需要的参数即可，也是一种提升代码安全性的方法。继承：是一种体现代码重用性的特性，减少代码的冗余，但在 C#语言中仅支持单继承。多态：不仅体现了代码的重用性，也体现了代码的灵活性，它主要通过继承和实现接口的方式，让类或接口中的成员表现出不同的作用。

(3) 支持跨平台。最早的 C#语言仅能在 Windows 平台上开发并使用，目前最新的 C# 6.0 版本已经能在多个操作系统上使用，如在 Mac、Linux 等操作系统上使用。此外，还能将其应用到手机、PDA 等设备上。

(4) 开发多种类型的程序。使用 C#语言不仅能开发在控制台下运行的应用程序，也能开发 Windows 窗体应用程序、网站、手机应用等多种应用程序，并且其提供的 Visual Studio 2019 开发工具中也支持多种类型的程序，让开发人员能快速地构建 C#应用程序。

1.2.3 C#与.NET 的关系

.NET 是一个开发平台，而 C# 是一种在 .NET 开发平台上使用的编程语言，目前能在 .NET 平台上使用的开发语言很多，如 Visual Basic.NET、Python、J#、Visual C++.NET 等。但在 .NET 平台上使用最多的是 C# 语言。.NET 框架是一个多语言组件开发和执行环境，它提供了一个跨语言的统一编程环境。.NET 框架的目的是便于开发人员容易地建立 Web 应用程序和 Web 服务，使得 Internet 上的各应用程序之间可以使用 Web 服务进行沟通。

1.2.4 面向对象程序设计基础

(1) 类与对象。类是对象的模板，它定义了对象的特征和行为规则，对象是通过类产生的，类和对象都由唯一的名字进行标识，即类名和对象名。

(2) 属性。属性是类或对象的一种成分，它反映类创建的对象特征，如对象的名称、大小、标题等。

(3) 方法与事件。对象功能就是方法，能够响应的刺激就是事件。

(4) 事件驱动的程序设计。面向对象的程序设计语言的基本编程模式是事件驱动。程序的执行是由事件驱动的，一旦程序启动后就根据发生的事件执行相应的程序代码（事件过程），如果无事件发生，程序就空闲着，等待事件的发生，此时用户也可以启动其他的应用程序。在这种程序设计模式下，程序员只需考虑发生了某事件时，系统该做什么，从而编制出相应的事件过程代码。图 1-2 为面向对象编程过程。

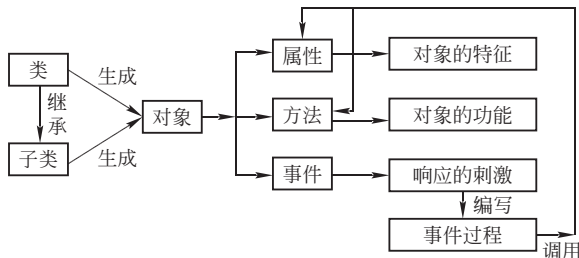


图 1-2 面向对象编程过程

(5) 可视化程序设计的一般步骤。首先利用系统给定的可视化类设计出程序运行界面，然后设计窗口和可视化工具的属性，最后编写事件过程代码。

1.3 C#安装与使用

1.3.1 Visual Studio 2022 的安装

在使用 C# 语言进行应用程序开发之前，首先需要安装 Visual Studio 2022，官网地址为：<https://visualstudio.microsoft.com/zh-hans/downloads/>，进行下载，下载页面如图 1-3

所示，其中社区版免费使用，Professional 和企业版都需要输入密钥。

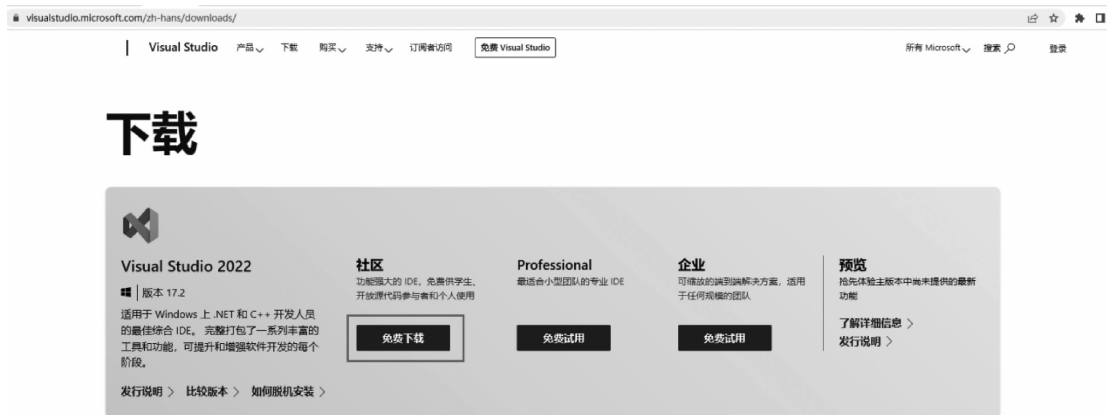


图 1-3 下载页面

安装步骤如下。

(1) 检查安装环境。在开始安装 Visual Studio 前，查看系统要求。应用最新的 Windows 更新，可确保计算机包含最新的安全更新程序和 Visual Studio 所需的系统组件。重新启动，可确保挂起的任何安装或更新都不会影响 Visual Studio 安装。释放空间，通过运行磁盘清理应用程序等方式，从 %SystemDrive% 删除不需要的文件和应用程序。

(2) 下载 Visual Studio 2022。下载 Visual Studio 2022 安装程序文件 VisualStudioSetup.exe。

(3) 运行 Visual Studio 2022 安装程序。双击 VisualStudioSetup.exe 文件，安装 Visual Studio 2022 Community（社区版），将出现如图 1-4 所示界面，要求确认 Microsoft 软件许可条款和 Microsoft 隐私声明，单击“继续”按钮。

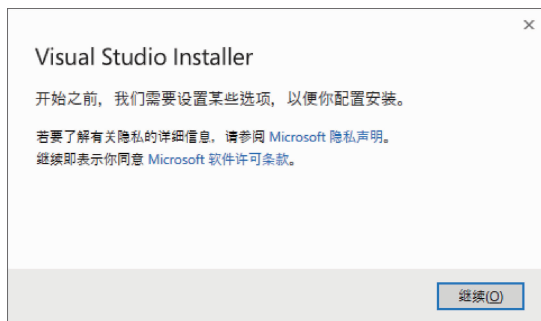


图 1-4 安装条款阅读页面

(4) 选择工作负荷。安装该安装程序后，可以通过选择所需的功能集或工作负荷来使用该程序自定义安装。操作方法如图 1-5 所示，在“Visual Studio 安装程序”中找到所需的工作负荷。

例如，选择“ASP.NET 和 Web 开发”工作负荷。它附带默认核心编辑器，该编辑器针对超过 20 种语言提供基本代码编辑支持，能够打开和编辑任意文件夹中的代码（而无



图 1-5 安装负荷选择

须使用项目)，还提供集成的源代码管理。选择所需的工作负荷后，单击“安装”按钮。接下来，会出现多个显示 Visual Studio 安装进度的状态屏幕。图 1-6 为安装提示。

💡 提示

在安装之后，可以随时安装最初未安装的工作负荷或组件。如果已打开 Visual Studio，请转到“工具” > “获取工具和功能...”，这会打开 Visual Studio 安装程序。或者，从“开始”菜单打开“Visual Studio 安装程序”。在此处可以选择要安装的工作负荷或组件。然后，选择“修改”。

图 1-6 安装提示

(5) 选择各个组件（可选）。如果不想使用工作负荷功能来自定义 Visual Studio 安装，或者想要添加比工作负载安装更多的组件，可通过从“各个组件”选项卡上安装或添加各个组件来完成此操作。选择所需组件，然后按照提示进行操作，如图 1-7 所示。

(6) 安装语言包（可选）。默认情况下，安装程序首次运行时尝试匹配操作系统语言。若要以所选语言安装 Visual Studio，请从 Visual Studio 安装程序中选择“语言包”选项卡，然后按照提示进行操作，如图 1-8 所示。

(7) 选择安装位置（可选）。为减少系统驱动器上 Visual Studio 的安装占用，可以选择将下载缓存、共享组件、SDK 和工具移动到不同驱动器，并将 Visual Studio 安装在其运行速度最快的驱动器上，如图 1-9 所示。只有首次安装才可选择其他驱动器。

(8) 安装完成。



图 1-7 组件选择



图 1-8 安装语言选择



图 1-9 安装位置

1.3.2 C#的控制台应用程序

例 1-1 在 Visual Studio 2022 中，创建控制台应用程序。

步骤：(1) 首先打开桌面“开始菜单”，单击“Visual Studio 2022”的图标，将出现该软件的起始界面（见图 1-10）。



图 1-10 起始界面

(2) 然后在出现的 Visual Studio 2022 窗口（见图 1-11），选择“创建新项目”。



图 1-11 创建新项目

(3) 在“创建新项目”窗口的搜索框中输入或键入“控制台”。接下来，从“语言”列表中选择 C#，然后从“平台”列表中选择 Windows。应用语言和平台筛选器之后，选择“控制台应用 (.NET Core)”模板，然后单击“下一步”按钮，如图 1-12 所示。

注意：此处有 2 个控制台桌面模板，1 个用于 .NET Framework，另 1 个用于 .NET Core，本书建议选择 .NET Core 用于开发。

(4) 在“配置新项目”窗口中，在“项目名称”框中键入或输入“EX1_1”，“位置”选择当前文件保存的位置，也可以用默认位置，“解决方案名称”默认跟项目名称一样，单击“下一步”按钮，如图 1-13 所示。



图 1-12 控制台模板

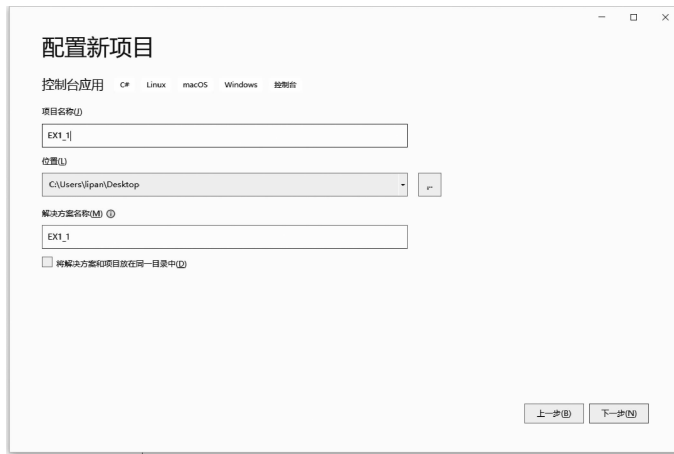


图 1-13 项目名称

(5) 在“其他信息”窗口，直接选择“创建”进行项目的新建，如图 1-14 所示。

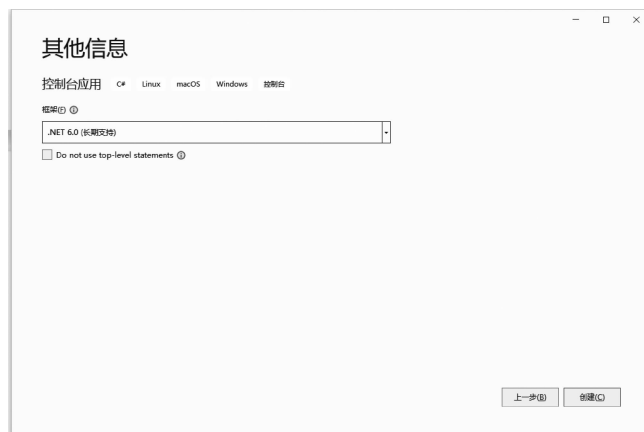


图 1-14 项目创建

(6) Visual Studio 随即打开新项目，出现界面如图 1-15 所示，在 Program.cs 文件中包含默认的 1 行“Console.WriteLine (“Hello, World!”);”代码，向控制台输出一行文字，以往版本中的引用和命名空间都没有了，代码窗口被大大的简化。

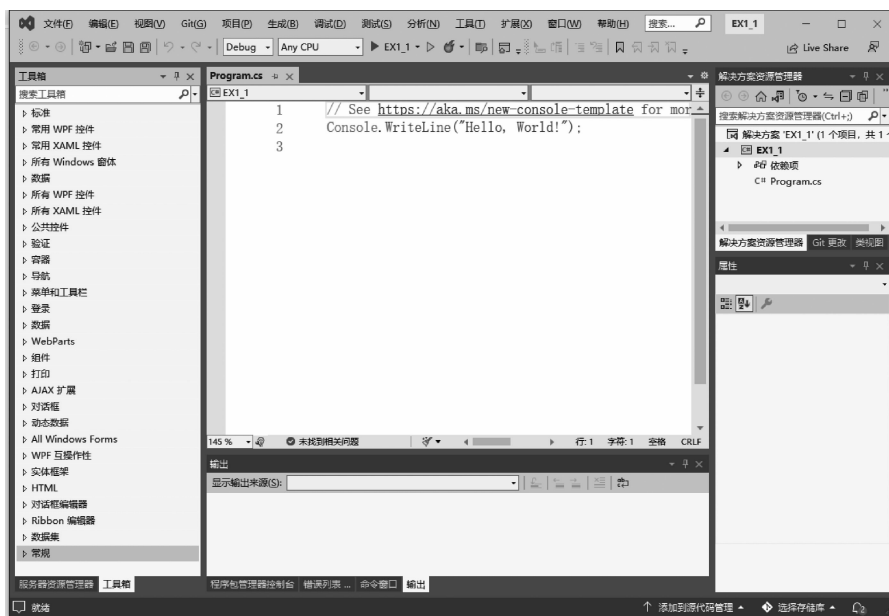


图 1-15 项目界面

程序中的语句“Console.WriteLine (“Hello, World!”);”作用是输出“Hello, World!”信息。该语句通过 Console 的 WriteLine () 方法输出文字，WriteLine 是一个定义在 System 命名空间中的 Console 类的一个方法，该语句会在屏幕上显示消息“Hello, World!”。

(7) 按“F5”键，该项目的运行结果如图 1-16 所示。

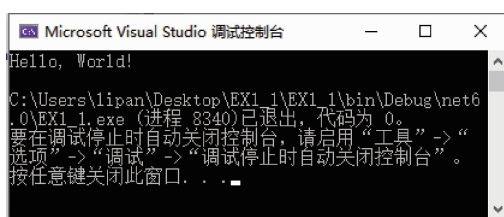


图 1-16 项目运行结果

1.3.3 C#的 Windows 应用程序

1. 案例

例 1-2 新建 Windows 应用程序，实现显示和退出功能。

步骤：(1) 在 Visual Studio 2022 集成开发环境中，依次选择“文件”|“新建”|“项目”菜单，如图 1-17 所示。弹出“创建新项目”对话框，选择“Windows 窗体应用”，单击“下一步”按钮，如图 1-18 所示。



图 1-17 新建项目



图 1-18 创建新项目

(2) 在“配置新项目”窗口，输入项目的名称，选择保存路径，然后单击“下一步”按钮，再单击“创建”按钮，即可创建 1 个 Windows 窗体应用程序，如图 1-19 所示。

(3) 打开的窗口为 Visual Studio 2022 集成开发环境，如图 1-20 所示。

2. Visual Studio 2022 集成开发环境介绍

Visual Studio 是微软公司的开发工具包系列产品。VS 是一个基本完整的开发工具集，它包括了整个软件生命周期所需要的大部分工具，如 UML 工具、代码管控工具、集成开发环境（IDE）等。所写的目标代码适用于微软支持的所有平台，包括 Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework 和 Microsoft Silverlight 及 Windows Phone。



图 1-19 项目配置

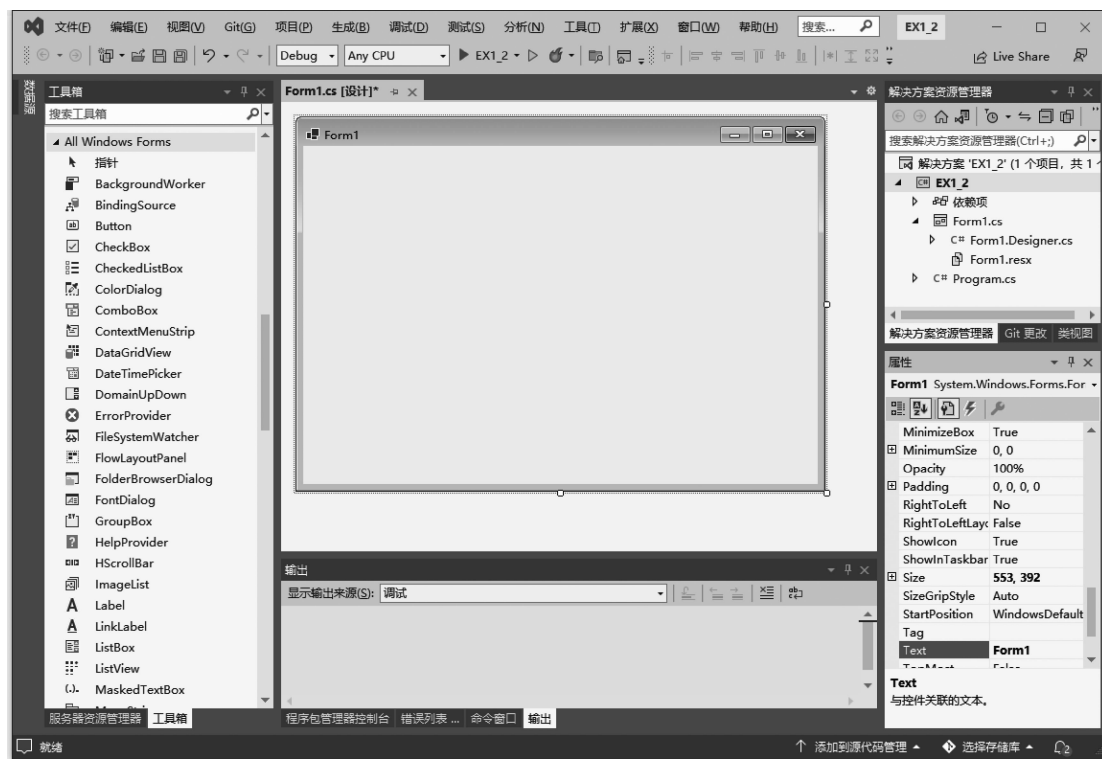


图 1-20 Visual Studio 2022 的集成开发环境

1) “决方案资源管理器”窗口

项目可以视为编译后的一个可执行单元，可以是应用程序、动态链接库等，而企业级的解决方案往往需要多个可执行程序的合作，为便于管理，在 Visual Studio.NET 集成环境中引入了解决方案资源管理器，如图 1-21 所示。如果没有出现“解决方案资源管理器”窗口，可以通过“视图”|“解决方案资源管理器”命令来显示。

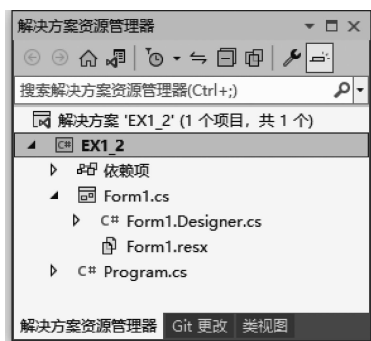


图 1-21 解决方案资源管理器

2) “工具箱”窗口

Visual Studio 2022 的“工具箱”功能强大，里面含有各类控件，如图 1-22 所示。“工具箱”控件种类众多，展开“常规”控件，显示如 Button、Text、Label、CheckBox 等命令。如果集成环境里面没有“工具箱”，可以通过“视图”|“工具箱”命令来实现。

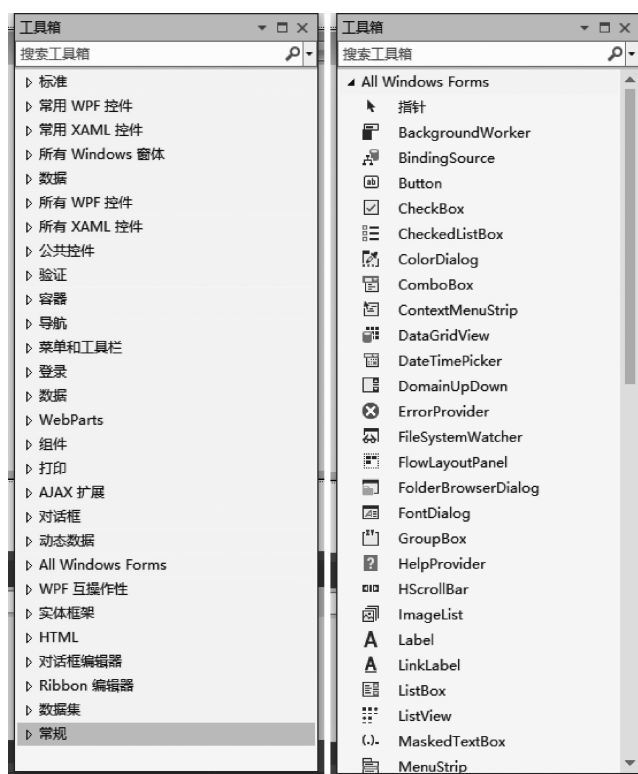


图 1-22 工具箱

Visual Studio 2022 里面还可以添加新工具，通过选择工具 | 选项 | Windows 窗体设计器 | 常规 | 工具箱 | 自动填充工具箱 | True 来添加，如图 1-23 所示。

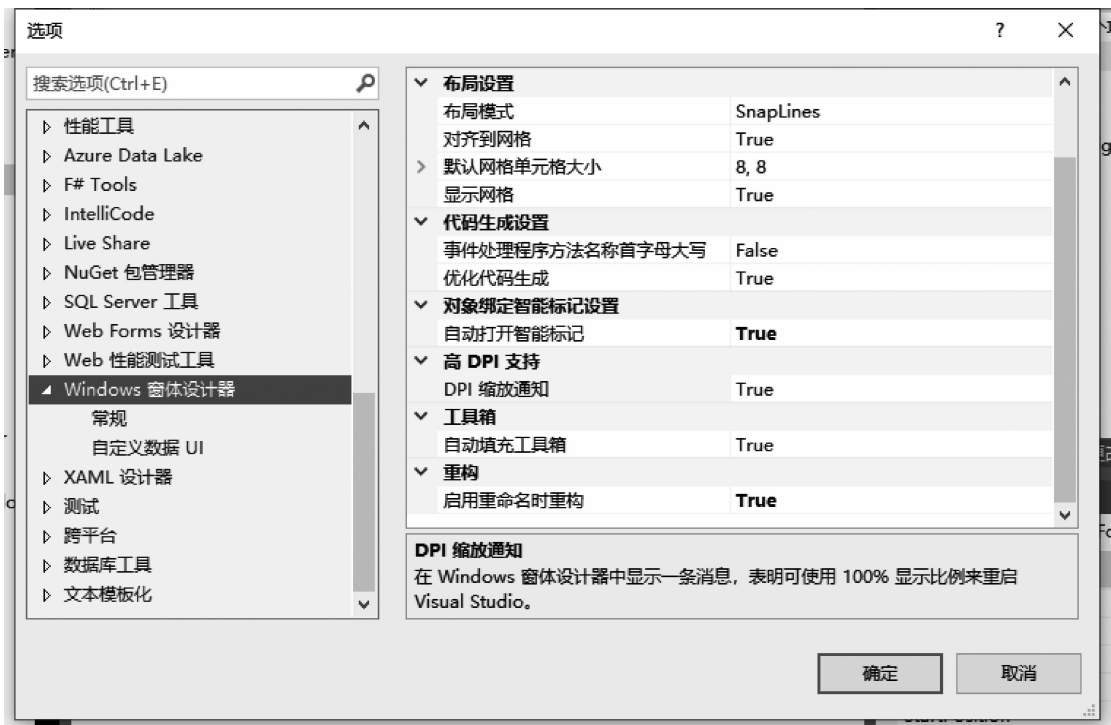


图 1-23 自动填充工具箱

“工具箱”中包含了可重用的控件或组件，用于自定义应用程序，当使用可视化的方法编程时，可在窗体中拖放控件，绘制出应用程序界面，而不用自己去写代码。

3) “属性”窗口

“属性”窗口如图 1-24 所示。如果界面里没有出现这个窗口，可以通过选择菜单“视图”|“属性窗口”命令或按 F4 键命令来显示该窗口。“属性”窗口设置窗体或控件的属性，选择窗体或控件要处理的事件并编写事件代码。属性确定了控件的基本信息，如字

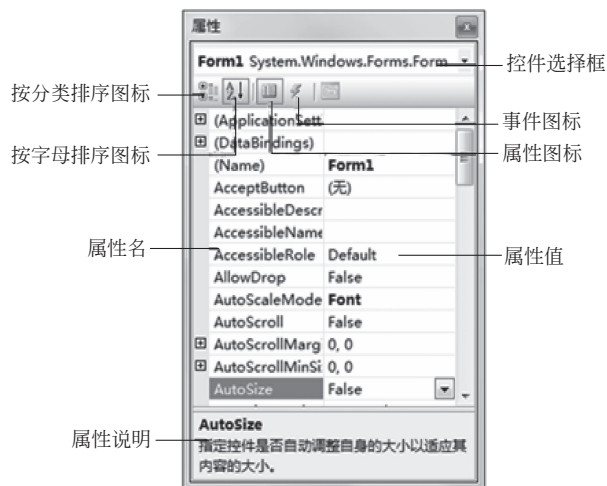


图 1-24 属性窗口

体、大小、边框、数据等，每个控件都有自己的属性。事件与具体的对象相关，由系统或用户激发，事件一旦激发将执行相应的事件代码。

当“属性”窗口中的“属性”图标处于被按下状态时，左边一栏显示选中对象（窗体或控件）的属性名，右边一栏显示属性的当前值，如图 1-24 所示。当“属性”窗口中“事件”图标处于被按下状态时，左边一栏显示选中窗体或控件的事件名，右边一栏显示事件过程名，如图 1-25 所示。属性分别有“按字母排序图标”和“按分类排序图标”两种排序。其中“按分类排序图标”是指属性名或事件名按照分类顺序排列，是默认方式，“按字母排序图标”是属性名或事件名按照字母顺序排列。

在“属性”窗口顶部是一个下拉列表，被称为控件或组件选择框，是一个列表框。列表框显示当前正在修改的控件或组件，可以使用该列表框来选择一个控件或组件进行修改。例如，窗体上有几个 Button 按钮，可以通过它选择指定按钮的 Text 属性来进行修改。“属性”窗口的底部有 1 个小窗口，可以显示选中属性或事件的说明，也就是属性的作用和事件发生的时机。

4) 界面设计

创建完项目后，在 VisualStudio 2022 开发环境中会有 1 个默认的窗体，可以通过工具箱向其中添加各种控件来设计窗体界面。具体步骤是：用鼠标按住工具箱中要添加的控件，然后将其拖放到窗体中的指定位置即可。本实例分别向窗体中添加 1 个 Label 控件和 1 个 Button 控件，设置窗体和控件的属性（见表 1-1），界面设计效果如图 1-26 所示。

表 1-1 例 1-2 的对象设置属性值

名称	属性	设置值
Form1	Text	C#窗口
Label1	Text	“ ”
	Font	四号
Button1	Text	显示

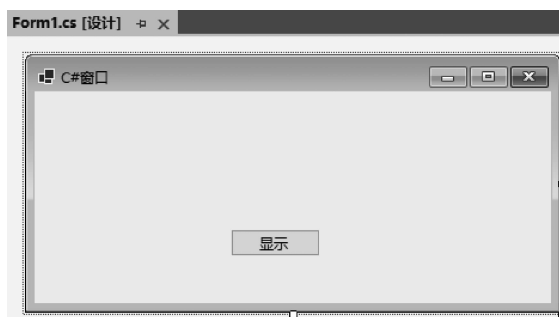


图 1-26 界面设计效果

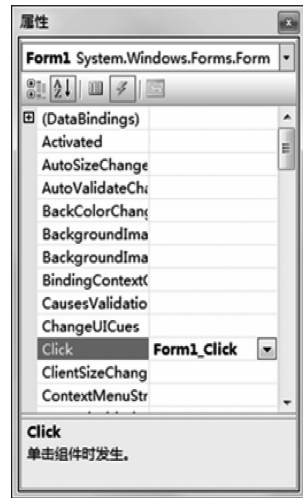


图 1-25 属性事件

5) 编写程序代码

双击 Button 控件，即可进入代码编辑器，并自动触发 Button 控件的 Click 事件，在该事件中即可编写代码，也可以选中 Button，在属性窗口单击事件按钮，然后双击 Click 事件（见图 1-27）进入代码窗口，Button 控件的默认代码如下：

```
private void button1_Click(object sender, EventArgs e)
{
    label1.Text = "欢迎进入 Visual C#2022 编程世界!";
}
```

6) 保存项目

单击 VisualStudio 2022 开发环境工具栏中的按钮，或者选择“文件”|“全部保存”菜单，即可保存当前项目。

7) 运行程序

单击 VisualStudio 2022 开发环境工具栏中的按钮，或者选择“调试”|“开始调试”菜单，即可运行当前程序，程序效果如图 1-28 所示。

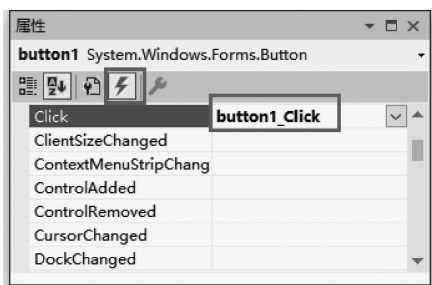


图 1-27 button1 事件



图 1-28 程序效果

1.4 C#程序代码

结合例 1-2 中出现的代码，下面进行 C#语言的简单语法介绍，在 Windows 应用程序项目中，默认有 1 个 Program.cs 代码文件，自动生成代码如图 1-29 所示。

1. 命名空间

用 namespace 声明，它包含一系列的类。命名空间 EX1_2 包含了类 Program。

2. 类

程序的“class Program”是类的声明，它声明的类的名字为 Program，程序的功能就是依靠该类来完成的。类 Program 包含了程序使用的数据和方法声明，类一般包含多个方法，方法定义了类的行为，在这里 Program 类只有一个 Main 方法。C#要求程序中的每个元素都要属于一个类。C#中用大括号“{}”构成一个程序块，大括号应成对出现，可以嵌套。

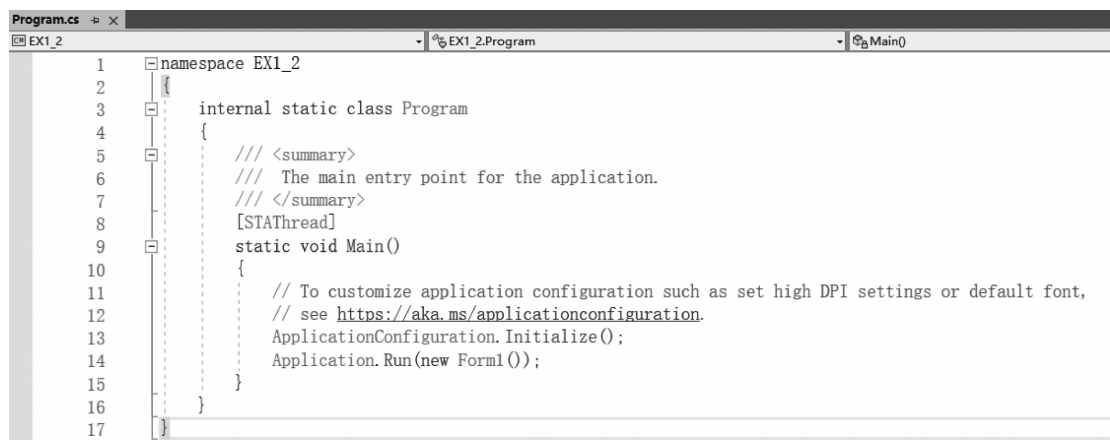


图 1-29 控制台程序代码

3. 类的方法

程序中的语句“`public static void Main (string [] args)`”为 Program 类声明了一个方法。程序的执行总是从 Main() 方法开始的，是 C#程序的入口点。一个程序中不允许出现两个或两个以上的 Main() 方法，而且 C#中 Main() 方法必须被包含在一个类中。

4. 注释

(1) 使用“//”。“//”符号后面的内容就是注释的内容（单行）。

(2) 使用“/* */”符号对。“/* */”符号对之间的内容都是注释内容（单行或多行）。

5. 语句书写规则

(1) C#程序对大小写是区分的。

(2) C#的每条语句都必须是分号“;”结尾。

(3) 可以在一行上写多条 C#的语句，也可以把一条 C#语句写在多行上。

(4) 程序的执行从 Main 方法开始。

(5) 与 Java 不同的是，文件名可以不同于类的名称。

6. using 关键字

代码窗口中可以在一开始添加语句“`using System;`” using 关键字用于在程序中包含 System 命名空间。一个程序一般有多个 using 语句。它的作用是导入命名空间，该语句类似于 C 和 C++中的#include 命令。导入命名空间之后，就可以自由地使用其中的元素。

7. 整体排版

单击 F7 键可直接从窗体设计进入窗体的代码窗口，删去最后一个大括号，然后再次输入一个大括号“}”，可以自动对代码进行整体排版。

8. 删除自动生成控件代码

在窗体 Form1 上删除控件，有时需要在 Form1.Designer.cs 代码中同步删除自动生成控件代码，否则程序报错。


```

{
    Console.WriteLine("Hello,这是我的第一个C#程序");
}
}

```

(2) 在 D 盘保存文件为 HelloWorld.cs，保存类型选择“所有文件”。然后打开“开始”菜单（见图 1-30），进行选择：开始 | Visual Studio 2022 | Developer Command Prompt for VS 2022，打开命令提示窗口。在命令提示窗口输入命令：CSC/?，然后单击回车键，显示出编译器的各种相关命令参数，如图 1-31 所示。

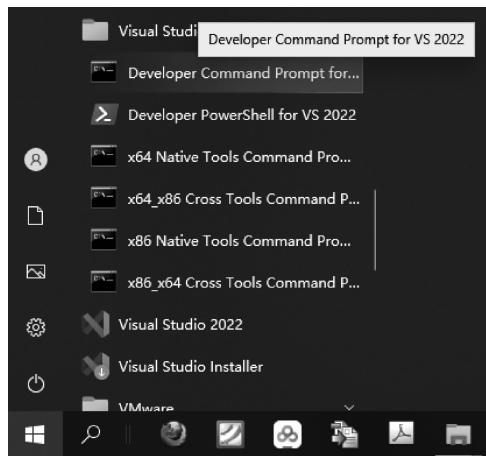


图 1-30 “开始”菜单选择



图 1-31 编译器的命令参数

(3) 输入 CSC D: \ HelloWorld.cs 命令，单击回车键，如果程序没有出错，然后再输

入文件名 HelloWorld，单击回车键，输出字符串如图 1-32 所示。



图 1-32 HelloWorld 运行结果

实验 1-2 用 C#语言编写 Windows 应用程序，在窗体界面上显示一行字符：“你好”

```
using System;
using System.Windows.Forms;
namespace WinFormsApp1
{
    public partial class Form1 : Form
    {
        TextBox txtTest;
        public Form1 ()
        {
            InitializeComponent();
            Form myfrm = new Form();
            txtTest = new TextBox();
            txtTest.Size = new System.Drawing.Size(100, 20);
            txtTest.Location = new System.Drawing.Point(20, 20);
            Button btnOk = new Button();
            btnOk.Size = new System.Drawing.Size(80, 25);
            btnOk.Location = new System.Drawing.Point(140, 20);
            btnOk.Text = "Click Me";
            btnOk.Click += new System.EventHandler(btnClick);
            this.Controls.Add(txtTest);
            this.Controls.Add(btnOk);
        }
        void btnClick(object sender, System.EventArgs e)
        {
            string userInput;
```

```
        userInput = txtTest.Text;
        MessageBox.Show(userInput);
    }
}
```

运行结果如图 1-33 所示。

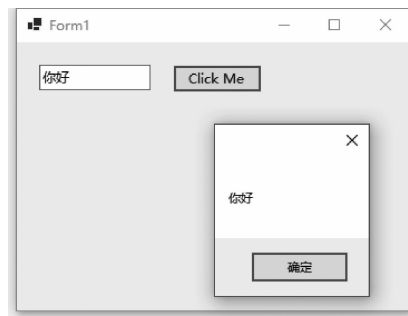


图 1-33 运行结果

实验 1-3 编写一个应用程序，输入一个圆的半径，打印该圆的直径、周长和面积，使用下面的公式： R 为半径，直径等于 $2R$ ，周长等于 $2\pi R$ ，面积等于 πR^2 。

实验 1-4 编写 1 个应用程序，输入 1 个五位数，把这 5 个数分成单独的阿拉伯数字，并且把这 5 个数字用 4 个空格彼此分开逆序输出，例如，用户输入 31245，则输出 5 4 2 1 3。