第十一章 視窗事件處理技巧



11.1 事件介紹

11.2 鍵盤事件

11.3 滑鼠事件

11.4 共享事件

11.1 事件介紹

- 事件(Event)是物件受到外力因素的影響,而發生某種動作。
- 將觸發事件的物件稱為「事件傳送者」。
- 將捕捉事件並且回應它的物件稱為「事件接收者」。
- 當事件發生時,這物件的事件處理函式會被啟動。
- 事件處理函式內的程式碼是依程式的需求而撰寫的,等於 告訴電腦,當某物件發生某事件就去執行所撰寫的程式碼。
- 在事件驅動應用程式中,不像程序化的應用程式控制程式執行 部分和所產生的結果,而是程式碼執行流程是由操作者來決定, 以回應事件的發生。
- 事件的引發有可能是使用者的動作,也可能是來自作業系統或其它應用程式、甚至來自應用程式本身的訊息。
- 在設計事件驅動程式時,瞭解整個事件驅動模式是很重要。



11.2 鍵盤事件

11.2.1 KeyPress 事件

- 當某個物件取得 Focus(駐點或稱控制權)時,按下鍵盤某個按鍵,就會觸動該物件的鍵盤事件。
- 當做完按下按鍵再放開動作時,會依序觸動 KeyDown、KeyPress 和 KeyUp 三個事件。
- **企**動物件取得駐點時,收到按鍵被按下又放開後所觸動的事件, 會傳回按鍵的字元。
- □ 果使用者按的不是字元按鍵(例如、、、鍵),是不會觸動 KeyPress 事件,但 KeyDown 和 KeyUp 事件仍會觸發。
- ₩ ReyPress 事件處理函式寫法:
 private void 物件_KeyPress(object sender, KeyPressEventArgs e)
 {

// 程式區塊





■ KeyPress 事件處理函式的 第一個參數 sender 代表觸發 KeyPress 事件的物件; 第二個參數 e 為 KeyPressEventArgs 型別,e用來 表示事件相關資料的物件。

■ e常使用的屬性如下說明:

引数 e 屬性	說明
KeyChar	可以取得輸入的字元,其資料型別寫字元(Char)。
Handled	設定是否不接受輸入的字元。屬性値寫 false 表示接受;true 表示不接受字元。





- KeyPress 的功能主要是在取得由鍵盤所按鍵的字元,所以 檢查字元是否合法的程式碼會寫在KeyPress 事件中。
- 事件中引數e的KeyChar屬性值可以取得輸入的字元,其資料型別為字元(Char)。
- 如果想將字元轉為ASCII碼(鍵盤碼),可用 (byte)e. Keychar 將所按鍵盤的字元轉成ASCII碼。另外事件 中引數 e 的 Handled 屬性值也是非常重要,如檢查 輸入的字元為不合法時,只將 Handled 屬性值設為true, 該字元就會被清除且插入點游標停留在原處。
- 例如希望使用者只能輸入英文字母,其判斷式寫法: (字母的ASCII碼由65~90)

```
屬 寫法1:
  int n = (byte )e.KeyChar;
   if (n < 65 || n > 90)
     e.Handled = true;
屬 寫法2:
  char ch = e.KeyChar;
   if (ch < 'A' || ch > 'Z')
      e.Handled = true;
```



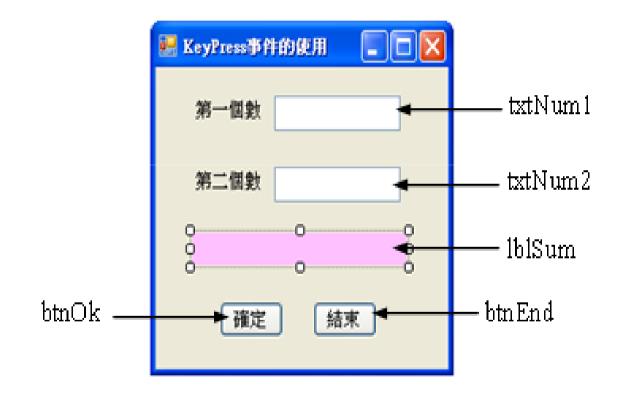
範例演練

■ KeyPress事件的使用	
第一個數	12.5
第二個數	5236.88
兩數加相為	: 5249.38
確定結束	

錯誤!!	X
8	請輸入數字
	確定



輸出入介面





```
// FileName : KeyPressEvent.sIn
01 private void txtNum1_KeyPress(object sender, KeyPressEventArgs e)
02 {
03 char ch = e.KeyChar;
    //判斷是否輸入數字、「.」、倒退鍵和 Enter 鍵
04
    if ((ch < '0' || ch > '9') && (ch != '.' && ch != '\b'))
05
06
        if ((byte)ch == 13)
07
80
         txtNum2.Focus();
09
10
11
       else
12
          e.Handled = true;
13
          MessageBox.Show("請輸入數字", "錯誤!!", MessageBox
14
                         Buttons. OK, Message BoxIcon. Error);
15
16 }
17 }
```



19 private void txtNum2_KeyPress(object sender, KeyPressEventArgs e)

```
20 {
    char ch = e.KeyChar;
21
22
    //判斷是否輸入數字、「.」或倒退鍵和 Enter 鍵
    if ((ch < '0' || ch > '9') && (ch != '.' && ch != '\b'))
23
24
25
       if ((byte)ch == 13) //按Enter鍵游標跳到btnOk鈕
26
         btnOk.Focus();
27
28
29
       else
30
         e.Handled = true;
31
32
         MessageBox.Show("請輸入數字", "錯誤!!", MessageBox
                 Buttons.OK, MessageBoxIcon. Error);
33
34
```



```
37 private void btnOk_Click(object sender, EventArgs e)
38 {
    IblSum.Text = "雨數加相為:" + (double.Parse(txtNum1.Text) +
39
                double.Parse(txtNum2.Text)).ToString();
40 }
41
42 private void btnEnd_Click(object sender, EventArgs e)
43 {
    Application.Exit();
44
45 }
```



11.2.2 KeyDown 和 KeyUp 事件

一. KeyDown 事件

- 當控制項收到按鍵被按下時所觸動的事件。
- 寫法:

```
private void 物件_KeyDown(object sender, KeyEventArgs e) {
```

// 程式區塊

}

二. KeuUp 事件

- 當控制項收到按鍵被放開時所觸動的事件。
- 寫法: private void 物件_KeyUp(object sender, KeyEventArgs e) {

// 程式區塊

- 第一個參數 sender 代表觸發該事件的物件。
- 第二個參數e 為 KeyEventArgs 型別。
- e 用來表示事件相關資料的物件。





■事件中引數 e 有些屬性可取得使用者按鍵情形其屬性:

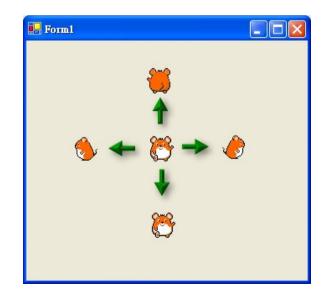
引数e屬性值	說明
e.KeyCode	取得由鍵盤所按鍵的鍵盤碼,其資料型別寫整數。
e.Alt	取得使用者是否按 All 鍵,若值寫 true 表示按All 鍵。
e.Control	取得使用者是否按 Com 鍵,若屬性值寫 true 表示按 Com 鍵。
e.Shift	取得使用者是否按 <mark>◆ Shift</mark> 鍵,若屬性値寫 true 表示按 <mark>◆ Shift</mark> 鍵。



■ 鍵盤上每個按鍵,有對應鍵盤碼 (KeyCode) 也可用Keys 列舉常數:

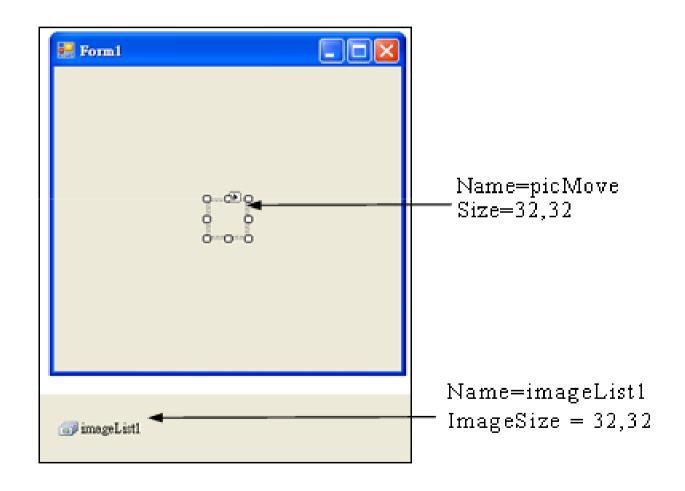
按鍵	Keys 列舉常數值	鍵盤碼(KeyCode)
0 ~ 9	Keys.D0 ~ Keys.D9	48~57
A ~ Z	Keys.A~Keys.Z	65~90
F1 ~ F10	Keys.F1 ~ Keys.F12	112~121
← 、 →	Keys.Left、Keys.Right	37、39
↑ 、↓	Keys.Up \ Keys.Down	38、40
Enter+ 、 空白鍵	Keys.Enter \ Keys.Space	13、32
☆ Shift	Keys.ShiftKey	16
Ctrl	Keys.ControlKey	17
Esc	Keys.Escape	27

範例演練





輸出入介面





```
// FileName : KeyUpDownEvent.sln
01 private void Form1_Load(object sender, EventArgs e)
02 {
    imageList1.Images.Add(Image.FromFile("up.gif"));
03
    imageList1.lmages.Add(Image.FromFile("down.gif"));
04
    imageList1.lmages.Add(lmage.FromFile("left.gif"));
05
06
    imageList1.lmages.Add(Image.FromFile("right.gif"));
07
    picMove.lmage = imageList1.lmages[1];
08 }
09
```



```
10 private void Form1_KeyDown(object sender, KeyEventArgs e)
11 {
    switch (e.KeyCode)
12
13
14
       case Keys.Up:
15
           picMove.Image = imageList1.Images[0];
16
           picMove.Top -= 5;
           break;
17
       case Keys.Down:
18
19
          picMove.lmage = imageList1.lmages[1];
          picMove.Top += 5;
20
          break;
21
       case Keys.Left:
22
23
          picMove.lmage = imageList1.lmages[2];
          picMove.Left -= 5;
24
          break;
25
```

```
() (1) (-)
```

```
26    case Keys.Right:
27       picMove.Image = imageList1.Images[3];
28       picMove.Left += 5;
29       break;
30    }
33    private void Form1_KeyUp(object sender, KeyEventArgs e)
34 {
35       picMove.Image = imageList1.Images[1];
36 }
```





11.3 滑鼠事件

事件	說明
Click	當使用者按一下滑鼠左鍵就觸發該控制項的事件。
DoubleClick	當使用者快按兩下滑鼠左鍵就觸發該控制項的事件。
MouseDown	當使用者按下滑鼠按鍵就觸發該控制項的事件。
MouseUp	當使用者放掉滑鼠按鍵就觸發該控制項的事件。
MouseEnter	當滑鼠游標進入控制項所觸發的事件。
MouseMove	當滑鼠游標在控制項中移動所觸發的事件。
MouseLeave	當滑鼠游標離開控制項所觸發的事件。



11.3.1 Click 和 DoubleClick 事件

- 當使用者按滑鼠左鍵,會依序觸發 MouseDown、Click和 MouseUp 三個事件。
- 可依照需求將程式碼寫在適當事件處理函式。當使用者 快按兩下滑鼠左鍵,也會依序觸發 MouseDown、Click、 DoubleClick 和 MouseUp 四個事件。

11.3.2 MouseDown 和 MouseUp事件

■ 當使用者按下和放開滑鼠按鍵,會分別觸發 MouseDown 事件和 MouseUp事件。在事件中引數e 有一些屬性,可提 供一些重要的訊息。



引数 ⊖ 屬性值	說明
e.Button	取得使用者按下滑鼠的哪個按鍵,其屬性值爲:
	① e.MouseButtons.Left(左鍵)
	② e.MouseButtons.Middle (中鍵)
	③ e.MouseButtons.Right(右鍵)
	若要判斷使用者是否按下滑鼠左鍵,其寫法如下:
	if(e.Button==MouseButtons.Left)
	{
	// 程式區段
	}
e.X	取得滑鼠游標的X座標值。
e.Y	取得滑鼠游標的 Y 座標值。

11.3.3 MouseEnter、MouseMove和MouseLeave事件

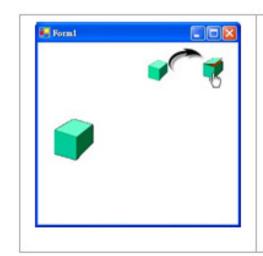
- 當使用者移動滑鼠游標到一個控制項時,會觸發 MouseEnter 事件。
- 🦏 在控制項中移動會觸發 MouseMove事件。
- ᄤ 離開控制項會觸發 MouseLeave 事件。
- 以上三個事件中引數 e 有一些屬性,可提供一些重要的訊息。

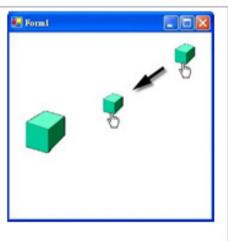
引数 ⊖屬性值	說明
e.Button	取得使用者按下滑鼠的哪個按鍵,其屬性值為: ① e.MouseButtons.Left (左鍵) ② e.MouseButtons.Middle (中鍵) ③ e.MouseButtons.Right (右鍵)
e.X	取得滑鼠游標的X座標值。
e.Y	取得滑鼠游標的Y座標值。
e.Clicks	取得使用者按鍵的次數。

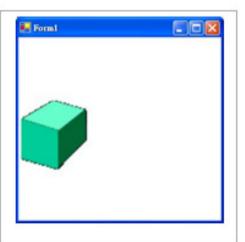
() (A) (A) (A)

範例演練

視窗左邊有個大箱子,右邊有個小箱子。當滑鼠指到小箱子時, 游標形狀會呈手型,並且蓋子會掀開。滑鼠離開時,蓋子合上。 拖曳小箱子時,小箱子會隨之移動。拖曳小箱子到大箱子中間時, 小箱子會消失且大箱子會變大,但小箱子有一部分不在大箱子中 是沒有動作。

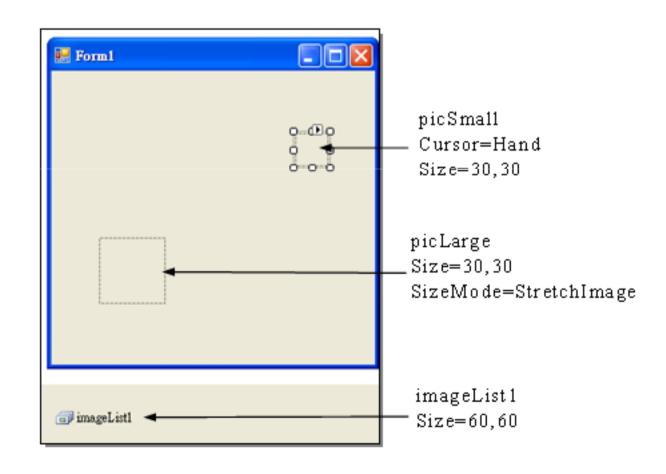








輸出入介面





```
// FileName : MouseUpDownEvent.sIn
01 bool drag = false;
02 int x down, y down;
03 private void Form1_Load(object sender, EventArgs e)
04 {
05
    imageList1.lmages.Add(lmage.FromFile("box1.gif"));
06
    imageList1.lmages.Add(lmage.FromFile("box2.gif"));
    picSmall.lmage = imageList1.lmages[0];
07
80
    picLarge.lmage = imageList1.lmages[0];
09 }
10
11 private void picSmall_MouseEnter(object sender, EventArgs e)
12 {
13
    picSmall.lmage = imageList1.lmages[1];
14 }
15
```



```
16 private void picSmall_MouseLeave(object sender, EventArgs e)
17 {
18
    picSmall.lmage = imageList1.lmages[0]:
19 }
20
21 private void picSmall MouseDown(object sender, MouseEventArgs e)
22 {
23 drag = true;
24 \times down = e.X;
25
   y_down = e.Y;
    picSmall.Image = imageList1.Images[0];
26
27 }
28
29 private void picSmall_MouseMove(object sender, MouseEventArgs e)
30 {
31
    if (drag)
32
33
                  picSmall.Top += (e.Y - y_down);
                  picSmall.Left += (e.X - x_down);
34
35
36 }
```





```
38 private void picSmall MouseUp(object sender, MouseEventArgs e)
39 {
40
    bool x_in, y_in;
41
    drag = false;
    y in = (picSmall.Top > picLarge.Top) && (picSmall.Top +
42
            picSmall.Height) < (picLarge.Top + picLarge.Height);</pre>
    x in = (picSmall.Left > picLarge.Left) && (picSmall.Left +
43
           picSmall.Width) < (picLarge.Left + picLarge.Width);
44
    if (x_in && y_in)
45
46
        picSmall.Visible = false;
        picLarge.Top -= 20;
47
        picLarge.Left -= 20;
48
49
        picLarge.Width += 40;
        picLarge.Height += 40;
50
51
52 }
```

() (A) (A) (A)

11.4 共享事件

11.4.1 使用共享事件的好處

- 若多個按鈕控制項的 Click 事件處理函式內的程式碼 除物件名稱外其它完全相同,但還要逐一輸入相同 的程式碼,非常沒效率?
- ■本節將介紹如何使多個控制項的事件去共用(共享) 一個事件處理函式,以達到簡化程式提高程式的維護。



■ 當在表單載入時觸發Form1_Load事件處理函式時,可使用+=運算子及new EventHandler來指定事件處理函式是由哪個物件的事件來觸發,例如按鈕btn1的Click事件被觸發時即執行btn1_Click事件處理函式,寫法:

```
private void Form1_Load(object sender, EventArgs e)
{
    btn1.Click += new EventHandler(btn1_Click);
}
private void btn1_Click(object sender, EventArgs e)
{
    // 程式區塊
}
```

() (1) (+) (+)

■ 如希望 btn1、btn2 和 btn3 的 Click 事件,能共同使用 btn1_Click事件處理函式,程式碼寫法:

```
private void Form1_Load(object sender, EventArgs e)
{
   btn1.Click += new EventHandler(btn1_Click);
   btn2.Click += new EventHandler(btn1_Click);
   btn3.Click += new EventHandler(btn1_Click);
}
```



■ 另外不同類別的控制項或事件也可共用事件,如希望 btn1 的 Click、btn2 的 Enter 和 textBox1 的TextChanged 事件,能一起共用 btn1_Click 事件處理函式,程式碼寫法:

```
private void Form1_Load(object sender, EventArgs e)
{
   btn1.Click += new EventHandler(btn1_Click);
   btn2.Enter += new EventHandler(btn1_Click);
   textBox1.TextChanged += new EventHandler(btn1_Click);
}
```

11.4.2 動態新增與移除事件

- 一. 新增事件語法
- 語法:物件.事件+=new EventHandler(事件處理函式);
- [例] 指定當按 btnOk 鈕觸發該鈕的 Click 事件時所要執行的是Z MyClick事件處理函式,寫法如下: btnOk.Click+=new EventHandler(MyClick);

二. 移除事件語法

- 語法:物件.事件-=new EventHandler(事件處理函式);
- [例] 欲移除 btnOk 鈕 Click 事件所要執行的 MyClick 事件處理函式, 寫法:

btnOk.Click-=new EventHandler(MyClick);



11.4.3 控制項來源的判斷

- 若當觸發共用事件時要知道是哪個物件被按下,就可在事件處理函式中將 sender 轉型成要使用的物件,接著再利用 Equals 方法來判斷。
- 例如:btn1 和 btn2 共用同一個事件處理函式,如果按 btn1 時 label1 顯示「Visual C#」;按 btn2 時 lable1 顯示「2008」,寫法

```
Button btn=(Button)sender; //sender轉型成Button再指定給btn if (btn.Equals(btn1)) {
    label1.Text = " Visual C# ";
} else {
    label1.Text = " 2008";
}
```





- Button btn=(Button)sender;
- MessageBox.Show("你按下 "+btn.Text+"鈕");

上一題



範例演練

共有16種國旗亂數出現,使用者按國名鈕選擇答案,答錯時顯示答案。

若按 🗺 鈕時,顯示下一種國旗;

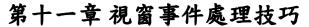
若按 上题 鈕時,顯示上一種國旗,16題答完就顯示答對題數。

當按壓斷鈕後,重新做答。



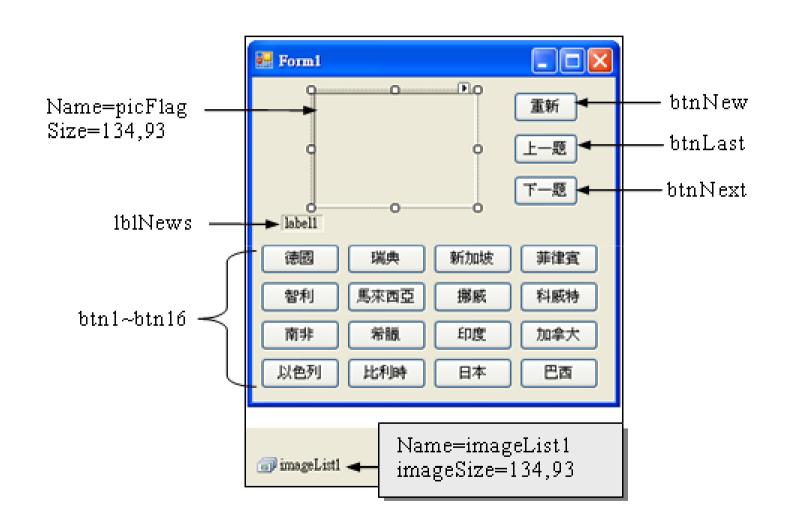














() (A () ()

```
// FileName : FlagTest.sln
01 string[] flag_name = new string[] { "德國", "瑞典", "新加坡", "菲律賓",
      "智利", "馬來西亞", "挪威", "科威特", "南非", "希臘", "印度", "加拿大",
     "以色列", "比利時", "日本", "巴西" };
02 int test_num, right_num;
03 int[] question = new int[] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 };
04
05 private void Form1 Load(object sender, EventArgs e)
06 {
    this.Text = "國旗常識大考驗!":
    for (int i = 0; i <= flag name.GetUpperBound(0); i++)
09
10
        imageList1.lmages.Add(lmage.FromFile(flag_name[i] + ".jpg"));
11
```

```
1000
```

```
12
    btn1.Click += new EventHandler(btn1 Click);
13
    btn2.Click += new EventHandler(btn1 Click);
    btn3.Click += new EventHandler(btn1 Click);
14
    btn4.Click += new EventHandler(btn1 Click);
15
16
    btn5.Click += new EventHandler(btn1 Click);
    btn6.Click += new EventHandler(btn1 Click);
17
18
    btn7.Click += new EventHandler(btn1 Click);
19
    btn8.Click += new EventHandler(btn1 Click);
    btn9.Click += new EventHandler(btn1 Click);
20
    btn10.Click += new EventHandler(btn1 Click);
21
    btn11.Click += new EventHandler(btn1_Click);
22
    btn12.Click += new EventHandler(btn1 Click);
23
    btn13.Click += new EventHandler(btn1 Click);
24
    btn14.Click += new EventHandler(btn1 Click);
25
    btn15.Click += new EventHandler(btn1 Click);
26
    btn16.Click += new EventHandler(btn1 Click);
27
28
    btnNew.Click += new EventHandler(btnNew Click);
    btnLast.Click += new EventHandler(btnNext Click);
29
30
    btnNew Click(sender, e);
31 }
```



1000

```
33 private void btn1_Click(object sender, EventArgs e)
34 {
    Button btn = (Button)sender;
35
36
    if (btn.Text == flag_name[question[test_num - 1] - 1])
37
        IblNews.Text = "你答對了!";
38
        right_num += 1;
39
40
41
    else
42
        lblNews.Text = " 你答錯了!答案是" + flag_name[question
43
                      [test num - 1] - 1];
44 }
45 }
46
```



() (t) (c) (c)

```
47 private void btnNew_Click(object sender, EventArgs e)
48 {
49
    int j, k, t;
50
    Random rnd = new Random();
    for (int i = 1; i <= 12; i++)
51
52
53
        do
54
              j = rnd.Next(16);
55
56
              k = rnd.Next(16);
          } while (j == k);
57
          t = question[j];
58
          question[j] = question[k];
59
60
        question[k] = t;
61
62
    test_num = 1; right_num = 0;
    picFlag.lmage = imageList1.lmages[question[0] - 1];
63
    lblNews.Text = "第 " + test_num + " 題";
64
65 }
```



(i) (ii) (c)

```
67 private void btnNext_Click(object sender, EventArgs e)
68 {
69
    Button btn = (Button)sender;
    if (btn.Equals(btnNext)) {
70
72
        test num += 1;
73
    else {
74
76
        test num -= 1;
        if (test_num == 0) {
77
79
          test num = 1;
80
81
    if (test_num <= 16) {
82
        lblNews.Text = "第 " + test_num + " 題";
84
        picFlag.lmage = imageList1.lmages[question[test_num - 1] - 1];
85
86
87
    else {
89
        test_num = 16;
90
        lblNews.Text = " 你已經答完全部題目!共答對" + right_num + "題";
91
92 }
```