



高职高专“十一五”规划教材

C#

CHENGXU SHEJI
XIANGMU SHIXUN JIAOCHENG

程序设计项目实训教程

黄锐军 编著



化学工业出版社



高职高专“十一五”规划教材

C#

CHENGXU SHEJI
XIANGMU SHIXUN JIAOCHENG

程序设计项目实训教程

● 黄锐军 编著



化学工业出版社

· 北京 ·

本书根据高职高专的教学特点,用项目驱动的方式进行编排和讲解,全书包含八个项目及一个综合项目,每个单元都设计成案例展示、技术要点、程序设计、模拟训练、应用拓展等几个部分,使读者在明确要完成一个任务的前提下去学习知识,训练技能,边学边做,不断学习与提高,实践性很强。

本书通过大量的项目案例,主要介绍了 C#语言基础知识、分支程序的设计、各种循环程序的设计方法与语句结构、数组的应用、函数的设计、面向对象的程序设计方法、文件操作、用库存储类与数据访问类两种访问数据库的方法等知识。同时书中还配有大量练习,希望通过这些练习能进一步锻炼和培养读者编程的能力。本书的所有案例都在 Visual Studio 2005 的集成环境下运行通过。

本书主要作为高职高专计算机专业的教材,也可以作为计算机培训教材和相关技术人员参考。

图书在版编目(CIP)数据

C#程序设计项目实训教程 / 黄锐军编著. —北京: 化学工业出版社, 2010.1

高职高专“十一五”规划教材

ISBN 978-7-122-07522-2

I. C… II. 黄… III. C语言-程序设计-高等学校: 技术学院-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2010) 第 000330 号

责任编辑: 王听讲

文字编辑: 陈元

责任校对: 战河红

装帧设计: 刘丽华

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 化学工业出版社印刷厂

787mm×1092mm 1/16 印张 19¼ 字数 514 千字

2010 年 2 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686)

售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书, 如有缺损质量问题, 本社销售中心负责调换。

定 价: 36.00 元

版权所有 违者必究

前 言

C#语言是微软.Net技术的核心编程语言，它继承了C语言的编程风格，语法严谨、功能强大、内容丰富、使用灵活方便，既可以编写Windows桌面程序，又可以编写Web网络程序，可视化效果好、编程速度快，所以得到广泛应用。这本书以项目化、工作过程化的方式，深入浅出地介绍了C#语言的Windows程序开发方法，读者可以把它作为学习C#程序设计的基础教材。

要学好一门语言，不应该拘泥于该语言的语法细节，重点是要大量使用该语言来编写程序，从实践中来学习与巩固它的基本知识。因此本书的特点是实践性强，采用项目驱动的方式讲解要掌握的知识与内容，每个单元都设计成案例展示、技术要点、程序设计、模拟训练、应用拓展等几个部分，使读者在明确要完成一个任务的前提下去学习知识，训练技能，边学边做，不断学习与提高。同时书中还配有大量练习，希望通过这些练习能进一步锻炼和培养读者编程的能力。本书的所有案例都在Visual Studio 2005的集成环境下运行通过。

全书包含八个项目及一个综合项目。项目实训1是一个完成学生成绩统计的项目，主要介绍了C#语言基础知识。项目实训2是一个所得税计算器的项目，主要介绍了分支程序的设计。项目实训3是关于整数分解的项目，介绍了各种循环程序的设计方法与语句结构。项目实训4是关于单词统计的项目，主要介绍了数组的应用。项目实训5是自编日历的项目，介绍了函数的设计。项目实训6是有关学生信息管理的项目，介绍了面向对象的程序设计方法。项目实训7是一个记事本程序的项目，介绍了文件操作。项目实训8是有关学生记录管理的项目，介绍了用库存储类与数据访问类两种访问数据库的方法。综合实训设计了一个日记本程序，讲解了综合应用C#开发Windows的基本方法，目的是巩固所学的知识，重点培养读者的编程能力。

本书可以作为高职高专计算机相关专业的教材，建议总学时安排在60~80学时之间，其中讲授与上机实习的学时比例在1:1左右。读者可以到化学工业出版社教学资源网站(<http://www.cipedu.com.cn>)免费下载使用本书的相关资源和电子教案。

本书在编写过程中得到了田维琪、曾月芬、余珊珊、黄兵、曾子英、黄晓羽、赖唤勇、黄博建、黄子昱等人的帮助，在此表示衷心的感谢！

由于作者水平有限，书中难免有些不足之处，敬请广大读者批评指正。

黄锐军

2009年10月于深圳

目 录

项目实训 1 学生成绩统计	1
项目案例 1.1 学生信息输入输出	1
1.1.1 案例展示	1
1.1.2 技术要点	1
1.1.3 程序设计	4
1.1.4 模仿训练	7
1.1.5 应用拓展	8
项目案例 1.2 学生平均成绩计算	8
1.2.1 案例展示	8
1.2.2 技术要点	9
1.2.3 程序设计	11
1.2.4 模仿训练	12
1.2.5 应用拓展	12
项目案例 1.3 学生成绩累计计算	14
1.3.1 案例展示	14
1.3.2 技术要点	14
1.3.3 程序设计	15
1.3.4 模仿训练	16
1.3.5 应用拓展	17
项目案例 1.4 字符与字符编码	18
1.4.1 案例展示	18
1.4.2 技术要点	18
1.4.3 程序设计	20
1.4.4 模拟训练	20
1.4.5 应用拓展	20
实训 1 学生成绩统计程序	21
练习题	23
项目实训 2 所得税计算器	24
项目案例 2.1 所得税缴纳情况判断	24
2.1.1 案例展示	24
2.1.2 技术要点	24
2.1.3 程序设计	27
2.1.4 模仿训练	27
2.1.5 应用拓展	28

项目案例 2.2 所得税最高税率判断	28
2.2.1 案例展示	28
2.2.2 技术要点	28
2.2.3 程序设计	30
2.2.4 模仿训练	31
2.2.5 应用拓展	32
项目案例 2.3 所得税输入异常捕捉	33
2.3.1 案例展示	33
2.3.2 技术要点	33
2.3.3 程序设计	34
2.3.4 模仿训练	35
2.3.5 应用拓展	35
项目案例 2.4 所得税覆盖税率判断	36
2.4.1 案例展示	36
2.4.2 技术要点	37
2.4.3 程序设计	37
2.4.4 模仿训练	39
2.4.5 应用拓展	40
实训 2 所得税计算器程序	41
练习题	43
项目实训 3 整数的分解	45
项目案例 3.1 整数的因数分解	45
3.1.1 案例展示	45
3.1.2 技术要点	45
3.1.3 程序设计	47
3.1.4 模拟训练	47
3.1.5 应用拓展	48
项目案例 3.2 整数是否是素数的判断	49
3.2.1 案例展示	49
3.2.2 技术要点	49
3.2.3 程序设计	52
3.2.4 模拟训练	53
3.2.5 应用拓展	53
项目案例 3.3 整数的素数因数分解	54
3.3.1 案例展示	54
3.3.2 技术要点	54
3.3.3 程序设计	56
3.3.4 模拟训练	57
3.3.5 应用拓展	57
实训 3 整数的分解	58

练习题	61
项目实训 4 单词统计	62
项目案例 4.1 字符数组与字符串转换	62
4.1.1 案例展示	62
4.1.2 技术要点	62
4.1.3 程序设计	63
4.1.4 模拟训练	64
4.1.5 应用拓展	64
项目案例 4.2 字母统计与 ListBox 列表显示	66
4.2.1 案例展示	66
4.2.2 技术要点	66
4.2.3 程序设计	67
4.2.4 模拟训练	68
4.2.5 应用拓展	69
项目案例 4.3 字母统计与数组排序	72
4.3.1 案例展示	72
4.3.2 技术要点	72
4.3.3 程序设计	74
4.3.4 模拟训练	76
4.3.5 应用拓展	76
项目案例 4.4 字母统计与 ListView 列表显示	77
4.4.1 案例展示	77
4.4.2 技术要点	77
4.4.3 程序设计	80
4.4.4 模拟训练	81
4.4.5 应用拓展	81
项目案例 4.5 单词统计与动态存储	83
4.5.1 案例展示	83
4.5.2 技术要点	83
4.5.3 程序设计	84
4.5.4 模拟训练	86
4.5.5 应用拓展	86
实训 4 单词统计程序	88
练习题	92
项目实训 5 我的日历	94
项目案例 5.1 日历某年是否是闰年的判断	94
5.1.1 案例展示	94
5.1.2 技术要点	94
5.1.3 程序设计	97
5.1.4 模拟训练	98

5.1.5 应用拓展	99
项目案例 5.2 日历某日期是第几天的计算	100
5.2.1 案例展示	100
5.2.2 技术要点	100
5.2.3 程序设计	102
5.2.4 模拟训练	104
5.2.5 应用拓展	105
项目案例 5.3 日历某日期是星期几的计算	107
5.3.1 案例展示	107
5.3.2 技术要点	107
5.3.3 程序设计	109
5.3.4 模拟训练	112
5.3.5 应用拓展	115
项目案例 5.4 日历与时间的显示	116
5.4.1 案例展示	116
5.4.2 技术要点	116
5.4.3 程序设计	118
5.4.4 模拟训练	121
5.4.5 应用拓展	122
实训 5 我的日历程序	123
练习题	128
项目实训 6 学生信息管理	129
项目案例 6.1 学生类与对象的建立	129
6.1.1 案例展示	129
6.1.2 技术要点	129
6.1.3 程序设计	134
6.1.4 模拟训练	135
6.1.5 应用拓展	136
项目案例 6.2 学生类的变量与属性	137
6.2.1 案例展示	137
6.2.2 技术要点	137
6.2.3 程序设计	139
6.2.4 模拟训练	141
6.2.5 应用拓展	142
项目案例 6.3 学生类的继承与派生	143
6.3.1 案例展示	143
6.3.2 技术要点	143
6.3.3 程序设计	146
6.3.4 模拟训练	150
6.3.5 应用拓展	152

项目案例 6.4 学生类的照片处理	153
6.4.1 案例展示	153
6.4.2 技术要点	153
6.4.3 程序设计	155
6.4.4 模拟训练	158
6.4.5 应用拓展	160
项目案例 6.5 学生类对象与数组存储	160
6.5.1 案例展示	160
6.5.2 技术要点	160
6.5.3 程序设计	162
6.5.4 模拟训练	164
6.5.5 应用拓展	165
项目案例 6.6 学生类对象与列表存储	166
6.6.1 案例展示	166
6.6.2 技术要点	166
6.6.3 程序设计	167
6.6.4 模拟训练	171
6.6.5 应用拓展	172
项目案例 6.7 学生信息对话框的建立	172
6.7.1 案例展示	172
6.7.2 技术要点	172
6.7.3 程序设计	175
6.7.4 模拟训练	178
6.7.5 应用拓展	179
实训 6 学生信息管理程序	180
练习题	189
项目实训 7 我的记事本	191
项目案例 7.1 记事本程序文件的读写	191
7.1.1 案例展示	191
7.1.2 技术要点	191
7.1.3 程序设计	193
7.1.4 模拟训练	194
7.1.5 应用拓展	195
项目案例 7.2 记事本程序的菜单设计	197
7.2.1 案例展示	197
7.2.2 技术要点	197
7.2.3 程序设计	199
7.2.4 模拟训练	201
7.2.5 应用拓展	202
项目案例 7.3 记事本程序工具栏与状态栏设计	203

7.3.1	案例展示	203
7.3.2	技术要点	203
7.3.3	程序设计	205
7.3.4	模拟训练	207
7.3.5	应用拓展	207
项目案例 7.4 记事本程序的模态对话框设计		209
7.4.1	案例展示	209
7.4.2	技术要点	209
7.4.3	程序设计	210
7.4.4	模拟训练	211
7.4.5	应用拓展	212
项目案例 7.5 记事本程序的非模态对话框设计		213
7.5.1	案例展示	213
7.5.2	技术要点	213
7.5.3	程序设计	214
7.5.4	模拟训练	217
7.5.5	应用拓展	218
实训 7 我的记事本程序		220
练习题		233
项目实训 8 学生记录管理		236
项目案例 8.1 学生记录的数据访问类读取		236
8.1.1	案例展示	236
8.1.2	技术要点	236
8.1.3	程序设计	239
8.1.4	模拟训练	241
8.1.5	应用拓展	242
项目案例 8.2 学生记录的数据访问类更新		244
8.2.1	案例展示	244
8.2.2	技术要点	244
8.2.3	程序设计	245
8.2.4	模拟训练	247
8.2.5	应用拓展	249
项目案例 8.3 学生记录的数据存储类读取		251
8.3.1	案例展示	251
8.3.2	技术要点	251
8.3.3	程序设计	253
8.3.4	模拟训练	255
8.3.5	应用拓展	256
项目案例 8.4 学生记录的数据存储类更新		257
8.4.1	案例展示	257

8.4.2	技术要点	257
8.4.3	程序设计	259
8.4.4	模拟训练	262
8.4.5	应用拓展	263
实训 8	学生信息管理程序	264
	练习题	274
综合实训	我的日记本	277
实训 1	基于磁盘文件存储的“我的日记本”	277
实训 2	基于数据库存储的“我的日记本”	294
参考文献		306

项目实训 1 学生成绩统计

项目功能：统计学生的平价成绩、总成绩等。

学习范围：语言基础、Windows 程序基本控件的使用、数据类型、信息输入输出对话框、数据类型转换、算术计算。

练习内容：针对该知识与能力范围的知识练习与多个项目实训练习。

项目案例 1.1 学生信息输入输出

1.1.1 案例展示

编写一个 C# 程序，输入学生的学号与姓名，按确定按钮后显示出输入的值，如图 1-1 所示。

1.1.2 技术要点

计算机存储的数据多种多样，但每种数据是有类型的，有些数据可以做传统意义的加减乘除运算，有些数据只能做简单的输入输出。



图 1-1 学生信息

1. 字符串与变量

学生学号、姓名这样的数据，只是一个个字符（char）构成的一串文字，称为字符串（string）。这种数据一般只用来输入输出进行显示，字符串与字符串之间不做传统意义上的加减乘除运算。字符串用一对英文的双引号引起来，例如“张三”、“李四”、“1234”、“a”都是 string 类型的字符串数据。注意不能用中文的双引号引起来。这里学号完全由数字组成，也是字符串，因为学号的数字并不代表一个数学中的数，例如“1234”并不代表数学中的 1234 这个整数。

计算机中为要存储一个字符串数据，就必须在内存中开辟一个空间来存储它。显然空间的大小与字符串的长度有关，字符串越长，所要的空间越大，反之要的空间越小。为了在这个空间中存放字符串，还必须为这个空间取一个名字，这个名字就是变量名。在 C# 中用 string 定义变量空间，例如：

```
string a,b,c;  
a="1000";  
b="张三";  
c="";
```

第一条语句定义了 a、b、c 这样三个字符串变量空间，第二、三、四条语句分别使变量 a、b、c 中存储了字符串“1000”、“张三”及一个空字符串（空字符串不包含任何字符）。

语句是程序的基本单位，每条语句都执行一定的功能。在 C# 中语句用分号结束，C# 中程序是一条条语句组成的，执行时也是按语句的先后顺序一条条执行的。每条语句不一定占一行，一行可以有多条语句，一条语句也可以分几行来写，但一条语句占一行会使程序美观，

例如下列程序语句是正确的:

```
string a,b,c; a="1000"; b="张三"; c="";
```

在指定的空间中存储的字符串数据是可以随时改变的,新的数据进来后旧的数据就被覆盖了,例如:

```
string a,b,c;
```

```
a="1000";
```

```
b="张三";
```

```
a=b;
```

那么执行后 a 中的数据为"张三",原来的数据"1000"已经不存在了。

2. 变量名称

变量是有名字的存储单元,它的值可以通过程序改变。程序中用到的变量要定义或声明,定义的方式为:

数据类型 变量 1,变量 2,.....;

每个变量都必须有一个名称,变量名称是一个用户定义的变量名字。变量的名字一般遵循下面的规则:

(1) 变量名一般以字母开始,后面可以跟若干个英文字母或数字或下划线,不能包含空格;

(2) 变量名区分大小写,例如变量 A 与变量 a 是不相同的;

(3) 变量名不宜太长,一般最好有一定的含义,例如用 math 及 english 分别表示学生的数学成绩与英语成绩就是比较好的命名方法;

(4) C#定义了很多关键字,用户不能定义与关键字重名的变量,否则会引起混乱。

根据这些原则,sno、sname、age、math、english 等变量名字是合法的,而 2x、x y 等变量名是不合法的。

在定义变量时,可以同时定义几个同类型的变量,还可以在定义时说明它的初始值,例如:

```
string sno,sname="张三";
```

则定义变量 sno 及 sname,而且 sname 有初始值为"张三"。

3. 认识 C#开发环境

启动 Visual Studio 2005 后会出现一个起始页面,执行“文件→新建→项目”后出现如图 1-2 所示的对话框。

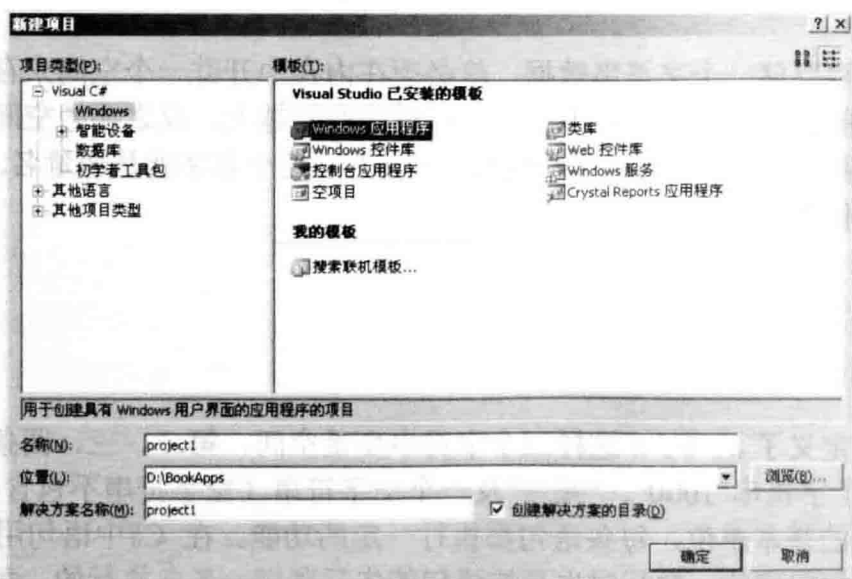


图 1-2 新建项目

在对话框中选择项目编程语言为“Visual C#”，在模板中选择“Windows 应用程序”，输入项目的名称（例如 project1），选择项目存放的位置（例如 D:\BookApps）后确定，随后出现图 1-3 所示的界面。

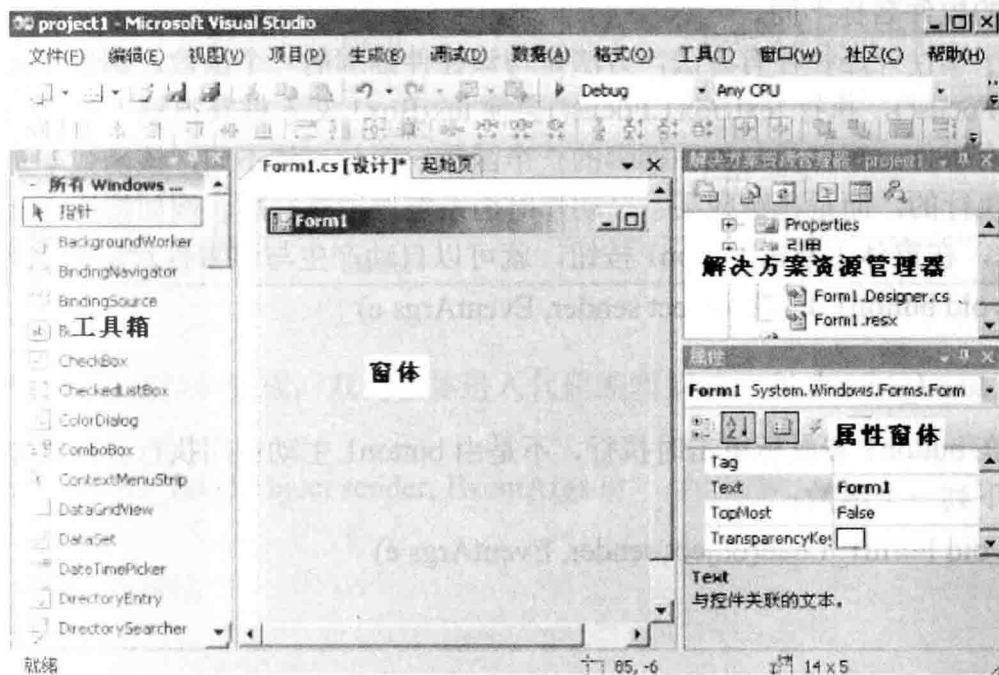


图 1-3 Visual Studio 2005 开发环境

在这个界面中有几个子窗体：

(1) 工具箱

工具箱中有 Windows 常用的控件，例如项目中用到的标签（Label）、按钮（Button）等。控件往往是为了完成输入输出的一些部件，例如标签控件 Label 常用来显示一个字符串，文本框（TextBox）控件用来输入字符串，按钮（Button）用来响应鼠标的单击事件并使程序执行一段代码。

(2) 窗体

窗体是设计 Windows 程序的基本界面，系统已经建立了一个名称 Name 为 Form1 的窗体，用户只要把控件（对象）放在这个窗体上就可以。

(3) 属性窗体

它是设置控件属性的窗体，窗体与控件的属性都可以在这个窗体中设置。要改变窗体及标签的文字就要改变它们的属性，例如把 label1 的 Text 属性设置为“学号”，则窗体上 label1 的显示文本就变为“学号”。

(4) 解决方案资源管理器

该窗体包含项目中使用的文件，例如 Form1.cs 就是其中一个文件，C#的源程序文件扩展名都为“.cs”。

4. 窗体与控件

在窗体上放任何一个控件，该控件都有一个名称（Name），例如放一个标签（Label），系统为它设置的名称为 label1，再放一个标签在窗体上，系统为它设置的名称为 label2，如再放一个文本输入框（TextBox），系统为它设置的名称为 textBox1 等。名称是程序识别不同控件的标记，每个控件都有一个名称，不同的控件名称不同。

每个控件都有很多的属性，属性是与控件捆绑的一个变量参数，因此属性也是有不同数

据类型的。例如标签上的文字“学号”就是标签的 `Text` 属性决定的，文本框中的文字也是 `Text` 属性决定的，而窗体的 `Text` 属性决定了窗体的标题，它们的类型都是 `string`。一般一个控件有多个属性，不同的对象有不同的属性，但任何控件都有一个属性 `Name`，它是对象的名称，不同的控件名称不同。

控件除了属性外还往往有方法，方法是与该控件捆绑的一个函数。例如 `TextBox` 文本框有一个方法 `Focus()`，执行 `textBox1.Focus()` 会使 `textBox1` 文本框获得输入焦点。

控件还有事件，事件是与控件捆绑的一个函数，但与方法不同，该函数不是通过控件本身主动调用执行的，而是在响应某一个动作时由系统调用执行的。例如按钮 `button1` 有一个事件为 `Click`，在窗体上双击 `button1` 按钮，就可以自动产生与该事件对应的下列函数：

```
private void button1_Click(object sender, EventArgs e)
{
}
```

该函数在 `button1` 被鼠标单击时执行，不是由 `button1` 主动调用执行。再比如双击窗体本身，会产生下列一个函数：

```
private void Form1_Load(object sender, EventArgs e)
{
}
```

该函数对应于窗体的 `Load` 事件，表示窗体程序被装载到内存开始执行时就执行该函数，这是一个重要的函数，常用它来执行程序的初始化。

1.1.3 程序设计

1. 程序设计步骤

用 C# 设计程序的步骤一般分为以下几步：

(1) 界面的设计

既在窗体界面上安排所需的控件。

(2) 设置控件的属性

在属性窗口中设置控件的属性以满足程序的需要。

(3) 编写程序代码

在代码窗口中编写 C# 的程序语句。

(4) 调试与执行程序

找出并修正程序的错误，执行程序查看是否达到目的。

(5) 保存程序文件

最后保存程序文件，关闭项目。

根据不同的程序，设计的步骤可能有所不同，但这几个基本步骤大致相同。

2. 界面设计

C# 是可视化的程序设计，界面设计简单，一切都是所见即所得。界面设计在窗体上完成，把控件放置在窗体上，调整它们的位置，设置它们的属性，就可以做出一个漂亮的程序界面。程序界面用于响应鼠标操作与键盘等的输入操作，同时用来显示数据与结果。

在窗体 `Form1` 上放置三个标签控件 `Label`，名称 (`Name`) 分别为 `label1`、`label2`、`label3`，其中 `label1`、`label2` 依次设置它们的 `Text` 属性为“学号”、“姓名”，在每个标签的旁边依次放置两个文本输入框 (`TextBox`)，名称 (`Name`) 依次为 `textBox1`、`textBox2`，分别用于输入对应的数据。最后在窗体上放一个按钮 (`Button`)，名称为 `button1`，设置 `Text` 属性为“确定”。

注意这些控件的 Text 属性都是 string 类型的数据，这些属性设置归纳为表 1-1。

表 1-1 属性设置

控 件	名 称	属 性
窗体 Form	Form1	Text="文本框与字符串"
文本框 TextBox	textBox1	Text=""
	textBox2	Text=""
标签	label1	Text="学号"
	label2	Text="姓名"
按钮 Button	button1	Text="确定"

3. 代码设计

代码设计在代码窗口中完成，双击窗体进入代码编辑窗口，在 Form1_Load 函数中编写代码：

```
private void Form1_Load(object sender, EventArgs e)
{
    textBox1.Focus();
    label3.Text = "";
}
```

双击按钮 button1，进入代码编辑窗口，在 button1_Click 函数中编写代码：

```
private void button1_Click(object sender, EventArgs e)
{
    string sno, sname;
    sno = textBox1.Text;
    sname = textBox2.Text;
    label3.text = "学号:" + sno + " 姓名:" + sname;
}
```

执行该程序，效果如图1-1。

4. 程序说明

为了说明各个语句或一段语句的功能，可以在程序中插入注释语句。注释语句是不被计算机执行的语句，它仅对程序进行注解作用，目的是方便阅读程序。C#中的注释语句有两种形式：

- (1) 用/*及*/包含的一段文字；
- (2) 用“//”开始一直到本行结束都看成是注释语句。

在编辑器中注释语句默认为绿颜色。在学生信息输入程序中可看到程序代码由几个部分组成，下面把程序代码加上注释语句，就可以看到各个部分的结构与功能如下：

```
/*
    C#程序的开始部分总有一些 using 的语句，它们是对 .Net 系统的库引用，例如
    using System;
    表示对系统的引用。这部分系统自动产生，用户不要改变它们。
*/
using System;
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
/*
```

namespace 是程序的命名空间，project1 是该空间的名称，它在建立项目时由用户确定，一旦确定，不要轻易改变。

```
*/
namespace project1
{
/*
```

Form1 是程序窗体的名称，该名称默认为 Form1，Form1 是从系统的 Form 派生出来的一个窗体，大多数程序都只用到这个窗体。

```
*/
public partial class Form1 : Form
{
/*
```

程序由一组函数组成，函数是一段程序代码的总和，这段代码被包含在一对大括号中，每个函数都有一个名字。该程序包含了 Form1、button1_Click、Form1_Load 三个函数，它们并列存放在 public partial class Form1:Form 的框架中，出现的先后顺序可以改变，不影响程序功能。

```
*/
    public Form1()
    {
```

该函数是系统启动时的初始化函数，用户不要改动它。

```
*/
    InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
/*
```

这个函数 button1_Click 是在 button1 按钮被点击时执行的，函数是一段程序代码的总和。这段代码被包含在一对大括号中。一个 C# 程序往往由多个函数组成，函数与函数之间并列排列，前后顺序不影响程序功能。

```
*/
    string sno, sname, sex; //sno 学号, sname 姓名
    sno = textBox1.Text; //从 textBox1 输入学号
    sname = textBox2.Text; //从 textBox2 输入姓名
    //用 label3 显示学号、姓名
    label3.Text = "学号:" + sno + " 姓名:" + sname;
```