# Estándar de codificación

Convenciones de diseño

Un buen diseño utiliza el formato para enfatizar la estructura de su código y hacer que el código sea más fácil de leer. Los ejemplos y ejemplos de Microsoft se ajustan a las siguientes convenciones:

Use la configuración predeterminada del Editor de código (sangría inteligente, sangrías de cuatro caracteres, tabulaciones guardadas como espacios). Para obtener más información, consulte Opciones, Editor de texto, C #, Formato.

Escriba solo una declaración por línea.

Escriba solo una declaración por línea.

Si las líneas de continuación no están sangradas automáticamente, sangra una tabulación (cuatro espacios).

Agregue al menos una línea en blanco entre las definiciones de métodos y las definiciones de propiedades.

Use paréntesis para hacer que las cláusulas en una expresión sean aparentes, como se muestra en el siguiente código:

if ((val1 > val2) && (val1 > val3))

{

// Take appropriate action.

}

Convenciones que comentan

Coloque el comentario en una línea separada, no al final de una línea de código. Comience el texto del comentario con una letra mayúscula. Termina el texto del comentario con un punto. Inserte un espacio entre el delimitador de comentarios (//) y el texto del comentario, como se muestra en el siguiente ejemplo.

No cree bloques de asteriscos formateados alrededor de los comentarios:

// The following declaration creates a query. It does not run

// the query.

Arrays

Use la sintaxis concisa cuando inicialice matrices en la línea de declaración:

// Preferred syntax. Note that you cannot use var here instead of string[].

string[ ] vowels1 = { "a", "e", "i", "o", "u" };

// If you use explicit instantiation, you can use var.

var vowels2 = new string[] { "a", "e", "i", "o", "u" };

// If you specify an array size, you must initialize the elements one at a time.

var vowels3 = new string[5];

vowels3[0] = "a";

vowels3[1] = "e";

// And so on.

try-catch y uso de sentencias en el manejo de excepciones

Use una declaración try-catch para la mayoría de los controles de excepciones:

static string GetValueFromArray(string[] array, int index)

{

try

{

return array[index];

}

catch (System.IndexOutOfRangeException ex)

{

Console.WriteLine("Index is out of range: {0}", index);

throw;

}

}

Simplifique su código usando el C # usando la declaración. Si tiene una declaración try-finally en la que el único código en el bloque finally es una llamada al método Dispose, use una instrucción using en su lugar:

// This try-finally statement only calls Dispose in the finally block.

Font font1 = new Font("Arial", 10.0f);

try

{

byte charset = font1.GdiCharSet;

}

finally

{

if (font1 != null)

{

((IDisposable)font1).Dispose();

}

}

// You can do the same thing with a using statement.

using (Font font2 = new Font("Arial" , 10.0f))

{

byte charset = font2.GdiCharSet;

}