

Expresiones y estructuras de control

Sentencias de control

- **Estructuras de control:** Regulan el flujo de ejecución de un programa o función.
- Las estructuras de control que existen son:
 - Secuencial
 - Selección
 - Repetición



Estructuras de control selectivas

- **Sentencia if:** Es la estructura de control de selección principal.

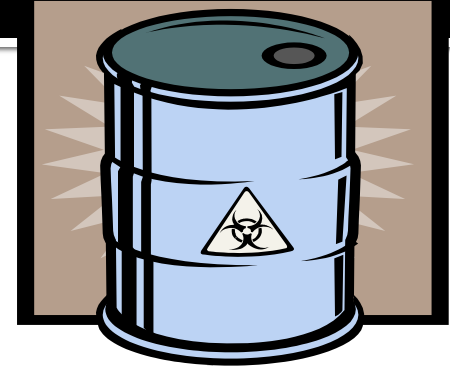
- **Formato:**

`if (condición) acción`



`condición` es una expresión entera (lógica).
`acción` es cualquier sentencia o sentencias ejecutables, que se ejecutarán sólo si la condición toma un valor distinto de cero.

Estructuras de control selectivas



■ Ejemplo:

```
class Program
{
    static void Main(string[] args)
    {
        int n, d;
        Console.WriteLine("Introduce dos enteros");
        n = Convert.ToInt32(Console.ReadLine());
        d = Convert.ToInt32(Console.ReadLine());
        //Console.WriteLine(n % d == 0);

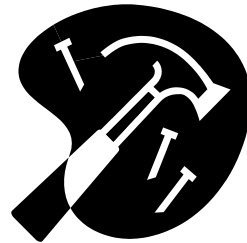
        if (n % d == 0)
            Console.WriteLine("{0} es divisible entre {1}", n, d);
    }
}
```

Expresiones simples y complejas

- **Ejercicio:**

Modificar el programa anterior para mostrar si dos números positivos son divisibles

.

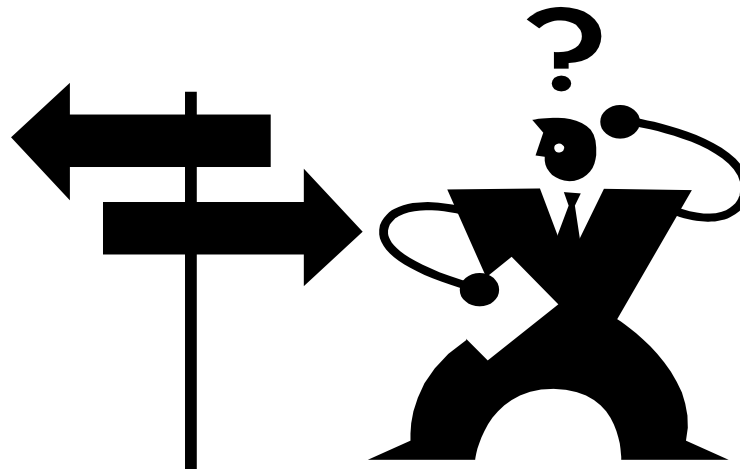


Estructuras de control selectivas

- Sentencia if de dos opciones: `if-else`

- **Formato:**

```
if (condición) acción1     else acción2
```

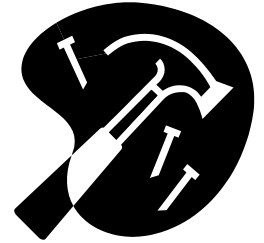


Estructuras de control selectivas

Ejemplo:

```
class Program
{
    static void Main(string[] args)
    {
        int n, d;
        Console.WriteLine("Introduce dos enteros");
        n = Convert.ToInt32(Console.ReadLine());
        d = Convert.ToInt32(Console.ReadLine());
        //Console.WriteLine(n % d == 0);

        if (n % d == 0)
            Console.WriteLine("{0} es divisible entre {1}",
                               n, d);
        else
            Console.WriteLine("{0} no es divisible entre
                               {1}", n, d);
    }
}
```



Ejercicio: Generar 2 números aleatorios e imprimir cuál es el mayor.

Estructuras de control selectivas

- **Sentencias `if-else` anidadas:** Las sentencias `if-else` anidadas permiten implementar decisiones que impliquen diferentes opciones.

- **Formato:**

```
if (condición1)  
    acción1  
else if (condición2)  
    acción2  
...  
else if (condiciónn)  
    acciónn  
else  
    acciónn+1
```

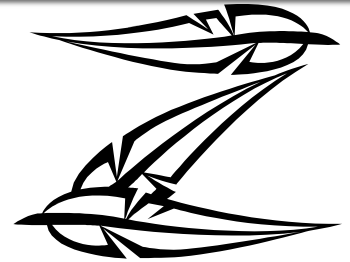


Estructuras de control selectivas

- Ejemplo:

```
class Program
{
    static void Main(string[] args)
    {
        int n, d;
        double z;
        Console.WriteLine("Introduce dos enteros");
        n = Convert.ToInt32(Console.ReadLine());
        d = Convert.ToInt32(Console.ReadLine());

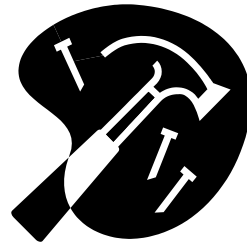
        if (n > 0)
        {
            z = 2 * Math.Log(n);
            Console.WriteLine("Valor de z: {0}", z);
        }
        else if (n < 0 && d > 0)
        {
            z = Math.Abs(n) + Math.Sqrt(d);
            Console.WriteLine("Valor de z: {0}", z);
        }
        else
            Console.WriteLine("Imposible calcular z");
    }
}
```



Expresiones simples y complejas

- **Ejercicio:**

Encender el ventilador de una automóvil cuando la temperatura sea mayor de la temperatura ambiente, pero si la temperatura es menor a 10°C , prender la calefacción del automóvil, en caso contrario apagar tanto la calefacción como el ventilador.



Estructuras de control selectivas

- Las sentencias if-else anidadas también se pueden implementar con sentencias if-else.
- **Ejemplo:**

if-else	if-else anidadas
<pre>if(x > 0) numero_pos++; else if(x < 0) numero_neg++; else numero_ceros++;</pre>	<pre>if(x > 0) numero_pos++; else if(x < 0) numero_neg++; else numero_ceros++;</pre>

Estructuras de control selectivas

- **Sentencia switch:** Es una sentencia de control útil para seleccionar una de entre múltiples alternativas. El *selector* puede ser de tipo int o char.

- **Formato:**

```
switch(selector) {  
    case etiqueta1:  sentencias1;  
                      break;  
    case etiqueta2:  sentencias2;  
                      break;  
    ...  
    case etiquetan:  sentenciasn;  
                      break;  
    default: sentencias; //opcional  
}
```



Estructuras de control selectivas



- **Ejemplo:**

```
class Program
{
    static void Main(string[] args)
    {
        int nota;

        Console.WriteLine("Introduce una nota");
        nota = Convert.ToInt32(Console.ReadLine());

        switch (nota)
        {
            case 10: Console.WriteLine("Felicidades, examen superado");
                    break;
            case 9: Console.WriteLine("Notable");
                   break;
            case 8: Console.WriteLine("Aprobado");
                   break;
            default: Console.WriteLine("Necesitas estudiar más");
                    break;
        }
    }
}
```

- **Ejercicio:** Programa que imprima si una letra es una vocal o no.

Estructuras de control selectivas

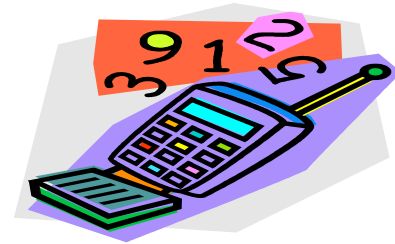
- **Caso particular case:** Se pueden tener varios case en una alternativa.

- **Ejemplo:**

```
class Program
{
    static void Main(string[] args)
    {
        char letra;
        Console.WriteLine("Introduce un caracter");
        letra = char.Parse(Console.ReadLine());

        switch (letra)
        {
            case '0': case '1': case '2': case '3': case '4': case '5':
            case '6': case '7': case '8': case '9':

                Console.WriteLine("Es un digito");
                break;
            default: Console.WriteLine("No es un digito");
                break;
        }
    }
}
```



Estructuras de control repetitivas

- Las estructuras repetitivas realizan la repetición o iteración de acciones.
- **Sentencia `while`:** Repite una sentencia o sentencias n de veces.

- **Formato:**

```
while (condición)
    sentencia;
```

```
while (condición) {
    sentencia1;
    sentencia2;
    ...
    sentencian;
}
```

Funcionamiento:

1. Se evalúa la condición
2. Si la condición es verdadera:
 - a. Se ejecuta la sentencia o sentencias dentro del `while`
 - b. Se regresa al paso 1
3. Si la condición es falsa: se sale del bucle o ciclo `while` y se continúa con el programa.



Estructuras de control repetitivas

- **Ejemplo:**

```
class Program
{
    static void Main(string[] args)
    {
        int x = 0;
        while (x < 10)
            Console.WriteLine("x: {0}", ++x);
    }
}
```



- **Ejercicio:**

- Programa que pida cinco números del teclado y que imprima al final su multiplicación.

Estructuras de control repetitivas

- **Ejemplo:** Programa que genera aleatorios entre 1 y 199

```
class Program
{
    static void Main(string[] args)
    {
        Random r = new Random();
        int x=0;

        while (x < 100)
        {
            //Genera aleatorios entre 1 y 199
            x = r.Next(1, 200);
        }
        Console.WriteLine("Genero un valor
aleatorio mayor o igual a 100; {0}",x);
    }
}
```



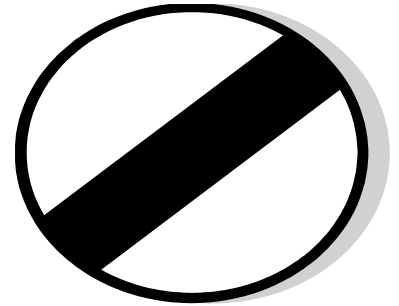
Estructuras de control repetitivas

- `while` infinito:

- **Ejemplo:**

```
class Program
{
    static void Main(string[] args)
    {
        int contador=0;

        while (true)
        {
            Console.WriteLine("Contador: {0}",
                               contador++);
        }
    }
}
```



Estructuras de control repetitivas

■ Ejemplo:

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        int contador=0;
```

```
        while (true)
```

```
        {
```

```
            contador++;
```

```
            if (contador > 10)
```

```
                break;
```

```
        }
```

```
        Console.WriteLine("Contador: {0}",  
            contador++);
```

```
    }
```

```
}
```



Estructuras de control repetitivas

- **Bucle for:** La sentencia for sirve para ejecutar un bloque de sentencias un número fijo de veces.

- **Ejemplo:**

```
class Program
{
    static void Main(string[] args)
    {
        int i;

        for (i = 0; i < 10; i++)
            Console.WriteLine("Hola");
    }
}
```



- **Ejercicio:** Programa que imprima 10 múltiplos de 3.

Estructuras de control repetitivas

- Las variables de control del bucle for también pueden ser decimales.

- **Ejemplo:**

```
double x;
```

```
for(x = Math.pow(y, 3.0); x >  
2.0; x = Math.sqrt(x))
```

```
    Console.WriteLine("X: {0}", x);
```



Estructuras de control repetitivas

- **Bucle do-while:** Se utiliza para especificar un bucle condicional que se ejecuta al menos una vez.

- **Formato:**

```
do sentencia while (expresión);
```

```
do
```

```
    sentencia
```

```
while (expresión);
```



Estructuras de control repetitivas

■ Ejemplo:

```
class Program
{
    static void Main(string[] args)
    {
        int num = 0;

        do
        {
            Console.WriteLine("Introduzca un  
digito entre 0-9");
            num = int.Parse(Console.ReadLine());
        } while (num >= 0 && num <= 9);
        Console.WriteLine("Adios");
    }
}
```



Diagramas de Flujo

- **Juego de azar: craps**

Un jugador tira los dados. Cada dado tiene 6 caras. Una vez que los dados dejan de moverse se calcula la suma de los puntos de cada cara superior. Si la suma es 7 u 11 en el primer tiro el jugador GANA. Si la suma es 2, 3 o 12 en el primer tiro el jugador PIERDE. Si la suma es 4, 5, 6, 8, 9 o 10 en el primer tiro, se convierte en el **punto** del jugador. Para GANAR el jugador debe seguir tirando hasta que salga su **punto** y PIERDE si tira 7 antes de llegar a su **punto**.

