Ejercicios sobre Controles - 3.
Curso 2010-2011

Práctica 1. Matrices de controles en tiempo de diseño

En algunas aplicaciones es necesario utilizar muchos controles con funcionalidad similar (cajas de texto, botones, etc). Cuando son muchos controles, puede ser eficiente almacenarlos en una matriz y hacer referencia a ellos en forma indexada. Piense en una calculadora matricial (matrices m x n) en la que cada elemento de la matriz se introduce mediante una caja de texto.

En esta práctica se creará una aplicación sencilla basada en Windows. En el formulario principal se insertarán un conjunto de controles del mismo tipo que después serán almacenados en una matriz de controles.

Iniciar una nueva aplicación en Visual Basic .NET

1. Abra Visual Studio .NET.
2. Añada un nuevo proyecto de tipo **Windows Form**.
3. Asígnele un nombre al proyecto y guárdelo en una carpeta del ordenador. Recuerde guardar los cambios cada cierto tiempo.
4. Haga clic en el formulario para seleccionarlo y establezca las siguientes propiedades:

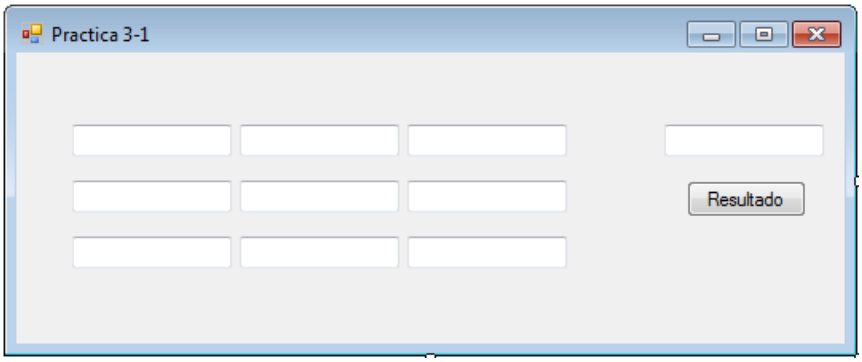
Propiedad	Nuevo Valor
Name	frmPrincipal
Text	Práctica 3-1
Size	535; 220

Añadir controles al formulario principal

1. Añada un control **TextBox** y un **Button**. Establezca las siguientes propiedades:

Control	Propiedad	Nuevo Valor
Textbox	**Name**	txtRes
	Text	*Dejar en blanco*
	Location	406; 45
Button	**Name**	btnRes
	Text	Resultado
	Location	420; 80

2. Añada 9 controles **TextBox**, cuya propiedad **Size** tenga el valor 100; 20, y colóquelos, comenzando por el primero, en las siguientes localizaciones:
   (35; 45), (140; 45), (245; 45), (35; 80), (140; 80), (245; 80), (35; 115), (140; 115) y (245; 115).
   Llegados a este punto, la interfaz de usuario (*formulario principal*) *debe tener el siguiente aspecto*:



Añadir los manejadores de eventos

1. Declare en el formulario (clase Form1) una variable de la siguiente forma:

```
Private mattext(8) As TextBox
```

2. Agregue un procedimiento manejador para el evento **Load** del formulario. Copie el código siguiente:

```
Dim i As Integer
Dim txt As TextBox

For Each ctrl As Control In Me.Controls
     If ctrl.Name <> "btnRes" And ctrl.Name <> "txtRes" Then
          txt = CType(ctrl, TextBox)
          mattext(i) = txt
          mattext(i).Text = i.ToString
          i += 1
     End If
Next
```

   El evento Load ocurre cuando se carga un formulario. Por ello el procedimiento manejador de este evento siempre se ejecuta cuando se utiliza el formulario en el cual está definido.
   Cada control tiene una propiedad asociada, llamada **Controls**, en la que se almacena una colección con todos los controles contenidos en él. En este caso, dicha propiedad del formulario principal contiene una referencia a todas las cajas de texto y al botón.
   En el bucle se accede a la colección de controles del formulario (**Me.Controls**) y, después de descartar la primera caja de texto que se insertó y el botón en la instrucción **If**, se asigna la referencia a cada caja de texto encontrada a la posición actual de la matriz **mattext**. Observe que el índice **i** de la matriz se incrementa en cada iteración. Dentro del bucle también se asigna el valor de **i** a la propiedad **Text** de cada caja de texto.
   Al finalizar el bucle se tienen todas las cajas de texto, excepto la que se colocó a la derecha, en una matriz. Debido a ello se puede acceder a la información almacenada en todas las cajas de texto mediante un simple bucle.
3. Agregue un procedimiento manejador para el evento **Click** del botón. Copie el código siguiente:

```
Dim i, suma As Integer

For i = 0 To 8
     suma += Convert.ToInt16(mattext(i).Text)
Next
txtRes.Text = suma.ToString
```

En este procedimiento se accede a la información almacenada en cada caja de texto mediante un bucle. Al finalizar este se asigna a la caja de texto txtRes, el resultado de sumar todos los datos leídos. Pregúntese cómo se pudiera realizar este mismo algoritmo sin tener todas las cajas de texto almacenadas en una matriz.
Nótese que no se realiza validación alguna de los datos por lo que si cambia el contenido de alguna caja y no escribe un número, el programa puede causar un error de ejecución.

Comprobar la funcionalidad

1. Ejecute la aplicación y compruebe la funcionalidad de la misma.

Práctica 2. Matrices de controles en tiempo de ejecución

En esta práctica también se utiliza una matriz de controles. A diferencia de la práctica anterior, los controles se insertan en tiempo de ejecución, es decir, el programador previamente no añade los controles en tiempo de diseño, solo prepara el formulario para la inserción de los mismos mediante código.

Iniciar una nueva aplicación en Visual Basic .NET

1. Abra Visual Studio .NET.
2. Añada un nuevo proyecto de tipo **Windows Form**.
3. Asígnele un nombre al proyecto y guárdelo en una carpeta del ordenador. Recuerde guardar los cambios cada cierto tiempo.
4. Haga clic en el formulario para seleccionarlo y establezca las siguientes propiedades:

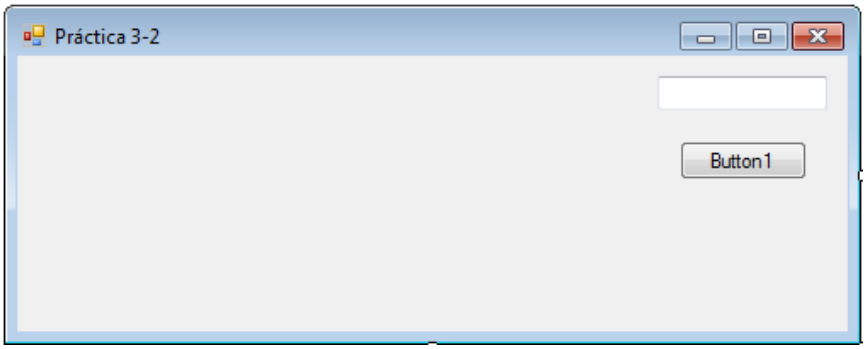
Propiedad	Nuevo Valor
Name	frmPrincipal
Text	Práctica 3-2
Size	505; 200

Añadir controles al formulario principal

1. Añada un control **TextBox** y un **Button**. Establezca las siguientes propiedades:

Control	Propiedad	Nuevo Valor
Textbox	**Name**	txtRes
	Text	*Dejar en blanco*
	Location	406; 45
Button	**Name**	btnRes
	Text	Resultado
	Location	420; 80

Llegados a este punto, la interfaz de usuario (*formulario principal*) *debe tener el siguiente aspecto*:



Añadir los manejadores de eventos

1. Declare en el formulario (clase Form1) una variable de la siguiente forma:

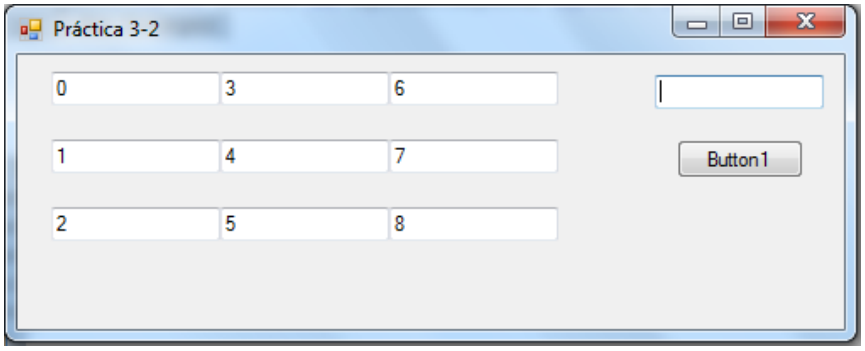
```
Private mattext(8) As TextBox
```

2. Agregue un procedimiento manejador para el evento **Load** del formulario. Copie el código siguiente:

```
Dim i, x, y, j As Integer

j = 0
x = 20
y = 10
For i = 0 To 8
      mattext(i) = New TextBox
      mattext(i).Text = i.ToString
      mattext(i).Location = New Point(x, y)
      Controls.Add(mattext(i))
      y += 40
      j += 1
      If j = 3 Then
            x += 100
            j = 0
            y = 10
      End If
Next
```

   En este procedimiento se ejecuta un bucle en el que, en cada iteración, se accede a un elemento de la matriz al que se asigna una caja de texto creada. También a la caja de texto se le asigna un valor a su propiedad **Text**, se le asigna una posición dentro del formulario y por último se añade a la propiedad **Controls** de este.
   Las variables *j*, *x* e *y* se utilizan para obtener una posición tal que al terminar la ejecución del procedimiento, la interfaz de usuario tenga el siguiente aspecto:



3. Agregue un procedimiento manejador para el evento **Click** del botón. Copie el código siguiente:

```
Dim i, suma As Integer

For i = 0 To 8
      suma += Convert.ToInt16(mattext(i).Text)
Next
txtRes.Text = suma.ToString
```

Comprobar la funcionalidad

1. Ejecute la aplicación y compruebe la funcionalidad de la misma.