

7.4 NumericUpDown 數字鈕控制項

 **NumericUpDown** 數字鈕控制項可以使用上/下鈕選擇數值，或是直接在文字方塊輸入一個數值，以避免輸入的資料不合乎指定的資料型別或是超出範圍而發生輸入錯誤。數值資料型別為 `Decimal`，所以可以輸入帶有小數的資料。

文字方塊 →  ← 上/下鈕

7.4.1 NumericUpDown 控制項的常用屬性

1. Value 屬性

可以設定或取得所選取或文字方塊內的數字，其資料型別為 `Decimal`，預設值為最小值(`Minimum` 屬性值)。

2. Maximum 屬性(預設值為 100)

可以設定或取得數字鈕控制項的最大值，如果使用者輸入值大於最大值時，其值會自動改為原設定值以避免超出範圍。

3. Minimum 屬性(預設值為 0)

可以設定或取得數字鈕控制項的最小值，如果使用者輸入值小於最小值時，其值會自動改為原設定值以避免超出範圍。

4. Increment 屬性(預設值為 1)

可以設定或取得按上/下鈕時每次增減的值。若屬性值為 2，則每次數字增減 2。屬性值可以為小數，但必須配合 `DecimalPlaces` 屬性才能顯示。若屬性值為 0.5，則每次數字增減 0.5。

5. DecimalPlaces 屬性(預設值為 0)

可以設定文字方塊內的數字允許顯示的小數位數，例如屬性值等於 2 表示顯示到小數二位。

6. ReadOnly 屬性(預設值為 false)

可以設定是否讓使用者自行輸入數值，若屬性值為 `false` 時，可讓使用者輸入。若屬性值為 `true` 時，則只能使用上/下鈕來選擇數值。

7.4.2 NumericUpDown 控制項的常用事件

ValueChanged 事件是 NumericUpDown 數字鈕控制項的預設事件，當在數字鈕控制項上按上/下鈕，或是直接輸入數值後，改變 Value 屬性值時，就會觸動 ValueChanged 事件。

實作 FileName : Plus.slnx

使用 NumericUpDown 控制項設計一個兩數(0 ~ 100)的加法程式，其中可以設定小數位數(0 ~ 2)，按 鈕計算兩數相加的結果。

► 輸出要求

► 解題技巧

Step 1 建立輸出入介面

1. 新增專案並以「Plus」為新專案名稱。
2. 依照輸出要求，在表單上建立控制項，如下圖所示。

Step 2 問題分析

NudPoint 的 Value 值改變時會觸動 NudPoint_ValueChanged 事件處理函式，在事件中先用 Convert.ToDecimal() 和 ToString() 方法，將

NudNo1 和 NudNo2 的 Value 屬性值轉成指定位數的數值。然後用 if 結構，分別根據 NudPoint 的 Value 屬性值來設定 NudNo1 和 NudNo2 的 Increment 屬性值，來指定增減值；以及設定 DecimalPlaces 屬性值，來指定顯示的小數位數。舉例來說：要設定 NudNo1 的增減值為 0.1，若寫法為 NudNo1.Increment = 0.1；會發生錯誤，出現下列訊息：

```
readonly struct System.Double
Represents a double-precision floating-point number.

CS0664: 不可將類型 double 的常值，隱含轉換成類型 'decimal'; 請使用 'M' 後置詞來建立此類型的常值
```

正確寫法為 NudNo1.Increment = 0.01M; (參閱第 2.4 節說明)。

Step 3 編寫程式碼

FileName : Plus.slnx

```
01 namespace Plus
02 {
03     public partial class Form1 : Form
04     {
05         public Form1()
06         {
07             InitializeComponent();
08         }
09         private void Form1_Load (object sender, EventArgs e)
10         {
11             LblAns.Text = "0";           //預設答案為 0
12             NudPoint.Maximum = 2;      //設 NudPoint 的最大值 = 2
13             NudPoint.ReadOnly = true;  //設 NudPoint 只能按上下鈕
14         }
15         private void BtnOK_Click(object sender, EventArgs e)
16         {
17             decimal n1 = NudNo1.Value;  //n1=NudNo1 的 Value 值
18             decimal n2 = NudNo2.Value;  //n2=NudNo2 的 Value 值
19             //根據 NudPoint.Value 來格式化顯示的數值
```

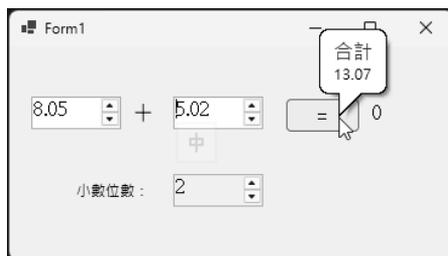
```

20         LblAns.Text = (n1 + n2).ToString($"F{NudPoint.Value}");
21     }
22     private void NudPoint_ValueChanged (object sender, EventArgs e)
23     { //NudPoint 的 Value 值改變時
24         string frm = $"F{NudPoint.Value}"; //轉換為帶格式的字串
25         NudNo1.Value = Convert.ToDecimal((NudNo1.Value).ToString(frm));
26         NudNo2.Value = Convert.ToDecimal((NudNo2.Value).ToString(frm));
27         if (NudPoint.Value == 0) //若 NudPoint.Value = 0
28         { //設 NudNo1 和 NudNo2 的增減值為 1
29             NudNo1.Increment = NudNo2.Increment = 1;
30             //設 NudNo1 和 NudNo2 顯示到整數
31             NudNo1.DecimalPlaces = NudNo2.DecimalPlaces = 0;
32         }
33         else if (NudPoint.Value == 1)
34         { //設 NudNo1 和 NudNo2 的增減值為 0.1
35             NudNo1.Increment = NudNo2.Increment = 0.1M;
36             //設 NudNo1 和 NudNo2 顯示到小數一位
37             NudNo1.DecimalPlaces = NudNo2.DecimalPlaces = 1;
38         }
39         else
40         { //設 NudNo1 和 NudNo2 的增減值為 0.01
41             NudNo1.Increment = NudNo2.Increment = 0.01M;
42             //設 NudNo1 和 NudNo2 顯示到小數二位
43             NudNo1.DecimalPlaces = NudNo2.DecimalPlaces = 2;
44         }
45     }
46 }
47 }

```

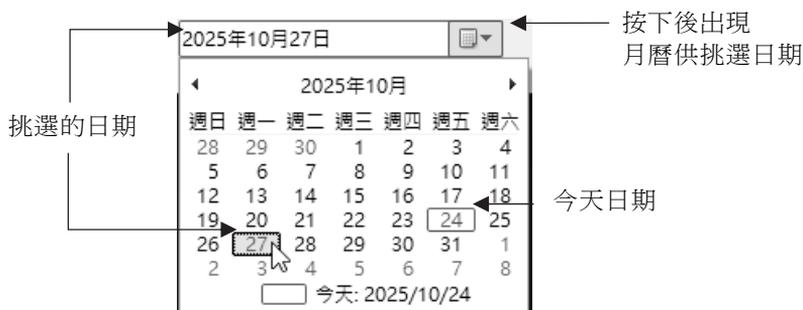
► 馬上練習

延續實作，加入 ToolTip 提示控制項。當滑鼠指標停留在 鈕 0.5 秒後，會以氣球方式顯示快顯視窗，視窗內容為兩數相加的結果。



7.6 DateTimePicker 日期時間挑選控制項

 **DateTimePicker** 日期時間挑選控制項，可以快速建立一個供使用者挑選日期或時間的介面。為節省佔用表單空間 DateTimePicker 控制項右邊有一個下拉鈕，按下後才出現月曆供使用者挑選日期。文字方塊中會顯示使用者挑選的日期，使用者也可以自行用鍵盤輸入日期。MonthCalendar 控制項可以選擇一段日期，但 DateTimePicker 控制項只能挑選一天。



7.6.1 DateTimePicker 控制項的常用屬性

1. Value 屬性(預設值為今天日期)

可以設定或取得挑選的日期。Value 屬性值的資料型別為 DateTime，其值含日期和時間。雖然將 Format 屬性設為 Short，其屬性值依然會包含時間。若只要部分日期或時間資料，就必須使用日期時間類別或 Format 格式。

例 使用 Hour 屬性取得使用者挑選的小時。

```
int what_hour = dateTimePicker1.Value.Hour;
```

例 使用 Format 格式取得使用者挑選的日期字串。

```
string what_date = dateTimePicker1.Value.ToString("yyyy/MM/dd");
```

2. Format 屬性(預設值為 Long)

可以設定日期和時間的顯示格式，其屬性值有：

① Long：以長日期格式顯示日期

② Short：以短日期格式顯示日期

③ Time：只顯示時間

④ Custom：配合 CustomFormat 屬性，自訂格式顯示日期或時間。

3. CustomFormat 屬性(預設值為空白)

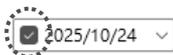
可以自訂日期或時間顯示格式，但必須將 Format 屬性值設為 Custom 才有效。例如：將屬性值設為「西元 yyyy 年 MM 月 dd 日 (ddd)」，則會顯示為：

例 在程式執行階段設定 dateTimePicker1 為自訂的時間格式，執行結果為：

```
dateTimePicker1.Format = DateTimePickerFormat.Custom;
dateTimePicker1.CustomFormat = "HH 點 mm 分 ss 秒";
```

4. ShowCheckBox 屬性(預設值為 false)

可以設定是否顯示核取方塊。若屬性值為 true 時表示顯示。



5. Checked 屬性(預設值為 true)

當 ShowCheckBox 屬性值為 true 時，Checked 屬性可設定或取得核取方塊是否勾選。

6. ShowUpDown 屬性(預設值為 false)

可以設定日期的挑選方式。

① false：用下拉鈕形式顯示月曆供挑選日期

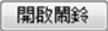
② true：用上下鈕來挑選日期。使用時先點選項目，然後按上下鈕調整

7.6.2 DateTimePicker 控制項的常用事件

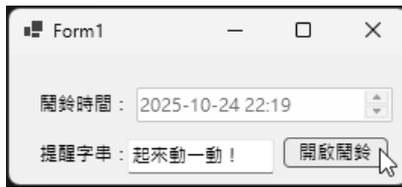
ValueChanged 事件是 DateTimePicker 控制項的預設事件，當控制項的 Value 屬性值改變時，就是當使用者改變挑選日期或時間時就觸動本事件。但是如果使用者所挑選的日期早於 MinDateTime 或晚於 MaxDateTime 的時候，則不會觸動 ValueChanged 事件。



實作 FileName: Alarm.slnx

設計一個定時提醒程式。最小提醒時間是操作時的時間加 1 分鐘，最大時間是 7 天後。使用者可以設定提醒時間及提醒字串，設定後按  鈕開啟鬧鈴。當指定時間到達時，會發出系統音及提示訊息。

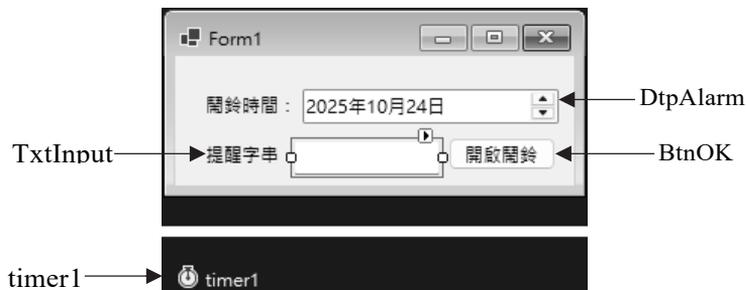
► 輸出要求



► 解題技巧

Step 1 建立輸出入介面

1. 新增專案並以「Alarm」為新專案名稱。
2. 在表單上建立下列各控制項：



Step 2 分析問題

1. 在 Form1_Load 事件處理程序中，設定 DtpAlarm 控制項以自訂的格式顯示日期和時間，及最小、最大可挑選時間。
2. 在 BtnOK_Click 事件處理程序中，設定計時器啟動，並設定 DtpAlarm 不動作。
3. 計時器啟動後，timer1_Tick 事件會每秒觸動一次，事件中要檢查目前時間是否大於等於設定時間，如果是的話，就關閉計時器，並使用 Beep.Play()函式發出系統音，最後顯示提示訊息。

Step 3 編寫程式碼

FileName : Alarm.slnx

```

01 namespace Alarm
02 {
03     public partial class Form1 : Form
04     {
05         public Form1()
06         {
07             InitializeComponent();
08         }
09
10         private void Form1_Load(object sender, EventArgs e)
11         {
12             DtpAlarm.CustomFormat = "yyyy-MM-dd HH:mm"; // 自訂格式
13             DtpAlarm.Format = DateTimePickerFormat.Custom; // 顯示自訂格式
14             DtpAlarm.MinDate = DateTime.Now.AddMinutes(1); //目前時間加 1 分鐘
15             DtpAlarm.MaxDate = DateTime.Now.AddDays(7); // 目前時間加 7 天
16         }
17
18         private void BtnOK_Click(object sender, EventArgs e)
19         {
20             timer1.Interval = 1000; // 每秒啟動計時器一次
21             timer1.Enabled = true; // 啟動計時器
22             DtpAlarm.Enabled = false; // 避免操作者更動時間

```

12.3 使用 ChatGPT 協作開發 C# 視窗程式

本小節以 C# Windows Forms 為範例，逐步帶領如何與 ChatGPT 協作，從輸出入需求、撰寫程式提示詞，到實作與測試，一步一步完成 C# 視窗程式的開發流程。



實作 FileName : lot.slnx

與 ChatGPT 協作開發一個大樂透模擬程式。系統會以亂數產生 1~49 之間的 6 個不重複號碼作為開獎結果。使用者需在介面中勾選 6 個號碼 進行投注。當使用者所選的號碼中，若有 3 個 (含) 以上與開獎號碼相同，則顯示「你中獎了，共中 x 個號碼，請到投注站領獎」訊息；否則顯示「沒中獎，請再接再勵」。

► 輸出要求

The image shows two screenshots of a Windows Forms application window titled "Form1".

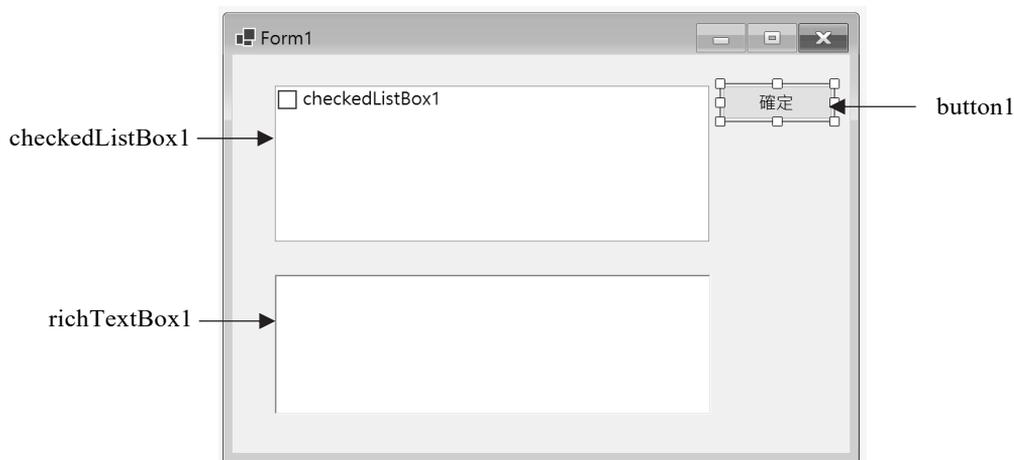
Left Screenshot: The window contains a grid of checkboxes for numbers 1 through 30. The following numbers are checked: 6, 8, 9, 10, 14, and 15. A "確定" (OK) button is visible on the right. Below the grid, a text box displays: "開獎號碼 : 31, 44, 9, 13, 26, 3", "你選的號碼 : 6, 8, 9, 10, 14, 15", and "沒中獎，請再接再勵".

Right Screenshot: The window shows the same grid after clicking "確定". The following numbers are checked: 7, 8, 12, 13, 14, and 15. The "確定" button is now disabled. The text box displays: "開獎號碼 : 4, 6, 38, 39, 40, 41", "你選的號碼 : 4, 7, 12, 13, 38, 39", and "你中獎了，共中 3 個號碼，請到投注站領獎".

► 解題技巧

Step 1 建立輸出入介面

1. 使用 Button、RichTextBox 以及 CheckedListBox 製作下圖輸出入介面。
2. CheckedListBox 結合了 清單(ListBox) 與 核取方塊(CheckBox) 的功能，可進行多重勾選。本例使用此控制項讓使用者勾選要投注的號碼



Step 2 使用 ChatGPT 生成需要的程式碼

1. 請 ChatGPT 扮演資深的 C# 程式設計師，提示詞如下：

你是一位資深的 C# 程式設計師，請您之後以此身份回答問題

2. 依據本例的需求，請 ChatGPT 提供對應的程式碼。

在撰寫提示詞 (Prompt) 時，建議明確說明輸入與輸出介面名稱、功能需求等細節，以確保生成的程式更貼近實際需求。若功能內容較多、不易以一段文字完整描述，建議以條列方式清楚列出各項需求。提示詞如下：

依下面說明提供程式碼，並在程式碼後面加上註解

在 Form1 表單中有以下控制項：

1. checkedListBox1
2. richTextBox1
3. button1 ([確定] 按鈕)

請撰寫一段 C# 程式，完成以下功能：

1. 表單載入時在 checkedListBox1 中顯示 1 至 49 的號碼，checkedListBox1 中的號碼使用水平排列。
2. 按下 [確定] 按鈕 (button1) 時，需執行下列步驟：

- (1) 檢查使用者是否勾選 6 個號碼；若未勾選 6 個號碼，則在 richTextBox1 顯示「無法投注，請檢查是否勾選 6 個號碼」，並結束事件處理。
- (2) 隨機產生 1~49 不重複的亂數號碼，並存入 lot 陣列中。
- (3) 逐一比較 checkedListBox1 被勾選的號碼是否存在於 lot 陣列中。若有超過 3 (含)個號碼相同，則在 richTextBox1 顯示：「你中獎了，共中 x 個號碼，請到投注站領獎」。否則顯示：「沒中獎，請再接再勵」。

3. 接著 ChatGPT 會生成 Form1_Load 和 button1_Click 事件處理函式，可將生成的程式碼複製到對應的事件內。若提供的程式碼有誤，可進行修正。

Step 3 本例中筆者生成的程式碼如下

要特別注意，ChatGPT 每次所生成的程式碼可能略有不同，建議讀者應具備基本的程式判讀與調整能力，在使用 ChatGPT 生成程式碼前先檢視並確認程式內容是否符合專案需求。

FileName : Form1.cs

```

01 namespace lot
02 {
03     public partial class Form1 : Form
04     {
05         public Form1()
06         {
07             InitializeComponent();
08         }
09
10         private void Form1_Load(object sender, EventArgs e)
11         {
12             // 設定 CheckedListBox 顯示 1~49 的號碼
13             for (int i = 1; i <= 49; i++) // 由 1 跑到 49
14             {
15                 checkedListBox1.Items.Add(i.ToString()); // 將數字轉為字串加入清單中
16             }
17         }

```

```
18         // 設定 CheckedListBox 為水平排列模式
19         checkedListBox1.MultiColumn = true;    // 啟用多欄顯示
20         checkedListBox1.ColumnWidth = 100;    // 每欄寬度設定為 40 像素
21     }
22
23     private void button1_Click(object sender, EventArgs e)
24     {
25         // 取得使用者勾選的項目數量
26         int selectedCount=checkedListBox1.CheckedItems.Count; // 計算勾選數量
27
28         // 檢查是否剛好勾選 6 個號碼
29         if (selectedCount != 6)
30         {
31             // 顯示錯誤訊息
32             richTextBox1.Text = "無法投注，請檢查是否勾選 6 個號碼";
33             return; // 結束事件，不再繼續執行
34         }
35
36         // 建立亂數物件
37         Random rnd = new Random();
38
39         // 建立 1~49 的號碼清單
40         List<int> pool = Enumerable.Range(1, 49).ToList();
41
42         // 用於存放亂數開獎結果的陣列
43         int[] lot = new int[6]; // 開獎 6 個號碼
44
45         // 從 pool 隨機抽取 6 個不重複號碼
46         for (int i = 0; i < 6; i++)
47         {
48             int index = rnd.Next(pool.Count); //隨機取一個索引值
49             lot[i] = pool[index]; //將該號碼放入開獎結果
50             pool.RemoveAt(index); //從池中移除該號碼以避免重複
51         }
52         //指定 lot 陣列中獎號碼為 1,2,3,4,5,6，測試用
53         //lot = new int[6] { 1,2,3,4,5,6};
54
55         // 將勾選的號碼轉為整數清單
56         List<int> selectedNumbers = new List<int>();
57         foreach (var item in checkedListBox1.CheckedItems)
58         {
59             // 將字串轉為整數並加入清單
```

```
60         selectedNumbers.Add(int.Parse(item.ToString()));
61     }
62
63     // 計算中獎號碼數量，比對交集並計算相同數量
64     int matchCount = selectedNumbers.Intersect(lot).Count();
65
66     // 清空 richTextBox1
67     richTextBox1.Clear();
68
69     // 顯示開獎結果
70     richTextBox1.AppendText("開獎號碼：" + string.Join(", ", lot) +
        Environment.NewLine); // 顯示開獎號碼
71     richTextBox1.AppendText("你選的號碼：" + string.Join(", ", selectedNumbers)
        + Environment.NewLine); // 顯示玩家選號
72
73     // 判斷是否中獎
74     if (matchCount >= 3) // 若中 3 個（含）以上
75     {
76         // 顯示中獎訊息
77         richTextBox1.AppendText
            ($"你中獎了，共中 {matchCount} 個號碼，請到投注站領獎");
78     }
79     else
80     {
81         richTextBox1.AppendText("沒中獎，請再接再勵"); //顯示未中獎訊息
82     }
83     }
84 }
85 }
```

► 說明

本例中，系統每次投注時都會隨機產生 1~49 之間的 6 個號碼。由於亂數結果不可預測，測試時要選中系統號碼相對困難。因此，可在第 53 行暫時設定固定的中獎號碼，以方便測試與驗證程式邏輯。待確認程式運作正常後，再將該行程式碼註解或移除。