Controles de usuario

Apunte Nro 1

PROGRAMACIÓN II

Profesor: DARIO CARDACCI

Controles de usuario

A medida que diseña y modifica la interfaz de usuario de las aplicaciones de Windows Forms, deberá agregar, alinear y colocar los controles de usuario. Los controles son objetos contenidos dentro de objetos de formularios. Cada tipo de control tiene su propio conjunto de propiedades, métodos y eventos que lo hacen adecuado para un fin determinado. Puede manipular los controles del diseñador y escribir código para agregar controles de forma dinámica en el entorno de tiempo de ejecución.

En particular nos concentraremos en los siguientes controles aunque existen muchos más que lo invitamos a que los investigue cuando agote los que están a continuación:

**1 Textbox.**

Los cuadros de texto se utilizan para obtener la entrada del usuario o para mostrar texto. El control se utiliza para texto editable, aunque también se puede convertir en de solo lectura. Los cuadros de texto pueden mostrar varias líneas, ajustar el texto al tamaño del control y agregar formato básico. El TextBox control permite un único formato para el texto que se muestra o se escribe en el control.

<https://docs.microsoft.com/es-es/dotnet/framework/winforms/controls/textbox-control-windows-forms>

**2 Label.**

Se usan para mostrar texto o imágenes que el usuario no puede editar. Se usan para identificar objetos en un formulario, para proporcionar una descripción de lo que hará un determinado control si se hace clic en él, por ejemplo, o para mostrar información en respuesta a un evento o proceso en tiempo de ejecución de la aplicación. El control de [Label](https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.label) no puede recibir el foco, también se puede usar para crear teclas de acceso para otros controles.

<https://docs.microsoft.com/es-es/dotnet/framework/winforms/controls/label-control-windows-forms>

**3 Button.**

Se utiliza para colocar un botón que al oprimirlo produce que se ejecute una acción.<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.button?view=netcore-3.1>

**4 Checkbox.**

Use para CheckBox dar al usuario una opción, como true/false o yes/no. El CheckBox control puede mostrar una imagen, texto o ambos.

CheckBox Los controles y tienen una función similar: permiten RadioButton al usuario elegir entre una lista de opciones. CheckBox Los controles permiten al usuario elegir una combinación de opciones. En cambio, RadioButton los controles permiten a un usuario elegir entre opciones mutuamente excluyentes.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.checkbox?view=windowsdesktop-6.0>

**5 Radiobutton.**

Cuando el usuario selecciona un botón de opción dentro de un grupo, los demás se borran automáticamente. Todos RadioButton los controles de un contenedor determinado, como, constituyen un Form grupo. Para crear varios grupos en un formulario, coloque cada grupo en su propio contenedor, como un GroupBox control o Panel.

Use la Checked propiedad para obtener o establecer el estado de un RadioButton . La apariencia del botón de opción se puede modificar para que aparezca como un botón de estilo de alternancia o como un botón de opción estándar estableciendo la Appearance propiedad

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.radiobutton?view=windowsdesktop-6.0>

**6 GrupBox.**

Muestra GroupBox un marco alrededor de un grupo de controles con o sin un título. Use para GroupBox agrupar lógicamente una colección de controles en un formulario. El cuadro de grupo es un control de contenedor que se puede usar para definir grupos de controles.

El uso típico de un cuadro de grupo es contener un grupo lógico de RadioButton controles. Si tiene dos cuadros de grupo, cada uno de los cuales contiene varios botones de opción, cada grupo de botones es mutuamente excluyente, estableciendo un valor de opción por grupo.

Puede agregar controles a GroupBox mediante el método de la propiedad Add Controls.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.groupbox?view=windowsdesktop-6.0>

**7 Listbox.**

El control permite mostrar una lista de elementos al usuario que el usuario ListBox puede seleccionar haciendo clic en él. Un ListBox control puede proporcionar una o varias selecciones mediante la propiedad SelectionMode. también proporciona la propiedad para habilitar la presentación de elementos en columnas en lugar de ListBox una lista vertical recta de MultiColumn elementos. Con esto, el control puede mostrar elementos más visibles y el usuario ya no necesita desplazarse a un elemento.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.listbox?view=windowsdesktop-6.0>

**8 Combobox.**

Muestra un cuadro de texto combinado con, que permite al usuario ComboBox seleccionar elementos de la lista o escribir un nuevo ListBox valor.

La propiedad especifica si la lista siempre se muestra o si la lista DropDownStyle se muestra en una lista desplegable. La DropDownStyle propiedad también especifica si la parte del texto se puede editar. No hay ninguna opción para mostrar siempre la lista y no permitir la entrada de un nuevo valor.

Para agregar o quitar objetos de la lista en tiempo de ejecución, use métodos de la clase (a través de ComboBox.ObjectCollection Items la propiedad de ComboBox). Puede asignar una matriz de referencias de objeto con el AddRange método. La lista muestra el valor de cadena predeterminado para cada objeto. Puede agregar objetos individuales con el Add método. Puede eliminar elementos con el Remove método o borrar toda la lista con el método Clear.

Además de la funcionalidad de visualización y selección, también proporciona características que permiten agregar eficazmente elementos a y para buscar texto dentro de los elementos ComboBox ComboBox de la lista. Con los métodos y, puede agregar un gran número de elementos a sin que el control se vuelva a dibujar cada vez que se agregue un elemento BeginUpdate EndUpdate a la ComboBox lista. Los FindString FindStringExact métodos y permiten buscar un elemento en la lista que contiene una cadena de búsqueda específica.

https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.combobox?view=windowsdesktop-6.0

**9 Monthcalendar.**

El MonthCalendar control permite al usuario seleccionar una fecha mediante una presentación visual. Puede limitar la fecha y las horas que se pueden seleccionar estableciendo las propiedades MinDate y MaxDate.

El MonthCalendar sistema operativo dibuja el control, por lo que el evento nunca se Paint genera. Si necesita proporcionar una apariencia personalizada al control, debe invalidar el método, llamar a la implementación base de y, a continuación, MonthCalendar OnPrint realizar un dibujo OnPrint personalizado.

Si necesita un formato de fecha personalizado y una selección limitada a una sola fecha, considere la posibilidad de usar DateTimePicker un control en lugar de MonthCalendar . El uso DateTimePicker de elimina gran parte de la necesidad de validar los valores de fecha y hora.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.monthcalendar?view=windowsdesktop-6.0>

**10 ProgressBar.**

Un control consta de una ventana que se rellena, de forma predeterminada de izquierda a derecha, a medida ProgressBar que avanza una operación. El control tiene un intervalo y una posición actual.

ProgressBar invalida los metadatos de la Maximum propiedad y establece su valor predeterminado en 100.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.controls.progressbar?view=windowsdesktop-6.0>

**11 DataGridView.**

El DataGridView control proporciona una tabla personalizable para mostrar datos. La DataGridView clase permite la personalización de celdas, filas, columnas y bordes mediante el uso de propiedades como DefaultCellStyle, ColumnHeadersDefaultCellStyle, CellBorderStyle y GridColor.

Puede usar un DataGridView control para mostrar datos con o sin un origen de datos subyacente. Sin especificar un origen de datos, puede crear columnas y filas que contengan datos y agregarlos directamente a DataGridView mediante las propiedades y Rows Columns. También puede usar la colección Rows para tener acceso a objetos y a la propiedad para leer o escribir valores de celda DataGridViewRow DataGridViewRow.Cells directamente. El Item[] indexador también proporciona acceso directo a las celdas.

Como alternativa a rellenar manualmente el control, puede establecer las propiedades y para enlazar a un origen de datos y rellenarlo automáticamente DataSource DataMember con DataGridView datos.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.datagridview?view=windowsdesktop-6.0>

**12 ColorDialog.**

El miembro heredado ShowDialog se debe invocar para crear este cuadro de diálogo común específico. HookProc se puede invalidar para implementar la funcionalidad específica de enlace del cuadro de diálogo. Use Color para recuperar el color seleccionado por el usuario.

Cuando se crea una instancia de ColorDialog , algunas de las propiedades de lectura y escritura se establecen en valores iniciales.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.colordialog?view=windowsdesktop-6.0>

**13 FontDialog.**

El miembro heredado debe ShowDialog invocarse para crear este cuadro de diálogo común específico. HookProc se puede invalidar para implementar la funcionalidad de enlace de cuadro de diálogo específica.

Al crear una instancia de, algunas de las propiedades de lectura FontDialog y escritura se establecen en valores iniciales.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.fontdialog?view=windowsdesktop-6.0>

**14 Linklabel.**

El LinkLabel control es similar a un control con la Label excepción de que puede mostrar un hipervínculo. Se pueden especificar varios hipervínculos en el texto del control. Cada hipervínculo puede realizar una tarea diferente dentro de una aplicación. Por ejemplo, puede usar un hipervínculo para mostrar un sitio web en Microsoft Internet Explorer o para cargar un archivo de registro asociado a una aplicación.

Cada hipervínculo que se muestra en LinkLabel el control es una instancia de la clase LinkLabel.Link. La LinkLabel.Link clase define la información de visualización, el estado y la ubicación del hipervínculo. Además, la propiedad de la clase permite asociar información, como una LinkData dirección URL que se va a LinkLabel.Link mostrar, con el hipervínculo. Cuando un usuario hace clic en un hipervínculo dentro del control, se genera el evento y el objeto que representa el hipervínculo en el que se hizo clic se pasa como parte del objeto que se pasa como parámetro al controlador de LinkClicked LinkLabel.Link LinkLabelLinkClickedEventArgs eventos. Puede usar este objeto para obtener el objeto asociado al hipervínculo en el que el usuario LinkLabel.Link hizo clic. Todos los hipervínculos contenidos en el LinkLabel control se almacenan en la instancia de clase para el control LinkLabel.LinkCollection.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.linklabel?view=windowsdesktop-6.0>

**15 Checkedlistbox.**

Este control presenta una lista de elementos que el usuario puede navegar mediante el teclado o la barra de desplazamiento del lado derecho del control. El usuario puede colocar una marca de verificación por uno o varios elementos y los elementos activados se pueden navegar con CheckedListBox.CheckedItemCollection y CheckedListBox.CheckedIndexCollection.

Para agregar objetos a la lista en tiempo de ejecución, asigne una matriz de referencias de objeto con el AddRange método. A continuación, la lista muestra el valor de cadena predeterminado para cada objeto. Puede agregar elementos individuales a la lista con el Add método.

El CheckedListBox objeto admite tres estados a través de la CheckState enumeración: y Checked Indeterminate Unchecked . Debe establecer el estado de en el código porque la interfaz de usuario de un no proporciona Indeterminate CheckedListBox un mecanismo para hacerlo.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.checkedlistbox?view=windowsdesktop-6.0>

**16 DateTimePicker. Clase 3**

**17 ListView. Clase 3**

**18 NotifyIcon. Clase 3**

**19 NumericUpDown. Clase 3**

**20 PictureBox. Clase 3**

**21 RichtextBox. Clase 3**

**22 ToolTip. Clase 4**

**23 TreeView. Clase 4**

**24 WebBrowser. Clase 4**

**25 MenuStrip. Clase 4**

**26 FolderBrowseDialog. Clase 4**

**27 OpenFileDialog. Clase 4**

**28 SaveFileDialog. Clase 4**

**29 PrintDialog. Clase 4**

Tipos de Datos y Variables

<https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/types-and-variables>

Arrays

<https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/arrays/>

Estructuras de decisión

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/if-else>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/switch>

Estructuras de repetición

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/do>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/for>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/foreach-in>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/while>

Instrucciones de salto

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/break>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/continue>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/return>