

## Seminario de Lenguajes (.NET)

### Práctica 5

**Importante:** Los ejercicios 2 y 3 de esta práctica deben entregarse para su evaluación. La fecha de entrega se establecerá más adelante durante el transcurso de la cursada.

1) Modifique la definición de la clase Matriz realizada en la práctica 4. Elimine los métodos SetElemento, GetElemento, GetDiagonalPrincipal y GetDiagonalSecundaria. Defina las propiedades de sólo lectura DiagonalPrincipal y DiagonalSecundaria. También defina un indizador adecuado para leer y escribir los elementos de la matriz.

A modo de ejemplo observe la salida del siguiente fragmento de código:

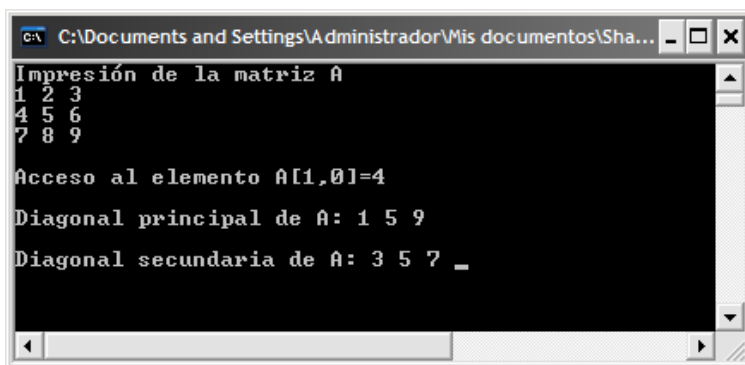
```
public static void Main(string[] args)
{
    Matriz A=new Matriz(3,3);
    for(int i=0;i<9;i++) A[i/3,i%3] = (i+1);
    Console.WriteLine("Impresión de la matriz A");
    A.imprimir();

    Console.Write("\nAcceso al elemento A[1,0]={0}",A[1,0]);

    Console.Write("\n\nDiagonal principal de A: ");
    foreach(double d in A.DiagonalPrincipal) Console.Write("{0} ",d);

    Console.Write("\n\nDiagonal secundaria de A: ");
    foreach(double d in A.DiagonalSecundaria) Console.Write("{0} ",d);

    Console.ReadKey(true);
}
```



2) Defina la clase Persona con las siguientes propiedades de lectura y escritura: Nombre, Sexo, DNI, FechaNacimiento. Además defina una propiedad de sólo lectura (calculada) Edad. Defina un indizador de lectura/escritura que permita acceder a las propiedades a través de un índice entero. Así, si p es un objeto Persona, con p[0] se accede al nombre, p[1] al sexo p[2] al DNI, p[3] a la fecha de nacimiento y p[4] a la edad. En caso de asignar p[4] simplemente el valor es descartado.

3) Dada la siguiente definición de clase:

```
class ListaDePersonas{
    private ArrayList lista=new ArrayList();
    public void Agregar(Persona p){
        lista.Add(p);
    }
}
```

Complétela agregando dos indizadores de sólo lectura

- a) Un índice entero que permite acceder a las personas de la lista por número de documento. Por ejemplo `p=lista[30456345]` devuelve el objeto `Persona` que tiene `DNI=30456345` o null en caso de que no exista en la lista.
- b) Un índice de tipo `char` que devuelva un arreglo de strings con todos los nombres de las personas de la lista que comienzan con el carácter índice. Por ejemplo `arreglo=lista['L']` devuelve todos los nombres de las personas que comienzan con L.

4) ¿Cuál es el error en el siguiente programa?

```
using System;
class Program{
    static void Main(){
        Auto a=new Auto();
        a.Marca = "Ford";
        Console.WriteLine(a.Marca);
        System.Console.ReadKey();
    }
}
class Auto{
    private string marca;
    public string Marca{
        set{
            Marca=value ;
        }
        get{
            return marca;
        }
    }
}
```

5) Observe la siguiente definición de clase. Verifique que línea (o líneas) de código provoca error de compilación. Analice cuándo es posible acceder a miembros estáticos y de instancia.

```
class ClaseA{
    char c; static string st;
    void metodo1(){
        st="string";
        c='A';
    }
    static void metodo2(){
        new ClaseA().c='a';
        st="st2";
        c='B';
    }
}
```

6) Codifique una clase `Registro` con una propiedad `idRegistro` de sólo lectura que pueda utilizarse como clave del mismo. Utilice el constructor para guardar en el objeto creado un número único que puede corresponderse con la cantidad de instancias creadas, de tal manera que el siguiente código:

```
...
System.Console.WriteLine(new Registro().IdRegistro);
System.Console.WriteLine(new Registro().IdRegistro);
System.Console.WriteLine(new Registro().IdRegistro);
...
```

produzca la siguiente salida:

```
0
1
2
```