25-10-2019

Programación

**NOMBRE:** José Alejandro Ibarra Miranda

**MATERIA:** Programación

**MATRICULA:** 1944102

**GRUPO:** 32

**IMPARTE:** Felipe Ramírez

**Programa 1:** Hola mundo: estructura del programa y salida por consola

Elabore un proyecto de consola que muestre el mensaje “hola mundo, ¡ahora en c#!.

//autor: José Alejandro Ibarra Miranda

//fecha de creacion: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace holamundo

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("¡¡¡HOLA MUNDO, AHORA EN C#!!!");

Console.ReadKey();

}

}

}

**Programa 2:** 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 dará type. Mostrar el número que se convirtió.

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace conversiones

{

class Program

{

static void Main(string[] args)

{

string num = "1234";

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

int intnum = Convert.ToInt32(num);

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

Console.WriteLine(string.Format("el numero es {0}", intnum));

Console.ReadKey();

}

}

}

**Programa 3:** 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 2 variables, usando el mensaje “la suma de x y y es z”.

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Aleatorio

{

class Program

{

static public float NUM1 = 38.5f;

static void Main(string[] args)

{

//se hizo una declaración dentro de un método, y como resultado hace que la variable sea local.

float NUM2 = 0.0f;

//se da un valor aleatorio

Random numeroAlea = new Random();

NUM2 = (float)numeroAlea.Next(1, 11);

Console.WriteLine(string.Format("la suma de {0} y {1} es {2}", NUM1, NUM2, NUM1 + NUM2));

Console.ReadKey();

}

}

}

**Programa 4:** Entrada: revisando si un dato es 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.”

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Entrada

{

class Program

{

static void Main(string[] args)

{

//se declaró una variable para preguntar dicha información y la otra para recibir el valor entero equivalente, si llega a ser la conversión posible

string VALOR;

int R = 0;

Console.Write("Escribe: ");

VALOR = Console.ReadLine();

//Se checa bien si cuyo valor registrado puede convertirse a int

if (int.TryParse(VALOR, out R))

{

//Si la conversión es posible, el valor convertido se almacena en la variable int y se muestra

Console.WriteLine(string.Format("DATO ENTERO {0}, MUY BIEN!!", R));

}

else

{

Console.WriteLine("DATO NO ES ENTERO. INTENTAR DE NUEVO");

//si el dato no es entero se manda un mensaje de que la conversión no fue exitosa

}

Console.WriteLine("PRESIONE CUALQUIER TECLA PARA CONTINUAR...");

Console.ReadKey();

}

}

}

**Programa 5:** 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 deber ser eficiente, por lo que no se puede usar +.

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace nombre

{

class Program

{

static void Main(string[] args)

{

string Nom;

string ape;

Console.Write("escribe un nombre: ");

Nom = Console.ReadLine();

Console.Write("escribe los apellidos: ");

ape = Console.ReadLine();

//se le asigna a las variables su versión en Mayúsculas

Nom = Nom.ToUpper();

ape = ape.ToUpper();

//los datos se declaran en string por lo cual son inmutables, es decir que si se concatenan con +, realmente se que crean copias de lo que se suma, que de tal manera se consume mas memoria. El stringbuilder permite evitar esto. Se declara un objeto stringbuilder y se asigna una nueva instancia de la clase (new)

StringBuilder nombcomplet = new StringBuilder(Nom);

nombcomplet.Append("");

nombcomplet.Append(ape);

Console.WriteLine(nombcomplet);

Console.WriteLine("presiona cualquier tecla para continuar...");

Console.ReadKey();

}

}

}

**Programa 6:** Tabla: uso de ciclos finitos (for)

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

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace tabla

{

class Program

{

static void Main(string[] args)

{

string NUM;

int NUMERO;

//se pregunta el dato como string, y se convierte a su equivalente numérico.

Console.Write("ingresa un numero del 1 al 9: ");

NUM = Console.ReadLine();

NUMERO = Convert.ToInt32(NUM);

//se genera un ciclo mejor conocido de iteraciones

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

{

Console.WriteLine(string.Format("{0} x {1} = {2}", NUMERO, i, NUMERO \* i));

}

Console.WriteLine("presiona cualquier tecla para terminar...");

Console.ReadKey();

}

}

}

**Programa 7:** 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.

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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("");

//se realiza un for dentro de otro for, permite combinar sus variables de secuencia

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

{

Console.WriteLine(string.Format("{0} x {1} = {2}", i, j, i \* j));

}

}

Console.WriteLine("presiona cualquier tecla para terminar...");

Console.ReadKey();

}

}

}

**Programa 8:** 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).

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace compara

{

class Program

{

static void Main(string[] args)

{

//los datos se declaran como string, se convierten a int y se almacenan en variables de trabajo. Se declaran 2 variables

string NUM1, NUM2;

int NUMM1, NUMM2;

Console.Write("Numero 1: ");

NUM1 = Console.ReadLine();

Console.Write("Numero 2: ");

NUM2 = Console.ReadLine();

NUMM1 = Convert.ToInt32(NUM1);

NUMM2 = Convert.ToInt32(NUM2);

//un if dentro del otro, ambos con una salida por falso

if (NUMM1 == NUMM2)

{

Console.WriteLine(string.Format("Numeros proporcionados {0} y {1}. {2}", NUMM1 , NUMM2, "los numeros son iguales!"));

}

else

{

if (NUMM1 > NUMM2)

{

Console.WriteLine(string.Format("Numeros proporcionados {0} y {1}. {2}", NUMM1, NUMM2, "el Mayor es el primero."));

}

else

{

Console.WriteLine(string.Format("Numeros proporcionados {0} y {1}. {2}", NUMM1, NUMM2, "el Mayor es el segundo."));

}

}

Console.WriteLine("");

Console.WriteLine("Presiona cualquier tecla para terminar...");

Console.ReadKey();

}

}

}

**Programa 9:** Acumulado: ciclo infinito y operadores incluyentes

Elabore un proyecto llamado acumulado, que pregunte números enteros indefinidamente. Solo 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.

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace acumulado

{

class Program

{

static void Main(string[] args)

{

string NUM\_ERO;

int NUM;

int ACU = 0;

Console.WriteLine("INGRESE LOS ENTEROS A ACUMULAR: ");

Console.WriteLine("Dejar vacio y dar enter para salir...");

Console.WriteLine("");

//el ciclo while infinito, no concluye sino hasta que se ejecuta de manera explicita un break

while (true)

{

Console.Write("ingresa un numero entero: ");

NUM\_ERO = Console.ReadLine();

if (NUM\_ERO == "")

{

break;

}

else

{

if (int.TryParse(NUM\_ERO, out NUM))

{

ACU += NUM;

Console.WriteLine(string.Format("monto acumulado: {0}", ACU));

}

else

{

Console.WriteLine("el dato proporcionado no es un numero entero.");

Console.WriteLine("intenta de nuevo.");

}

}

}

Console.WriteLine("");

Console.WriteLine("presiona cualquier tecla para terminar...");

Console.ReadKey();

}

}

}

**Programa 10:** Múltiplo: residuales, operadores lógicos y manejo de errores.

Elabore un proyecto llamado múltiplo, que pregunte un numero 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.

//autor: José Alejandro Ibarra Miranda

//Fecha de creación: 25/septiembre/19

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Multiplo

{

class Program

{

static void Main(string[] args)

{

string NUMERO;

int Num;

bool esmultiplo3, esmultiplo5, esmultiplo7;

try

{

Console.Write("ingresa un numero entero: ");

NUMERO = Console.ReadLine();

Num = Convert.ToInt32(NUMERO);

//si un numero tiene un residual de cero con respecto a un numero, es que es un multiplo

esmultiplo3 = ((Num % 3) == 0);

esmultiplo5 = ((Num % 5) == 0);

esmultiplo7 = ((Num % 7) == 0);

//si es multiplo de 3 y de 5 al mismo tiempo, o si es multiplo de 7 es correcto.

if ((esmultiplo3 & esmultiplo5) | esmultiplo7)

{

Console.WriteLine("ES CORRECTO");

}

else

{

Console.WriteLine("ES INCORRECTO");

}

}

catch (Exception e)

{

Console.WriteLine("el dato proporcionado arroja errores");

Console.WriteLine(e.Message);

}

finally

{

Console.WriteLine("");

Console.WriteLine("PRESIONA CUALQUIER TECLA PARA TERMINAR...");

Console.ReadKey();

}

}

}

}