**Módulo: Características del Lenguaje C#**

**Tutorial: Definición de Clases**

**Ejemplo 1**

1. Definición de clases. Crear las siguiente definición:

class MiPrimeraClase

{

}

1. Agregar un miembro de la clase - campo

class MiPrimeraClase

{

public string nombre = "Alejo";

}

1. Crear un objeto del tipo de la clase definida. Instanciacion de la clase.

static void Main(string[] args)

{

MiPrimeraClase objMiPrimeraClase = new MiPrimeraClase();

Console.WriteLine(objMiPrimeraClase.nombre);

Console.ReadLine();

1. Crear una propiedad de la clase:

//2. Propiedad

private int \_edad = 0;

public int edad

{

get

{

return edad;

}

set

{

edad = value;

}

}

1. Indicar la diferencia entre los iconos, cuando aparece el IntelliSense.
2. Uso del método Get {}

Console.WriteLine("Edad: " + objMiPrimeraClase.Edad);

1. Uso del método Set {}

objMiPrimeraClase.edad = 51;

1. Crear un método de la clase:

//3. Metodos

public void MiInfo()

{

Console.WriteLine(nombre + " tiene " + edad + " años.");

}

1. Crear un método con retorno de valor, con parámetros

public string MiInfo(string nombre, int edad)

{

return nombre + " tiene " + edad + " años.";

}

1. Uso de los campos, propiedades y métodos de la clase definida. Agregar el siguiente código

objMiPrimeraClase.Edad = 51;

Console.WriteLine("Nombre: " + objMiPrimeraClase.nombre);

Console.WriteLine("Edad: " + objMiPrimeraClase.Edad);

Console.ReadLine();

Console.WriteLine("Llama MyInfo");

objMiPrimeraClase.MiInfo();

Console.ReadLine();

Console.WriteLine("Llama MyInfo(nombre,edad)");

Console.WriteLine(objMiPrimeraClase.MiInfo("Franco",12));

Console.ReadLine();

**Ejemplo 2**

1. Definir propiedad de solo lectura o solo escritura. Solo se especifica el método Get para lectura.
2. Probar asignar una propiedad y ver que ocurre.
3. Lo misma para solo escritura.
4. Modificar el ámbito de los métodos Get o Set (indicarlo como privado).
5. Uso de la propiedad Indexador. Agregar la siguiente propiedad

public string this[int index]

{

get

{

}

set

{

}

}

1. Agregar la colección de elementos que define la propiedad y completar los métodos Get y Set

private string[] nombreDocentes = new string[] { "Diego", "Marcos", "Victor" };

public string this[int index]

{

get

{

return nombreDocentes[index];

}

set

{

nombreDocentes[index] = value;

}

}

1. Agregar en el objeto creado del tipo de clase definido el uso de la propiedad indexador. Para eso de crea el siguiente método en la clase.

public void listarDocentes()

{

foreach (string i in nombreDocentes)

{

Console.WriteLine(i);

}

}

1. Llamar al método desde el objeto instanciado.

objMiPrimeraClase.listarDocentes();

**Tutorial: Instanciación de Clases**

**Ejemplo 1**

1. Crear en la clase Programa08b, el siguiente campo y propiedad como miembros de la clase.

private string nombreValue = string.Empty;

public string Nombre

{

get

{

return nombreValue;

}

set

{

nombreValue = value;

}

}

1. Agregar el siguiente método que corresponde a una sobre carga del constructor de la clase

public Programa08b (string nuevoNombre)

{

this.nombreValue = nuevoNombre;

}

1. Instanciar una clase para ver los diferentes constructores.

static void Main(string[] args)

{

Program p = new Program("Alejo");

Console.WriteLine(p.Nombre);

Console.ReadLine();

}

1. Crear varios métodos sobrecargados haciendo uso del polimorfismo :

public void Saludar()

{

Console.WriteLine("Bienvenidos a Paraná");

}

public void Saludar(string slogan)

{

Console.WriteLine(slogan);

}

public void Saludar(string slogan, int temporada)

{

Console.WriteLine (slogan + " - " + temporada.ToString());

}

1. Con la clase creada llamar a los métodos sobrecargados.

//Ejemplo de Sobrecarga

p.Saludar();

p.Saludar("Ciudad Paisaje");

p.Saludar("Pase la mejor temporada", 2011);

Console.ReadLine();