# 项目1 创建 VB.NET 程序开发环境

得益于广阔应用场景及巨大市场需求的有力支撑,2022年,我国软件和信息技术服务 业(以下简称软件业)运行稳步向好,软件业务收入跃上10万亿元台阶,盈利能力保持稳定, 软件业务出口保持增长。

本项目介绍 VB.NET 应用程序开发环境及简单程序设计流程,以此来介绍 VB 开发 Windows 应用程序开发技术的结构、特点和开发流程。让读者对 Windows 应用程序开发 有一个直观的认识,为本教材后续介绍的 Windows 应用程序开发打下基础。

## 知识目标:

- (1) 了解.NET 平台的基本结构;
- (2) 了解 Windows 应用开发技术的原理;
- (3) 掌握 VB.NET 简单语句的结构;
- (4) 掌握 VB.NET 程序调试的流程。

# 能力目标:

- (1) 掌握安装 Visual Studio 2022 的方法;
- (2) 掌握创建 VB.NET 应用程序的方法;
- (3)掌握简单 VB.NET 应用程序的设计流程;
- (4) 掌握简单 VB.NET 控件的使用方法。

#### 课程思政目标:

- (1)引导学生树立正确的世界观、人生观、价值观,自觉践行社会主义核心价值观;
- (2)培养学生发现问题、分析问题和解决问题的能力;
- (3)明确软件开发专业人员工作性质的社会价值。

# 任务 1.1 创建 VB.NET 程序开发环境

# 1.1.1 安装 Visual Studio 2022 编程环境

Visual Studio 2022 能够开发的程序包括常见的 Visual Basic、Visual C#、Visual C++ 和 Python 等。Visual Basic 开发是 Visual Studio 2022 一个重要的组成部分。

微软于 2021 年发布 Visual Studio 2022 的首个预览版。2022 版本更快、更易于使用、 更具有轻量级特点,专为学习者和构建工业规模解决方案的人员设计。

#### 1. Visual Studio 2022 编程环境可以开发的常用项目类型

(1) 桌面应用程序:可以使用 Visual Studio 2022 开发 Windows 桌面应用程序,如 Windows Forms 应用程序、WPF 应用程序和 Universal Windows Platform (UWP) 应用程序。

(2) Web 应用程序:可以使用 Visual Studio 2022 开发 Web 应用程序,如 ASP.NET 应 用程序、ASP.NET Core 应用程序和 Node.js 应用程序。

(3)移动应用程序:可以使用 Visual Studio 2022 开发移动应用程序,如 Android 应用程序、iOS 应用程序和 Windows Phone 应用程序。

(4) 云应用程序:可以使用 Visual Studio 2022 开发云应用程序,如 Azure 应用程序和 AWS 应用程序。

(5) 游戏开发程序:可以使用 Visual Studio 2022 开发游戏,如 Unity 游戏和游戏插件。

(6)数据库应用程序:可以使用 Visual Studio 2022 开发数据库应用程序,如 SQL Server 应用程序和 MySQL 应用程序。

#### 2. 安装 Visual Studio 编程环境

Visual Studio 2022 最低系统要求如表 1-1 所示。

类 别	说明
	在以下 64 位操作系统上支持 Visual Studio 2022:
	(1) Windows 11 版本 21H2 或更高版本,如家庭版、专业版、专业教育版、专业工作站
	版、企业版和教育版。
	(2) Windows 10 版本 1909 或更高版本,如家庭版、专业版、教育版和企业版。
古共的晶化系统	( 3 ) Windows Server Core $2022_{\circ}$
又时的採作矛犯	(4) Windows Server Core $2019_{\circ}$
	(5)Windows Server 核心 2016。
	(6) Windows Server 2022 标准和数据中心。
	(7) Windows Server 2019 标准和数据中心。
	(8) Windows Server 2016 标准和数据中心
	(1)建议使用四核或更好的处理器,如 ARM64 或 x64 处理器。不支持 ARM 32 处
	理器。
	(2)内存至少4GB。许多因素都会影响所使用的资源。对于典型的专业解决方案,建
石田 /开	议使用 16GB 内存。
硬件	(3)硬盘要有 850MB~210GB 可用空间,具体取决于安装的功能;典型安装需要
	20~50GB的可用空间。建议在固态硬盘上安装 Windows 和 Visual Studio 以提高性能。
	(4) 支持最低显示分辨率为 WXGA(1366 像素×768 像素)的显卡。Visual Studio 最
	适宜的分辨率为 1920 像素×1080 像素或更高。

表 1-1 Visual Studio 2022 安装所需系统

以下是 Visual Studio Community 2022 版本安装步骤。



- (1) 启动安装程序,如图 1-1 所示。
- (2)单击"继续"按钮,进入如图 1-2 所示的界面,显示下载及安装进度。
- (3)下载安装完成以后,进入如图 1-3 所示的界面。
- (4)选择安装的选项,如图 1-4 所示。

境的安 装过程

2



# 图 1-1 启动安装程序界面

#### 图 1-2 显示下载及安装进度



#### 图 1-3 安装选项选择界面

在选择安装项时需要帮助? 详细信息			>	× 安装详细信息
Veb 和云 (4)	17			<ul> <li>.NET 桌面开发</li> <li>こ包含     <li>✓ .NET 桌面开发工具     </li> </li></ul>
		Azure From 用心理 From NetT an NET Framework 开发云应用和创建变深 的 Azure SDK、工具和项目。还包含用于实现应用程序管		<ul> <li>✓ .NET Framework 4.7.2 开发工具</li> <li>✓ C≠和 Visual Basic</li> <li>可透</li> <li>○ 用于.NET的开发工具</li> <li>② NET Framework 4.8 开发工具</li> </ul>
Python 开发           对 Python 进行编辑、请试、交互式开发和语代码管理。		Node js 开发 使用 Node js (一个由异步事件驱动的 JavaScript 运行时)生 成可编版的网络应用程序。		<ul> <li>Z Entity Framework 6 工具</li> <li>MET 分析工具</li> <li>IntelliCode</li> <li>实时间试器</li> <li>Uve Share</li> </ul>
『面应用和移动应用 (5)				MLNET Model Builder
INET Multi-platform App UI 开发 使用 Ce 和 NET MAUI 从单个基本代码库生成 Android、 IOS、Windows 和 Mac 应用。		INET 桌籤开発 H C4、Visual Basic 和 F4 ら NET 和 NET Framework 一起 使用、生成 WFF、Windows 窗体和控制台应用程序。		■ BinHo Notadia Stabilio F F 桌面電気内 PreEmptive Protection - Dotfuscator NET Framework 4.62-4.7.1 开发工具 NET 可修理用目标包 Windows Communication Foundation
置 \Program Files\Microsoft Visual Studio\2022\Community 更改				移除(E)不受支持的组件

图 1-4 选择需要安装的模块



(5)单击"安装"按钮,进入如图 1-5 所示的界面,显示下载和安装进度。

			<b>平台</b> 人员 新闻
Visual Studio Community 2022		暂停(P)	开及入员和国
正在下载并验证: 650 MB/2.76 GB	(11 MB/秒)		无法下载内容,因为网络存在问题。
正在安装句:106/525			1. CO(15)
5%			
Microsoft.VisualStudio.Debugger.Concord.Managed			
发行说明			

图 1-5 下载和安装界面

(6)下载和安装需要一段时间。安装完成后,会出现如图 1-6 所示的界面,选择是否重启操作系统。

需要重启		
成功了! 但还有一步,请在启动 Visual Studio 前重启计算机。		
	重启( <u>R</u> )	以后再说(N)

图 1-6 确定是否重启系统

(7)重启操作系统之后,重新打开 Visual Studio,进入如图 1-7 所示的界面。



图 1-7 开发设置界面

(8)从中选择"开发设置""颜色主题",进入如图 1-8 所示的界面。

Visual Studio 2022		- 0	×
打开最近使用的内容( <u>R</u> )	开始使	使用	
使用 Visual Studio 时,你打开的任何项目、文件夹成文件都将显示在此处供你快速访问。 可固定任何频繁打开的对象,使其始终位于列表页部。	↓	<b>克隆存储库(<u>C</u>)</b> 从 GitHub 或 Azure DevOps 等联机存储库获取 代码	
	ď	<b>打开项目或解决方案(P)</b> 打开本地 Visual Studio 项目或.sln 文件	
	f	打开本地文件夹(E) 导航和编辑任何文件夹中的代码	
	御	<b>创建新项目(N)</b> 选择具有代码基架的项目模板以开始	
		继续但无须代码(W) →	

图 1-8 启动后的 Visual Studio

# 1.1.2 熟悉 Visual Studio 2022 编程环境

Visual Studio 2022 将程序开发中用到的各种功能集成在一个公共的工作环境中,称为 IDE。在该编程开发环境中提供了各种控件、窗口和方法,用户可以方便地进行各种应用 程序的开发,以及在各种开发界面之间切换,可以在很大程度上节约开发时间。

#### 1. Visual Studio 2022 创建 VB.NET Windows 应用程序

(1) 启动 Visual Studio 2022, 初始界面如图 1-9 所示。

Visual Studio 2022	- 🗆 X
打开最近使用的内容(B)	开始使用
<ul> <li>22素最近使用的项目(Alt+S)(S)</li> <li>P・</li> <li>P・</li> </ul>	→ <u>克隆存储库(C)</u> 从 GitHub 或 Azure DevOps 等联机存储库获取 代码
▶ 更早	打开项目或解决方案(P) 打开本地 Visual Studio 项目或.sln 文件
	行开本地文件夹(E) 导航和编辑任何文件夹中的代码
	(引建新项目(N)) 造線具有代码基架的项目機板以开始
	建荣但无须代码(W)→

#### 图 1-9 Visual Studio 2022 启动后的初始界面

(2)选择"创建新项目",然后进入如图 1-10 所示的创建新项目的界面。





图 1-10 创建 VB.NET Windows 应用程序

(3)在"所有语言"下拉列表中选择 Visual Basic,在"所有项目类型"下拉列表中选择 "桌面",然后在出现的可以创建项目的模板里选择"Windows 窗体应用",如图 1-11 所示。

创建新项目		- ロ × 復変模版(Alt+5)(5) タ・ 会感素強(r)
最近使用的项目模板(R)		Visual Basic - 所有平台(2) - 桌面 -
🗂 Windows 窗体应用	C#	Windows 窗体並用 用于部建、NET Windows 窗体(WinForms)应用的项目模板。
■ 控制台应用	C#	Visual Basic Windows 桌面
		■ 第 WPF 幾席 用于创建目标为 .NET WPF 应用程序的拠虑的项目
		Visual Basic Windows 桌面 库
		■ MVF应用程序 用于创建、NET WPF应用程序的项目
		Visual Basic Windows 桌面
		♥■ WPF 用户控件库 ■ 用于为、NET WPF 应用程序创建用户控件库的项目
		Visual Basic Windows 桌面 库
		WFF 自定义按件库 用于为、NET WPF 应用程序创建自定义按件库的项目
		Visual Basic Windows 桌面 库
		₩₩ WPF 应用(.NET Framework)
		上一步(图) 下一步(图)

图 1-11 选择创建"Windows 窗体应用"模板

(4) 单击"下一步"按钮,进入如图 1-12 所示的配置新项目的界面。

(5)设置好项目名称,选择项目的保存位置,设置好解决方案名称,然后单击"下一步"按钮,进入如图 1-13 所示的界面选择框架版本。



项目1 创建 VB.NET 程序开发环境

		-		
配置新项目				
Windows 窗体应用 Visual Basic Windows 桌面				
须目名称(J)				
1-3-1				
位置(L)				
D:\2023年上半年\2023年5月8日清华大学出版社改版2023年\VB.NET源代码\项目1用户登录界面\ *	2**			
₩決方案名称(M) ①				
1-3-1				
将解决方案和项目放在同一目录中(D)				
商日 彼左" Pui 2022年 ト半年 2022年5日9日達火十学出版社造版 2022年 VID NIET 海伊政 16日1日占登寻				
※ロ h11 いたのこうナディーにのこうナジョンロ /月ナノテエルK11 K01 K2023キ / VD. NET /新 いら(※日 /円/・豆※ 界面/1-3-1/1-3-1/『中创建				
	⊢(₽)	Τ-	_++(N)	
	T 20	- P	200	

图 1-12 配置新项目

其他信息		-	
Windows 窗体应用 Visual Basic Windows 桌面			
框架 [] ①			
.NET 6.0 (长期支持)	*		
		上	₽(C)

图 1-13 选择框架版本

文件日 編編E 税限の GitG 项目の 生成B 調成D 格式の 熟试S 注意 (Ctrl) 1-3-1 - ロ ×         分析(N) 工具口 扩展(X) 窗口(M) 税助(H)         ● ● 御・宮 四 り・ ● Debug • Any CPU • ▶ 1-3-1 • ▶ ● ● 師 。 。 白 Live Share 皮 雪気         ● ● 御・宮 四 り・ ● Debug • Any CPU • ▶ 1-3-1 • ▶ ● ● 師 。 。 白 Live Share 皮 雪気         ● ● 御・宮 四 り・ ● Debug • Any CPU • ▶ 1-3-1 • ▶ ● ● 師 。 。 白 Live Share 皮 雪気         ● ● 御・宮 四 り・ ● Debug • Any CPU • ▶ 1-3-1 • ▶ ● ● 師 。 。 白 Live Share 皮 雪気         ● ● 御・宮 四 り・ ● Debug • Any CPU • ▶ 1-3-1 • ▷ ● ● 師 。 。 白 Live Share 皮 雪気         ● ● 御・宮 四 り・ ● Debug • Any CPU • ▶ 1-3-1 • ▷ ● ● 師 。 。 白 Live Share 皮 雪気         ● ● 御・宮 四 り・ ● Debug • Any CPU • ▶ 1-3-1 • ▷ ● ● 師 。 。 白 小 ● • ○ 白 小 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●				
分析(N) 工具① 扩展(X) 室口(M) 報助(H)         ● ● 御・留 回 り・マ Debug Any CPU ● 1-3-1 * ● ● ● ● ****       企 tive Share 尼 留 四         ● ● 御・留 回 り・マ Debug Any CPU ● 1-3-1 * ● ● ● ● ****       企 tive Share 尼 留 四         ● ● 御・留 回 り・マ Debug Any CPU ● 1-3-1 * ● ● ● ● ****       企 tive Share 尼 回 四         ● ● 御・留 回 り・マ Debug Any CPU ● 1-3-1 * ● ● ● ● ****       企 tive Share 尼 回 ● ****         ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	00	文件(E) 编辑(E) 视图(V) Git(C	j) 项目(P) 生成(B) 调试(D) 格式(O) 测试(S) 搜索	(Ctrl 🖓 1-3-1 — 🗆 🗡
●・● 御・留 回 り・・・ Debug • Any CPU • 1-3-1 • ● の 時 :: ① Live Share 反 強調         ● ● 御・留 回 り・・・ Debug • Any CPU • 1-3-1 • ● の 時 :: ② Live Share 反 強調         ● ● 御・留 回 り・・・ Debug • Any CPU • 1-3-1 • ● の 時 :: ② Live Share 反 強調         ● ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○		分析(N) 工具(I) 扩展(X) 窗口	(W) 帮助(日)	
・ 取録         ・ 単本は       ● cm1.vb [L9:H]* ③ × ● Φ         ・ Φ          ・ 単本は          ・ 単本は          ・ ● cm1.vb [L9:H]* ③ × ● Φ         ・ ● cm1.vb [L9:H]* ④ × ● Φ         ・ ● cm1.vb [L9:H]* ⊕ F         ・ ● VB Form1.besigner.vb         ・ ● Form1.vb [L9:H]* ⊕ F         ・ ● Cm1.vb [L9:H]* ⊕ F         ・ ● Form1.vb [L9:H]* ⊕ F         ● Cm1.vb [L9:H]* ⊕ F         ● Form1.vb [L9:H]* ⊕ F         ● Fo		- • • • • • • • • • • •	- Debug - Any CPU - ▶ 1-3-1 - ▷ ④ - □	ā Ț Ţ ld Live Share 👂 管理员
<ul> <li>              登室工具相</li></ul>	数据	工具箱 → ┦ X	Form1.vb [设计]*   ⇒   ×	■ 解决方案资源管理器
ト All Windows Forms       ● 公共登件         ● 容器       ● 容器         ● 容器       ● 案報本和工具性         ● 追相       ● 打印         ● 方面       ● 方面         ● 対応値       ● 改振         ● 数据       ● 必托         ● 数据       ● 必括         ● 対応値       ● び 後載頭         ● 数据       ● び 後載頭         ● 対応値       ● び 後載頭         ● 数据       ● び 後載頭         ● 成素頭       ● び 後載頭         ● 数       ● び 後載頭         ● 「日 1,-26       ● 101,-26	三	搜索工具箱 ・ ター		ⓒ ⊘ ☆ ┩ ఀ⊙ - ≒ 🗇 吨 - "
> 公共控件         > 容器         > 算業和工具だ         > 組件         > 打印         > 対活框         > 数据         * 常規         此組中没有可用的控件,将某项施 至此文本可将具添加到工具编。         輸出         ・ 中 大         指规列表 程序包管理器控制台 输出         学 101,-26         11       个 添加到保存者 ( 回 法经合案 ) ①		▷ All Windows Forms	📲 Form1 📃 🗷	搜索解決方案资源管理器(Ctrl+;) ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
<ul> <li>&gt; 容器</li> <li>&gt; 菜単和工具栏</li> <li>&gt; 組件</li> <li>&gt; 対活框</li> <li>&gt; 数据</li> <li>- 常規</li> <li>- 単面 1.3-1</li> <li>- 粉 VB Poject</li> <li>&gt; 47 依拠項</li> <li>- * 26 依拠項</li> <li>- * 26 体拠項</li> <li>- * 4 ×</li> <li>- * 4 ×</li> <li>- * 5 × 5</li> <li>- * 5 × 5</li> <li>- * 4 ×</li> <li>- * 5 × 5</li> <li>-</li></ul>		▶ 公共控件		😡 解决方案 '1-3-1' (1 个项目, 共 1 个) 🔺
<ul> <li>&gt; 菜単和工具栏</li> <li>&gt; 組件</li> <li>&gt; 招付</li> <li>&gt; 対活框</li> <li>&gt; 数据</li> <li>- 数据</li> <li>- 数据</li> <li>- 常規</li> <li>- 数据</li> <li>- 常規</li> <li>- 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一</li></ul>		▷ 容器		▲ VB 1-3-1
> 組件         > 打印         > 対流症         > 数据         > 数据         / 常规         此組中没有可用的拉件。将某项指 至此文本可将具添加到工具箱。         輸出         ・ 中、、         借具別表 程序包管理器控制台 輸出         〇         101,-26         111         * 法         * 法         * 法         * 101,-26		▶ 菜单和工具栏		💭 My Project
▶ 打印         ▶ 对话框         ▶ 数据         ▲ 常规         此组中没有可用的拉件、将某项施         室此文本可将具添加到工具编。         輸出         ◆ 取         備限列速         但序包管理器控制台 输出         〇 101,-26         1 1		▶组件		▶ 8日 依赖项
> 対弦性         > 数据         / 常規         / 常規         此组中没有可用的拉件。将某项指 至此文本可将具添加到工具箱。         輸出         輸出         輸出         ・ 中 x、         輸出         輸出         ・ 中 x、		▶ FJED	P P	♦ VB ApplicationEvents.vb
▶ 改強 ▲ 常規 此組中没有可用的控件、将莫项拖 至此文本可将其添加到工具箱。     ▶ VB Form1,Designer,vb 前 Form1,rex       輸出     ・ 中 × 輸出       輸出     ・ 中 × 輸出       輸出     ・ 中 × ・ 指误列表 程序包管理器控制台 輸出       〇     101,-26				▲ I Form1.vb
				VB Form1.Designer.vb
此组中没有可用的控件。将某项施 至此文本可将其添加到工具箱。 輸出 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		4 带死		解决方案资源管理器 Git 更改
Form1 System.Windows.Forms.Form     -       輸出     ・     中、、、       輸出     ・     中、、、       調量・     ・     ・       出現別表 程序包管理器控制台 輸出     一     ご       「101,-26     通1x1     个 添加到源代码管理 へ 回 选择合库 へ い 。		此组中没有可用的控件。将某项拖 至此文本可將其添加到丁目第		属性 <b>▼</b> ₽ ×
		THE CT. PLAN COMPANY OF THE		Form1 System.Windows.Forms.Form
世報報告報告報告報告報告報告報告報告報告報告報告報告報告報告報告報告報告報告報			· 输出 - ↓ ×	III 💱 🖗 🗲 🎾
Text         与控件关联的文本。           错误列表 程序包管理器控制台 输出         小 法加到源代码告理 本 团 选择仓库 + ①1 //2 //2 //2 //2 //2 //2 //2 //2 //2 //				, UseWaitCursor False 🚊
<				Text 与按件关联的文本。
🗊 📋 101,-26 🔄 1 x 1 🕆 添加頭腺代码管理 🗕 団 选择合库 🔺 🗘 🚪			错误列表 程序包管理器控制台 输出	27711222002211
	Ţ	101 , -26	<u></u>	添加到源代码管理 🔺 🔟 选择仓库 🔺 🖓 👘

(6) 单击"创建"按钮,将完成项目的创建,进入如图 1-14 所示的界面。

图 1-14 新建的 VB.NET Windows 应用程序





#### 图 1-15 "文件"菜单的下拉菜单项

编辑(E) 视图(V) Git(G) 项目(P)

转到(G)

▶ All Win
 ▷ 公共控
 ▷ 容器
 ▷ 菜単和:
 ◇ 類切(T)
 ▷ 344
 ○ 复制(C)

り 撤销(U)

□ 粘贴(P)

× 删除(D)

嗡 全选(A)

高级(V)

书签(K)

图 1-16 "编辑"菜单的下拉菜单项

搜索工具新 (~ 重做(R)

查找和替换(F)

生成

.

Þ

Ctrl+Z

Del

1

I

文件(F)

- 🕣 🖞

L具箱

> 打印

> 对话框

数据

常规

此组中没

此文2

2. Visual Studio 2022 的开发环境的组成

Visual Studio 2022 的开发环境主要由菜单、 工具栏、窗体、工具箱、属性窗口和解决方案资 源管理器等几部分组成。

(1)菜单。菜单包括"文件""编辑""视图" 等,其中包含了开发 VB.NET 程序常见的命令。

 "文件"菜单如图 1-15 所示,它包含的主 要菜单项说明如下。

新建:支持新建项目、仓库、文件等。

打开:支持打开已有的项目/解决方案、文件 夹、网站、文件等。

关闭:关闭正在编写的项目。

关闭解决方案:关闭正在编写的解决方案。

退出:退出 Visual Studio 2022 编程环境。

② "编辑"菜单包含的主要菜单项有"转到""查找和替换" "撤销" "重做" "剪切" "复制" "粘贴" 等,如图 1-16 所示。

庸(E) 视图(V) Git(G) 项目(P) 生成(B) 调试(D) 格式(O)

F7

测

-1

③ "视图"菜单包含的主要菜单项有"代码""设计器""解决方案资源管理器""服务器资源管理器""类视图""工具箱"等,如图 1-17 所示。具体说明如下。

	<u></u>	反口留(D)	Shiit+F7
	$\mathcal{C}$	打开(O)	
		打开方式(N)	
/s Fori		解决方案资源管理器(P)	Ctrl+Alt+L
	$\overline{\mathbb{Q}}$	Git 更改(G)	Ctrl+0, Ctrl+G
±	2	Git 存储库(S)	Ctrl+0, Ctrl+R
	RR	团队资源管理器(M)	Ctrl+∖, Ctrl+M
		服务器资源管理器(V)	Ctrl+Alt+S
	80	SQL Server 对象资源管理器(S)	Ctrl+ Ctrl+S
	Ē	测试资源管理器(T)	Ctrl+E, T
	司	书签窗囗(Q)	Ctrl+K, Ctrl+W
用的控	\$	调用层次结构(H)	Ctrl+Alt+K
5 <del>34</del> /89/	Ôŝ	类视图(A)	Ctrl+Shift+C
	$\langle \cdot \rangle$	代码定义窗口(D)	Ctrl+ D
	1	对象浏览器(J)	Ctrl+Alt+J
	Ĝ	错误列表(I)	Ctrl+∖, E
	₿	输出(O)	Ctrl+Alt+O
	Ê	任务列表(K)	Ctrl+∖, T
	÷	工具箱(X)	Ctrl+Alt+X
	$\square$	通知(N)	Ctrl+ Ctrl+N
	>	终端	Ctrl+`
		其他窗口(E)	
		工具栏(T)	
	K 3	全屏幕(U)	Shift+Alt+Enter
	ø	所有窗囗(L)	Shift+Alt+M
	8	Tab 键顺序(B)	
		向后导航(B)	Ctrl+-
		向前导航(F)	Ctrl+Shift+-
		下一个任务(X)	

图 1-17 "视图"菜单的下拉菜单项

代码:打开代码编辑界面。

设计器:打开设计器编辑界面。

解决方案资源管理器:打开解决方案资源管理器窗口。

服务器资源管理器:打开和服务器以及数据库相关内容的操作界面。

类视图:打开类视图窗口。

工具箱:打开工具箱窗口。

④ Git 菜单包含的菜单项有"克隆""创建 Git 仓库""本地存储 库""设置",如图 1-18 所示。

⑤"项目"菜单包含的主要菜单项有"添加窗体(Windows 窗体)""添加用户控件(Windows 窗体)""添加类""设为启动项目"等, 如图 1-19 所示。具体说明如下。

添加窗体 (Windows 窗体): 向已有的应用程序中添加新的

Oiti	(0)	项目(P)	土小K(B)	ų	可にい
÷	克陶	肇(L)			
۰	创建	畫 Git 仓库(	G)		
	本地	也存储库(L)		۲	L
	提到	を或存储(C)			
ŝĝ	设计	髶(E)			1
	÷] *	<ul> <li>▲ 克鼎</li> <li>◆ 创级</li> <li>本加</li> <li>提び</li> <li>役2</li> <li>役3</li> <li>役4</li> </ul>	<ul> <li>▲ 克隆(L)</li> <li>◆ 创建 Git 仓库( 本地存储库(L) 提交或存储(C)</li> <li>◆ 设置(E)</li> </ul>	<ul> <li>▲ 丙隆(L)</li> <li>◆ 创建 Git 仓库(G)</li> <li>本地存储库(L)</li> <li>提交或存储(C)</li> <li>磣 设置(E)</li> </ul>	<ul> <li>→ 京康()</li> <li>→ 京康(L)</li> <li>◆ 创建 Git 仓库(G)</li> <li>本地存储库(L)</li> <li>▶</li> <li>提交或存储(C)</li> <li>愛 设置(E)</li> </ul>

图 1-18 Git 菜单的下 拉菜单项

窗体。

添加用户控件(Windows 窗体):添加用户自定义的控件。

添加类:添加 VB.NET 的类。

设为启动项目:将正在编辑的项目设为启动项。

⑥ "调试" 菜单包含的主要菜单项有"开始调试""开始执行(不调试)"等,如图 1-20 所示。具体说明如下。

j) 项	∃( <u>P</u> )	生成( <u>B</u> )	调试( <u>D</u> )	格式( <u>O</u> )	测试( <u>S</u> )	分析( <u>N</u>			
- 13	添加	窗体(Wind	ows 窗体)(	E)					
_ to	添加	用户控件(V	Vindows 窗	h体)( <u>U</u> )					
10	添加组件( <u>N</u> )								
習	添加	模块( <u>M</u> )							
tig	添加	)类( <u>C</u> )							
+0	添加	新数据源(上	<u>4</u> )						
ĉ	添加	新项( <u>W</u> )			Ctrl+S	hift+A			
t0	添加	现有项( <u>G</u> ).			Shift+	Alt+A			
¢	编辑	项目文件							
۰.	新建	文件夹( <u>D</u> )							
	从项	目中排除(」	)						
ø	显示	所有文件(0	<u>2</u> )						
	添加	项目引用(E	3)						
	添加	共享项目引	l用( <u>S</u> )						
	添加	COM 引用	l( <u>C</u> )						
	连接	服务( <u>C</u> )				+			
*	配置	启动项目…							
	设为启动项目( <u>A</u> )								
	导出	模板( <u>E</u> )							
6	管理	NuGet 程	序包( <u>N</u> )						
¢	1-3-	1 属性(P)							



调词	载(D)	格式(O)	测试(S)	分析(N)	工具	
	窗口	(W)		+		
►	开始调试(S) 开始执行(不调试)(H) 应用代码更改(A)			F5 Ctrl+F5 Alt+F10		
$\triangleright$						
G						
25	性能探查器(F)			Alt+F2		
25	重启	性能探查器				
°o	附加到进程(P)			Ctrl+Alt+P		
°0	重新附加到进程			Shift+Alt+P		
	其他调试目标(H)				÷	
$\stackrel{\downarrow}{\bullet}$	逐语	句(S)		F11		
ି	逐过程(O)			F10		
	切换	断点(G)		F9		
	新建	断点(B)			•	
č	删除	所有断点(D	)	Ctrl+Shift	+F9	
÷	选项(O)					
x	1-3-	1 调试属性				

图 1-20 "调试"菜单的下拉菜单项

开始调试:启动正在编辑的项目进行调试。

开始执行(不调试):启动正在运行的项目。

(2)工具栏。工具栏在菜单栏的下面,如图 1-21 所示。工具栏提供了常用命令的快速访问按钮,单击某个按钮,可执行对应的操作,效果和使用菜单是一样的。

🔄 🔄 - 🕘 词 - 🚅 💾 🗐 🤌 - 🖓 - 🖓 - Debug - Any CPU - ▶ 1-3-1 - ▶ 🐠 - 閘 🖓 - 👘 🖓 - 👘 + 日 = = 日 + 山 | 米 王 🖾 🔍 | 帅 용 📑 😜

图 1-21 工具栏的界面



(3)窗体。在创建了一个 Windows 应用程序后,系统会自动生成一个默认的窗体, 也就是应用程序界面。在开发过程中,用户编程使用的各种控件就是布局在窗体之上的, 当程序运行时,用户所看到的就是窗体。应用程序的设计界面及各功能窗口如图 1-22 所示。



图 1-22 应用程序的设计界面及各功能窗口

(4)工具箱。工具箱中提供了各种控件、容器、菜单和工具栏、组件、对话框和数据等。 在默认情况下,工具箱将控件和各种组件按照功能的不同进行了分类,如图 1-23 所示。

用户在编程过程中可以根据需要选择各种控件和组件。如果所需要的控件或者组件在 工具箱中找不到,可以在工具箱区域内右击,在出现的快捷菜单中选择"选择项"命令, 打开"选择工具箱项"对话框,在该对话框中选择相应的控件或组件,如图 1-24 所示。



图 1-23 工具箱

图 1-24 "选择工具箱项"对话框

