



第 3 章 流程控制

程序有三种执行流程：顺序执行、分支执行和循环执行。顺序执行是指代码按照它们出现的顺序依次执行；分支执行是指依据一定的条件选择执行路径，而不是严格按照语句出现的顺序。循环是指对代码段的重复执行，可以减少源程序重复书写的工作量。C# 中使用分支结构和循环结构来实现对程序流程的控制。

3.1 分支语句

分支语句又叫作条件语句。当程序中需要两个或两个以上的选择时，可以根据条件判断结果来选择将要执行的语句。分支结构程序设计方法的关键在于构造合适的分支条件和分析程序流程，根据不同的程序流程选择适当的分支语句。C# 中的分支语句包含两类：if 语句和 switch 语句。

3.1.1 if 语句

程序执行时，当所做选择较少时，可使用 if 语句进行控制。if 语句根据布尔类型的表达式的值来选择要执行的语句。if 语句包含如下几种形式。

1. 单分支 if 语句

格式：

```
if (条件表达式)  
    语句；
```

如果条件表达式为真，则执行语句。若想执行一组语句，则将这一组语句用 {} 括起来构成一个语句块。

例如，求 x 的绝对值，可以用如下的语句。

```
if (x < 0)    x = -x;
```

2. 双分支 if-else 语句

格式：

```
if (条件表达式)
```

```

    语句 1;
else
    语句 2;

```

如果条件表达式为真,则执行语句 1,否则执行语句 2。

【例 3-1】 输入一个整数,判断其奇偶性,程序运行结果如图 3.1 所示。

(1) 创建一个新的 Windows 应用程序项目 3-1。并将 1 个 Label 控件、1 个 TextBox 控件和 2 个 Button 控件从工具箱添加到 Form1 窗体上。

(2) 在“属性”窗格中修改属性。将窗体 Form1 的 Text 属性修改为“判断奇偶性”;将标签 Label1 控件的 Text 属性修改为“请输入一个整数:”;将按钮 button1 和 button2 的 Text 属性,分别修改为“清空”和“确定”,效果如图 3.2 所示。



图 3.1 例 3-1 运行结果



图 3.2 例 3-1 界面设计

(3) 将按钮 button1 和 button2 的 Name 属性分别修改为 btnClear 和 btnOK。

(4) 修改“清空”和“确定”按钮的 Click 事件中的代码。完成后 Form1.cs 文件的代码如下。

```

namespace _3_1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void btnClear_Click(object sender, EventArgs e)
        {
            textBox1.Text = ""; //清空文本框中的内容
        }
        private void btnOK_Click(object sender, EventArgs e)
        {
            int t = Convert.ToInt32(textBox1.Text);
            if (t % 2 == 0)
                MessageBox.Show("输入的数是偶数");
            else

```

```
        MessageBox.Show("输入的数是奇数");  
    }  
}  
}
```

这里介绍一下 WinForm 控件的命名规则：用控件简写后跟控件功能名，如将“清空”按钮命名为 btnClear。

WinForm 常用控件命名举例，如表 3.1 所示。

表 3.1 WinForm 常用控件命名举例

控件类型	控件类型简写	命名举例	控件类型	控件类型简写	命名举例
Label	lbl	lblMessage	TreeView	tvw	tvwType
Button	btn	btnSave	ListView	lvw	lvwBrowser
TextBox	txt	txtName	Timer	tmr	tmrCount
MenuStrip	mns	mnsStart	ToolStrip	tsr	tsrBold
CheckBox	chk	chkStock	StatusStrip	ssr	ssrTime
RadioButton	rbtn	rbtnSelected	OpenFileDialog	odlg	odlgFile
PictureBox	pic	picImage	SaveFileDialog	sdlg	sdlgSave
DataGrid	dgrd	dgrdView	ColorDialog	cdlg	cdlgColor
ListBox	lst	lstProducts	FontDialog	fdlg	fdlgFont
ComboBox	cbo	cboMenu			

注意：要区分控件的 Name 和 Text 属性。

在例 3-1 的程序代码中，两次出现了同一个语句“MessageBox.Show(“指定文本”)；”，其含义是显示具有指定文本的消息对话框。MessageBox.Show() 共有 21 种重载方法（有关方法的重载将在后续章节中介绍），此处是最简单的用法，只显示文本信息。还可以给消息框加上标题，如 MessageBox.Show(“输入的数是偶数”，“标题”)。

3. 多分支 if-else 语句

格式：

```
if(条件表达式 1)  
    语句 1;  
else if(条件表达式 2)  
    语句 2;  
else if(条件表达式 3)  
    语句 3;  
...  
else  
    语句 n;
```

执行时，首先判断条件表达式 1，其值若为 true，则执行语句 1，if 语句执行结束；否则判断条件表达式 2，若其值为 true，则执行语句 2……以此类推，若所有条件均不成立，则执行 else 后的语句 n。

【例 3-2】 判断输入的字符串的首字符是字母、数字还是其他字符,运行结果如图 3.3 所示。



图 3.3 例 3-2 运行结果

Form1.cs 文件的代码如下。

```
namespace _3_2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void btnClear_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
        }
        private void btnOK_Click(object sender, EventArgs e)
        {
            string s = textBox1.Text;
            //将 s 中的字符复制到字符串数组中
            char[] c = s.ToCharArray();
            if (c[0] >= '0' && c[0] <= '9')
                MessageBox.Show("首字符为数字!");
            else
                if ((c[0] >= 'a' && c[0] <= 'z') || (c[0] >= 'A' & c[0] <= 'Z'))
                    MessageBox.Show("首字符为字母!");
                else
                    MessageBox.Show("首字符为其他字符!");
        }
    }
}
```

4. 嵌套 if 语句

在 if 语句中可以嵌套 if 语句,以实现程序中复杂的逻辑判断。

【例 3-3】 使用嵌套 if 实现如下函数 $y = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$, 运行结果如图 3.4 所示。



图 3.4 例 3-3 运行结果

Form1.cs 文件的代码如下。

```
namespace _3_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnCal_Click(object sender, EventArgs e)
        {
            int x = Convert.ToInt32(textBox1.Text);
            int y = 0;
            if (x >= 0)
            {
                if (x > 0)
                {
                    y = 1;
                }
                else
                {
                    y = 0;
                }
            }
            else
            {
                y = -1;
            }
            MessageBox.Show("y 的值是: " + y);
        }
    }
}
```