

 普通高等教育计算机规划教材

Visual Basic.NET

程序设计教程

上机指导及习题解答

第 2 版

崔淼 刘瑞新 主编



机械工业出版社
CHINA MACHINE PRESS



013048088

TP312BA-42
57-2

普通高等教育计算机规划教材

Visual Basic .NET 程序设计教程

上机指导及习题解答

第2版

崔淼 刘瑞新 主编



TP312BA-42

57-2



机械工业出版社



北航

C1655415

880830810

本书既是《Visual Basic.NET 程序设计教程第2版》的配套教材,也可单独作为实例教程或者其他 Visual Basic 程序设计教程的配套书使用。本书除了对原教材中所有习题做了详细解答以外,还对每章内容增加了比较实用的上机实训项目。本书在设计程序时,将面向对象的思想贯穿于所有习题和实训项目中,使用“任务驱动”的方式,强调使用面向对象的程序设计方法实现程序功能,强调程序功能由类及其属性、方法等实现。本书中每章的上机实训项目均由“实训目的”、“实训要求”和“实训步骤”三部分组成,通过实例对本章重点进行了概括总结。对于原教材中没有详细介绍过的一些概念和使用技巧,在本书中利用习题的方式从实用的角度出发进行了补充介绍。本书以 Microsoft Visual Studio 2008 为开发平台,兼容 Microsoft Visual Studio 2005/2010。

本书既适合作为高等院校 Visual Basic 程序设计教材的练习册,也可作为广大计算机爱好者和各类 Visual Basic 程序设计培训班的教学辅导用书。

图书在版编目(CIP)数据

Visual Basic .NET 程序设计教程上机指导及习题解答 / 崔淼, 刘瑞新主编.
—2版. —北京: 机械工业出版社, 2013.3

普通高等教育计算机规划教材

ISBN 978-7-111-41865-8

I. ①V… II. ①崔… ②刘… III. ①BASIC 语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 053724 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 郝建伟 王 凯

责任印制: 张 楠

中国农业出版社印刷厂印刷

2013 年 6 月第 2 版·第 1 次印刷

184mm×260mm·15.25 印张·376 千字

0001—3000 册

标准书号: ISBN 978-7-111-41865-8

定价: 32.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

教材网: <http://www.cmpedu.com>

销售一部: (010) 68326294

机工官网: <http://www.cmpbook.com>

销售二部: (010) 88379649

机工官博: <http://weibo.com/cmp1952>

读者购书热线: (010) 88379203

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

- 1) 反映计算机技术领域的新发展和新应用。
- 2) 为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、多媒体光盘、课程设计和毕业设计指导等内容。
- 3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。
- 4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。
- 5) 注重教材的实用性、通用性，适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

许多人在刚开始学习程序设计类课程时,通常会出现“上课听得懂,下课不会做”的现象。为此我们编写了《Visual Basic.NET 程序设计教程(第2版)》的配套教材,全书共分为13章,主要包括 Visual Basic 语法基础,流程控制语句与控件,面向对象的程序设计方法,接口、事件、委托和泛型,异常处理和文件操作,数据库操作,LINQ to SQL 以及使用 Microsoft Excel 实现数据导入及报表输出等内容。此外,本书还专门增加了一章用来介绍三层架构应用程序开发的相关知识和程序设计方法,以满足读者逐步过渡到大中型应用系统开发的需要。

本书除了对原教材中所有习题做了详细的解答以外,还在每章内容中增加了比较实用的上机实训项目。每章的上机实训项目均由“实训目的”、“实训要求”和“实训步骤”三部分组成,通过实例对本章重点进行了概括总结。对于原教材中没有详细介绍过的一些概念和使用技巧,在本书中利用习题的方式从实用的角度出发进行了补充介绍。所以本书也可单独作为实例教程或者其他 Visual Basic 程序设计教程的配套书使用。本书以 Microsoft Visual Studio 2008 为开发平台,兼容 Microsoft Visual Studio 2005/2010。

本教材特色之一是将面向对象的思想贯穿于所有习题和实训项目中,在所有习题及上机实训项目中使用“任务驱动”的方式,强调使用面向对象的程序设计方法实现程序功能。强调程序功能由类及其属性、方法等实现,窗体中控件仅组成用户操作界面的“松耦合”程序设计方式(二层架构设计方式)。针对初学者的特点,全书在编排上采用由简到繁、由浅入深的循序渐进的方法,力求通俗易懂、简捷实用;为了便于读者阅读,在所有习题的原代码中均加入了大量的注释;程序设计中的操作以详尽的表述与图例相结合的方式来说明,以求读者对每一步操作都清清楚楚。对本书的重点内容:程序设计基础,面向对象的程序设计,泛型、接口、委托和事件,数据库访问技术,LINQ to SQL,以及使用 Microsoft Excel 实现数据导入及报表输出等,均通过简单易懂的实例进行了详尽叙述,突出对学生分析问题能力及解决问题能力的培养,强调面向对象的程序设计方法,避免了空洞的概念堆砌。

本书既适合作为高等院校 Visual Basic 程序设计教材的练习册,也可作为广大计算机爱好者和各类程序设计培训班的教学辅导用书。为了配合本书的教学,机械工业出版社为读者免费提供电子教案,可在机械工业出版社网站(www.cmpedu.com)上下载。

本书由崔淼、刘瑞新主编,崔淼编写第1~3章和第13章,刘瑞新编写第4章,吕振雷编写第5~6章,莫丽娟编写第7章,李建平编写第8.1章,刘克纯、翟丽娟、缪丽丽、李美嫦、孙洪玲编写第8.2章,曾赞编写第9、12章,王宁编写第10.1章,刘大学、万兆君、陈文明、万兆明编写第10.2章,张曦明编写第11.1章,孙明建、王金彪、骆秋容、崔瑛瑛、岳爱英、庄建新、张国胜、岳香菊、胡峰、丁新建、徐云林编写第11.2章,全书由刘瑞新教授统编定稿。由于作者水平有限,书中疏漏和不足之处难免,敬请广大师生指正。

编 者

目 录

出版说明

前言

第 1 章 Visual Studio 集成开发环境	1
1.1 实训 设计应用程序界面	1
1.1.1 实训目的	1
1.1.2 实训要求	1
1.1.3 实训步骤	1
1.2 习题解答	5
第 2 章 Visual Basic 语法基础	11
2.1 实训 Visual Basic 常用函数	11
2.1.1 实训目的	11
2.1.2 实训要求	11
2.1.3 实训步骤	12
2.2 习题解答	14
第 3 章 流程控制语句与控件	19
3.1 实训 流程控制语句综合练习	19
3.1.1 实训目的	19
3.1.2 实训要求	19
3.1.3 实训步骤	19
3.2 习题解答	25
第 4 章 面向对象的程序设计方法	38
4.1 实训 类的继承应用	38
4.1.1 实训目的	38
4.1.2 实训要求	38
4.1.3 实训步骤	38
4.2 习题解答	42
第 5 章 数组、结构与集合	55
5.1 实训 学生成绩查询程序	55
5.1.1 实训目的	55
5.1.2 实训要求	55
5.1.3 实训步骤	56
5.2 习题解答	58

第 6 章	接口、委托和事件	75
6.1	实训 接口的继承和事件的应用	75
6.1.1	实训目的	75
6.1.2	实训要求	75
6.1.3	实训步骤	76
6.2	习题解答	81
第 7 章	泛型	88
7.1	实训 List(Of T)泛型集合应用	88
7.1.1	实训目的	88
7.1.2	实训要求	88
7.1.3	实训步骤	89
7.2	习题解答	94
第 8 章	异常处理、程序调试和文件操作	107
8.1	实训 简单的在线测试程序	107
8.1.1	实训目的	107
8.1.2	实训要求	107
8.1.3	实训步骤	108
8.2	习题解答	115
第 9 章	数据绑定和数据访问控件	126
9.1	实训 简单数据库应用程序设计	126
9.1.1	实训目的	126
9.1.2	实训要求	126
9.1.3	实训步骤	127
9.2	习题解答	131
第 10 章	使用 ADO.NET 访问数据库	146
10.1	实训 使用 DataSet 设计一个用户管理程序	146
10.1.1	实训目的	146
10.1.2	实训要求	146
10.1.3	实训步骤	149
10.2	习题解答	163
第 11 章	使用 LINQ to SQL 访问数据库	199
11.1	实训 使用 LINQ to SQL 操作数据库	199
11.1.1	实训目的	199
11.1.2	实训要求	199
11.1.3	实训步骤	201

11.2 习题解答	206
第 12 章 数据的导入和报表输出	210
12.1 实训 使用 Excel 打印学生准考证	210
12.1.1 实训目的	210
12.1.2 实训要求	210
12.1.3 实训步骤	212
12.2 习题解答	217
第 13 章 三层架构应用程序设计	227
13.1 三层架构程序设计的概念	227
13.2 三层架构程序设计示例	228

第 1 章 Visual Studio 集成开发环境

1.1 实训 设计应用程序界面

1.1.1 实训目的

了解 Visual Studio 集成开发环境中各窗口的作用及使用方法；熟练掌握在 Visual Studio 环境中创建项目、保存文件、打开项目的方法；熟练掌握控件工具箱、窗体设计器、属性窗口的使用方法；熟练掌握使用属性窗口设置对象属性的操作过程；熟练掌握“格式”菜单或“格式”工具栏中的“对齐”、“使大小相同”、“水平间距”、“垂直间距”等控件布局功能，能够快速创建美观大方的应用程序界面。

1.1.2 实训要求

在 Visual Studio 的集成开发环境中，使用窗体设计器、控件工具箱及属性窗口创建如图 1-1 所示的应用程序界面。程序界面由排列在窗体上的 9 个标签控件、8 个文本框控件和两个按钮控件组成。各控件属性设置要求如下：



图 1-1 应用程序界面

1) 窗体不能改变大小，且没有最大化和最小化按钮，窗体标题栏中显示的文本为“编辑信息”。

2) 要求 9 个标签控件中分别显示界面标题和各文本框的输入提示信息。其中，用于显示界面标题的标签要求使用三号字、黑体。用于显示输入提示信息的 8 个标签控件要求其右侧距各自的文本框有相同的距离（各标签右侧对齐）。

3) 要求 8 个文本框中的文字对齐方式为右对齐。

4) 两个按钮控件的 Text 属性分别设置为“确定”和“取消”。

5) 为了在程序代码中方便地调用和设置各控件的属性，要求为 8 个文本框和两个按钮控件指定易于理解的名称。

1.1.3 实训步骤

1. 创建应用程序项目

启动 Visual Studio 后，在“起始页”中单击“新建项目”。然后在弹出的如图 1-2 所示的“新建项目”对话框的“项目类型”列表中选择“Visual Basic”，在“模板”列表中选择“Windows 窗体应用程序”，在对话框上方的下拉列表框中选择希望使用的 .NET Framework 版本（Visual Studio 2008 默认为 .NET Framework 3.5），在对话框下方为项目名称、解决方案名称命名，并指定相关文件的保存位置后，单击“确定”按钮完成项目创建，进入 Visual Studio 集成开发环境。

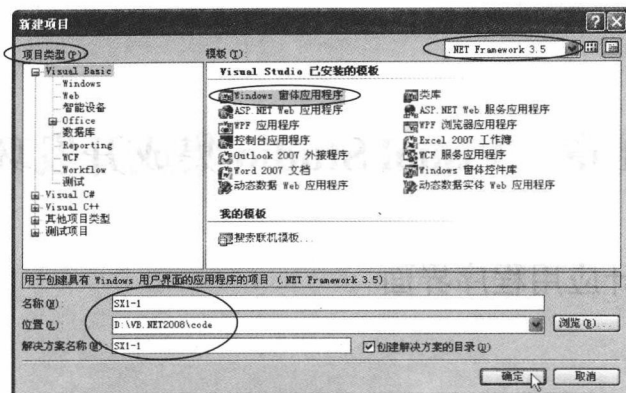


图 1-2 新建项目对话框

2. 设计程序界面

在窗体的设计图中，适当调整窗体的宽度和高度。通过控件工具箱向窗体添加两个标签控件 Label1、Label2 和 1 个文本框控件 TextBox1，并按如图 1-3 所示调整各控件的位置。

如图 1-4 所示，用鼠标拖出一个矩形框，同时选中标签 Label2 和文本框 TextBox1。当某控件被选中时，其周边将出现任务菜单按钮和用于调整大小的控制点，当多个控件被同时选中时，被选中的控件中只有一个控制点为白色，其他为黑色显示。具有白色控制点的控件称为“基准控件”或“当前控件”。此时执行“格式”菜单中“对齐”或“使大小相同”命令，则以该控件为基准。

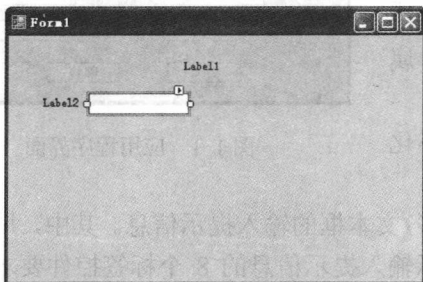


图 1-3 添加控件

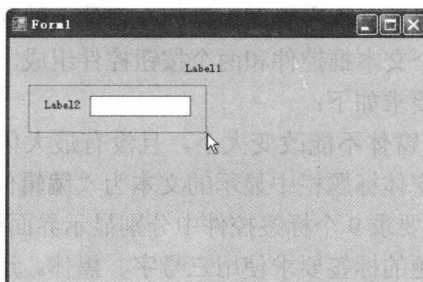


图 1-4 同时选中多个控件

鼠标指向被同时选中的标签和文本框控件，当鼠标指针变成十字箭头样式时，按下〈Ctrl〉键并拖动控件到其他位置，以达到多个控件同时复制的目的。注意，此过程中鼠标指针旁边会出现一个“+”号标记，表示当前的操作为复制操作，如图 1-5 所示。若没有按下〈Ctrl〉键，则操作为移动操作。

将上述操作重复执行 3 次，即可在窗体左侧得到 4 个标签和 4 个文本框控件。同时选中窗体左侧的 4 个标签和 4 个文本框控件，并通过〈Ctrl〉键配合下的鼠标拖动将它们复制到窗体的右侧，得到如图 1-6 所示的控件布局结果。

注意，通过鼠标拖动移动或复制多个控件时，屏幕上同样会出现位置对齐操作线，使用好这些参考线对快速创建程序界面是十分有益的。

最后，再向窗体下方的适当位置添加两个按钮控件 Button1 和 Button2，即可完成所有控件的添加、布局，效果如图 1-7 所示。

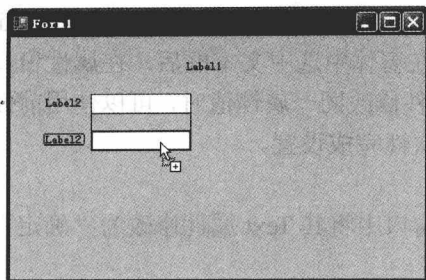


图 1-5 复制多个控件

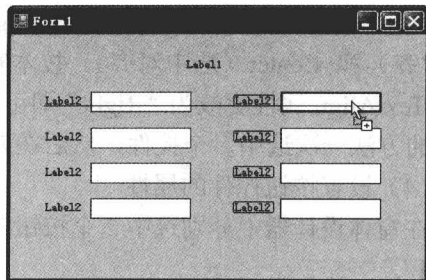


图 1-6 复制控件时的水平位置参考线

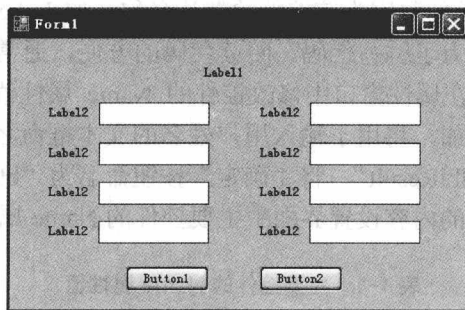


图 1-7 控件布局效果

3. 设置控件的属性

控件的属性决定着控件的外观。例如，标签中显示怎样的文字、文本框中用户输入的内容如何对齐等。控件的属性设置既可在设计时通过属性窗口完成，也可在程序运行时通过代码来完成。具体设置步骤如下：

(1) 窗体的属性设置

选中窗体，在属性窗口中设置其 `MaximizeBox` 属性和 `MinimizeBox` 属性为 `False`，设置 `MaximumSize`（最大尺寸）属性和 `MinimumSize`（最小尺寸）属性的值都等于 `Size`（当前尺寸）属性的值，实现窗体没有最大化 and 最小化按钮，也不能更改窗体的大小。将窗体的 `Text` 属性修改为“编辑信息”，该属性决定窗体的标题栏显示怎样的文本信息。

(2) 标签控件的属性设置

逐个选中用于提示文本框输入内容的 8 个标签控件，在属性窗口中设置它们的 `Text` 属性分别为“姓名”、“性别”、“年龄”、“工作单位”、“联系地址”、“邮编”、“电话”和“E-mail”。

选中用于显示界面标题的标签控件，在属性窗口找到 `Font` 属性，单击其右侧的打开对话框按钮 ，即可打开“字体”对话框，设置字体为黑体，字号为三号。在设计视图窗口中确认“填写个人信息”标签处于选中状态，单击工具栏中的“水平居中”按钮 .

分别选中窗体左侧和右侧的 4 个标签，单击布局工具栏中“右对齐”按钮  使之右侧对齐。对控件的位置进行微调时，可在选中一个或多个控件后按键盘上的上、下、左、右方向键来完成。

(3) 文本框控件的属性设置

文本框的文字对齐方式由其 `TextAlign` 属性决定，可选项有“`Left`”（左对齐，默认），`Right`（右对齐）和 `Center`（居中对齐）。按本实训的要求可在窗体中选中文本框后，在属性窗口中将其 `TextAlign` 属性更改为“`Right`”即可。多个同类控件修改同一属性值时，可以使用前面介绍过的方法一次选中多个控件，然后在属性窗口中一次性完成设置。

（4）设置按钮控件的属性

在窗体设计器中分别选中 2 个按钮控件，在属性窗口中将其 `Text` 属性修改为“确定”和“取消”。

（5）为控件命名

默认情况下，当控件被添加到窗体时，系统将自动以控件类型名（如 `TextBox`、`Button`、`Label` 等）加上一个自动增长的数字为该控件的默认名称（如 `Label1`、`Label2`、`Label3` 等）。当窗体中包含较多控件时，为了编写代码、阅读代码的方便，通常要求为控件起一个便于记忆和理解的名称，这就需要在属性窗口中修改控件的 `Name` 属性值。通常可用控件类型缩写加一个英文单词来表示。例如，将用于输入用户姓名的文本框命名为“`txtName`”，将用于输出计算结果的标签命名为“`lblResult`”，将“确定”按钮命名为“`btnOK`”等。

本实训要求按照表 1-1 的内容设置界面中主要控件的 `Name` 属性。

表 1-1 主要控件的 `Name` 属性值

控 件	<code>Name</code> 属性值
左侧文本框 <code>TextBox1~TextBox4</code>	<code>txtName</code> （姓名）、 <code>txtSex</code> （性别）、 <code>txtAge</code> （年龄）、 <code>txtUnit</code> （工作单位）
右侧文本框 <code>TextBox5~TextBox8</code>	<code>txtAdress</code> （地址）、 <code>txtPostalCode</code> （邮编）、 <code>txtTel</code> （电话）、 <code>txtEmail</code> （电子邮件）
“提交”按钮	<code>btnOK</code>
“取消”按钮	<code>btnCancel</code>

4. 试运行程序

单击工具栏中的  按钮或按 `<F5>` 键运行初步设计完毕的程序，逐一测试所有程序功能是否可以正确实现，以及程序界面是否还有需要完善的地方。

5. 保存或打开项目

程序界面设计完成后需要保存项目，退出 `Visual Studio` 环境后，如要对程序进行修改，则可以重新打开项目。

（1）保存项目文件

在 `Visual Studio` 集成开发环境中，若单击工具栏中的启动按钮  或按 `<F5>` 键运行程序，系统将自动保存所有项目文件到创建项目时指定的文件夹中。在设计过程中用户也可单击工具栏中的保存窗体文件按钮  或保存所有文件按钮  随时保存项目文件。

此外，用户在关闭 `Visual Studio` 窗口时，系统也会判断用户是否在修改了文件内容后没有执行保存操作，若没有执行保存操作，则显示如图 1-8 所示的对话框提示用户保存文件。

（2）打开项目

通常可使用以下 3 种方法打开已存在的项目：

1) 启动 `Visual Studio` 后，执行“文件”菜单中的“打开解决方案”命令，在打开的对话框中找到保存项目的文件夹，双击解决方案文件（*.sln）即可打开程序项目。

2) 启动 Visual Studio 后, 如图 1-9 所示若在“起始页”中的“最近的项目”区域中包含有要打开的项目名称, 则单击该名称也可打开对应的程序项目。

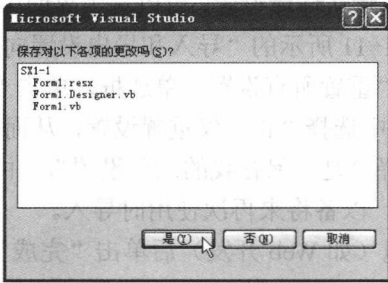


图 1-8 提示用户保存项目文件

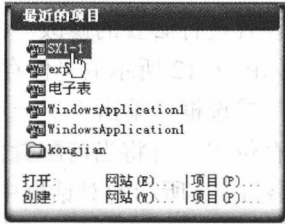


图 1-9 最近使用过的项目列表

3) 在“我的电脑”或“资源管理器”窗口中, 找到项目保存的文件夹, 双击其中扩展名为“.sln”的解决方案文件即可打开对应的程序项目。

1.2 习题解答

1. Visual Studio 集成开发环境中主要包含哪些子窗口? 其作用是什么?

解答: 如图 1-10 所示, Visual Studio 集成开发环境中除了具有菜单栏、工具栏外, 还包含了许多子窗口, 其中最常用的是“工具箱”、“解决方案资源管理器”、“属性”子窗口、输出窗口和窗体编辑区。

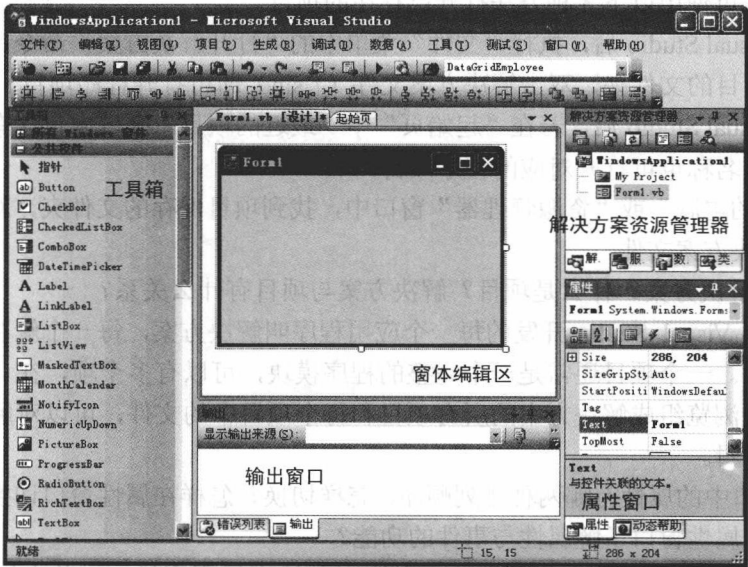


图 1-10 Visual Studio 的集成开发环境

用户的主要工作区域是窗体编辑区, 该子窗口用来显示 Windows 窗体的“设计”视图 (Form1.vb[设计]选项卡) 或双击窗体后打开的程序“代码窗口” (Form1.vb 选项卡)。更多用

于管理项目、调试程序等目的子窗口可以通过执行“视图”菜单中的相应命令将其打开。

2. 在使用 Visual Studio 的过程中，若希望改变默认环境设置，应如何操作？

解答：如果在使用 Visual Studio 的过程中想改变默认环境设置，可以在主界面中执行“工具”菜单下的“导入和导出设置”命令，通过如图 1-11 所示的“导入和导出设置向导”对话框对当前设置进行必要的修改。在对话框中，选择“重置所有设置”单选框，单击“下一步”按钮，在如图 1-12 所示的“保存当前设置”对话框中选择“否，仅重置设置，从而覆盖我的当前设置”单选框，单击“下一步”按钮。如果选择“是，保存我的当前设置”，并指定文件名称和保存位置，可将当前配置保存在指定文件中，以备将来再次使用时导入。

在如图 1-13 所示的对话框中选择新的设置环境（如 Web 开发）后单击“完成”按钮。

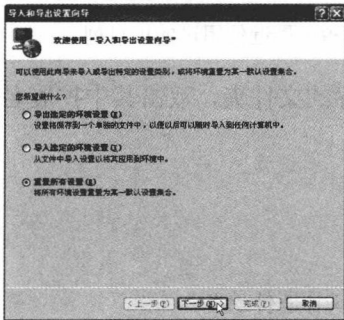


图 1-11 重置所有设置

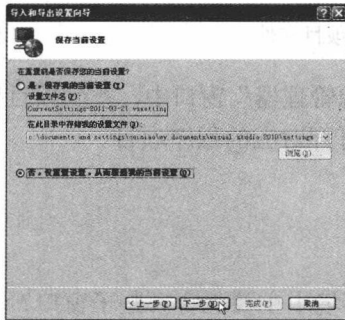


图 1-12 是否保存当前设置

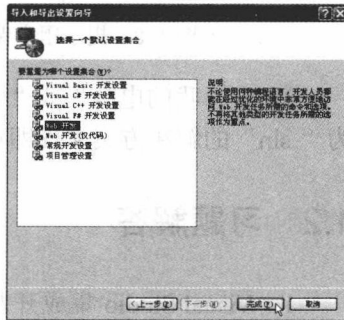


图 1-13 选择新的开发环境

3. 如何在 Visual Studio 集成开发环境中打开一个现有项目？

解答：通常可使用以下 3 种方法打开已存在的项目。

1) 启动 Visual Studio 后，执行“文件”菜单中的“打开解决方案”命令，在打开的对话框中找到保存项目的文件夹，双击解决方案文件 (*.sln) 即可打开程序项目。

2) 启动 Visual Studio 后，若在“起始页”中“最近的项目”区域内包含有要打开的项目名称，则单击该名称也可打开对应的程序项目。

3) 在“我的电脑”或“资源管理器”窗口中，找到项目保存的文件夹，双击其中扩展名为“.sln”的解决方案文件。

4. 什么是解决方案？什么是项目？解决方案与项目有什么关系？

解答：使用 Visual Studio 开发的每一个应用程序叫解决方案，每一个解决方案可以包含一个或多个项目。一个项目通常是一个完整的程序模块，可以有多个项，在“解决方案资源管理器”中可以浏览组成解决方案的所有项目和每个项目中的文件，可以对解决方案的各元素进行组织和编辑。

5. 属性窗口中的属性有哪两种排列顺序，怎样切换？怎样在属性窗口切换属性列表与事件列表？怎样在属性窗口了解属性与事件的功能？

解答：属性窗口中的属性有按分类顺序排列和按字母顺序排列，单击属性窗口的按分类顺序排列按钮  可以按分类顺序排列，或单击属性窗口中的按字母排序按钮  可以按字母顺序排列。单击属性窗口中的属性按钮  或事件按钮  可以在属性窗口与事件窗口之间切换。

如图 1-14 所示,属性与事件窗口的下部有一个属性或事件功能说明区域,当选择某一属性或事件时,说明区域显示文字说明属性或事件的作用。如果属性窗口的说明区域没有显示,可将鼠标指向属性窗口下部边框,当鼠标变为双向箭头时,向上拖动鼠标,属性窗口的说明区域即可显示。

6. Visual Studio 提供的“智能感知”功能主要包含哪些内容?

解答:在代码编写的过程中,Visual Studio 提供了“智能感知”的帮助方式,利用这种帮助方式不仅可以节省代码输入的时间,更重要的是避免了用户的输入错误。智能感知主要包括有提示类名或对象名、提示类成员或对象成员、提示方法的使用说明和实例化提示等。

7. 简述在 Visual Studio 集成开发环境中创建 Windows 应用程序的一般步骤。

解答:在 Visual Studio 中创建一个 Windows 应用程序,一般需要经过以下 6 个步骤。

- 1) 根据用户需求进行问题分析,构思出合理的程序设计思路。
- 2) 创建一个新的 Windows 应用程序项目。
- 3) 设计应用程序界面。
- 4) 设置窗体中所有控件对象的初始属性值。
- 5) 编写用于响应系统事件或用户事件的代码。

6) 试运行并调试程序,纠正存在的错误,调整程序界面,提高容错能力和操作的便捷性,使程序更符合用户的操作习惯。通常将这一过程称为提高程序的“友好性”。

8. 设计图 1-15 所示的应用程序界面。要求通过本题操作,掌握应用程序界面设计中常用技巧(如复制/粘贴控件、拖动控件时自动对齐等),掌握 Visual Studio “格式”菜单中的常用命令。此外,要求程序具有单击数字按钮时可将数字连续显示到文本框中,单击“C”按钮时可清除文本框中所有内容的功能。

程序设计步骤如下。

(1) 创建 Windows 应用程序项目

Visual Studio 启动后,单击“起始页”中“新建项目”,选择使用“Visual Basic”创建“Windows 应用程序”,在为项目命名、指定保存位置后,单击“确定”按钮进入 Visual Studio 集成开发环境。

(2) 设计程序界面

在窗体的设计图中通过双击工具箱中控件图标的方法,向窗体添加 1 个文本框控件和 1 个按钮控件,适当调整两控件的大小及位置。

选择按钮控件后单击工具栏上的复制按钮,再单击粘贴按钮创建与第一个按钮大小完全相同的按钮,并将其移动到适当的位置。Visual Studio 提供了窗体控件自动对齐的功能,如图 1-16



图 1-14 属性窗口下方的功能说明区

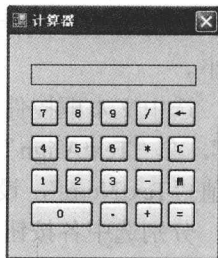


图 1-15 应用程序界面

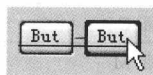


图 1-16 控件对齐参照线

所示，当拖动由复制/粘贴方法得到的第二个按钮靠近第一个按钮时，系统将自动显示对齐参照线。

继续单击粘贴按钮，得到第一行的 5 个按钮。拖动鼠标画出一个矩形框，将上述 5 个按钮全部选中，单击工具栏上的复制按钮后，单击粘贴按钮，得到与上述 5 个按钮大小、间隔均相同的一组按钮，并将其移动到适当位置（注意使用对齐参照线）。重复上述操作得到 4 组共 20 个按钮。删除最后一行第二个按钮，并调整最后一行第一个按钮（数字 0 按钮）的大小。

手工调整第一行 5 个按钮的间距后，按住〈Shift〉键依次选中第一行的 5 个按钮，注意应最后选中第一个按钮，使之作为调整依据。执行“格式”菜单中“对齐”项下的“中间对齐”命令使 4 个按钮水平对齐。

手工调整第一列 4 个按钮的间距后，按住〈Shift〉键依次选中第一列的 4 个按钮，注意应最后选中第一个按钮，使之作为调整依据。执行“格式”菜单中“对齐”项下的“左对齐”命令使 4 个按钮垂直对齐。

重复执行上述操作完成按钮控件所有行列对齐。

提示：由于窗体中的控件默认对齐网格，因此控件的对齐通常比较容易设置。如果需要精确调整控件位置，可按住〈Ctrl〉键后按↑、↓、←、→方向键按钮。图 1-17 所示的是控件布局完成后的界面效果。

(3) 设置对象属性

选中窗体后，在属性窗口中设置其 Text 属性为“计算器”，设置 MaximizeBox 属性和 MinimizeBox 为 False，设置 MaximumSize 属性和 MinimumSize 属性值为当前 Size 属性的值，使窗体没有最大化和最小化按钮，且不能改变大小。

选中文本框控件，在属性窗口中设置其 ReadOnly 属性为“True”，使之成为“只读文本框”，在“TextAlign”属性栏的下拉列表中选择属性值“Right”，在“(Name)”属性栏填入属性值“TextNum”，设置文本框的“BackColor”属性为与窗体背景色相同的颜色。

分别选中各按钮控件，参照程序界面设计目标和表 1-2 中对控件 Name 属性的规定依次设置其 Text 和 Name 属性值。



图 1-17 设计界面

表 1-2 各按钮控件的 Name 属性值

控 件	Name 属性值
0~9 数字按钮	btn0~btn9
运算符按钮 /、*、-、+	btnDiv、btnMult、btnSub、btnAdd
小数点按钮	btnDec
退格按钮 ←	btnBack
清除按钮 C	btnC
存储按钮 M	btnMemory
等于按钮 =	btnEq

(4) 创建控件的共享事件

在窗体的设计视图中双击某按钮控件，系统将自动创建其单击（Click）事件程序框架，例如，在窗体设计器中双击 btn0 按钮，系统将自动切换到代码窗口，并添加如下事件处理程序框架代码：

```
Private Sub btn0_Click(ByVal sender As System.Object, _  
                        ByVal e As System.EventArgs) Handles btn0.Click  
    '程序员编写代码的区域  
End Sub
```

将代码中的“btn0_Click”修改成“btn_Click”，并在第一行结尾处将 btn1.Click、btn2.Click、……、btn9.Click 添加进来。注意，每项之间应使用逗号“,”分隔。

经过上述修改，btn0~btn9 任何一个按钮被单击，都将转去执行 btn_Click 事件处理程序，从而使 btn_Click 事件处理程序成为 10 个按钮控件的共享 Click 事件处理程序。

(5) 编写程序代码

10 个数字按钮被单击时执行的共享事件处理代码如下：

```
Private Sub btn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles _  
                        btn0.Click, btn1.Click, btn2.Click, btn3.Click, btn4.Click, _  
                        btn5.Click, btn6.Click, btn7.Click, btn8.Click, btn9.Click  
    Dim btn As Button        '声明一个 Button 类型的变量 btn  
    btn = sender              '从 sender 中获取触发本事件的具体按钮  
    txtNum.Text = txtNum.Text + btn.Text    '将触发本事件按钮的文本追加到文本框中  
End Sub
```

“C”按钮被单击时执行的程序代码如下：

```
Private Sub btnC_Click(ByVal sender As System.Object, _  
                        ByVal e As System.EventArgs) Handles btnC.Click  
    txtNum.Text = ""        '设置文本框中显示的文本为空  
    txtNum.Focus()          '将插入点光标移动到文本框中  
End Sub
```

说明：

1) 按钮的 Click 事件可以通过对象（Object）类型的参数 sender 返回具体是哪个控件触发了本事件。下列语句的含义是，声明一个按钮类型的变量 btn 使之从 sender 中获取触发本事件的具体按钮。例如，用户单击了 btn9 按钮，则变量 btn 就得到 btn9。btn.Text 就等同于 btn9.Text。

```
Dim btn As Button        '声明一个 Button 类型的变量 btn  
btn = sender              '从 sender 中获取触发本事件的具体按钮
```

2) 在 Visual Basic 语法中使用“+”运算符连接两个字符串。例如，“11”+“12”的结果为“1112”而不是“23”。文本框的 Text 属性值是字符串类型的，故下列语句的含义是，将被单击按钮的 Text 属性值“追加”到文本框中。

```
txtNum.Text = txtNum.Text + btn.Text
```