

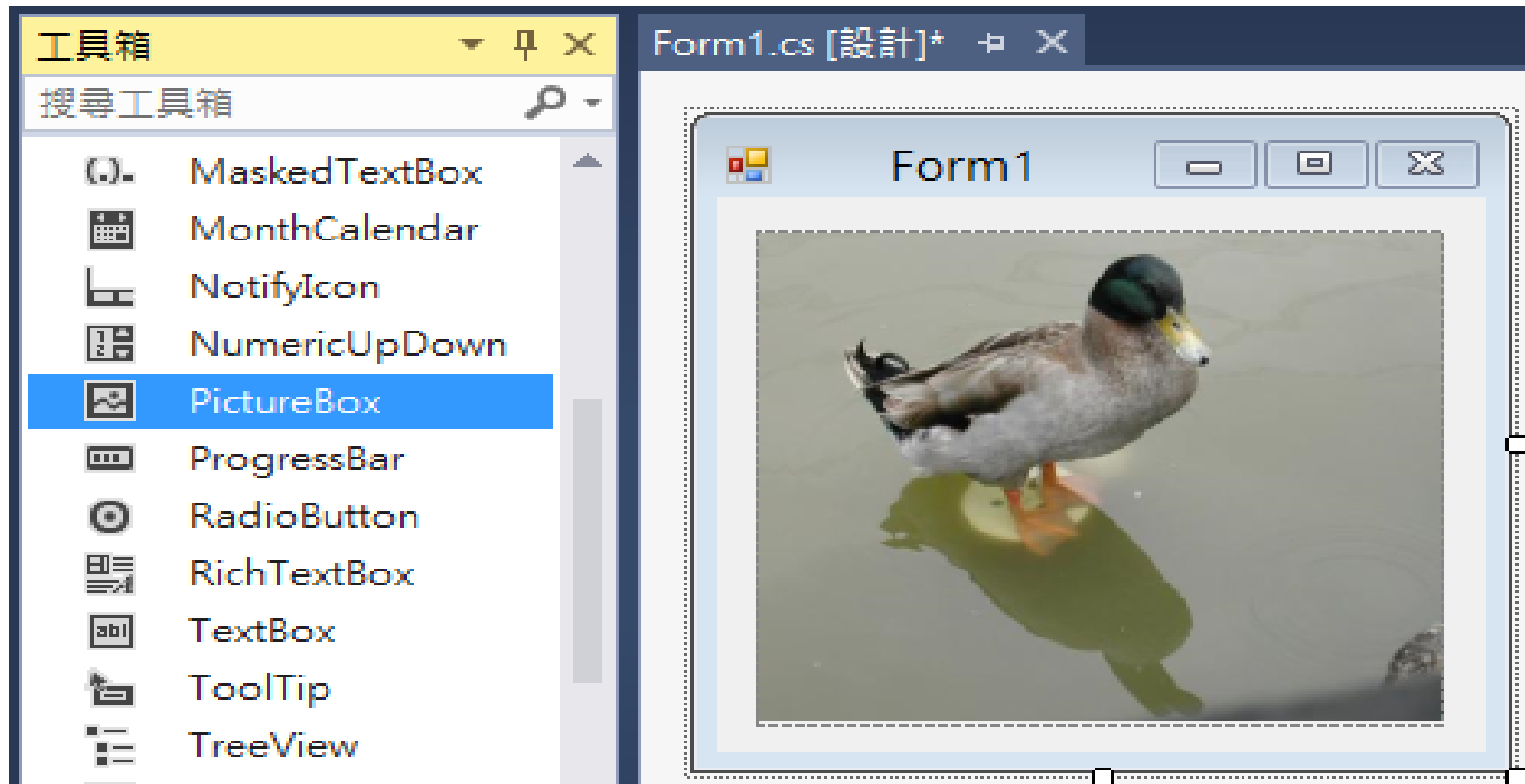


Chapter 7

Windows Basic Control Item
Multi Forms

7-1 PictureBox Control Item

- Usage
 - ⇒ draw graph or load image
 - ⇒ show animation



PictureBox Properties

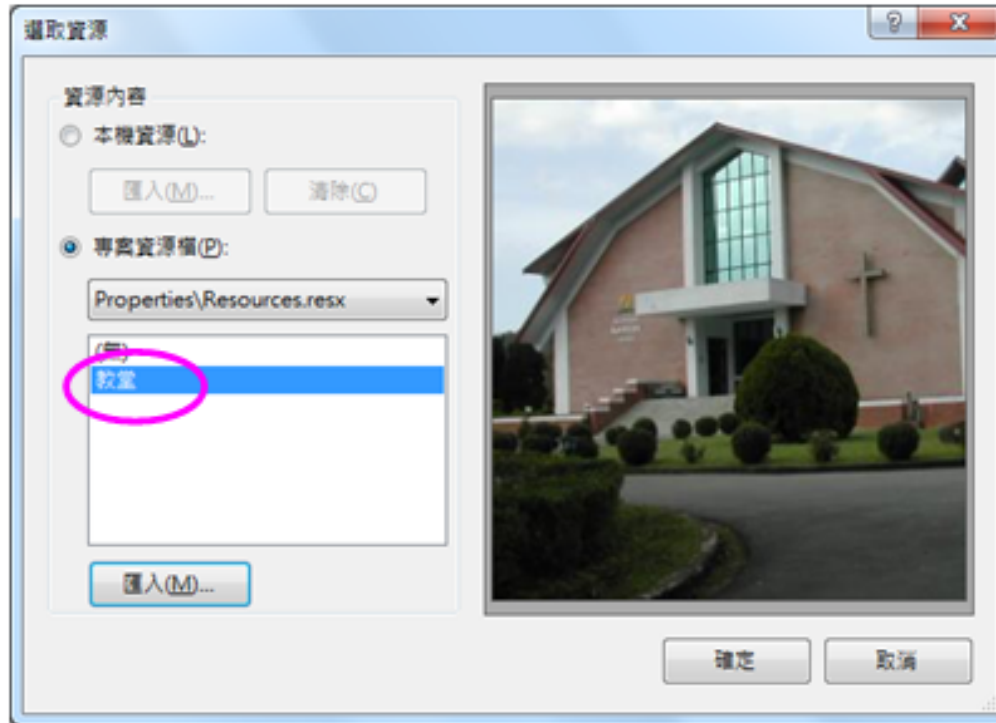
Property	Description
Image	Assign image file name
BorderStyle	None: no border (default) FixedSingle: single line Fixed3D: 3-D line
Location.X	The horizontal coordinates of the upper-left corner of the control relative to the upper-left corner of its container
Location.Y	The vertical coordinates of the upper-left corner of the control relative to the upper-left corner of its container
Size.Width	PictureBox's width (can be abridged as Width)
Size.Height	PictureBox's height (can be abridged as Height)

Property	Description
SizeMode	<p>Normal: The image is placed in the upper-left corner of the PictureBox. (default)</p> <p>StretchImage: The image is stretched or shrunk to fit the size of the PictureBox.</p> <p>AutoSize: The PictureBox is sized equal to the size of the image</p> <p>CenterImage: The image is displayed in the center</p> <p>Zoom: The size of the image is increased or decreased maintaining the size ratio.</p> <p>Ex: <code>pic.SizeMode = PictureBoxSizeMode.Zoom;</code></p>

3. Image Format of PictureBox

- **Acceptable image format:
Bitmap 、 GIF 、 JPEG 、 PNG...etc.**
- **Bitmap(.bmp) is an uncompressed bitmap image**
- **GIF(.gif) only contains 256 colors; animation and transparency is acceptable**
- **JPEG(.jpg) is a compressed bitmap image**
- **PNG(.png) combines the characteristics of GIF and JPEG**

Set Image File in Design Phase



CAUTION: Click on “本機資源” option, the chosen image file cannot be copied to project directory. Users have to copy image files separately if the project directory is copied. If “專案資源檔” is chosen, no additional duplication is required

PictureBox Methods

Method	Description
<code>Image.FromFile</code> or <code>new Bitmap</code>	Load image file with full path when program is running <code>Pic.Image = Image.FromFile(@"D:\fig\ex1.jpg");</code> or <code>Pic.Image = new Bitmap(@"D:\fig\ex1.jpg");</code> Ps: <code>@"D:\fig\ex1.jpg"</code> can also be written in <code>"D:\\fig\\ex1.jpg"</code>
<code>Point()</code> constructor	Move PictureBox object to the specific X-Y coordinate Ex1: <code>pic.Location = new Point(50, 100);</code> Ex2: <code>pic.Location = new Point(pic.Left - 5, pic.Top + 5);</code> or <code>pic.Left -= 5; pic.Top += 5;</code>

Method	Description
Size() constructor	Set the width and height of the image Ex: pic.Size = new Size(pic.Width / 2, pic.Height / 2); or pic.Width /= 2; pic.Height /= 2;
Dispose()	Clear the content of PictureBox and release the memory space Ex: pic.Dispose(); or pic.Image = null;

Load Image at Runtime

1. Image and executable file are under the same directory:

```
① pic.Image = Image.FromFile ("教堂.jpg") ;  
② pic.Image = new Bitmap ("教堂.jpg") ;
```

2. Image is at the parent directory of the executable file

```
① pic.Image = Image.FromFile (@"..\\教堂.jpg") ;  
② pic.Image = new Bitmap (@"..\\教堂.jpg") ;
```

3. Image and project file is under the same directory:

```
① pic.Image = Image.FromFile (@"..\\..\\教堂.jpg") ;  
② pic.Image = new Bitmap (@"..\\..\\教堂.jpg") ;
```

Image 類別

- **Make the file into the image class**

```
Image myPic = Image.FromFile(@"D:\fig\ex1.jpg");
```

- **Make the image show in the pictureBox**

```
pictureBox1.Image = myPic;
```

Practice(PictureBox):

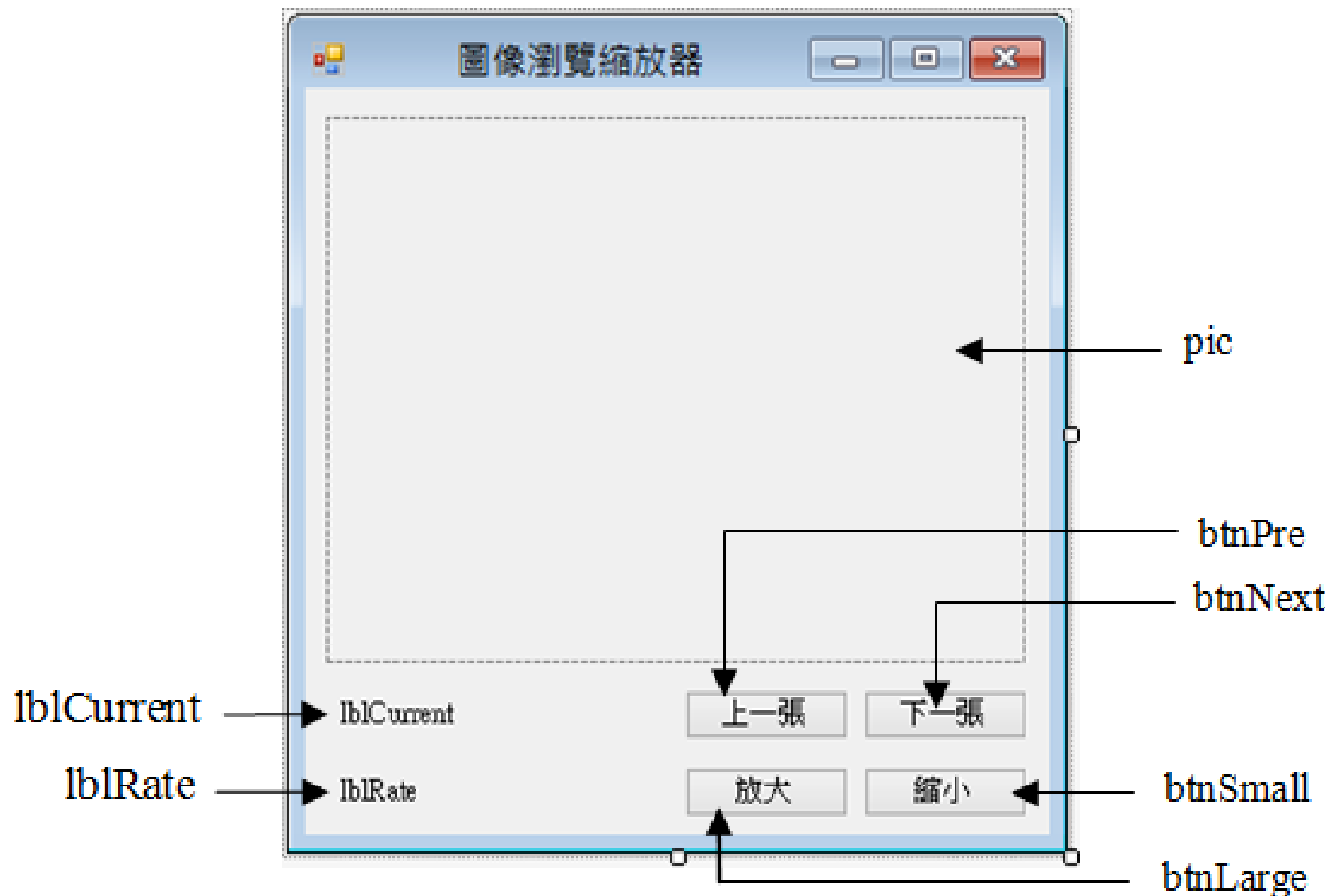
Design an image viewer, requirements:

1. There are 6 images (pic1.jpg ~ pic6.jpg). The program starts and loads the first image. The property SizeMode is set to Zoom
2. Change the showing image when pressing “上一張” or “下一張”. If the current image is the first image, the button “上一張” is unavailable. If the current image is the last image, the button “下一張” is unavailable
3. Every time press “放大” button, the image is enlarged 10%, the maximum is 100%.
Every time press “縮小” button, the image is shrunk 10%, the minimum is 10%.
The program shows image in 80% when starting. The image's center position is unchanged during zooming
4. Use lblCurrent label control item to show the order of images. Use lblRate label control item to show the rate of zooming

Result:

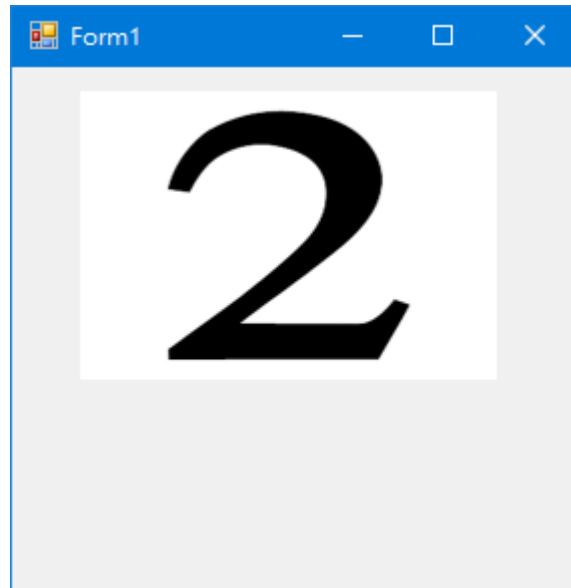
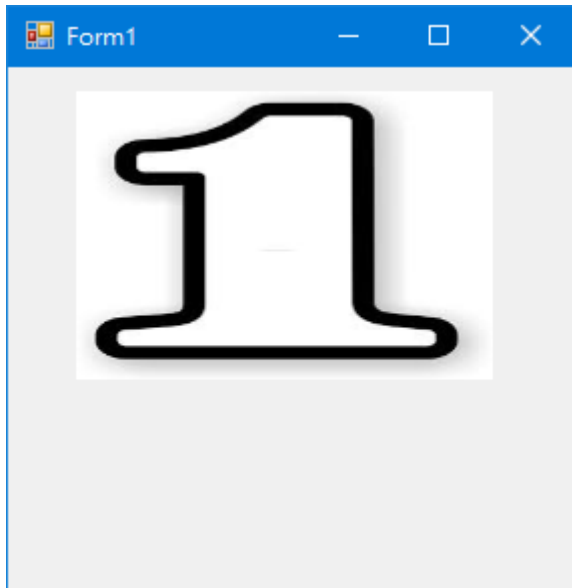


Design User Interface



Practice(pictureBox)

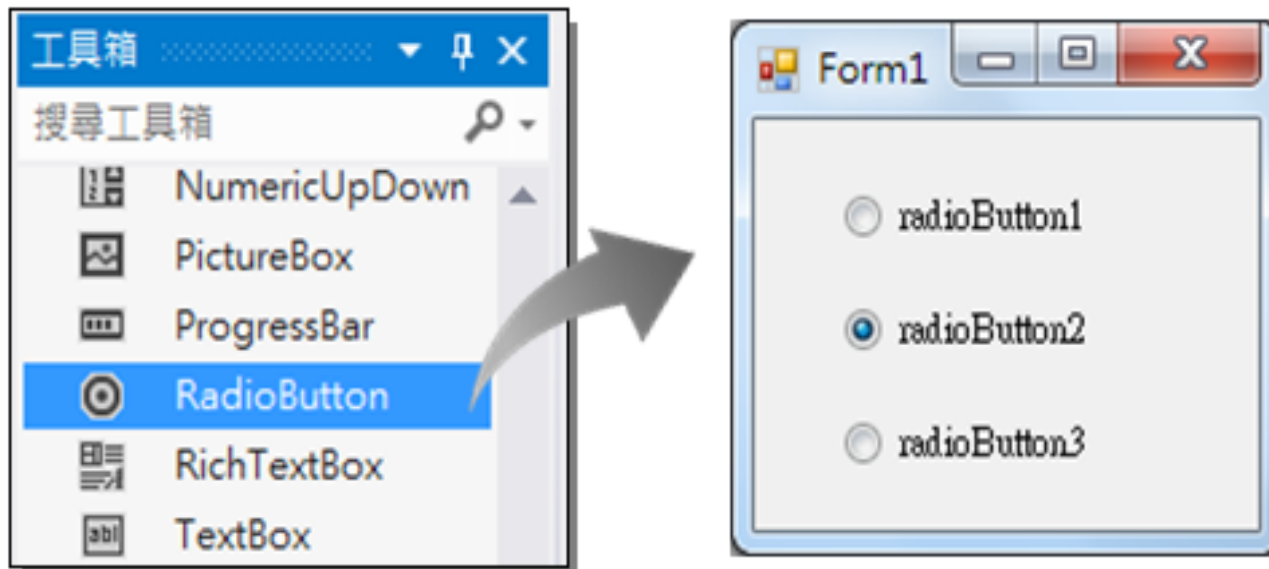
- Use pictureBox to load some pics.
- Use mouse click event to change the pics which are taken from the pictureBox.



7-2 RadioButton Control Item

Usage:

there are many related options and only a single choice is allowed



Form1

性別

☒ 男

☐ 女

學歷

☐ 小學 ☒ 大學

☐ 國中 ☐ 研究所

☐ 高中

2 choices

5 choices

Form1

☐ 男

☐ 女

☐ 小學 ☐ 大學

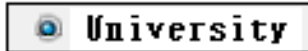



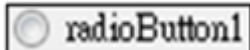

☐ 國中 ☐ 研究所

☒ 高中

7 choices

RadioButton Properties

Property	Description
Checked	<p>False: not selected (default) True: selected</p> <p>Usage: radioButton1.Checked = true;</p>
AutoCheck	<p>True: Checked property can be selected automatically as clicked False: Checked property cannot be selected automatically as clicked</p> <p>Usage: radioButton1.AutoCheck = true;</p>

Text	<p>Set the showing text of the control item</p> <p>Usage: radioButton1.Text = "University"; </p>										
TextAlign	<p>Set the alignment of the control item's showing text. Notice: AutoSize property has to be False</p> <table><tr><td>TopLeft</td><td>TopCenter</td><td>TopRight</td><td rowspan="3"></td></tr><tr><td>MiddleLeft</td><td>MiddleCenter</td><td>MiddleRight</td></tr><tr><td>BottomLeft</td><td>BottomCenter</td><td>BottomRight</td></tr></table> <p>Usage: radioButton1.TextAlign = ContentAlignment.MiddleRight;</p>	TopLeft	TopCenter	TopRight		MiddleLeft	MiddleCenter	MiddleRight	BottomLeft	BottomCenter	BottomRight
TopLeft	TopCenter	TopRight									
MiddleLeft	MiddleCenter	MiddleRight									
BottomLeft	BottomCenter	BottomRight									
Appearance	<p>Normal: small circle button and text (default) </p> <p>Button: text surrounded by a frame </p> <p>Usage: radioButton1.Appearance = Appearance.Button;</p>										
CheckAlign	<p>Set the alignment of small circle button when Appearance = Normal, 9 position like TextAlign property</p>										

Ex:

set the alignment of radioButton1's circle button TopCenter and the text's alignment BottomCenter

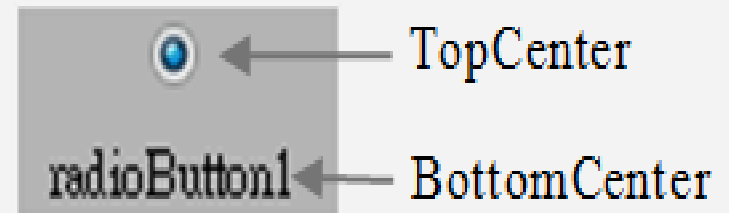
```
radioButton1.AutoSize = false;
```

```
radioButton1.Width = 100;
```

```
radioButton1.Height = 40;
```

```
radioButton1.CheckAlign = ContentAlignment.TopCenter;
```

```
radioButton1.TextAlign = ContentAlignment.BottomCenter;
```





RadioButton Events

1. **CheckedChanged**

⇒ default event

⇒ triggered when mouse clicked or Checked changed

⇒ if the radio button is already checked and mouse clicks – no event triggered because the value of Checked keeps True

2. **Click**

triggered when the radio button is clicked

Practice(RadioButton):

Design a program for computing 4 fundamental operations of arithmetic, requirements:

1. Choose a manner to calculate. After entering numbers and pressing “=” button, show the result
2. Use Label control item to show the result. Properties:
 - ① Background color: DarkRed
 - ② Foreground color: White
 - ③ Font size: 14pt
3. Label and TextBox control items align text to center
4. If 2 TextBoxs' value are not number as the “=” button is pressed, use try...catch statement to handle the exception and show the error message “請輸入數值!”
5. Digits after the floating point are not limited
6. The divisor cannot be 0 when using division. Show error message “除數不能為零” if the divisor is 0

Result:

四則運算

☒ 加法 ☐ 減法 ☐ 乘法 ☐ 除法

20 + 10

= 30

錯誤訊息

請輸入數值!

確定

錯誤訊息

除數不能為零

確定

Design User Interface



7-3 CheckBox Control Item

- **Single choice, multi choices and no choice are allowed**
- **As a check box is checked**
 - ⇒ **property Checked becomes True**
 - ⇒ **other check boxes are not affected**
- **If click on the checked check box**
 - ⇒ **become unchecked**
 - ⇒ **property Checked becomes False**

CheckBox Properties

Property	Description
Checked	False: unchecked (default) <input type="checkbox"/> True: checked <input checked="" type="checkbox"/>
ThreeState	True: 3 states: Checked <input checked="" type="checkbox"/> , Unchecked <input type="checkbox"/> , Indeterminate <input type="checkbox"/> False: 2 states: Checked, Unchecked (default)
CheckState	Set the current state of the check box according to ThreeState property 3 property values: Unchecked (default), Checked, Indeterminate



2. CheckBox Events

1. **CheckedChanged**

⇒ default event of **CheckBox**

⇒ triggered when **Checked** changes value

2. **CheckStateChanged**

triggered when **CheckState** changes value

3. **Click**

triggered when mouse click on the check box

Practice(exchange):

Design a change exchanger, requirements:

1. The check boxes of 3 types of change is unchecked when the program starts. \$1 is a necessary type. 4 TextBoxes for showing result is set read-only
2. Users can input the number of money and select the type of change they want

零錢兌換機

金額:

<input type="checkbox"/> \$50	<input type="checkbox"/> \$10	<input type="checkbox"/> \$5	\$1
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

零錢兌換機

金額:

<input checked="" type="checkbox"/> \$50	<input type="checkbox"/> \$10	<input checked="" type="checkbox"/> \$5	\$1
<input type="text" value="11"/>	<input type="text"/>	<input type="text" value="3"/>	<input type="text" value="2"/>

3. The program automatically shows the number of changes in TextBoxes. Use only \$1 to get all of money if no other types of change is selected
4. Automatically show the number of changes when the option of change is checked. Show 0 if there is no input number
5. Input number can only be unsigned integer. Show the error message as the figure if the format of input number is not fit



零錢兌換機

金額： 567

☐ \$50 ☐ \$10 ☐ \$5 ☒ \$1

0 0 0 567



零錢兌換機

金額：

☒ \$50 ☐ \$10 ☒ \$5 ☐ \$1

0 0

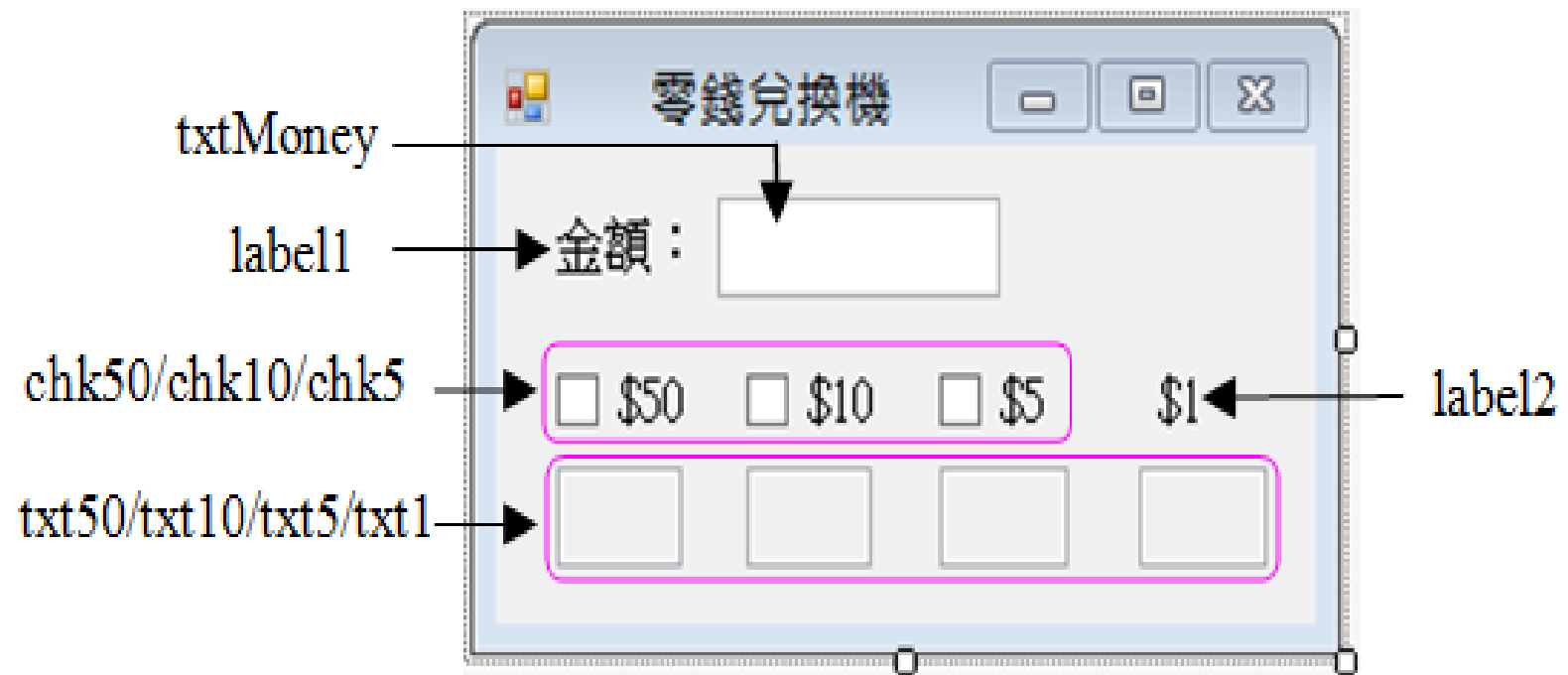


錯誤訊息

請輸入正整數值！

確定


User Interface



Practice(order):

Design a 3C shopping program, requirements:

1. Check products, enter the quantities, then press “結帳” button to get the total amount of money
2. When the form loads, set properties of the control item used to show total amount of money:
 - ① Foreground color: Color.Red
 - ② Background color: Highlight
 - ③ Font: 標楷體
 - ④ Font size: 14pt
 - ⑤ Font style: italic
 - ⑥ The total amount of money has a prefix “NT\$”, use ToString("#, #") to show formatted number
 - ⑦ Default: iPad2 check box is checked
3. Use text box for quantity input, default value is 1. Show error message “請輸入正整數值!” when the user inputs non-number character. Show error message “不能為負整數值!” when the user inputs minus number

- 
4. Use check boxes to show products, only one option is selected in the same time. If one product is checked, other products cannot be checked. Keep checked when click on checked check box, uncheck the checked check box when the use chooses other check box.
 5. When press “結帳” button, the label control item shows a message like “你購買 iPad Mini 共 2 台，共計 NT\$ 21,000 元” if the number of quantity is right.

Result:

3C 購物中心

Apple 系列產品	數量
<input type="checkbox"/> iPad2 (\$12,500元)	2
<input checked="" type="checkbox"/> iPad Mini (\$10,500元)	
<input type="checkbox"/> iPhone 5 (\$21,900元)	

結帳

你購買 iPad Mini 2台 共計:NT\$21,000元

錯誤訊息

請輸入正整數值！

確定

錯誤訊息

不能為負整數值！

確定

User Interface



Practice(radiobutton)

- Use radiobutton to create a program which can choose "clockwise rotation" or "counterclockwise rotation". The click event can decide which rotation should be chosen.
- Make a renew button to reset to the initial setting.





Multi Forms



Multi-form Programming

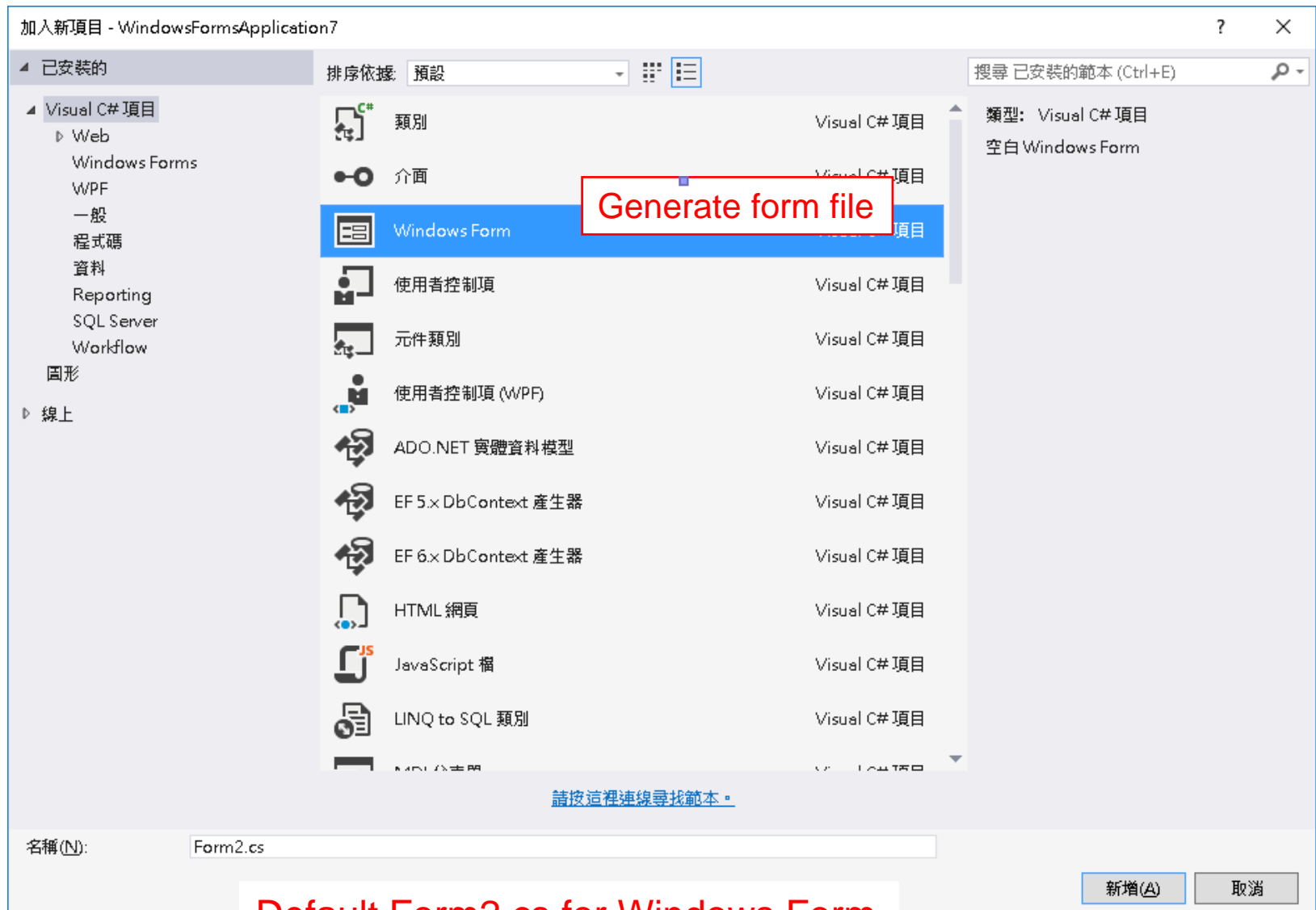
- A larger Windows program may require many forms to present user interface
- Multi-form application contains many forms in the project
- These forms many share member variables, member methods and so on. These variables and methods can be defined in class file

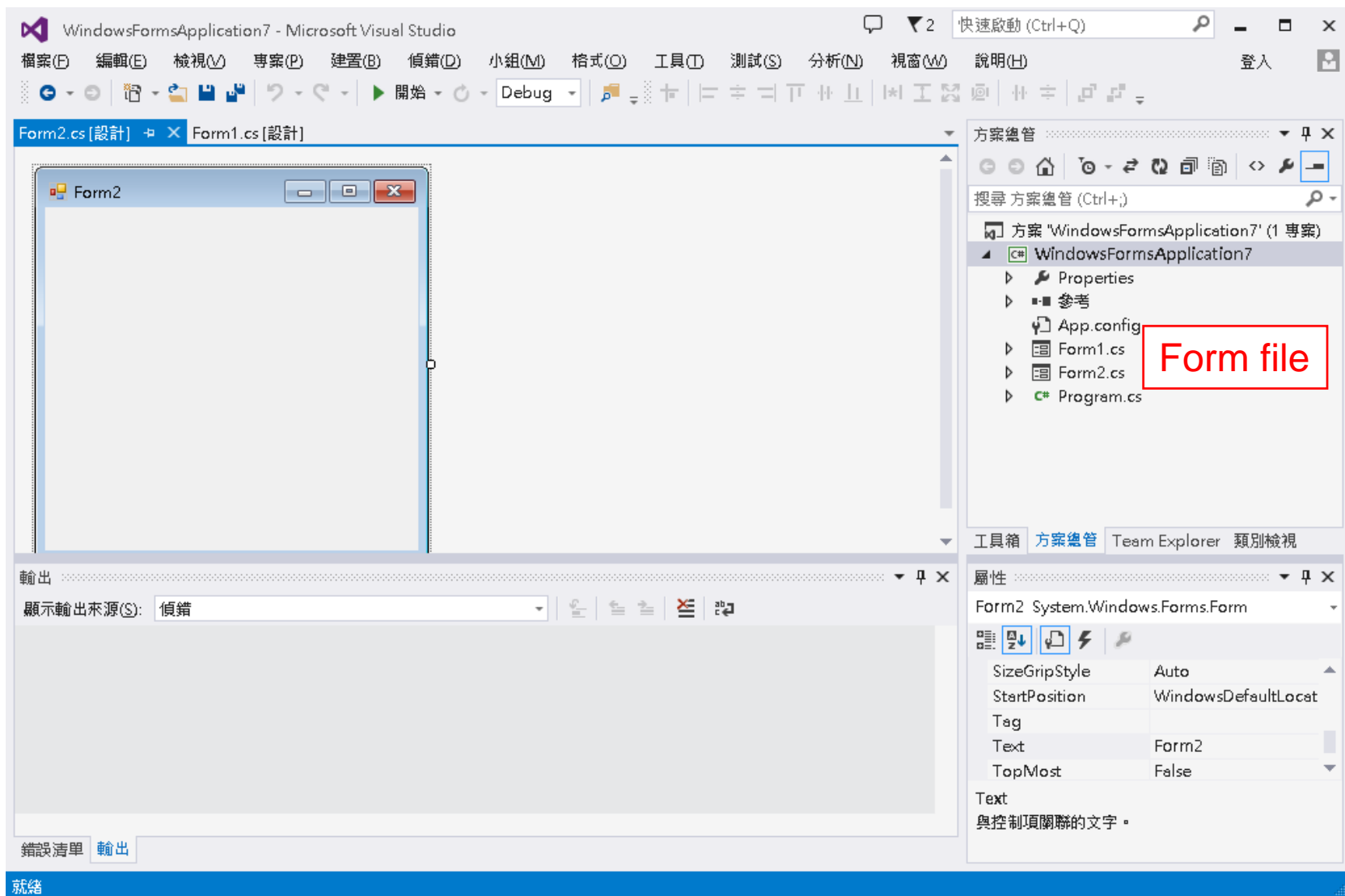


How to Add New Form

1. Add new form

Run command **【專案(P)/加入新項目(W)...】** in the main menu





How to Create Form Object

Grammar

```
formClass formName = new formClass();
```

Ex: create a Form2 object called f2, usage:

```
Form2 f2 = new Form2();
```




Multi-form Methods

1. Show()

display designated form, ex: `f2.Show();`

2. ShowDialog()

display designated form by dialog, ex:
`f2.ShowDialog();`

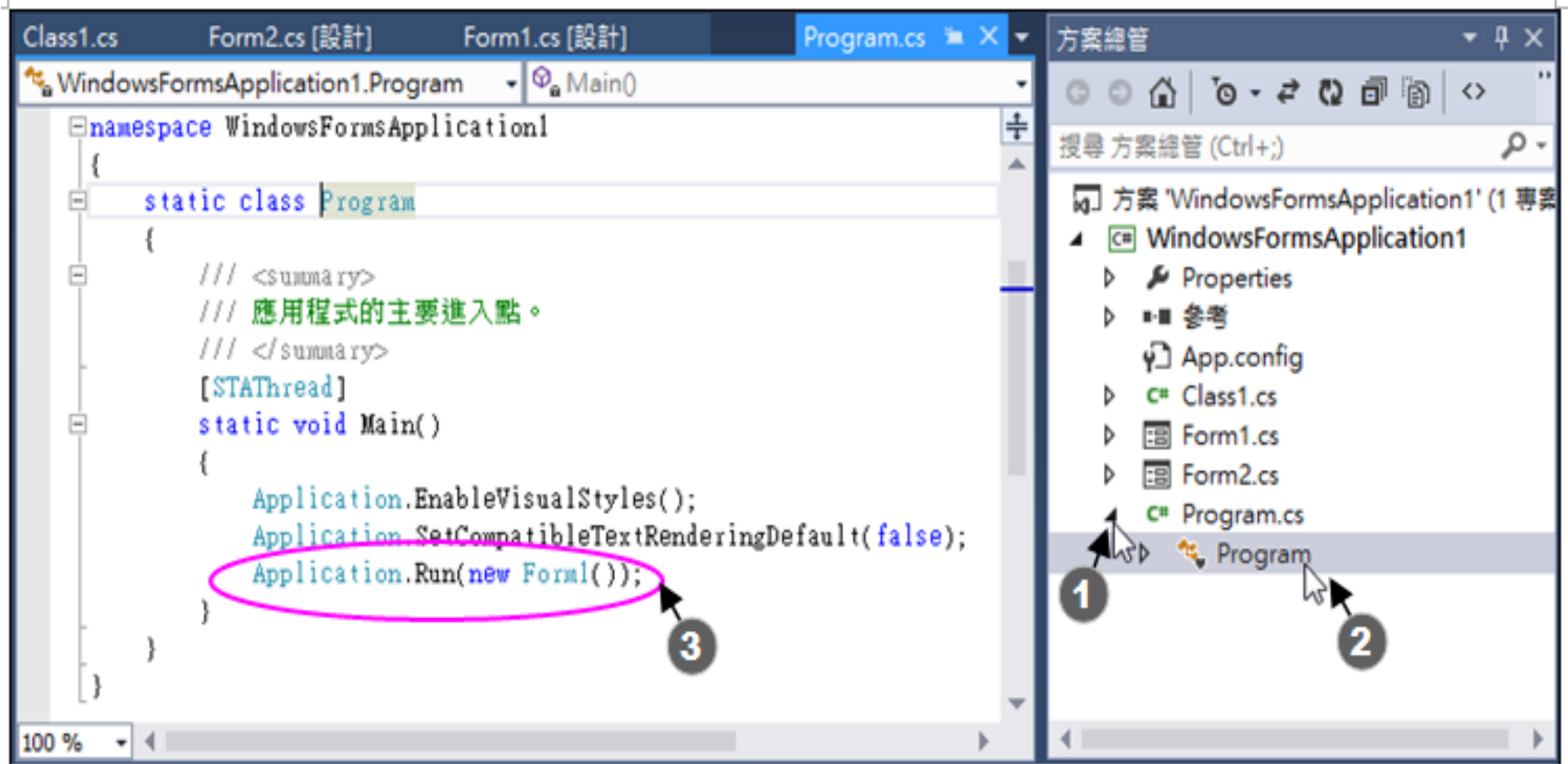
3. Hide()

hide designated form, ex: `f2.Hide();`

4. Close() 、 Dispose()

close designated form, ex: `f2.Close();` or `f2.Dispose();`

How to Set Up Initial Form

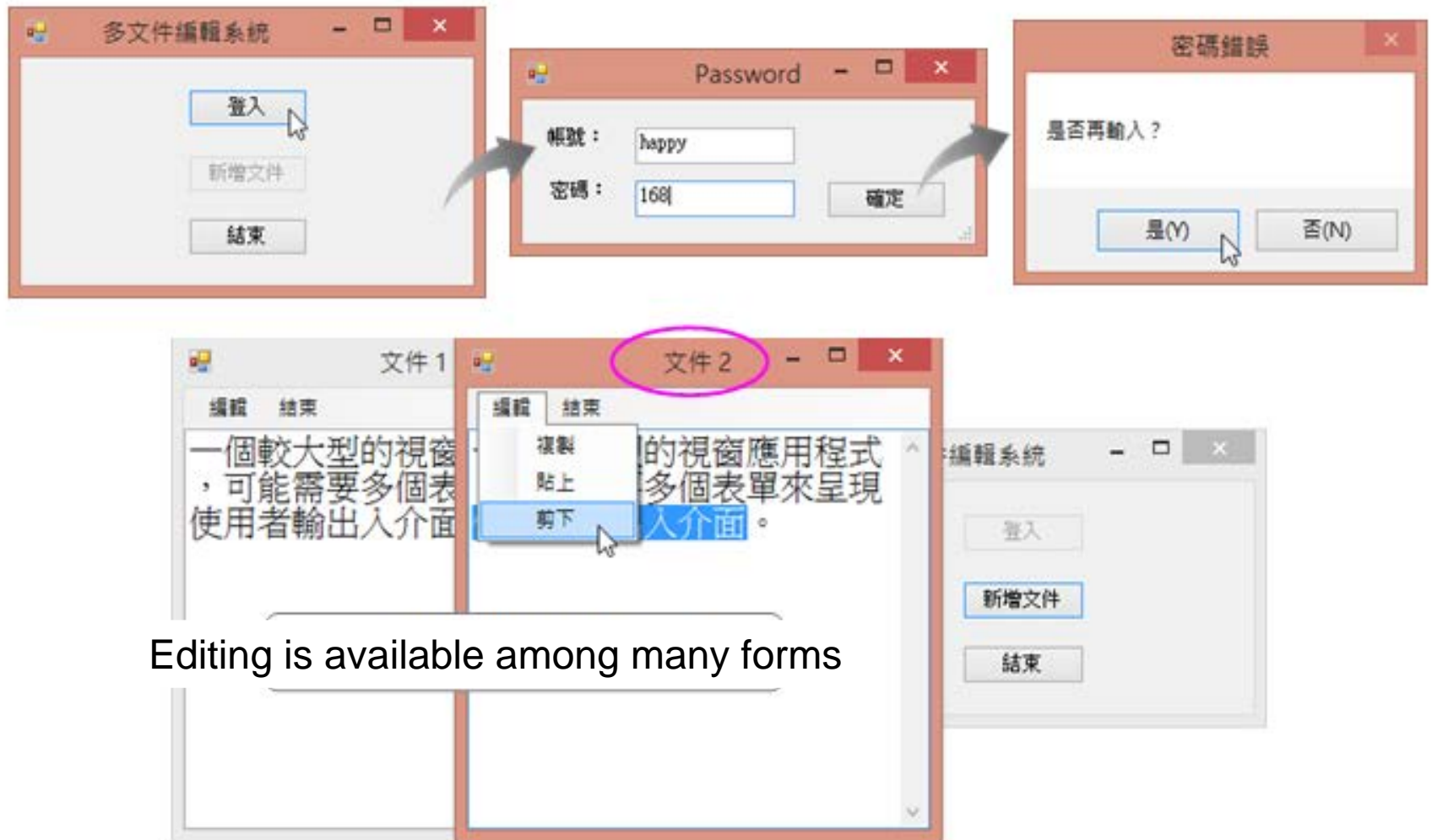


Practice(multiText):

Design a multi-document editing program, requirements:

1. Open “多文件編輯系統” window when the program starts. There are 3 buttons called “登入”, “新增文件” and “結束”. The “新增文件” button is disabled when the program starts
2. Press “登入” button to open log “Password” window, and “多文件編輯系統” window is not available
3. If the inputted account and password are correct (happy and 168), return to “多文件編輯系統” window. The “新增文件” button is enabled but “登入” button is disabled at this time. The program questions about trying again or not. The program terminates if the wrong passwords are inputted 3 times.
4. Press “新增文件” once to open a “Text” window. This window has “編輯” and “結束” main menu items. Press “結束” button to close this window
5. There are 3 sub menu items called “複製”, “貼上” and “剪下” in “編輯”. These functions can process “複製”, “貼上” and “剪下” among many “Text” windows

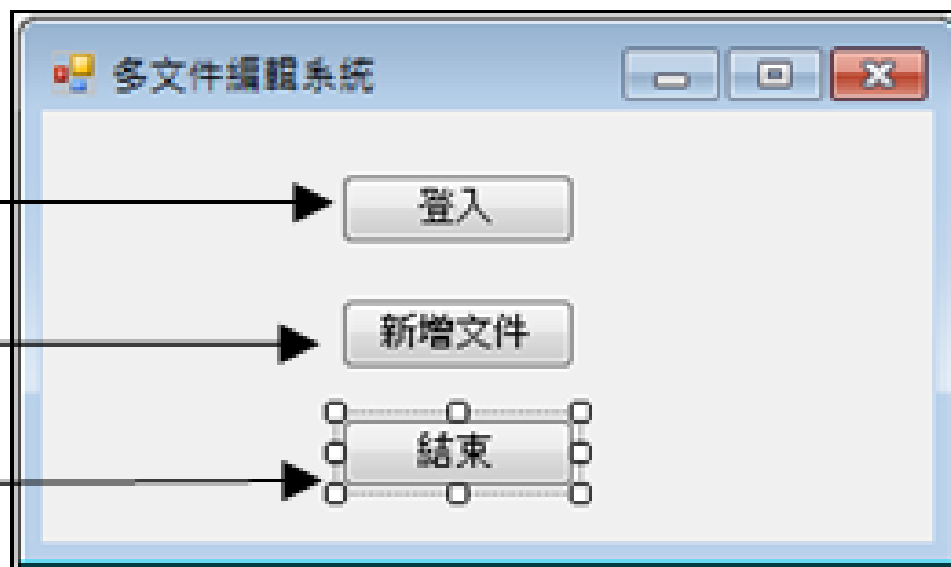
Result



btnPassword

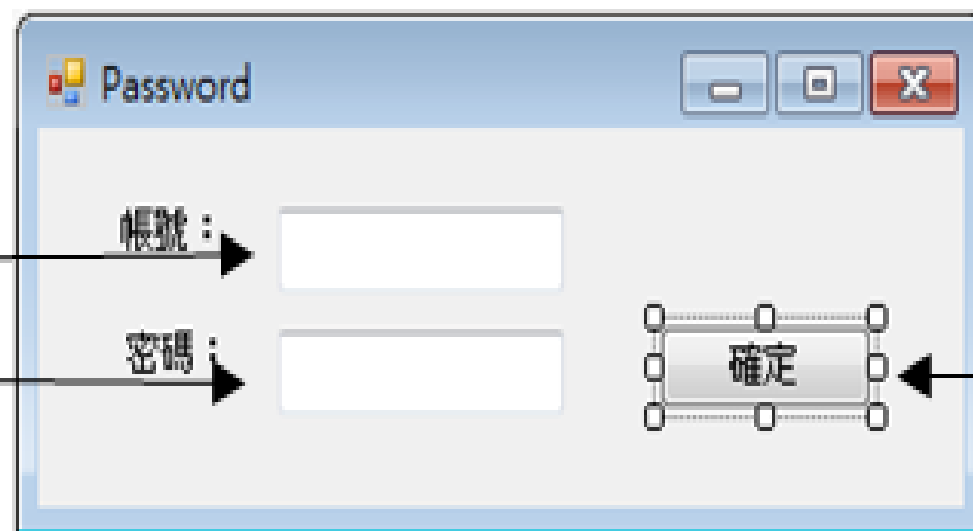
btnAdd

btnExit



txtID

txtPassword



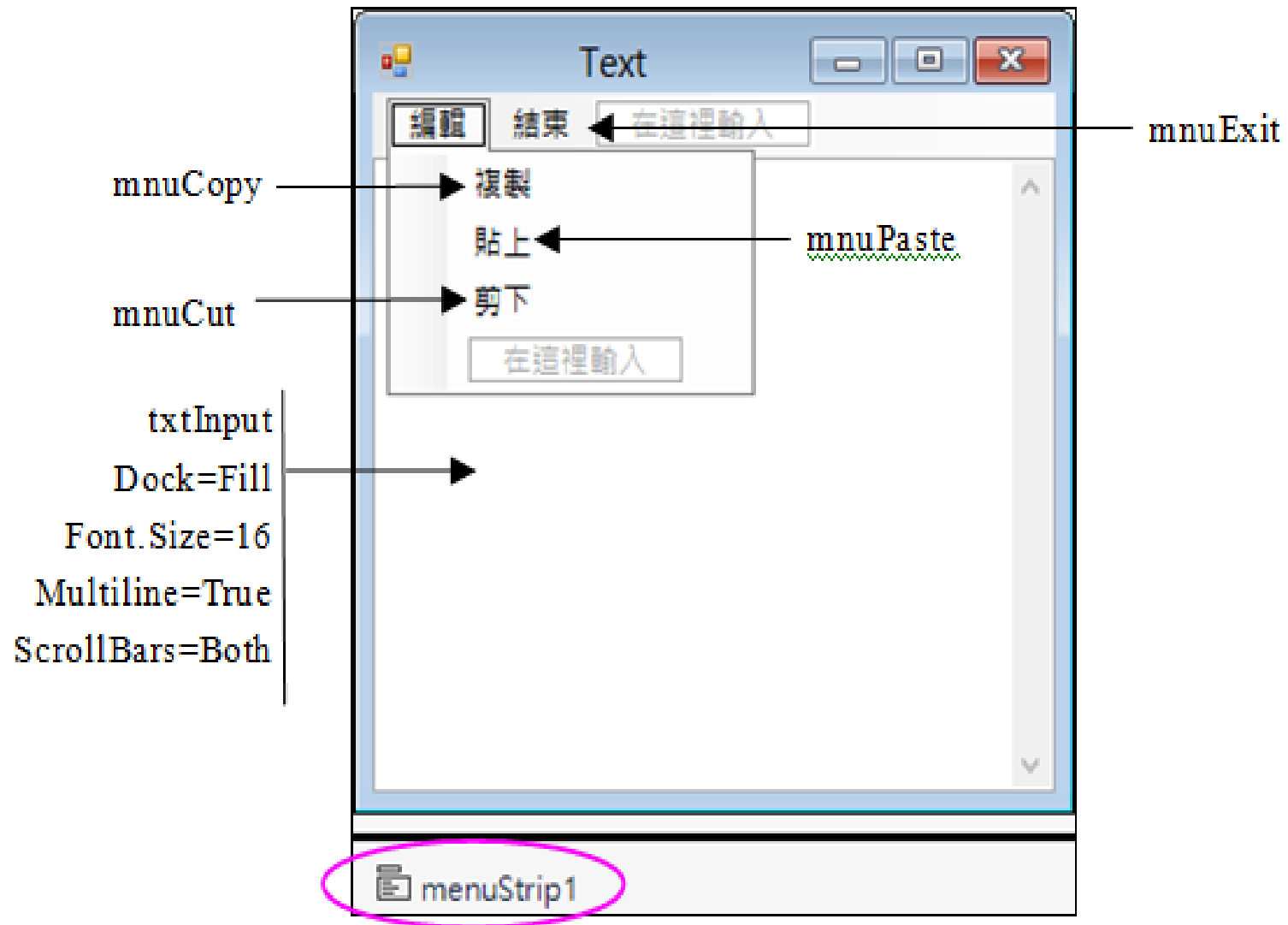
btnOK

A Windows-style dialog box titled "Password" with a standard title bar containing minimize, maximize, and close buttons. The dialog contains two text input fields and one button. The first field is labeled "帳號:" (Account) and the second is labeled "密碼:" (Password). The "密碼:" field has a small eye icon on its right side, indicating a password toggle. Below the fields is a button labeled "確定" (OK). Arrows from external labels point to each of these three elements.

txtID

txtPassword

btnOk

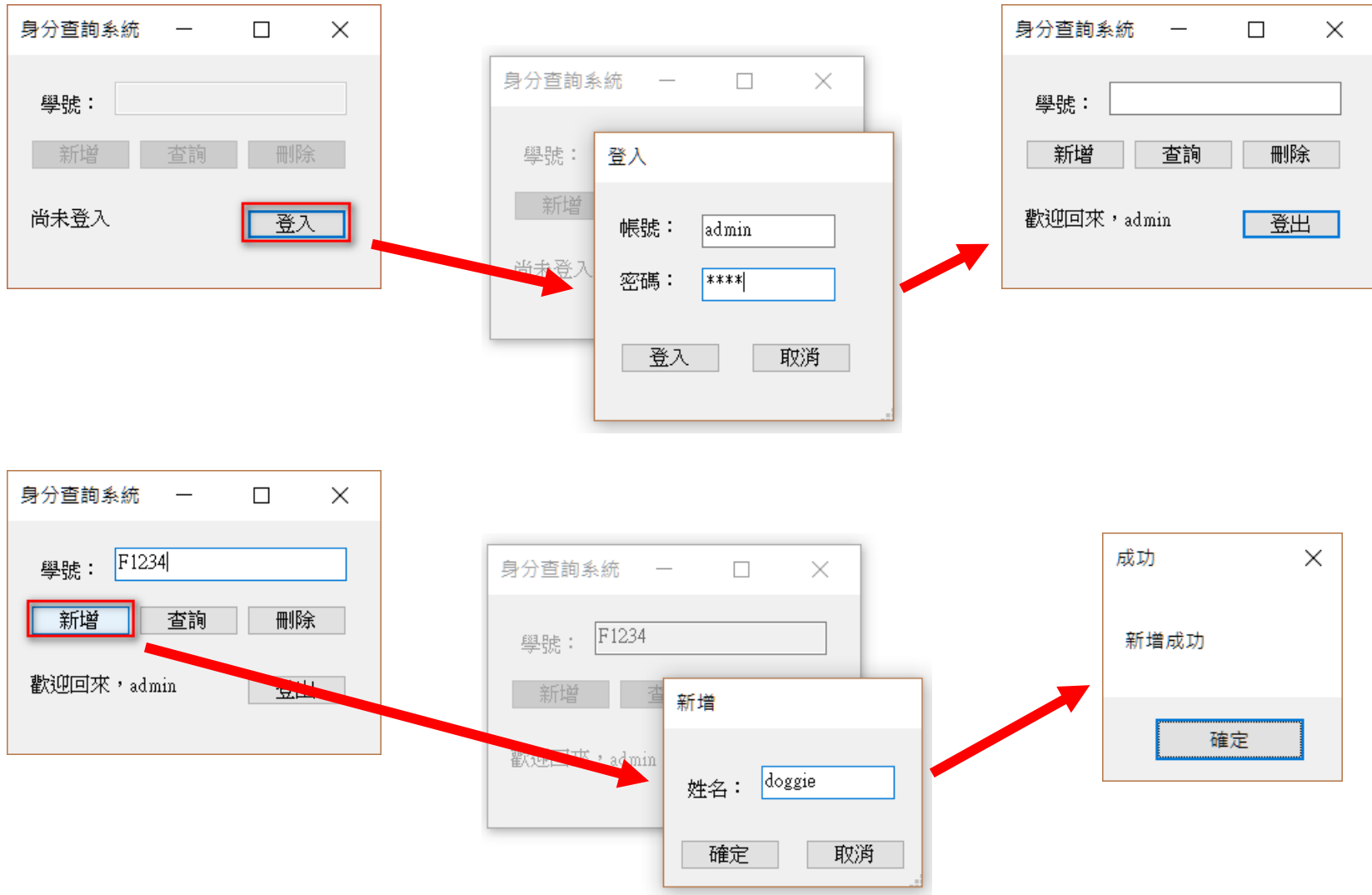


Practice(multiFormList):

Design a multi-form program, requirements:

1. A main form which contains 4 buttons (“新增”, “查詢”, “刪除”, “登入”) and 1 textbox for ID input.
2. When “登入” is pressed, a new form should show up and ask user for login. A user should login first to do further operation. (default: “admin”/“0000”. If you use a different setting, please attach a readme file to let TAs know what's yours)
3. When “新增” is pressed, a new form should show up and ask for the name which will be associated to the ID.
4. When “查詢” is pressed, print the name which has been associated to the ID.
5. When “刪除” is pressed, delete the data set of the ID.

Practice(multiFormList):

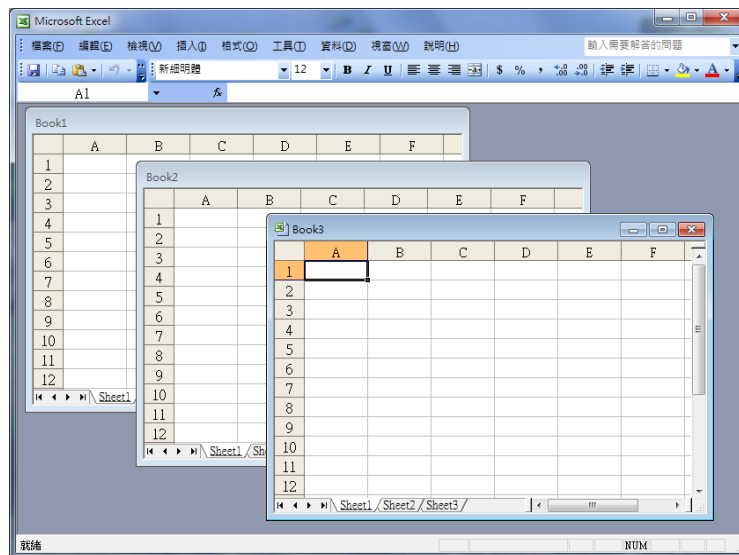


Practice(multiFormList):



MDI(Multiple Document Interface) Programming

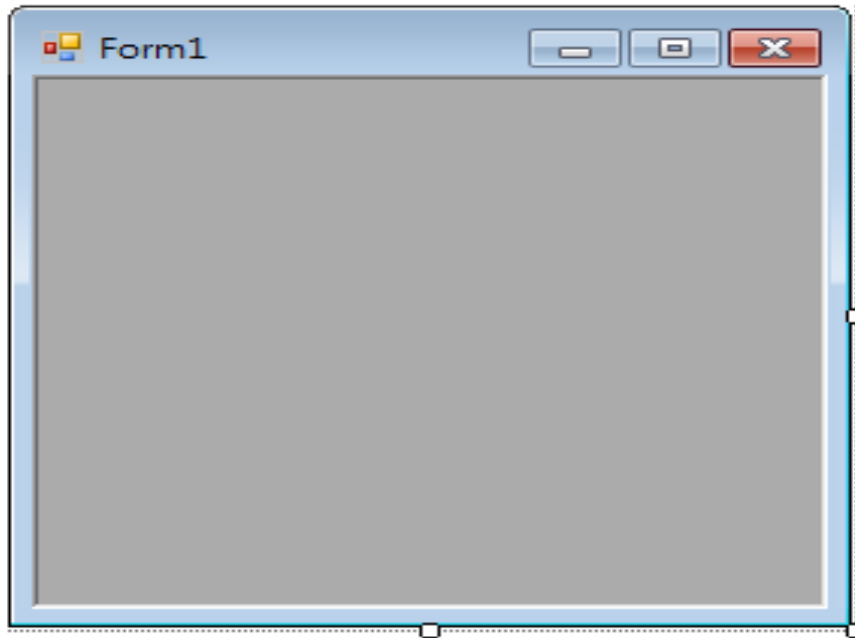
- **Single document interface**
only one form can be opened even there are many forms in the project
- **Multiple Document Interface**
open many form windows in the parent form
- **Ex: Microsoft Excel**



How to Create MDI Program

1. Assign parent form

- Set IsMdiContainer property to True so that the form is a parent form
- The content of MDI parent form is dark gray, and the BackColor property does not work



2. Assign which parent form belongs to

- Assign MdiParent property for child form of designated parent form during runtime
- Ex: show Form2 in the current parent form, usage:
`Form2 frmChild = new Form2();`
`frmChild.MdiParent = this;`
`frmChild.Show();`

MDI Form Properties

Property	Description
<code>IsMdiContainer</code>	Assign the form as a MDI container (parent form), default: False
<code>MdiChildren</code>	Get the child form array in MDI container. Ex: get the number of child forms, <code>int num = this.MdiChildren.Length;</code>
<code>ActiveMdiChild</code>	Get the current activated child form, data type is Form. Return Null if no child forms. Ex: get the current activated child form, <code>Form actForm = this.ActiveMdiChild;</code>

Ex: close all child forms in MDI program, usage:

```
int num = this.MdiChildren.Length;    // 取得子表單的數量
for (int x = 0; x < num; x++)
{
    Form frmChild = this.MdiChildren[x];
    frmChild.Close();
}
```

另一種方法

```
foreach (Form frmChild in this.MdiChildren)
{
    frmChild.Close();
}
```

MDI Form Methods

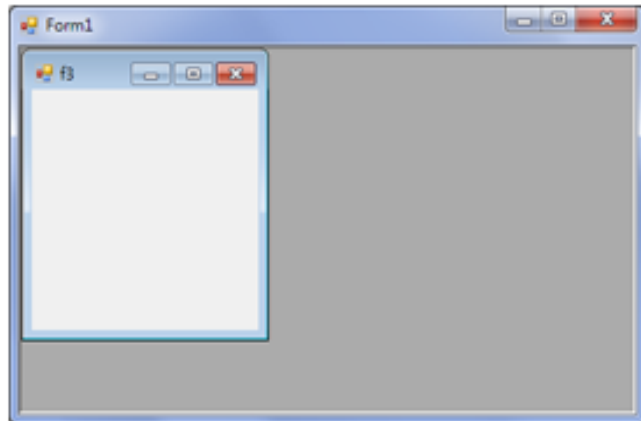
1. LayoutMdi()

assign the arrangement of child forms in parent form,
parameters:

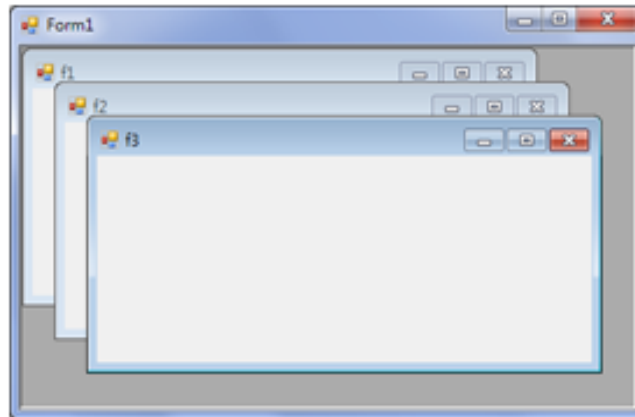
- ① Arrangelcons: child forms are arranged at original positions in workspace of parent form
- ② Cascade: child forms are arranged as cascade in workspace of parent form
- ③ TileHorizontal: child forms are horizontally arranged side by side
- ④ TileVertical: child forms are vertically arranged side by side

Ex: arrange child forms in current form as cascade:
`this.LayoutMdi(MdiLayout.Cascade);`

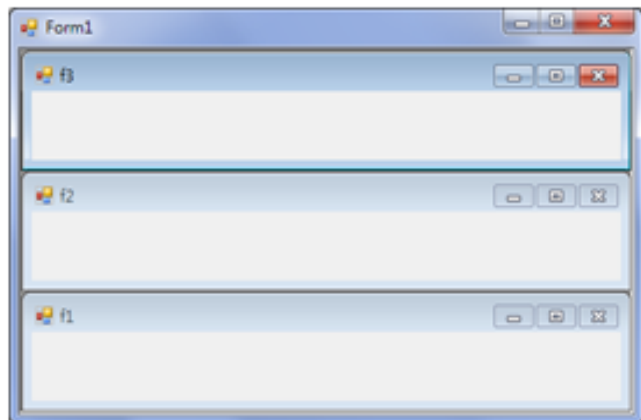
ArrangeIcons



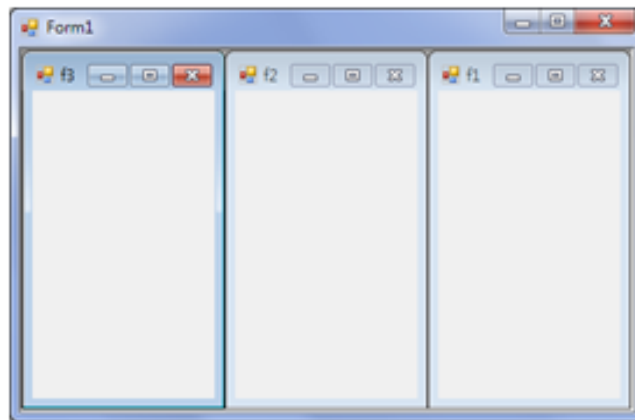
Cascade



TileHorizontal



TileVertical



ActiveMdiChild()

- Set designated child form as activated form
`this.ActiveMdiChild(frmChild);`
- Ex: set f1 form as the activated form in current parent form:
`this.ActiveMdiChild(f1);`



MenuStrip in MDI Form

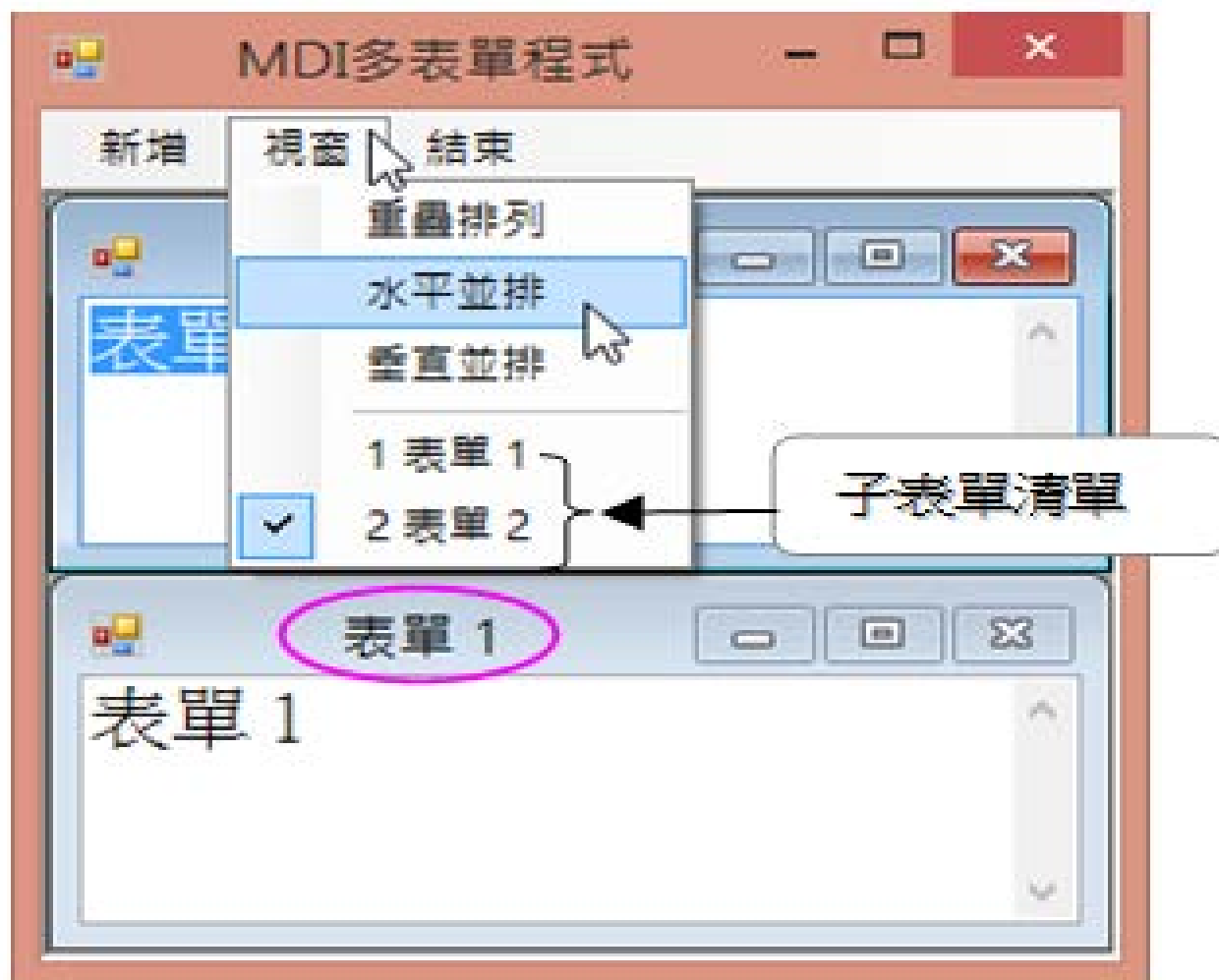
- In MDI program, if it is required that MenuStrip of parent form shows name of child form, MdiWindowListItem property can be used.
- System automatically manages name list of child form after the property is set
- Ex: set MdiWindowListItem property of menuStrip1 as mnuWindow, the mnuWindow sub menu will list all name of child forms, usage:
`menuStrip1.MdiWindowListItem = mnuWindow;`

Practice(mdiForm):

Design a MDI program, requirements:

1. Open “MDI多表單程式” window when the program starts, and there are 3 main menu item called “新增”, “視窗” and “結束”.
2. The parent form creates a new child form when press “新增” once. The new child form has the title and the text box with “表單1”, “表單2”, ... or “表單n”. All child forms are shown in “視窗” main menu item.
3. Press “結束” main menu item to end the program.
4. There are 3 sub menu items called “重疊排列”, “水平並排” and “垂直並排” under “視窗” main menu item for arranging child forms in relative order.

Result



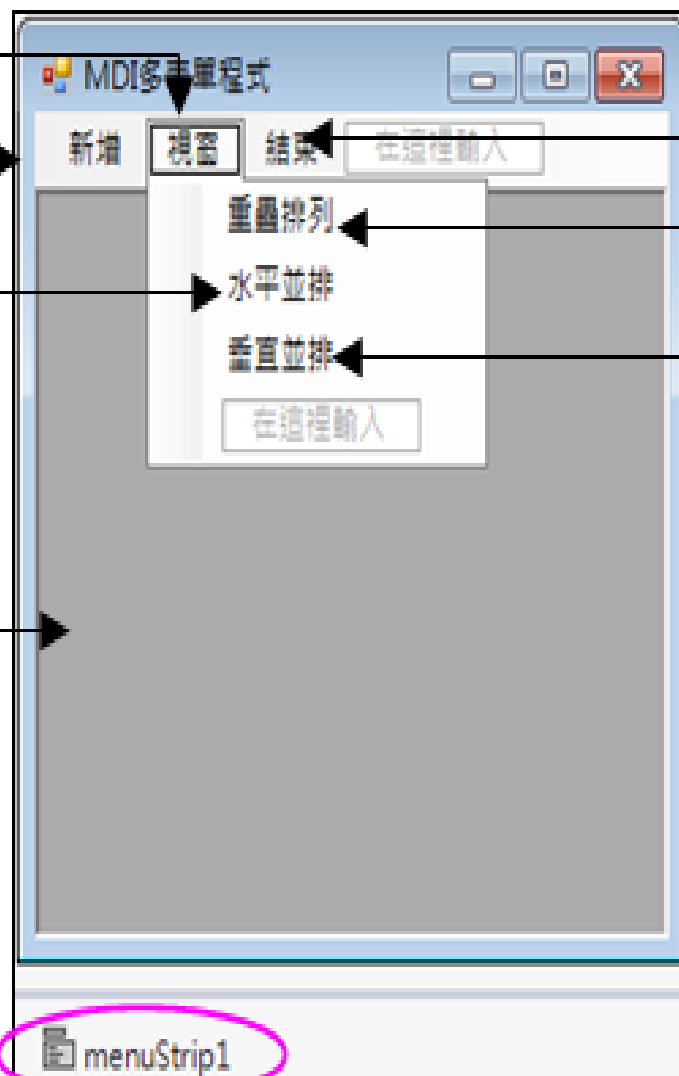
mnuWindow

mnuAdd

mnuHorizontal

Form1

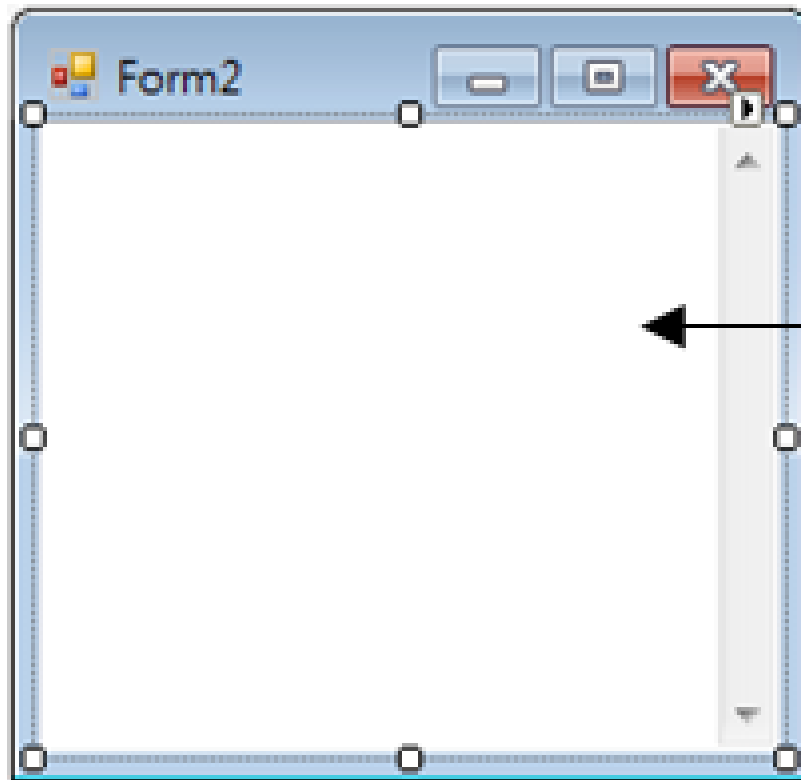
IsMdiContainer = True



mnuExit

mnuCascade

mnuVertical



Name=txtInput

Dock=Fill

Font.Size=16

Multiline=True

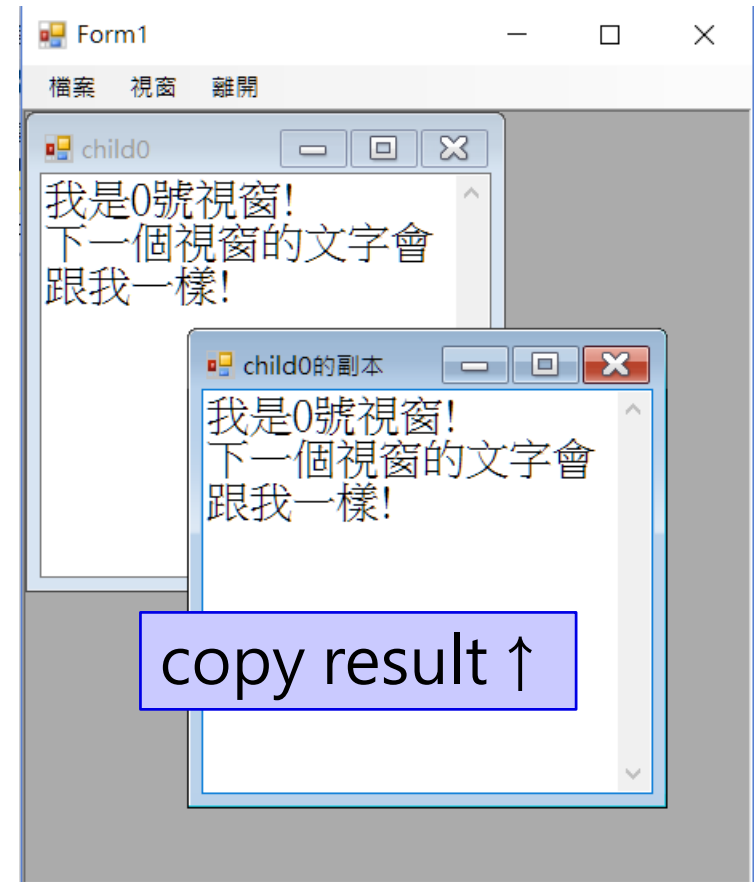
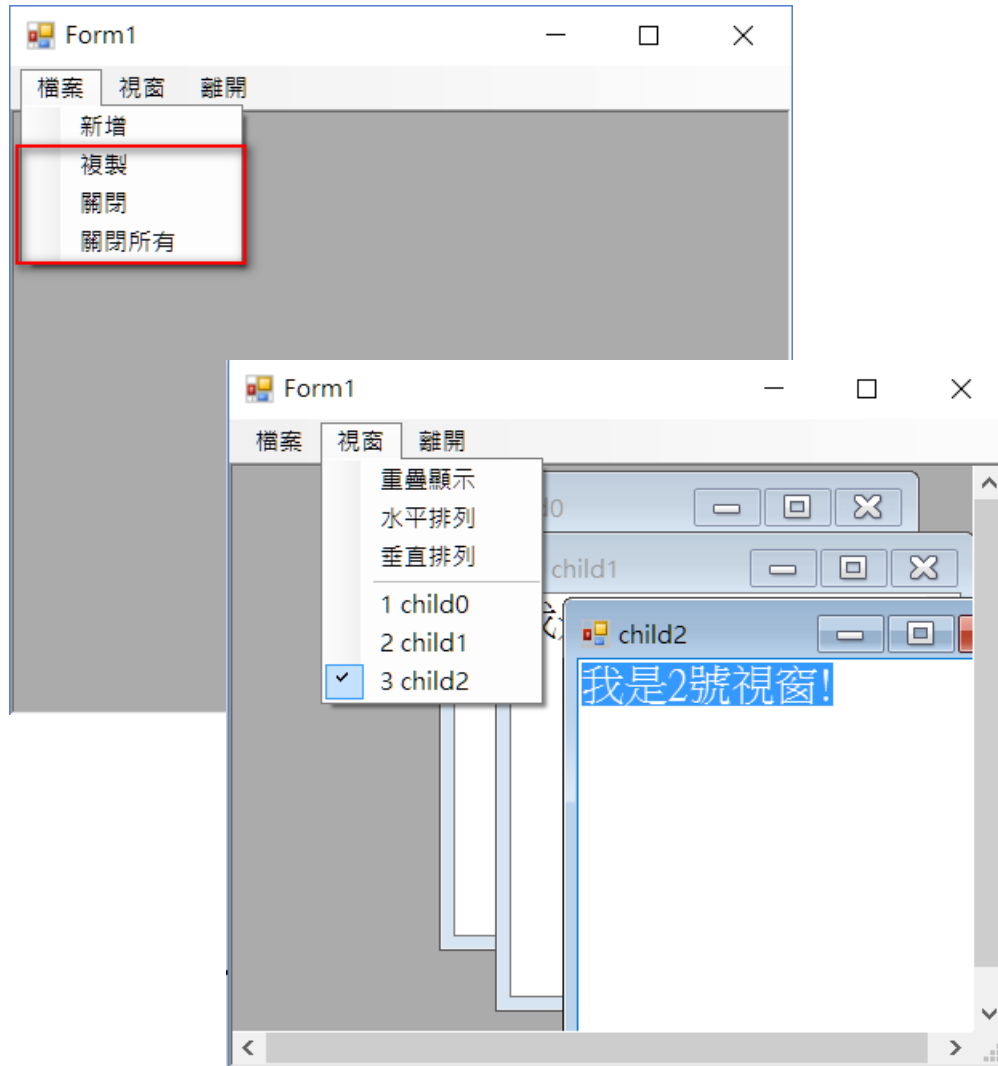
ScrollBars=Both

Practice(mdiFormAdvenced):

Design a MDI program, requirements:

1. Similar to previous practice, you have to implement “新增”, “視窗”, “結束”, but also implement “複製”, “關閉”, “關閉所有”.
2. When “複製” is pressed, you should open a new child form with the original text copied.
3. When “關閉” is pressed, the currently-actived child form should be closed. (i.e. ActiveMdiChild)
4. When “關閉所有” is pressed, all child form should be closed.

Practice(mdiFormAdvenced):





The End