

COLEXIO <b>VIVAS</b> S.L.	RAMA:	Computing	CICLO:	MAP		
	MÓDULO	Service and processes programming				
	PROTOCOLO:		AVAL:		DATA:	
	Autor	Francisco Bellas Aláez (Curro)				

## Ejercicios Archivos, delegados y lambdas

### Ejercicio 1:

Realizar una solución en Visual Studio con varios proyecto que funcionen como comandos de consola.

Recuerda que lo que se le pasa a continuación del comando de consola se recibe como array de strings en un parámetro del Main (`string[] args`). Puedes ver un ejemplo en el apartado *Lista de Argumentos* del Apéndice II del tema 3: Otros aspecto de C#.

Los comandos serán los siguientes:

1. Comando **ls**: se le puede pasar un directorio o una variable de entorno. Muestra el contenido de dicho directorio, tanto subdirectorios como archivos. Diferenciará por colores y en los archivos además indicará el tamaño de los mismos.

**Ejemplos:**

```
ls %appdata%
ls e:\temp
```

2. Comando **cat**: Se le pasa un archivo de texto. Muestra el contenido del mismo en la consola. Si justo tras el comando tiene el argumento -n con un número a continuación solo mostrará la cantidad de líneas indicadas por dicho número (en este caso evita leer el archivo entero si no es necesario).

**Ejemplos:**

```
cat myfile.txt
cat -n5 c:\windows\win.ini
```

3. Comando **newfile**: Crea un archivo con el texto que haya a continuación. Si justo tras el comando se pone -a, añade el texto en vez de sobreescribirlo.

**Ejemplos:**

```
newfile myfile.txt "This text goes in the myfile."
newfile -a miarchivo.txt "And this one is added."
```

<b>COLEXIO VIVAS S.L.</b>	RAMA:	Computing	CICLO:	MAP			
	MÓDULO	Service and processes programming					
	PROTOCOLO:			AVAL:		DATA:	
	Autor	Francisco Bellas Aláez (Curro)					

## Ejercicio 2

Ampliar el programa gestor de Astros (Ejercicio 8 de los temas 1, 2 y 3 genéricos) de forma que guarde la información de la base de datos que se está creando.

La colección de Astros debe cargarse al iniciar el programa y guardarse en el disco duro al finalizar el mismo de forma transparente al usuario (el usuario no participa). Esos son los únicos cambios en el programa, al inicio y fin del mismo, no cambies nada más. Hazlo en la carpeta %APPDATA% del usuario.

Si no hiciste el ejercicio completo hazlo al menos con la posibilidad de disponer de una colección donde se puedan añadir y eliminar Planetas y Cometas para poder hacer el apartado de archivos.

## Ejercicio 3

Crea un delegado que tenga parámetro double y devuelva también un double.

Haz un pequeño programa que pida un número al usuario y luego le pregunte si desea el cuadrado o el cubo del mismo.

Una vez que el usuario ha escogido, debe asignarse a una variable del tipo delegado definido previamente la función lambda que realice la operación de cuadrado o cubo.

Finalmente se ejecuta la variable con la función que se haya seleccionado.

Toma como referencia el ejemplo de Operaciones de delegados pues es similar (si cabe, más simple).

COLEXIO <b>VIVAS</b> S.L.	RAMA:	Computing	CICLO:	MAP		
	MÓDULO	Service and processes programming				
	PROTOCOLO:		AVAL:		DATA:	
	Autor	Francisco Bellas Aláez (Curro)				

## Ejercicio 4

Crea estos dos vectores en un Main:

```
int[] notas = { 5, 2, 8, 1, 9, 4 };

string[] palabras = { "Sol", "Luna", "Estrella", "Cielo" };
```

Usa métodos de la clase Array junto con lambdas para obtener directamente estos elementos del vector notas:

- Saber de si hay algún aprobado (Si existe o no) en notas.
- Mostrar los aprobados de notas.
- Indicar la posición en el array del último aprobado
- Mostrar la nota del último aprobado.
- Cuanto tienen nota par.

Y ahora haz lo siguiente con el vector palabras:

- Cual es la primera palabra de más de 3 caracteres.
- Mostrar todas las palabras en mayúsculas.
- Indica la posición de la primera palabra que empiece por E.

## Ejercicio 5

a) Desarrolla una función para la realización de cualquier menú mediante delegados. Dicha función se denominará **MenuGenerator** y tendrá dos parámetros: El primero un vector de strings con los nombres de las opciones. El segundo un vector de delegados (hazlos que devuelvan void y sin parámetros y el nombre del delegado que sea MyDelegate) que serán las funciones que correspondan a cada opción del menú.

Al ejecutar MenuGenerator saldrá un menú clásico con las opciones indicadas por el primer vector y una última opción Exit (independientemente del número de entradas del menú). Si se introduce una opción no válida (fuera del rango) dará un mensaje de error y vuelve a pedirlo. Solo finaliza el menú si se selecciona la última opción, que será la de salir.

COLEXIO <b>VIVAS</b> S.L.	RAMA:	Computing	CICLO:	MAP		
	MÓDULO	Service and processes programming				
	PROTOCOLO:		AVAL:		DATA:	
	Autor	Francisco Bellas Aláez (Curro)				

MenuGenerator debe ser robusto ante parámetros no válidos de forma que devolverá true si el menú se crea correctamente y false si ha habido algún problema con algún parámetro (longitudes de vectores distintas, parámetro null,...)

Para probarlo utiliza las siguientes funciones y el siguiente programa principal:

```

static void f1()
{
    Console.WriteLine("A");
}

static void f2()
{
    Console.WriteLine("B");
}

static void f3()
{
    Console.WriteLine("C");
}

static void Main(string[] args)
{
    MenuGenerator(new string[] { "Op1", "Op2", "Op3" },
                  new MyDelegate[] { f1, f2, f3 });
    Console.ReadKey();
}

```

- b) Elimina las funciones f1, f2 y f3 y pásalas al array MyDelegate como expresiones lambda.

<b>COLEXIO</b> <b>VIVAS</b> S.L.	RAMA:	Computing	CICLO:	MAP			
	MÓDULO	Service and processes programming					CURSO: 2º
	PROTOCOLO:			AVAL:		DATA:	
	Autor	Francisco Bellas Aláez (Curro)					

## OPCIONAL

Obtención de datos de un archivo BMP. Teniendo en cuenta las siguientes cabeceras resumidas de las que aparecen en

[https://en.wikipedia.org/wiki/BMP\\_file\\_format](https://en.wikipedia.org/wiki/BMP_file_format)

Offset hex	Offset dec	Size	Purpose
00	0	2 bytes	<b>BM</b>
02	2	4 bytes	The size of the BMP file in bytes
06	6	2 bytes	Reserved
08	8	2 bytes	Reserved
0A	10	4 bytes	The offset, i.e. starting address, of the byte where the bitmap image data (pixel array) can be found.

Offset (hex)	Offset (dec)	Size (bytes)	Windows BITMAPINFOHEADER[2]
0E	14	4	the size of this header, in bytes (40)
12	18	4	the bitmap width in pixels (signed integer)
16	22	4	the bitmap height in pixels (signed integer)
1A	26	2	the number of color planes (must be 1)
1C	28	2	the number of bits per pixel, which is the color depth of the image. Typical values are 1, 4, 8, 16, 24 and 32.
1E	30	4	the compression method being used. 0 if no compression.
22	34	4	the image size.
26	38	4	the horizontal resolution of the image. (pixel per metre, signed integer)
2A	42	4	the vertical resolution of the image. (pixel per metre, signed integer)
2E	46	4	the number of colors in the color palette, or 0 to default to $2^n$
32	50	4	the number of important colors used, or 0 when every color is important;

Realiza una clase con un constructor al que le puedes pasar un string que debería ser la ruta a un archivo BMP. Constará de las siguientes funciones:

**IsBmp:** Función que devuelve true si es un BMP válido. Para realizar esta comprobación debes tener en cuenta los siguientes:

- Que exista.
- Que tenga cabecera BMP (el principio del archivo).
- Que el tamaño del archivo coincida con el tamaño guardado en la primera cabecera.

<b>COLEXIO</b> <b>VIVAS</b> S.L.	RAMA:	Computing	CICLO:	MAP		
	MÓDULO	Service and processes programming				
	PROTOCOLO:		AVAL:		DATA:	
	Autor	Francisco Bellas Aláez (Curro)				

**InfoBmp:** Devuelve un pequeño objeto o estructura con los campos:

- Ancho en píxeles.
- Alto en píxeles.
- Si es un BMP comprimido o no.
- Bits por pixel.

Haz un programa (en consola o gráfico) donde se pueda seleccionar un archivo de alguna forma, diga si es o no un BMP válido y en caso afirmativo muestra la información del mismo.

**(Avanzado)** Crea en modo gráfico un visor de BMP con lectura a bajo nivel. En principio para no complicarte hazlo con BMPs pequeños y usa como pixel una label de tamaño pequeña (5x5 por ejemplo) con el color que corresponda al pixel. También es recomendable que en la primera versión lo hagas con BMPs de 24bits ya que con menos tendrías que pelearte con paletas, lo que te complicaría el código.