**HolaMundo: Estructura del programa y salida por consola.**

Elabore un proyecto de consola que muestre el mensaje **“Hola Mundo,**

**¡ahora en C#!”**. Al principio del programa, agregar comentarios que indiquen quién es el autor del programa, y la fecha de elaboración del programa.

Entradas y salidas:

Hola Mundo, ahora en C#!

Using System;

Namespace HolaMundo

{

Class Program

{

Static void Main(string[] args)

{

Console.WriteLine(“Hola mundo, ahora en C#!”);

Console.ReadLine();

}

}

}

**Conversiones: Convert y análisis de tipos.**

Elabore un proyecto de consola llamado **Conversiones**, que declare una variable de tipo **string** con un valor de **“1234”**, y que muestre el data type de la variable: realizar la conversión del dato a **int**, y mostrar el nueva *data type*. Mostrar también el número que se convirtió.

Entradas y salidas: System.String System.Int32

El número es 1234

Using System;

Namespace Conversiones

{

Class Program

{

Static void Main(string[] args)

{

String numero = “1234”;

Console.WriteLine(numero.GetType().ToString());

Int intNumero= Convert.ToInt32(numero);

Console.WriteLine(intNumero.GetType().ToString());

Console.WriteLine(String.Format(“El número es {0}”,

intNumero));

Console.ReadLine();

}

}

}

**Aleatorio: Números aleatorios y conversiones.**

Elabore un proyecto de consola llamado **Aleatorio**, que declare una variable a nivel clase, de tipo **float**, asignándole un valor cualquiera, explícitamente **float**; en el *entry point*, declare una variable local de tipo **float**, que adquiera un valor aleatorio entre 1 y 10, y que muestre en consola el resultado de la suma de las dos variables, usando el mensaje **“La suma de x y y es z”**.

Entradas y salidas:

La suma de 10.5 y 5.4 es 15.9

Using System;

Namespace Aleatorio

{

Class Program

{

Static public float numero1=24.5F;

Static void Main (string[] args)

{

Float numero2=0.0F;

Random numAleatorio = new Random();

Numero2= (float)numAleatorio.Next(1,11);

Console.WriteLine(string.Format(

“La suma de {0} y {1} es {2}”, numero1, numero2, numero1 + numero2));

Console.ReadLine();

}

}

}

**Entrada: Revisando si un dato es de un tipo.**

Elabore un proyecto llamado **Entrada**, que declare una variable que reciba un **valor**; si el valor puede ser transformado en **integer**, mostrar la leyenda **“Dato entero: x. ¡Muy bien!”** o de lo contrario, mostrar **“Dato no es entero. Intentar nuevamente.”**

Ejecutar proporcionando un entero, un flotante, y una cadena.

Entradas y salidas:

**a)**

Escribe algo: hola

Dato no es entero. Intentar de nuevo.

**b)**

Escribe algo: 12.5

Dato no es entero. Intentar de nuevo.

**c)**

Escribe algo: 10

Dato entero 10. Muy bien!

Using System;

Namespace Entrada

{

Class Program

{

Static void Main(string[] args)

{

String valor;

Int receptora = 0;

Console.Write(“Escrible algo: “);

Valor= Console.ReadLine();

If (int.Tryparse(valor, out receptora))

{

Console.WriteLine(String.Format(“Dato entero {0}. Muy bien!”, receptora));

}

Else

{

Console.WriteLine(“Dato no es entero. Intentar de nuevo.”);

}

Console.WriteLine(“”);

Console.WriteLine(“Presiona INTRO para continuar”);

Console.ReadKey();

}

}

}

**Nombre: Métodos de tipo y concatenación**

Elabore un proyecto llamado **Nombre**, que pregunte dos datos: nombre, y apellido. Los debe transformar a mayúsculas, y mostrar en forma de nombre completo (concatenación). La concatenación debe ser eficiente, por lo que no se puede usar **+**.

Entradas y salidas:

Captura un nombre: Felipe Captura los apellidos: Ramírez

FELIPE RAMÍREZ

Using System;

Using System.Text;

Namespace Nombre

{

Class Program

{

Static void Main(string[] args)

{

String nombre;

String apellidos;

Console.Write(“Captura un nombre: “);

nombre = Console.ReadLine();

Console.Write(“Captura los apellidos: “);

apellidos = Console.ReadLine();

nombre=nombre.ToUpper();

apellidos=apellidos.ToUpper();

StringBuilder nombreCompleto= new Stringbuilder(nombre);

nombreCompleto.Append(“ “);

nombreCompleto.Append(apellidos);

Console.WriteLine(nombreCompleto);

Console.WriteLine(“”);

Console.WriteLine(“Presiona INTRO para continuar”);

Console.ReadKey();

}

}

}

**Tabla: Uso de ciclos finitos (for)**

Elabore un proyecto llamado **Tabla**, que pregunte un número entero del 1 al 9, y muestre la tabla de multiplicar del número proporcionado.

Entradas y salidas:

Dame un número del 1 al 9: 4

4 x 1 = 4

4 x 2 = 8

4 x 3 = 12

4 x 4 = 16

*… así hasta terminar*

Using System;

Namespace Tabla

{

Class Program

{

Static void Main(string[] args)

{

String \_numero;

Int numero;

Console.Write(“Dame un número del 1 al 9: “);

\_numero = Console.ReadLine();

Numero = Convert.ToInt32(\_numero);

For (int i=1; i<=10; i++)

{

Console.WriteLine(

String.Format(“{0} x {1} = {2}”,

Numero, i, Numero\*1));

}

Console.WriteLine(“”);

Console.WriteLine(“Presiona INTRO para continuar”);

Console.ReadKey();

}

}

}

**Tablas: Ciclo anidado**

Elabore un proyecto llamado **Tablas**, que elabore las tablas de multiplicar del 1 al 10. Cada tabla deberá tener un encabezado **“Tabla del x”**. Entre una tabla y otra, debe haber un salto de línea.

Entradas y salidas:

Tabla del 1

1 x 1 = 1

1 x 2 = 2

1 x 3 = 3

1 x 4 = 4

1 x 5 = 5

1 x 6 = 6

1 x 7 = 7

1 x 8 = 8

1 x 9 = 9

1 x 10 = 10

Tabla del 2

2 x 1 = 2

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

2 x 5 = 10

Using System;

Namespace Tablas

{

Class Program

{

Static void Main(string[] args)

{

For int(i=1; i=10; i++)

{

Console.WriteLine(“”);

Console.WriteLine(String.Format(“Tabla del {0}: “, i));

Console.WriteLine(“”);

For (int j=1; j<=10; j++)

{

Console.WriteLine(String.Format(“{0} x {1} = {2}”, i,j,i\*j));

}

}

Console.WriteLine(“”);

Console.WriteLine(“Presiona INTRO para continuar”);

Console.ReadKey();

}

}

}

**Compara: Condicional anidado**

Elabore un proyecto llamado **Compara**, que pregunte dos números, y que muestre cuál de los dos es mayor, el primero o el segundo. También debe reportar si son iguales. El mensaje debe decir:

**“Números proporcionados: x y y. El mayor es primero.”** (o el segundo, o son iguales, según sea el caso).

Entradas y salidas:

**a)**

Número 1:10

Número 2:10

Números proporcionados: 10 y 10. Los números son iguales.

**b)**

Número 1:10

Número 2:20

Números proporcionados: 10 y 20. El mayor es el segundo.

**b)**

Número 1:20

Número 2:10

Números proporcionados: 20 y 10. El mayor es el primero.

Using System;

Namespace Compara

{

Class Program

{

Static void Main(string[] args)

{

String \_numero1, \_numero2;

Int numero1, numero2;

Console.Write(“Número 1: “);

\_numero1 = Console.ReadLine();

Console.Write(“Número 2: “);

\_numero2= Console.ReadLine();

numero1= Convert.ToInt32(\_numero1);

numero2= Converto.ToInt32(\_numero2);

if (numero1 == numero2)

{

Console.WriteLine(string.Format(“Números proporcionados {0} y {1}. {2}”,

Numero1, numero2, “Los números son iguales.”));

}

Else

{

If (numero1 > numero2)

{

Console.WriteLine(String.Format(“Números proporcionados {0} y {1}. {2}”,

Numero1, numero2, “El mayor es el primero.”));

}

Else

{

Console.WriteLine(String.Format(“Números proporcionados {0} y {1}. {2}”,

Numero1, numero2, “El mayor es el segundo.”));

}

}

Console.WriteLine(“”);

Console.WriteLine(“Presiona INTRO para continuar”);

Console.ReadKey();

}

}

}

**Acumulado: Ciclo infinito y operadores incluyentes.**

Elabore un proyecto llamado **Acumulado**, que pregunte números enteros indefinidamente. Sólo debe permitir números enteros, y notifica si no es así. Cada número que pregunte, deberá acumularlo, mostrando **“Acumulado hasta el momento: x”**. El programa no deja de preguntar números y acumularlos, hasta que se deje vacía la entrada.

Entradas y salidas:

Capture los enteros a acumular. Dejar vacío y dar INTRO, para salir

Dame un número entero: hola

El dato proporcionado no es un número entero. Intenta de nuevo.

Dame un número entero: 10 Monto acumulado: 10

Dame un número entero: 20 Monto acumulado: 30

Dame un número entero: 40 Monto acumulado: 70

Dame un número entero:

Presiona INTRO para continuar

Using System;

Namespace Acumulado

{

Class Program

{

Static void Main(string[] args)

{

String \_numero;

Int numero;

Int acumulado = 0;

Console.WriteLine(“Capture los enteros a acumular.”);

Console.WriteLine(“dejar vacío y dar INTRO, para salir”);

Console.WriteLine(“”);

While (true)

{

Console.Write(“Dame un número entero: “);

\_numero= Console.ReadLine();

If (\_numero== “”)

{

Break;

}

Else

{

If (int.TryParse(\_numero, out numero))

{

Acumulado +=numero;

Console.WriteLine(String.Format(“Monto acumulado: {0}”, acumulado));

}

Else

{

Console.WriteLine(“El dato proporcionado no es un número entero.”);

Console.WriteLine(“Intenta de nuevo.”);

}

}

}

Console.WriteLine(“”);

Console.WriteLine(“Presiona INTRO para continuar”);

Console.ReadKey();

}

}

}

**Multiplo: Residuales, operadores lógicos y manejo de errores.**

Elabore un proyecto llamado **Multiplo**, que pregunte un número entero. Si el número es múltiplo de 3 y múltiplo de 5, o múltiplo de 7, muestra el mensaje **“Correcto”**, de lo contrario, **“Incorrecto”**. Tip: Si un número es múltiplo de otro, residual es cero.

Entradas y salidas:

**a)**

Dame un número entero: 15 Correcto.

**b)**

Dame un número entero: 14 Correcto.

**b)**

Dame un número entero: 10

Incorrecto.

Using System;

Namespace Multiplo

{

Class Program

{

Static void Main(String[] args)

{

String \_numero;

Int numero;

Bool esMultiplo3, esMultiplo5, esMultiplo7;

Try

{

Console.Write(“Dame un número entero: “);

\_numero= console.ReadLine();

Numero= Convert.ToInt32(\_numero);

esMultiplo3 = ((numero % 3) == 0);

esMultiplo5 = ((numero % 5) == 0);

esMultiplo7 = ((numero % 7) == 0);

if ((esMultiplo3 & esMultiplo5) | esMultiplo7)

{

Console.WriteLine(“Correcto.”);

}

Else

{

Console.WriteLine(“Incorrecto.”);

}

Catch (Exception e)

{

Console.WriteLine(“El dato proporcionado causa errores.”);

Console.WriteLine(e.Message);

}

Finally

{

Console.WriteLine(“”);

Console.WriteLine(“Presiona INTRO para continuar”);

Console.Readkey();

}

}

}

}

}