

Tema 3: Identificadores, tipos y operadores de C#

- Comentarios
- Identificadores
- Variables
- Tipos de datos
- Operadores

07/10/2016

1

Comentarios

- `/*<texto>*/`
- `/*` Esto es un comentario que ejemplifica cómo se escribe comentarios que ocupen varias líneas `*/`
- `// <texto>`: Fin de línea

07/10/2016

2

Identificadores

- Sólo se pueden utilizar: letras mayúsculas, minúsculas, dígitos y carácter `_`.
- No puede comenzar por dígito.
- No puede contener blancos.
- No puede ser igual a una palabra reservada.
- Sensible a mayúsculas y minúsculas

07/10/2016

3

Palabras reservadas

abstract	As	base	bool	break	byte	case	catch
char	Checked	class	Const	continue	decimal	default	delegate
do	double	else	enum	event	explicit	extern	false
finally	fixed	float	For	foreach	goto	if	implicit
int	In	interface	internal	lock	is	long	namespace
new	Null	object	operator	out	override	params	private
protected	public	readonly	ref	return	sbyte	sealed	short
sizeof	stackalloc	static	String	struct	switch	this	throw
True	Try	typeof	uint	ulong	unchecked	unsafe	ushort
Using	virtual	void	while				

07/10/2016

4

Variables

- No utilizar _
- No crear identificadores que sólo se diferencien por mayúsculas/minúsculas: miResultado MiResultado.
- Comenzar por minúscula
- En un identificador de varias palabras: comenzar la segunda y sucesiva por letra Mayúscula: mediaAritmetica, valorTotal
- No utilizar notación húngara

07/10/2016

5

Declaración de variables

- Declaración simple
 - <tipo de datos> <identificador>;
- Declaración múltiple:
 - <tipo de datos> <identificador1>, <identificador2>, ... ;
- Declaración e inicialización:
 - <tipo de datos> <identificador1> = <valor>;

07/10/2016

6

Tipos de datos primitivos

Tipo	Descripción	Tamaño (bits)	Rango
int	Enteros	32	$-2^{31} \rightarrow 2^{31}-1$
long	Enteros largos	64	$-2^{63} \rightarrow 2^{63}-1$
float	Reales	32	$-3.4 \cdot 10^{38} \rightarrow 3.4 \cdot 10^{38}$
double	Reales prec. Doble	64	$-1.7 \cdot 10^{308} \rightarrow 1.7 \cdot 10^{308}$
decimal	Moneda	128	28 cifras signifs.
string	Cadena caracteres	16 por carácter	
char	Carácter	16	$0 \rightarrow 2^{16}-1$
bool	Lógico	8	true, false

07/10/2016

7

Operadores

- Operadores aritméticos:

Operador	Significado
+	Suma
-	Resta
*	Producto
/	División
%	Resto división entera (para int ó float) en C no
++	Incremento
--	Decremento

07/10/2016

8

Operadores

Operadores lógicos:

Operador	Significado
&&	AND lógico (sí evaluación perezosa)
	OR lógico (sí evaluación perezosa)
!	NOT lógico

07/10/2016

9

Operadores

Operaciones relacionales:

Operador	Significado
==	Igual a
!=	Distinto
<	Menor
<=	Menor o igual
>	Mayor
>=	Mayor o igual

07/10/2016

10

Operadores

Operaciones de manipulación de bits

Operador	Significado
&	AND binario
	OR binario
^	XOR binario
~	Complemento a 1
<<	Rotación izquierda
>>	Rotación derecha

07/10/2016

11

Operadores

Operaciones de asignación

Operador	Asignación abreviada
+=	Suma abreviada
-=	Resta abreviada
*=	Producto
/=	División
%=	Resto división
&=	AND binario
=	OR binario
^=	XOR binario
<<=	Desplazam. Izda
>>=	Desplazam. Dcha

07/10/2016

12

Operadores

■ Operaciones con cadenas

Operador	Significado
+	Suma
-	Resta
*	Producto
/	División
%	Resto división entera (para int ò float) en C no
++	Incremento
--	Decremento

07/10/2016

13

Operadores

■ Operador condicional

- <condición> ? <expresión1> : <expresión2>

07/10/2016

14

Operadores

■ Operaciones sobre tipos

- `typeof(<nombreTipo>)`
- `<expresión> is <nombreTipo>`
- `sizeof(<nombreTipo>)`

07/10/2016

15

Operadores

■ Operaciones de conversión

- `(<tipoDestino>) <expresion>;`
- `<expresión> as <tipoDestino>`

07/10/2016

16

Operadores

■ Operaciones sobre tipos

- As sólo es aplicable a tipos referencia y sólo a aquellos casos en que existan conversiones predefinidas en el lenguaje. Incluye conversiones entre un tipo y tipos padres suyos y entre un tipo y tipos hijos suyos.
- El programador puede definir cómo hacer conversiones de tipos por él definidos y otros mediante el operador (), pero no mediante as. Sin embargo, () sí que implica conversión si el <tipoDestino> no es compatible con el tipo del objeto referenciado.
- En caso de que se solicite hacer una conversión inválida as devuelve null mientras que () produce una excepción `System.InvalidCastException`.