**DataGridView (Control, formularios Windows Forms)**

El control DataGridView proporciona una forma eficaz y flexible de mostrar datos en formato de tabla. Puede utilizar el control DataGridView para mostrar vistas de sólo lectura de una cantidad pequeña de datos o puede ajustar su tamaño para mostrar vistas modificables de conjuntos muy grandes de datos.

Puede ampliar el control DataGridView de varias maneras para construir comportamientos personalizados en las aplicaciones. Por ejemplo, puede especificar mediante programación sus propios algoritmos de ordenación y crear sus propios tipos de celdas. Puede personalizar con facilidad la apariencia del control DataGridView eligiendo entre varias propiedades. Se pueden utilizar muchos tipos de almacenes de datos como origen de datos o el control DataGridView puede operar sin tener ningún origen de datos enlazado.

En los temas de esta sección se describen los conceptos y las técnicas que puede utilizar para compilar características de DataGridView en las aplicaciones.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pasar un argumento con ByVal y ByRef**

**Visual Studio .NET 2003**

Personas que lo han encontrado útil: 1 de 1 - [Valorar este tema](/l)

En Visual Basic, puede pasarse un argumento a un procedimiento *por valor* o *por referencia*, especificando las palabras clave **ByVal** o **ByRef**, respectivamente. Pasar un argumento por valor implica que el procedimiento no puede modificar el contenido del elemento variable en el código de llamada subyacente al argumento. Pasar un argumento por referencia permite que el procedimiento modifique el contenido de la misma forma que puede hacerlo el propio código de llamada.

La distinción que se hace entre pasar los argumentos por valor y por referencia no tiene nada que ver con la clasificación de los tipos de datos en tipos de valor y tipos de referencia. No obstante, ambas categorías interactúan entre sí.

**Argumentos variables y no variables**

El elemento de programación subyacente a un argumento puede ser un elemento variable, con capacidad para cambiar de valor, o un elemento no variable. La siguiente tabla muestra elementos variables y no variables.

|  |  |
| --- | --- |
| **Elementos variables** | **Elementos no variables** |
| Variables declaradas, incluidas las variables de objetos | Constantes |
| Campos (de clases) | Literales |
| Elementos matriciales | Enumeraciones |
| Elementos estructurales | Expresiones |

Los argumentos no variables no se modifican nunca en el código de llamada, aunque se pasen por referencia. El procedimiento al que se llama podría modificar su copia del argumento no variable, pero la modificación no afectaría al elemento subyacente al código de llamada.

**En esta sección**

[Pasar argumentos con ByVal](http://msdn.microsoft.com/es-es/library/cc436531(v=vs.71).aspx)

Explica cómo pasar argumentos por valor, lo que significa que el procedimiento no puede modificar la variable en sí.

[Pasar argumentos con ByRef](http://msdn.microsoft.com/es-es/library/cc436528(v=vs.71).aspx)

Explica cómo pasar argumentos por referencia, lo que significa que el procedimiento puede modificar la variable en sí.

[Mecanismo para pasar argumentos](http://msdn.microsoft.com/es-es/library/cc436535(v=vs.71).aspx)

Resume la interacción entre el tipo de datos del elemento y el mecanismo para pasar argumentos.

**Secciones relacionadas**

[Procedimientos](http://msdn.microsoft.com/es-es/library/cc437188(v=vs.71).aspx)

Proporciona vínculos a temas sobre los procedimientos de Visual Basic, información sobre los distintos tipos de procedimientos y la manera de llamarlos.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

split

Dividir separa strings. A menudo, los strings tienen caracteres delimitadores en sus datos. Los delimitadores son secuencias por comas y tabuladores " \ r \ n " secuencias de nueva línea y . Dividir maneja la división en cadena de caracteres y delimitadores .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dim

Declara y asigna espacio de almacenamiento para una o más variables .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

streamreader

Implementa un [TextReader](http://msdn.microsoft.com/es-es/library/system.io.textreader(v=vs.110).aspx) que lee los caracteres de una secuencia de bytes en una codificación determinada.

[System.Object](http://msdn.microsoft.com/es-es/library/system.object(v=vs.110).aspx)   
[System.MarshalByRefObject](http://msdn.microsoft.com/es-es/library/system.marshalbyrefobject(v=vs.110).aspx)  
[System.IO.TextReader](http://msdn.microsoft.com/es-es/library/system.io.textreader(v=vs.110).aspx)  
System.IO.StreamReader