

Anexo creación propiedades

Protección de datos y creación de propiedades

Con la clase cubo hemos visto cómo crear una clase sencilla que ya tiene funcionalidad.

Sin embargo, presenta un problema. El primero es que todos sus datos son públicos, lo que nos puede llevar a corrupción de información. Para proteger los datos tenemos que hacerlos privados y proveer una función de interfaz a aquellos a los que se necesita acceder por el exterior.

Para tener una comparación, trabajaremos sobre nuestra clase prisma. Lo primero que hacemos es cambiar el acceso de los datos a privado.

```
class prisma
{
    // Declaramos los datos
    private int ancho;
    private int alto;
    private int espesor;
    private int area;
    private int volumen;
}
```

Las propiedades son funciones de interfaz. Nos permiten acceder a los datos privados de una manera segura y controlada, pero van más allá de simples funciones ya que también nos brindan una forma de acceso intuitiva y sencilla.

La propiedad puede ser de varios tipos: lectura, escritura y la combinación de ambas (lectura-escritura). Una propiedad de lectura solamente nos permite leer el dato, pero no podemos agregarle información. Una propiedad que es de tipo escritura sólo nos permite colocar información en el dato, pero no podemos leerlo. La propiedad de lectura-escritura permite llevar a cabo ambas acciones.

Para lograr esto, la propiedad tendrá dos métodos. El método relacionado con la lectura se conoce como get y el relacionado con la escritura es set. Dependiendo de cuál método coloquemos en la propiedad, será su tipo.

La propiedad de éste tiene la siguiente forma de declaración:

```
public tipo nombre {
    get
    {
        ...
        ...
        return x;
    }
    set
    {
        ...
        ...
        x=value;
    }
}
```

Las propiedades son públicas para poder llamarlas desde el exterior de la clase. El tipo está referenciado al tipo del valor que leerá o colocará, ya sea entero, flotante, doble, etcétera. En su interior tenemos get, donde colocamos el código para sacar un valor de la clase por medio de return, y a set, donde ponemos el código necesario para introducir un valor en la clase.

Empecemos por crear propiedades para la clase prisma. Lo primero que tenemos que preguntarnos es a qué datos se necesita acceder por el exterior y qué tipo de acceso requieren.

Podemos ver que los datos ancho, alto y espesor necesitarán de escritura, pero también de lectura. Esto es en caso de que necesitemos saber las dimensiones.

A sus propiedades las llamaremos: Ancho, Alto y Espesor.

Los otros datos que necesitan tener una propiedad son area y volumen, pero en este caso solamente necesitamos leerlos. No tiene sentido escribir sobre esos datos, ya que la clase calculará sus propios valores.

Ahora que ya sabemos cuáles son las propiedades necesarias podemos decidir si es necesario algún tipo de validación. Sabemos que no podemos tener prismas con cualquiera de sus lados con valor de 0 o negativos y ése es un buen punto para validar.

Si el usuario diera un valor incorrecto, entonces colocaremos por default el valor 1. Hay que recordar que esto lo hacemos como ejemplo y cada aplicación puede tener sus propias reglas de validación para la información.

Cuando usemos el método set tendremos una variable previamente definida por el lenguaje que se llama value. Esta variable representa el valor que el usuario asigna y podemos usarlo en la lógica que necesitemos. Nuestras propiedades de lectura y escritura quedan de la siguiente forma:

// Definimos las propiedades

```
public int Ancho
{
    get
    {
        return ancho;
    }
    set
    {
        if (value <= 0)
            ancho = 1;
        else
            ancho = value;
    }
}

public int Alto
{
    get
    {
        return alto;
    }
    set
    {
        if (value <= 0)
            alto = 1;
        else
            alto = value;
    }
}

public int Espesor
{
    get
    {
        return espesor;
    }
    set
    {
        if (value <= 0)
            espesor = 1;
        else
            espesor = value;
    }
}
```

Ahora crearemos las propiedades de sólo lectura para el área y el volumen:

```
public int Area
{
    get
    {
        return area;
    }
}
public int Volumen
{
    get
    {
        return volumen;
    }
}
```

Como estas propiedades son de sólo lectura únicamente llevan el método `get`.

Cómo acceder a las propiedades

Acceder a las propiedades es muy sencillo ya que únicamente necesitamos colocar el objeto con el que queremos trabajar seguido del operador `.` y el nombre de la propiedad.

La asignación se lleva a cabo por medio del operador `=`.

Por ejemplo, si deseamos indicar que el ancho tiene 5 unidades, hacemos lo siguiente:

```
miPrisma.Ancho=5;
```

Y si deseamos imprimir el valor de la propiedad `Volumen`:

```
Console.WriteLine("El volumen es {0}",Volumen);
```



EL USO DE LAS PROPIEDADES

En ocasiones, dentro de los métodos `get` y `set` de la propiedad, tendremos un **return** o una asignación. Aunque esto es correcto, no debemos olvidar que en las propiedades podemos colocar cualquier lógica válida de C# que nos permita validar la información que sale o entra. Esto hará que nuestro código sea más seguro y evitaremos problemas con nuestra información.