

3. Tipos de datos, variables y expresiones

- Contenidos
 - Introducción
 - Concepto de variable
 - Tipos de datos
 - Operadores

Introducción

- En C# todos los objetos derivan de la clase System.Object
- Clasificación
 - Tipos valor: contienen directamente datos
 - Tipos referencia: almacenan la referencia a otro dato

Concepto de variable

- Zona de memoria donde se almacenan datos
- Atributos:
 - Nombre:
 - debe ser significativo
 - debe comenzar por una letra o guión bajo, se puede utilizar cualquier carácter UNICODE pero no debe contener espacios en blanco
 - C# es sensible a mayúsculas
 - Tipo: describe el uso de la variable
 - Valor que tiene asociado en un determinado momento

Concepto de variable

- Declaración de variables
 - Se le da un nombre a una variable para su utilización en el programa
 - Se le asigna un tipo de dato
 - Representación en memoria
 - Conjunto de valores posibles
 - Operaciones permitidas
 - Forma de realizar las operaciones
 - Sintaxis:
`tipo_de_dato nombre_de_variable;`

Tipos de datos

■ Tipos valor

Tipo C#	Rango
bool	{true, false}
byte, sbyte	[0, 255], [-128, 127]
ushort, short	[0, 65535], [-32768, 32767]
uint, int	[0, 4294967295], [-2147483648, 2147483647]
ulong, long	...
float	real con 7 dígitos
double	real con 15 dígitos
decimal	real con 28 dígitos
char	caracteres UNICODE

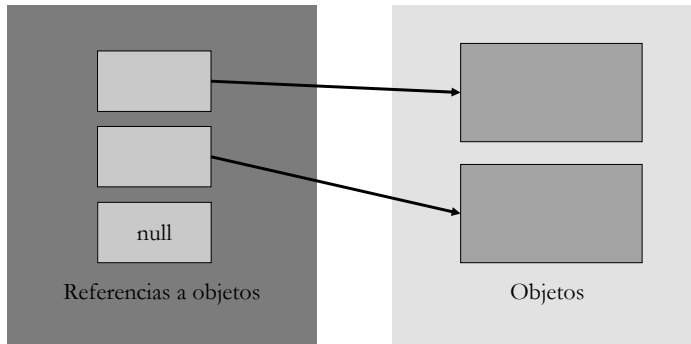
Tipos de datos

■ Declarar variables para almacenar:

- Edad de una persona
- Coordenada X de un punto
- Saldo de una cuenta bancaria
- Número de habitantes de Valencia
- Sexo de una persona
- Estado de una bombilla (encendida o apagada)

Tipos de datos

■ Tipos referencia



Tipos de datos

■ Tipo class

```
[modificadores] class identificador [:clase-base]
{
    //cuerpo de la clase
    //declaraciones de los miembros de la clase
}
```

Tipos de datos

- Ejemplo de clase

```
public class punto
{
    protected float x=0, y=0;

    public void trasladar (float dx, float dy)
    {
        x=x+dx;
        y=y+dy;
    }
}
```

Tipos de datos

- Tipo string

- Representa una cadena de caracteres UNICODE

- Declaración de variables:

```
string unaCadena="Hola";
```

- Concatenación de cadenas:

```
string unaCadena="Hola";
```

```
string otraCadena="mundo";
```

```
string cadenaFinal=unaCadena+" "+otraCadena;
```

Operadores

- Símbolo formado por uno o más caracteres que permite realizar una operación entre uno o más datos y produce un resultado
- Operadores aritméticos
 - +, -, *
 - / : división entera/real
 - % : resto de la división entera
 - -x, +x

Operadores

- Operadores relacionales
 - >, <, <=, >=
 - == : igualdad
 - != : desigualdad
 - Devuelven un valor tipo bool

Operadores

- Operadores lógicos

- `&&`, `||`, `!`

- Tabla de verdad

A	B	A&&B	A B	!A
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

Operadores

- Operadores de asignación

- Se utiliza para dar valores a una variable

- Sintaxis: `nombre_variable = expresion;`

- Operación destructiva

- Cuestión: Dadas dos variables a y b que contienen dos valores determinados, escribir las instrucciones de asignación necesarias para que a y b intercambien sus valores

Operadores

- Operadores de asignación compuestos
 - +=, -=, *=, /=
 - ++, --

Operadores

- Operadores de creación de objetos
 - new nombre_tipo (parámetros)
 - Constructor
 - Ejemplo con la clase Punto:
Punto p1 = new Punto();
- Operadores de acceso a objetos
 - objeto.miembro
 - Ejemplo: p1.trasladar(4, -1);

Operadores

- Operadores de conversión
 - Conversión implícita:
 - de un tipo de menor capacidad hacia un tipo de mayor capacidad
 - de un tipo sin signo a un tipo con signo
 - Ejemplos:
short corto=3;
int entero=corto;

int entero=300;
short corto=entero; ERROR

Operadores

- Operadores de conversión
 - Conversión explícita
 - Ejemplo:
int entero=300;
short corto=(short) entero;
- Precedencia de operadores

Ejercicios

Expresiones	Expresiones
2/8	4.0/8
3.2/0.8	6.0/3.0
35/35	0.75/0.25
10/3	10/-3
-10%3	10%3

Expresiones	Expresiones
1<15	2==(1+1)
(1+18)>=3	3.3<3
(2+1)>(2+5)	'a'!='A'
'A'<'Z'	5.5>5.3
12.5<0	1!=1

Ejercicios

Expresiones	Expresiones
(8>=6)&&(2==2)	(3>15) ((7.3<=10)
(8>=7)&&(2==2)	!(17>5)
(6.8<11)&&(1>100)	(!(13>2)) ((15>8)

Ejercicios

```
a = 2.0;  
b = 3.5;  
c = b*a/4.0;  
c = c+b;
```

```
a=8;  
b=15;  
c=4;  
r1=2*((a%5)*(4+(b-3)/(c+2)));
```

```
a=1;  
b=2;  
c=3;  
d=4;  
r1= a-b/c*d;  
r2= (a-b)/(c*d);
```

```
a=1;  
b=2;  
c='S';  
r1=c=='C';  
r2=b>=a;
```