

Este artigo foi traduzido por máquina. Para visualizar o arquivo em inglês, marque a caixa de seleção Inglês. Você também pode exibir o texto Em inglês em uma janela pop-up, movendo o ponteiro do mouse sobre o texto.

Convenções de maiúsculas e minúsculas

Publicado: abril de 2016

As diretrizes neste capítulo dispor de um método simples para uso de caso que, quando aplicadas de forma consistente, identificadores de marca para tipos, membros e parâmetros de fácil leitura.

Regras de maiúsculas e minúsculas de identificadores

Para diferenciar as palavras em um identificador, maiúscula a primeira letra de cada palavra no identificador. Não use sublinhados para diferenciar palavras, ou, na verdade, em qualquer lugar em identificadores. Há duas maneiras apropriadas para aproveitar os identificadores, dependendo do uso do identificador:

- PascalCasing
- camelCasing

A convenção de PascalCasing, usada para todos os identificadores, exceto os nomes de parâmetro, explora o primeiro caractere de cada palavra (incluindo acrônimos sobre duas letras), conforme mostrado nos exemplos a seguir:

```
PropertyDescriptor  
HtmlTag
```

Um caso especial é feito para abreviações de duas letras em que ambas as letras estão em maiúsculas, como mostra o seguinte identificador:

```
IOStream
```

A convenção de camelCasing, usada somente para os nomes de parâmetro, explora o primeiro caractere de cada palavra, exceto a primeira palavra, conforme mostrado nos exemplos a seguir. Como o exemplo também mostra, acrônimos de duas letras que começam um identificador camel case estão em minúsculas.

```
propertyDescriptor
```

`ioStream``htmlTag`

✓ **FAZER** usar PascalCasing para todos os membros, tipo e namespace nomes públicos consiste em várias palavras.

✓ **FAZER** usar camelCasing para nomes de parâmetro.

A tabela a seguir descreve as regras de maiúsculas e minúsculas para diferentes tipos de identificadores.

identificador	Maiúsculas e minúsculas	Exemplo
namespace	Pascal	<code>namespace System.Security { ... }</code>
Tipo	Pascal	<code>public class StreamReader { ... }</code>
Interface	Pascal	<code>public interface IEnumerable { ... }</code>
Método	Pascal	<code>public class Object { public virtual string ToString(); }</code>
Propriedade	Pascal	<code>public class String { public int Length { get; } }</code>
event	Pascal	<code>public class Process { public event EventHandler Exited; }</code>
Campo	Pascal	<code>public class MessageQueue { public static readonly TimeSpan InfiniteTimeout; } public struct UInt32 { public const Min = 0; }</code>
Valor de enumeração	Pascal	<code>public enum FileMode { Append, ... }</code>
PARAMETER	Minúscula	<code>public class Convert { public static int ToInt32(string value); }</code>

Capitalização de palavras compostas e termos comuns

A maioria dos termos compostos são tratados como palavras individuais para fins de maiúsculas e minúsculas.

X NÃO cada palavra em palavras compostas de forma fechada chamadas em maiúscula.

Essas são as palavras compostas escritas como uma única palavra, como ponto de extremidade. Para fins de diretrizes de maiúsculas e minúsculas, trate uma palavra composta de forma fechada como uma única palavra. Use um dicionário atual para determinar se uma palavra composta é escrita em forma fechada.

Pascal	Minúscula	Não
BitFlag	bitFlag	Bitflag
Callback	callback	CallBack
Canceled	canceled	Cancelled
DoNot	doNot	Don't
Email	email	EMail
Endpoint	endpoint	EndPoint
FileName	fileName	Filename
Gridline	gridline	GridLine
Hashtable	hashtable	HashTable
Id	id	ID
Indexes	indexes	Indices
LogOff	logOff	LogOut
LogOn	logOn	LogIn
Metadata	metadata	MetaData, metaData
Multipanel	multipanel	MultiPanel

MultiView	multiview	MultiView
Namespace	namespace	NameSpace
Ok	ok	OK
Pi	pi	PI
Placeholder	placeholder	PlaceHolder
SignIn	signIn	SignOn
SignOut	signOut	SignOff
UserName	userName	Username
WhiteSpace	whiteSpace	Whitespace
Writable	writable	Writeable

Diferenciação de maiúsculas e minúsculas

Idiomas que podem ser executados no CLR não são necessários para dar suporte a diferenciação de maiúsculas e minúsculas, embora alguns façam. Mesmo se o seu idioma oferece suporte a ele, outras linguagens que podem acessar a sua estrutura, não. Quaisquer APIs que são acessíveis externamente, portanto, não pode depender caso sozinho para distinguir entre os dois nomes no mesmo contexto.

X NÃO pressupor que todas as linguagens de programação diferenciam maiúsculas de minúsculas. Eles não são. Nomes não podem diferir somente maiúsculas e minúsculas.

Partes © 2005, 2009 Microsoft Corporation. Todos os direitos reservados.

Reimpresso com permissão de Pearson Education, Inc. de [as diretrizes de Design do Framework: convenções, linguagens e padrões para bibliotecas do .NET reutilizável](#), 2nd Edition por Krzysztof Cwalina e Brad Abrams, publicados 22 de outubro de 2008 pela Addison-Wesley Professional, como parte da série de desenvolvimento do Microsoft Windows.

Confira Também

[Diretrizes de Design de estrutura](#)

[Diretrizes de nomenclatura](#)

