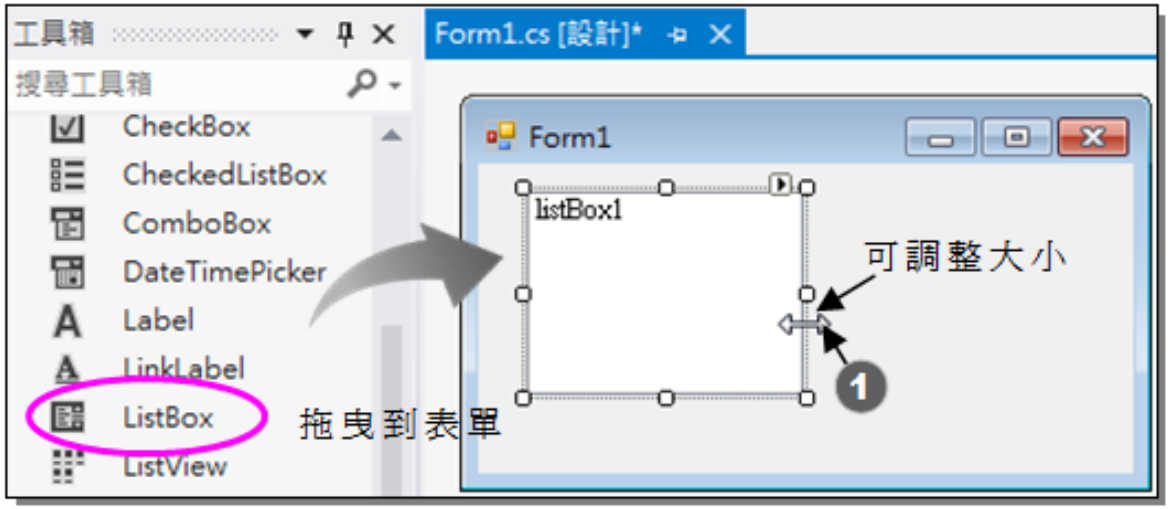
# Chapter 8

## Listed Control items

# 8-1 ListBox Control Item

- **Offer functions for choosing:**
  **① single or multiple choice**
  **② show in multiline**
  **③ insert into ListBox**
  **④ automatic sorting by alphabet**

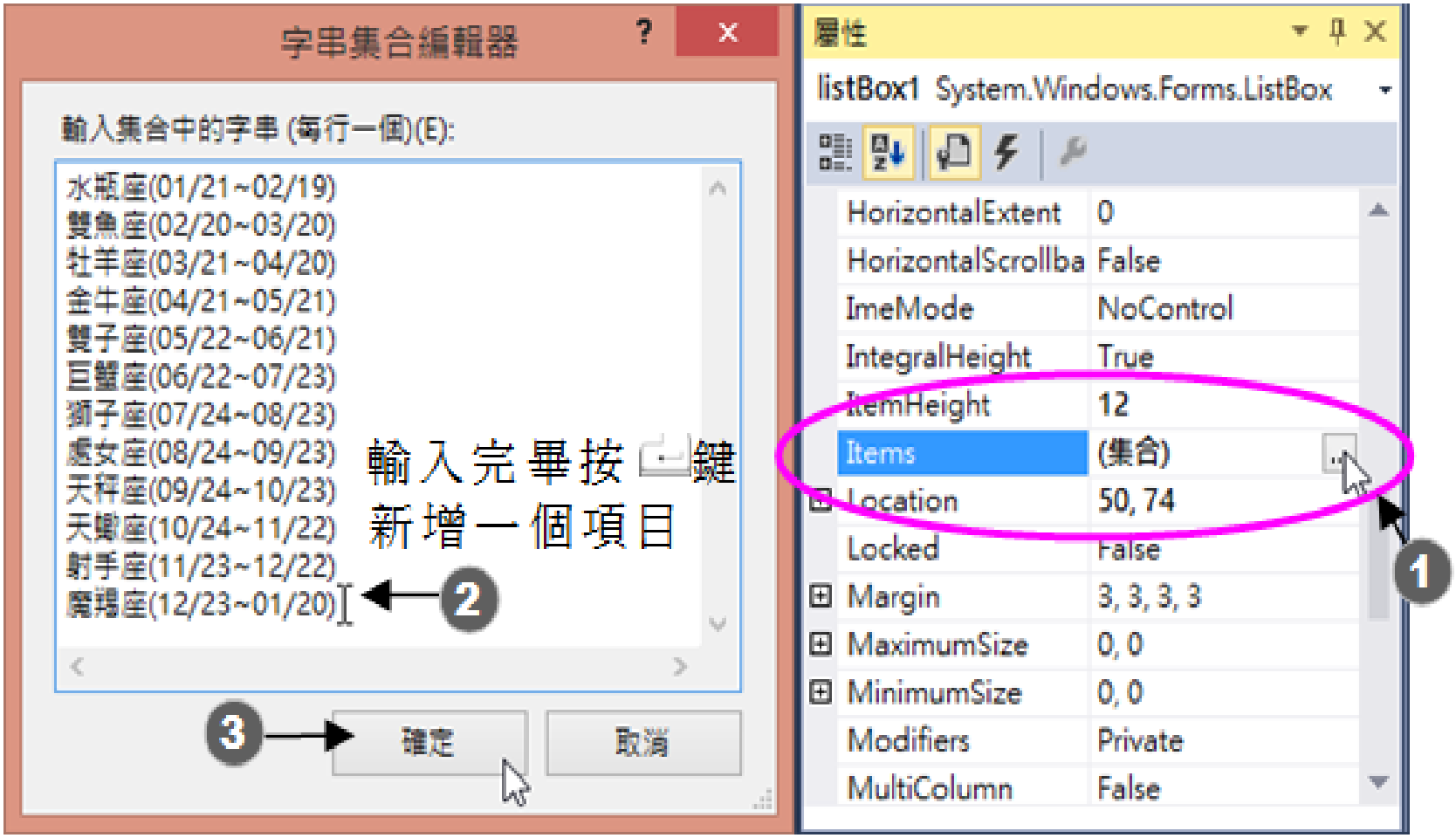


# ListBox Properties

| Property | Description |
|---|---|
| Text | Save selected item during runtime |
| Sorted | True: sort by alphabets' order<br>False: no sorting (default) |
| MultiColumn | Show in multiline or not, default is False<br>False<br>  <br>True<br>  |
| Items | Get or set all items of ListBox.<br>Ex: get the first item: 1stTest.Items[0].ToString(); |

| Property | Description |
|---|---|
| Items.Count | Number of items in ListBox |
| SelectionMode | One: only one can be selected (default)<br>None: cannot be selected<br>MultiSimple: multiple choices without Shift or Ctrl<br>MultiExtended: multiple choices, Shift or Ctrl is required |
| SelectedIndex | Get or set the index of selected item, start from 0 |
| SelectedIndices | Get or set the collection of selected items' indexes if SelectionMode is set to MultipleSimple or MultipleExteneded |
| SelectedItem | Get or set the selected item |
| SelectedItems | Get or set the collection of selected items if SelectionMode is set to MultipleSimple or MultipleExteneded |

# Add Item in ListBox



# Add Item in ListBox in Runtime

## 1. Add() Method
**add a string item to the end of ListBox**

| Grammar |
| --- |
| controlItemName.Items.Add(stringItem); |

**Ex: append “iPhone” and “iPad mini” to the end of 1stApple ListBox, usage:**

```
lstApple.Items.Add("iPhone");
lstApple.Items.Add("iPad mini");
```

## 2. AddRange() Method
## add all elements of the string array to the ListBox

| Grammar |
| --- |
| controlItemName.Items.AddRange(stringArrayName); |

**Ex: add 4 elements of string array ItemName to 1stApple ListBox, usage:**

```
string[] ItemName=new string[]{ "iPad" , "iPhone", "iMac" , "iTV" };
lstApple.Items.AddRange(ItemName);
```

## 3. Insert() method

**insert string item into designated index, the item at the original position and the following items are moved 1 position backward**

| Grammar |
| --- |
| controlItemName.Items.Insert(index, stringItem); |

**Ex: there are 3 items called “iPad”, “iPhone” and “iTV” in the ListBox. Insert “iMac” into the position with index 2, the item “iTV” at the original position is moved backward. Usage:**

```
lstApple.Items.Insert(2, "iMac",);
```

**Result:** "iPad"、"iPone"、"iMace"、"iTV"。

# Remove Item from ListBox in Runtime

## 1. Clear() method
## remove all items from ListBox

| Grammar |
| --- |
| controlItemName.Items.Clear(); |

**Ex: remove all items from 1stApple, usage:**

```
lstApple.Items.Clear();
```

## 2. Remove() method
## remove designated string item from ListBox, the following items move 1 position forward

| Grammar |
| --- |
| controlItemName.Items.Remove(stringItem); |

**Ex1: remove “iPad” item from 1stApple ListBox, usage:**

```
lstApple.Items.Remove("iPad");
```

**Ex2: remove selected item from 1stApple ListBox, usage:**

```
lstApple.Items.Remove(lstApple.SelectedItem);
```

# 3. RemoveAt() method
## remove the item with designated index from ListBox

| Grammar |
| --- |
| controlItemName.Items.RemoveAt(index); |

**Ex1: remove 3rd item from 1stApple ListBox, usage:**

```
lstApple.Items.RemoveAt(2);
```

**Ex2: remove selected-index item from 1stApple ListBox, usage:**

```
lstApple.Items.RemoveAt(lstApple.SelectedIndex);
```

# Other ListBox Methods

1. **SetSelected() Method**
   **set the item of ListBox to selected(true) or unselected(false)**

| Grammar |
| --- |
| controlItemName.SetSelected(index, true/false); |

**Ex: set the first item of 1stApple ListBox to selected, usage:**

```
lstApple.SetSelected(0, true);
```

## 2. ClearSelected() Method
## cancel all selected items and make them unselected

| Grammar |
| --- |
| controlItemName.ClearSelected(); |

**Ex: make all items in 1stApple ListBox unselected, usage:**

```
lstApple.ClearSelected();
```

## 3. GetSelected() Method

**See if the designated index of item is selected or not. Return true for selected; return false for unselected**

| Grammar |
| --- |
| returnValue = controlItemName.GetSelected(index); |

**Ex: merge all selected items in 1stApple ListBox into sel string variable, usage:**

```
string sel = "";
for (int i=0; i<lstApple.Items.Count; i++) {
    if (lstApple.GetSelected(i)) {
        sel += lstApple.Items[i] + "、";
    }
```

## 4. Contains() Method

**get whether the designated string is in the ListBox or not. Return true for existence; return false for nonexistence**

| Grammar |
| --- |
| returnValue = controlItemName.items.Contains(string); |

**Ex: get whether “iTV” exists in 1stApple controlItemName or not, usage:**

```
bool exist = comboBox1.Items.Contains("iTV");
```

# ListBox Events

1. **SelectedIndexChanged**

- **Default event**
- **Triggered when SelectedIndex changes value**

## Practice(apple):

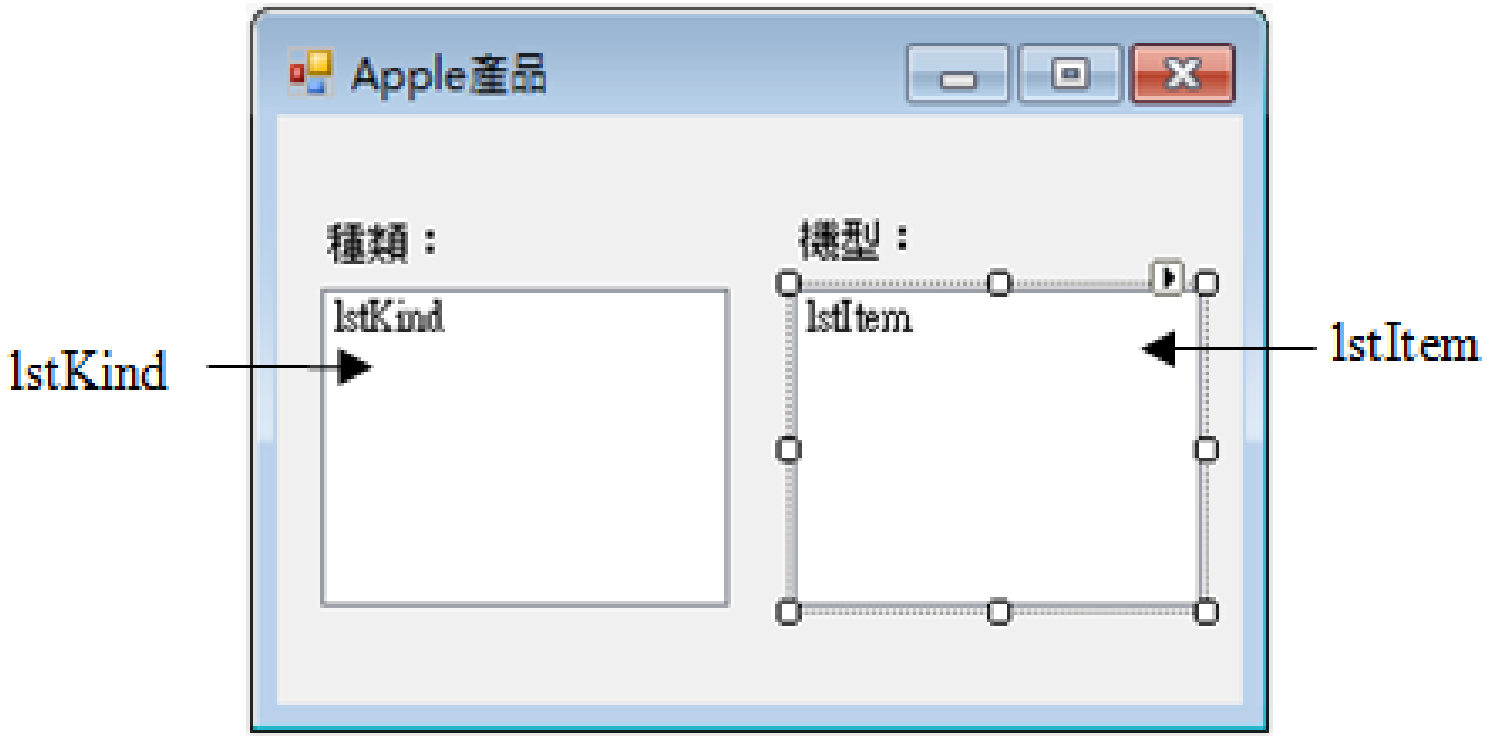
Design a program to show the apps of Apple. Requirements:

1. There are 4 options “Mac”, “iPod”, “iPhone” and “iPad” in the classification when the program starts. Mac is the default option.
2. Put all types of the designated classification into “Type List” when the option of classification is selected.
3. Every classification has corresponding types:
   - ① There are MacBook, Mac mini, iMac and MacPro in Mac classification.
   - ② There are iPod shuffle, iPod nano, iPod touch and iPod classic in iPod classification.
   - ③ There are iPhone 4, iPhone 4S and iPhone 5 in iPhone classification.
   - ④ There are iPad 2 and iPad mini in iPad classification.

**Result:**

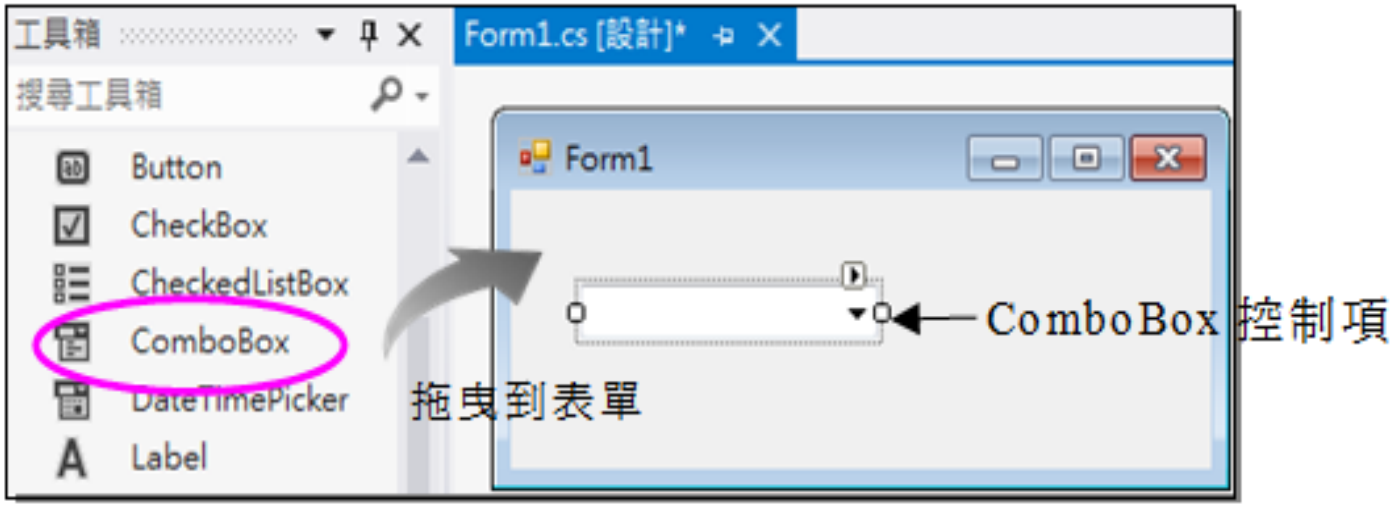

Apple產品
種類：
Mac
iPod
iPhone
iPad
機型：
iPod shuffle
iPod nano
iPod touch
iPod classic

# Design User Interface



# 8-2 ComboBox Control Item

- Provide many string items for choosing
- Only allow one item selected
- Input new item during runtime is allowed



# ComboBox Properties

| Property | Description |
|---|---|
| Text | Save the selected item or inputted string data during runtime. |
| MaxDropDownItems | The maximum number of visible items |
| DropDownStyle | Style of ComboBox<br>1. DropDown: show items when the button is pressed and customized input is allowed (default)<br>2. DropDownList: show items when the button is pressed and customized input is not allowed<br>3. Simple: always show items and customized input is allowed<br>  |

# ComboBox Events

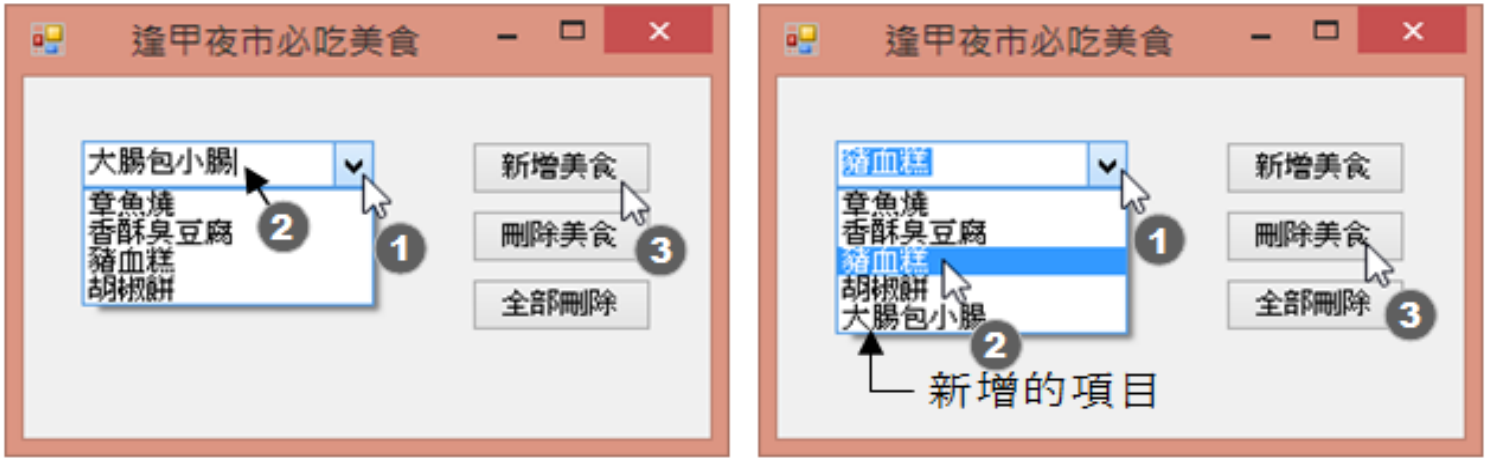
1. SelectedIndexChanged
   - Default event
   - Triggered when SelectedIndex changes value
2. TextChanged
   - Triggered when selected item is changed or input value changes Text property
   - Trigger TextChanged event ⇨ SelectedIndexChanged event in order when the user selects an item

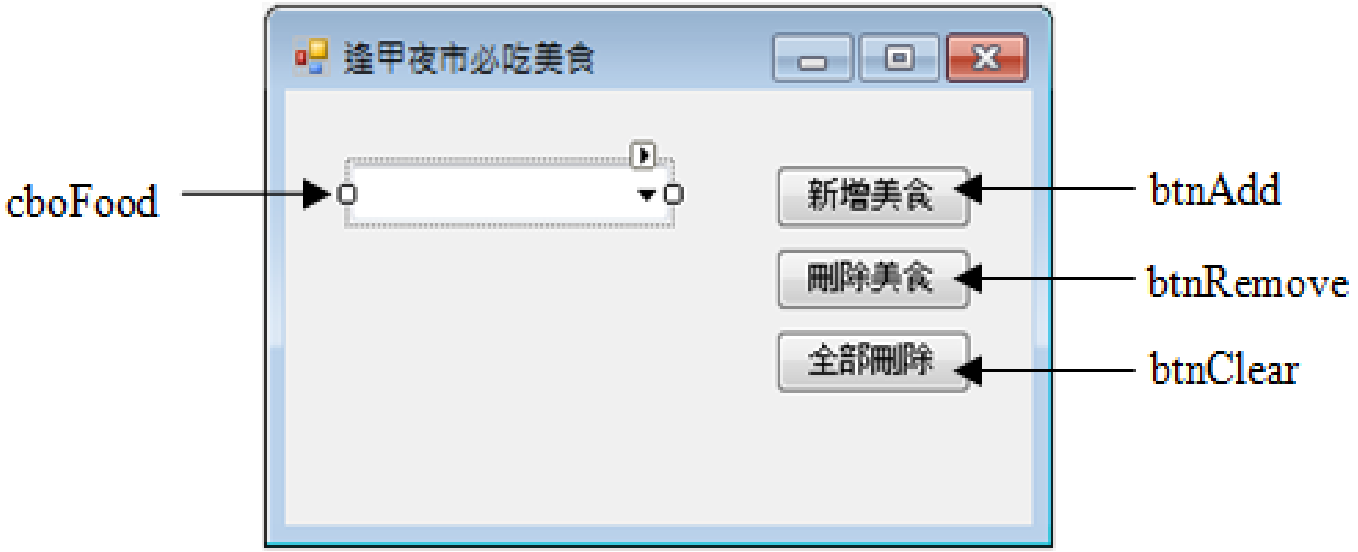
**Practice(nightMarket):**

Design a program for must-eat foods in Fengjia Night Market. Requirements:

1. There are 4 items called “章魚燒”, “香酥臭豆腐”, “豬米血糕” and “胡椒餅” in ComboBox when the program starts.
2. If inputted food name is not in the menu, press “新增美食” to add to the menu.
3. Press “刪除美食” to delete selected item.
4. Press “全部刪除” to delete all items.

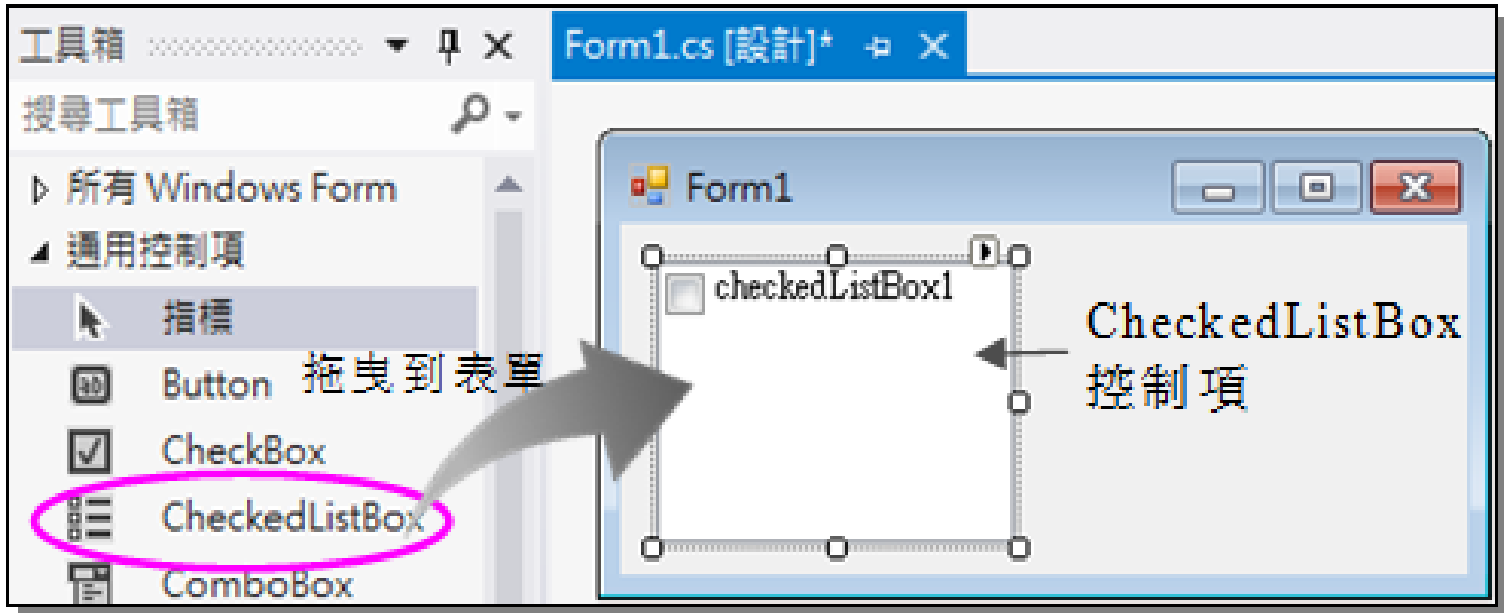
# Result:


逢甲夜市必吃美食
大腸包小腸
章魚燒
香酥臭豆腐
豬血糕
胡椒餅
新增美食
刪除美食
全部刪除
1
2
3
逢甲夜市必吃美食
豬血糕
章魚燒
香酥臭豆腐
豬血糕
胡椒餅
大腸包小腸
新增美食
刪除美食
全部刪除
1
2
3
新增的項目


逢甲夜市必吃美食
cboFood
新增美食
刪除美食
全部刪除
btnAdd
btnRemove
btnClear

# 8-3 CheckedListBox Control Item

- Similar to ListBox
- Additional check box for selection
- Can choose items, but is more suitable for multiple selections



# CheckedListBox Properties

| Property | Description |
|---|---|
| CheckOnClick | True: set checked when the item is clicked<br>False: set checked when the item is double-clicked |
| CheckedItems | Get collection of current selected items during runtime |
| CheckedIndices | Get index collection of current selected items during runtime |

# CheckedListBox Methods

## 1. add() method
**add new item and assign check status in the meantime**

| Grammar |
| --- |
| controlItemName.Items.Add(stringItem, true\|false); |

**Ex: add an item “台中” into c1stCity CheckedListBox and set checked. Add an item “新北” into c1stCity and set unchecked.**

```
clstCity.Items.Add("台中", true);
clstCity.Items.Add("新北", false);
```

## 2. SetItemChecked() method
## set the item with the designated index to be checked or unchecked

| Grammar |
| --- |
| controlItemName.SetItemChecked(index, true\|false); |

**Ex: set the 1st item of c1stCity CheckedListBox to be checked. Set the 3rd item of c1stCity to be unchecked, usage:**

```
clstCity. SetItemChecked(0,true);
clstCity. SetItemChecked(2,false);
```

## 3. GetItemChecked() method
get whether the item with the designated index is checked or not. Return true if checked, return false if unchecked.

| Grammar |
| --- |
| controlItemName.GetItemChecked(index); |

**Ex: set zipCode variable to be “400” when the 1st item of c1stCity is checked, usage:**

```
if(clstCity.GetItemChecked(0))
{
    string zipCode = "400";
}
```

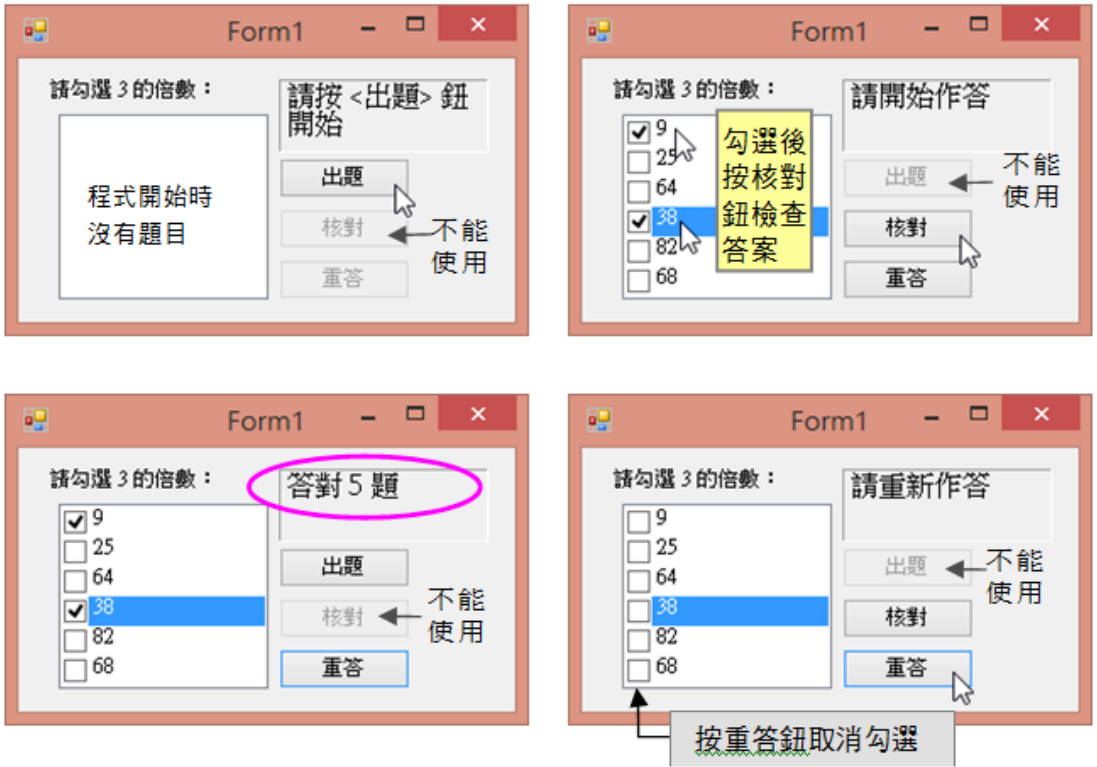
# SelectedIndexChanged Event

- Default event of CheckedListBox
- Triggered when the user selects an item and changes SelectedIndex property

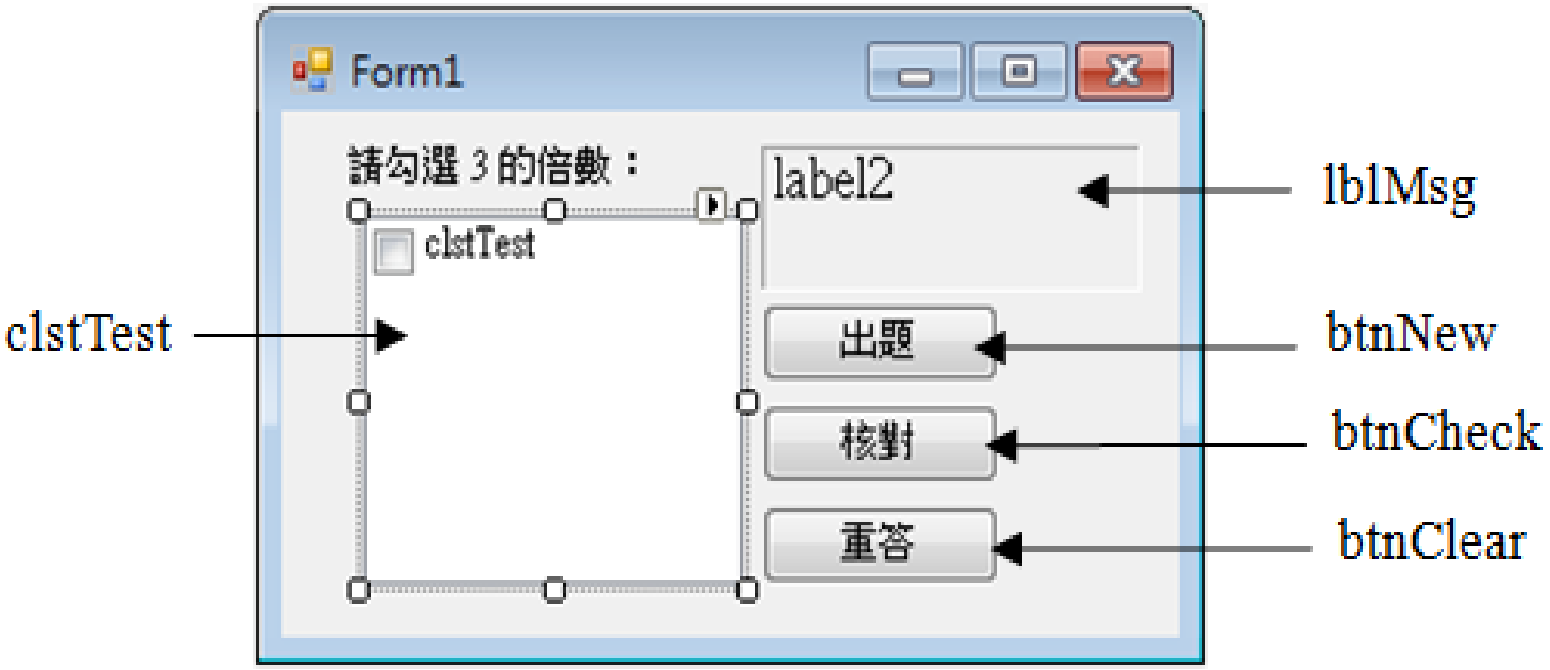
**Practice(multiple):**

Design a program to get multiples of 3, requirements:

1. The button “出題” is available when the program starts. The function of button is to generate 6 random between 1 and 99 and put these number into CheckedListBox. All numbers are unchecked.
2. The user can select which numbers are multiples of 3.
3. Press the button “核對” to show how many right answers they are.
4. Press the button “重答” to set all checked options to be unchecked.

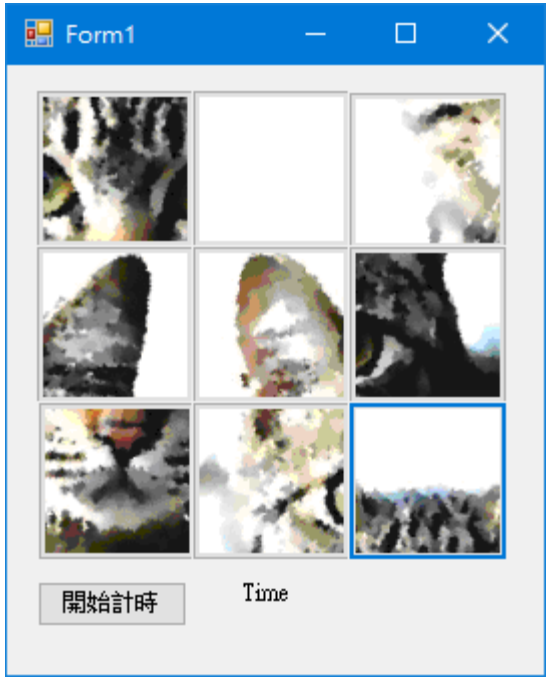


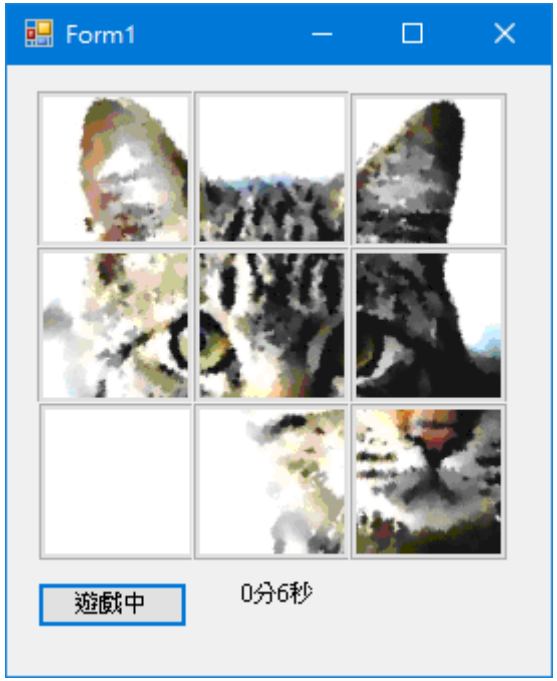
# Design User Interface

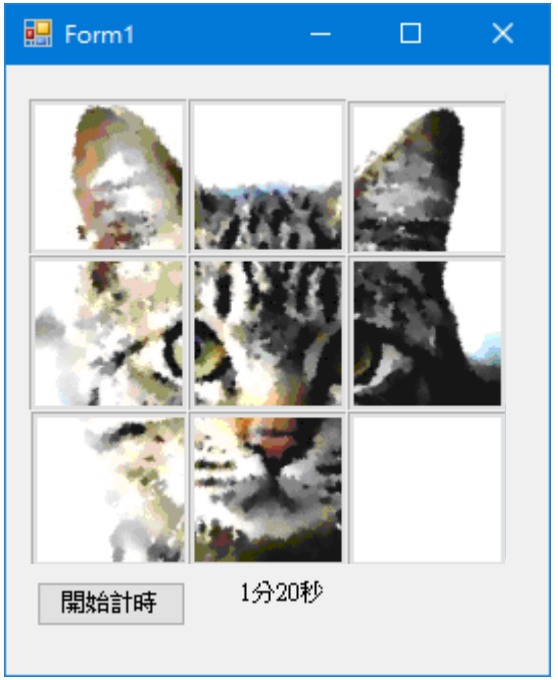


# Practice_Puzzle with Timer

- Use Timer to count how long the player play the game. Must be able to detect if the puzzle are in the correct places and stop the timer.

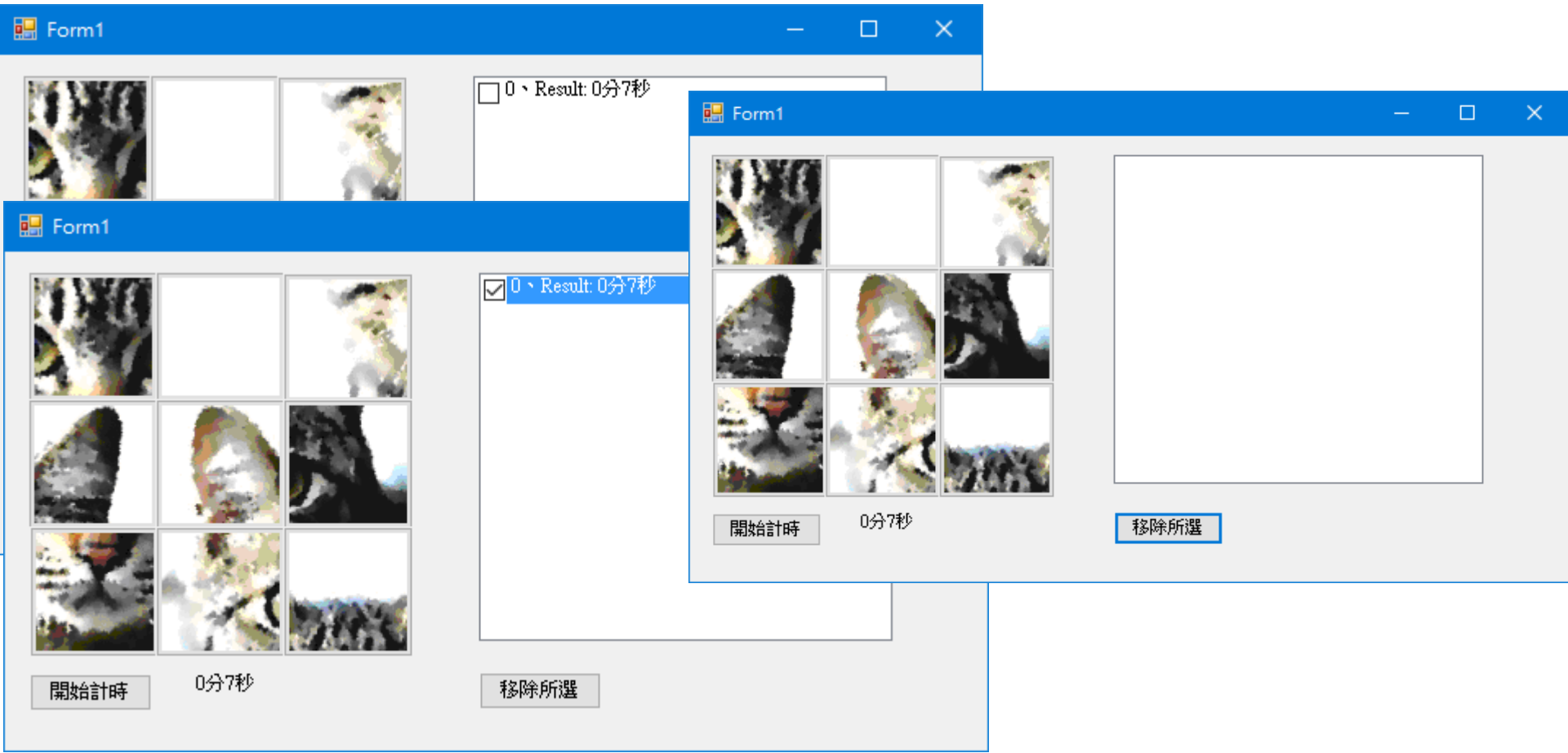


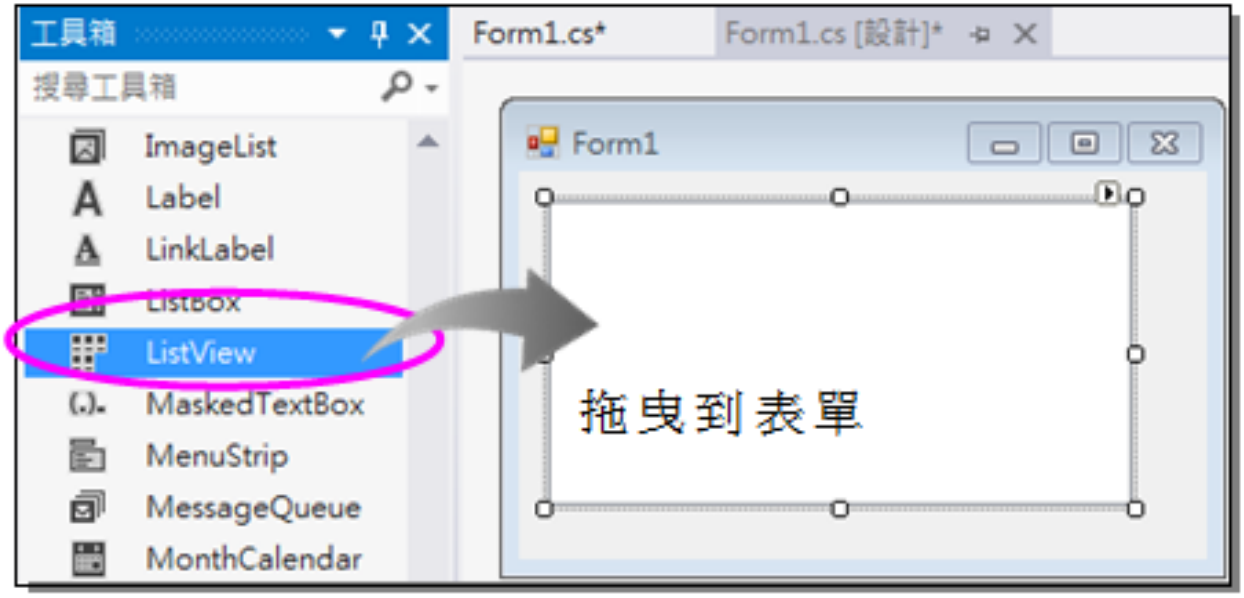
# Practice_CheckedListbox

- Add CheckedListbox to show everyone play time and let user delete the result which they choose



# 8-4 ListView Control Item

- Can show items in icon
- Can show text data, items can be selected
- The way to trigger is called “Activate”, like doing some actions. Use Activation property to set how to trigger



# ListView Properties

| Property | Description |
|---|---|
| Activation | How to trigger:<br>1. Standard, double-click fast, default<br>2. OneClick, click once<br>3. TwoClick, click twice |
| CheckBoxs | Show check box or not, default is False |
| Columns | Get or set the collection of columns |
| Items | Get or set the collection of items |
| Items[].SubItems | Set child collection of item with index |
| Items[].ImageIndex | Set image index of an item.<br>Ex: set the 1st item to show the 2nd image<br>listView1.Items[0].ImageIndex = 1; |
| LargeImageList | Set the ImageList source when show large icon |
| SmallImageList | Set the ImageList source when show small icon |
| View | LargeIcon(larget icon, default), Details(detailed list), SmallIcon(small icon), List(list), Tile(large icon and detailed list) |

# 1. Columns.Add() method
# add new column, assign column name and width

| Grammar |
| --- |
| controlItemName.Columns.Add(stringItem, columnWidth); |

**Ex: add columns “片名” with width 160 and “導演” with width 50 in 1stvMovies ListView, usage:**

```
lstvMovies.Columns.Add ("片名", 160);
lstvMovies.Columns.Add ("導演", 50);
```

# 2. SubItems.Add() method
# add new data after the end of items

| Grammar |
| --- |
| controlItemName.Items[index].SubItems.Add(stringData); |

**Ex: in 1stvBooks ListView, add “賽德克巴萊” and “逆光飛翔” to “片名” field, add “魏德聖” and “張榮吉” to “導演” field**

```
ListViewItem lvi1 = new ListViewItem("賽德克巴萊");
lvi1.SubItems.Add("魏德聖");
lstvMovies.Items.Add(lvi1);
ListViewItem lvi2 = new ListViewItem("逆光飛翔");
lvi2.SubItems.Add("張榮吉");
lstvMovies.Items.Add(lvi2);
```

## 3. BeginUpdate() method and EndUpdate() method

the screen twinkles because the display is reform repeatedly when use Add(). Therefore, use BeginUpdate to stop reform, then use EndUpdate to reform the screen.

**Ex: use BeginUpdate and EndUpdate methods in the previous example, usage:**

```
lstvBooks.BeginUpdate();
ListViewItem lvi1 = new ListViewItem("賽德克巴萊");
    …
lvi2.SubItems.Add("張榮吉");
lstvMovies.EndUpdate();
```

# ListView Events

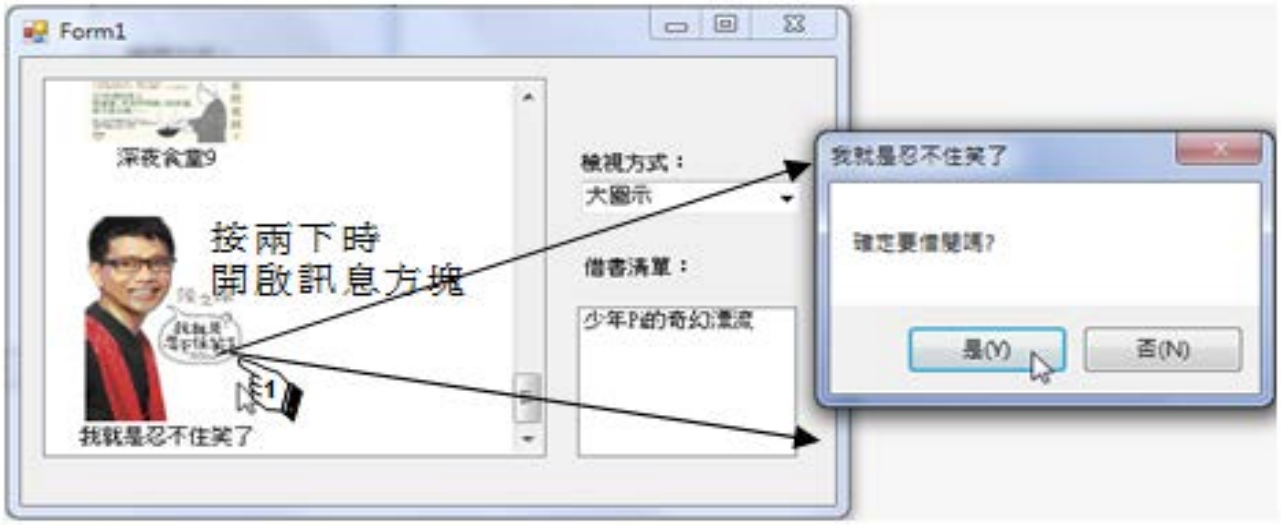
1. ItemActivate
   - Triggered when the item of ListView is activated
2. ColumnClick
   - Triggered when the column of data row is clicked

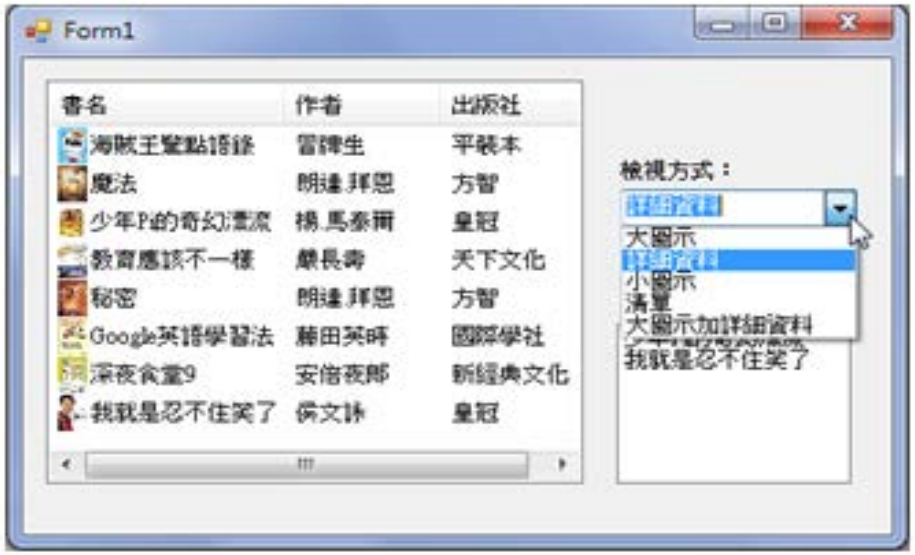
## Practice(library):

Design a program to choose and borrow books:

1. Show the message “確定要借閱嗎?” when the user chooses the book:
   ① If chosen book is in the borrow book list, no message
   ② Press “是” to add book into the list, press “否” for not borrow
2. Change view mode when select the items in the list
   ① The modes have “Large Icon”, “Detailed List”, “Small Icon”, “List” and “Title”
   ② Show book name, author and publisher when Detailed List is selected.
   ③ There are several books like the following figure.

# Result


Form1
深夜食堂9
按兩下時
開啟訊息方塊
我就是忍不住笑了
檢視方式：
大圖示
借書清單：
少年P的奇幻漂流
我就是忍不住笑了
確定要借閱嗎?
是(Y)
否(N)


Form1
書名
作者
出版社
海賊王驚點語錄
冒牌生
平裝本
魔法
朗達.拜恩
方智
少年P的奇幻漂流
楊.馬泰爾
皇冠
教育應該不一樣
嚴長壽
天下文化
秘密
朗達.拜恩
方智
Google英語學習法
藤田英時
國際學社
深夜食堂9
安倍夜郎
新經典文化
我就是忍不住笑了
侯文詠
皇冠
檢視方式：
大圖示
小圖示
清單
大圖示加詳細資料
我就是忍不住笑了

# Design User Interface

# The End

Take a Break …..