**UNIVERSIDAD TECNOLÓGICA DEL SUROESTE DE GUANAJUATO**

**Nombre y firma del Alumno:**

**David Mendoza Aguirre**

**Esmeralda Morales Conejo**

**Materia:**

**“Programación de Aplicaciones”**

**Carrera:**

**Tecnologías de la Información y Comunicación**

**Nombre del profesor(a):**

**MI. Ariana Gómez Contreras**

**Fecha:**

**Viernes 23 de agosto de 2019**

**Ejemplo básico:**

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**namespace ConsoleApplication1**

**{**

**public class ClaseBasica**

**{**

**public int i = 10;**

**}**

**class Program**

**{**

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

**{**

**object Objeto;**

**Objeto = new ClaseBasica();**

**ClaseBasica a;**

**a = (ClaseBasica)Objeto;**

**Console.WriteLine(a.i);**

**ClaseBasica b = new ClaseBasica();**

**Console.WriteLine(b.i);**

**Console.ReadLine();**

**}**

**}**

**}**

Comparar cómo hemos creado el objeto «**a**» y cómo el objeto «**b**«, el resultado es el mismo pero el objeto «**b**» fue creado directamente a partir de la clase «**ClaseBasica**» mientras que el objeto «**a**» fue creado a partir del objeto «**Objeto**«.

Veamos el ejemplo anterior ampliado:

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**namespace ConsoleApplication1**

**{**

**public class ClaseBasica**

**{**

**public int i = 10; //Campo**

**private int tantoporcien; //Campo**

**public int TantoPorCien //Propiedad**

**{**

**get { return tantoporcien; }**

**set {**

**if ((value &gt;= 0) &amp;&amp; (value &lt;= 100))**

**{**

**tantoporcien = value;**

**}**

**}**

**}**

**public void Resetea() //Método**

**{**

**tantoporcien = 0;**

**}**

**}**

**class Program**

**{**

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

**{**

**object Objeto;**

**Objeto = new ClaseBasica();**

**ClaseBasica a;**

**a = (ClaseBasica)Objeto;**

**a.i += a.i; // Expresion lambda. Equivalente a "a.i = a.i + a.i";**

**a.TantoPorCien=70;**

**Console.WriteLine(a.i);**

**Console.WriteLine(a.TantoPorCien+"%");**

**a.Resetea();**

**Console.WriteLine(a.TantoPorCien + "%");**

**ClaseBasica b = new ClaseBasica();**

**b.i =+ b.i; // ojo esto simplemente asigna el signo del entero es decir no hace nada en este caso;**

**b.TantoPorCien = 75;**

**b.TantoPorCien = 750;**

**Console.WriteLine(b.i);**

**Console.WriteLine(b.TantoPorCien + "%");**

**Console.ReadLine();**

**}**

**}**

**}**

Varias cosas a tener en cuenta, hemos ampliado la clase con un nuevo campo llamado «**tantoporcien**«, además como se trata de un campo que requiere un control sobre su valor ya que queremos que su valor esté comprendido entre 0 y 100, lo declararemos como **private** de forma que sólo pueda ser manipulado a partir de propiedades (**TantoPorCien**) o métodos (**Resetea**).

El resultado será:

**20**

**70%**

**0%**

**10**

**75%**

Podemos mejorar nuestra clase implementando un método que imprima directamente en pantalla «**Imprime**«:

**...**

**namespace ConsoleApplication1**

**{**

**public class ClaseBasica**

**{**

**...**

**public void Imprime()**

**{**

**Console.WriteLine(this.i.ToString() + " - " + this.tantoporcien.ToString()+"%");**

**}**

**}**

**class Program**

**{**

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

**{**

**object Objeto;**

**Objeto = new ClaseBasica();**

**ClaseBasica a;**

**a = (ClaseBasica)Objeto;**

**a.i += a.i; // Expresion lambda. Equivalente a "a.i = a.i + a.i";**

**a.TantoPorCien=70;**

**a.Imprime();**

**a.Resetea();**

**a.Imprime();**

**ClaseBasica b = new ClaseBasica();**

**b.i =+ b.i; // ojo esto simplemente asigna el signo del entero es decir no hace nada en este caso;**

**b.TantoPorCien = 75;**

**b.TantoPorCien = 750;**

**b.Imprime();**

**Console.ReadLine();**

**}**

**}**

**}**

Son una característica incluida a partir de la versión 3.0 de C# que hacen más concisa la declaración de una propiedad. Es posible aplicarla siempre que no exista «lógica» en los descriptores de acceso de la propiedad.

**public class Cliente {**

**private int idcliente;**

**public int IdCliente {**

**get {return idcliente;}**

**set {}**

**}**

**private string nombre;**

**public string Nombre {**

**get {return nombre;}**

**set {nombre = value;}**

**}**

**}**