Guía de estilos C#

No escribir código de varias lineas en una línea. Por ejemplo, si tengo un constructor

así: public Clase(int cantidad) { this.cantidad = cantidad; }

Se debe escribir

```
public Clase (int cantidad)
{
        this.cantidad = cantidad;
}
```

Nomenclatura

Las convenciones generales de nomenclatura explican la elección de los nombres más adecuados para los elementos de sus bibliotecas. Estas instrucciones se aplican a todos los identificadores.

Elección de palabra

- Elija los nombres fácilmente legibles para los identificadores. Por ejemplo, una propiedad denominada HorizontalAlignment es más legible en inglés que AlignmentHorizontal.
- Es preferible la legibilidad a la brevedad. El nombre de propiedad CanScrollHorizontally es mejor que ScrollableX (una referencia oculta al eje X). • No utilice guiones de subrayado, guiones ni ningún otro carácter no alfanumérico. • No utilice la notación húngara. La notación húngara consiste en incluir un prefijo en los identificadores para codificar ciertos metadatos sobre el parámetro, como puede ser el tipo de datos del identificador.
- Evite utilizar identificadores que están en conflicto con palabras clave de lenguajes de programación ampliamente utilizados. Aunque los lenguajes conformes a CLS deben proporcionar una manera de utilizar palabras clave como palabras normales, los procedimientos recomendados indican que no debería obligar a los desarrolladores a saber cómo hacerlo.

Abreviaturas y acrónimos

- En general, no debería utilizar abreviaturas ni acrónimos. Estos elementos hacen que los nombres sean menos legibles. De igual forma, es difícil saber cuándo es seguro suponer que un acrónimo es ampliamente reconocido.
- No utilice abreviaturas ni contracciones como parte de nombres de identificadores.

Por ejemplo, use OnButtonClick en lugar de OnBtnClick.

- No utilice ningún acrónimo que no esté ampliamente aceptado y, además, sólo cuando necesario.

Estilos de grafía

Las condiciones siguientes describen distintas maneras de usar mayúsculas y minúsculas de los identificadores.

- Grafía Pascal: La primera letra del identificador y la primera letra de las siguientes palabras concatenadas están en mayúsculas. El estilo de mayúsculas y minúsculas Pascal se puede utilizar en identificadores de tres o más caracteres. Por ejemplo: BackColor.
- Grafía Camel: La primera letra del identificador está en minúscula y la primera letra de las siguientes palabras concatenadas en mayúscula. Por ejemplo: backColor.
- Mayúsculas: Todas las letras del identificador van en mayúsculas. Por ejemplo: IO.

Reglas de uso de mayúsculas y minúsculas para los identificadores

Cuando un identificador está compuesto de varias palabras, no utilice separadores, como guiones de subrayado ("_") ni guiones ("-"), entre las palabras. En su lugar, utilice la grafía para señalar el principio de cada palabra.

Utilice la grafía Pascal para todos los nombres de miembros públicos, tipos y espacios de nombres que constan de varias palabras. Tenga en cuenta que esta regla no se aplica a las instancias de campos.

Utilice el uso combinado de mayúsculas y minúsculas tipo Camel para los nombres de parámetros.

Identificador	Estilo de grafía	Ejemplo
Clase	Pascal	AppDomain
Tipo de enumeración	Pascal	ErrorLevel
Valores de enumeración	Pascal	FatalError
Evento	Pascal	ValueChanged
Clase de excepción	Pascal	WebException
Campo estático de sólo lectura	Pascal	RedValue
Interfaz	Pascal	IDisposable
Método	Pascal	ToString
Espacio de nombres	Pascal	System.Drawing
Parámetro	Camel	typeName
Propiedad	Pascal	BackColor

Documentación

Utilizar la documentación de métodos. Colocando sobre la firma de un método /// se autocompletará el bloque de documentación:

```
/// <summary>
/// Descripción del método
/// </summary>
/// <param name="dato">Información sobre el parámetro</param>
/// <returns>Información sobre el retorno del método</returns>
private static int Metodo(string dato)
{
 // ...
}
```

También llevar comentarios dentro de los métodos describiendo porciones de código complejas o funcionalidades largas a fin de la comprensión actual y futura de dichas líneas. Para este fin utilizar // para comentar una línea simple o /* */ para comentar un bloque.

```
// Comentario de una línea

/*

Comentario de bloque.

Varias líneas

*/
```