

Introducción a C# .NET

Módulo 5

Enumeraciones

Enumeraciones

Aparte de las estructuras, existe otro tipo de datos que puede ser definido por el usuario desarrollador (**UDT**). Estos tipos se conocen como **enumeraciones**.

Las enumeraciones son un grupo de valores constantes relacionados a un posible valor que puede tener una variable/campo. Cada una de esas constantes tendrá un nombre, como así también la agrupación de esas constantes, es decir, la propia enumeración.

Por ejemplo, en una aplicación se deben realizar clasificaciones por tipo de cuenta bancaria: Caja de Ahorro Pesos, Caja de Ahorro Dólares, Cuenta Corriente, Cuenta Sueldo, etc.

Una manera sencilla de manipular estos valores, desde el código, podría ser identificar mediante números cada tipo de cuenta. De esta manera, cuando en un procedimiento se tenga que conocer la selección de tipo de cuenta que realizó el usuario de la aplicación, se utilizará el número identificador en lugar de la cadena de caracteres descriptiva del tipo de cuenta.

Declaración y asignación de valores en una enumeración

Una **enumeración** se declara de esta manera:

```
enum nombre {valor1, valor2, ..., valorn};
```

Se usa la palabra clave **enum** seguida del nombre de la nueva enumeración. Luego, entre **{ }** se debe codificar la lista de constantes agrupadas que conformarán los posibles valores. Esta lista de valores deberá ser finita y discreta.

El ejemplo de los tipos de cuentas bancarias continúa con la declaración de la enumeración:

```
enum tipo_cuenta {  
    CajaAhorroPesos,  
    CajaAhorroDolares,  
    CuentaCorriente,  
    CuentaSueldo};
```



Los valores de las enumeraciones

Cuando se declara una enumeración, se ingresa el listado de todos los valores posibles.

Internamente, a cada valor ingresado de la lista se le asigna un número de identificación. El primer elemento tendrá el valor 0, el segundo 1 y así sucesivamente. Sin embargo, también es posible personalizar el valor para el primer elemento de la enumeración y para cada uno de los elementos restantes.



Para utilizar una enumeración definida en la aplicación, debemos declarar una variable y asignarle el mismo tipo de dato que la enumeración. La declaración y asignación de la variable la hacemos de la forma tradicional.

```
tipo_cuenta cuenta;
```

Una vez creada, la forma de asignar un valor es simple, ya que cuando se escriba el operador de asignación `=`, el editor de código desplegará una lista con los posibles valores que admitirá la variable.

```
cuenta = tipo_cuenta.CajaAhorroPesos;
```

La asignación del valor a la variable se puede realizar en cualquier momento, durante la ejecución de la aplicación, luego de su declaración, o en el mismo momento de la declaración.

```
tipo_cuenta cuenta = tipo_cuenta.CuentaCorriente;
```



Visualizar el valor de una enumeración

Para mostrar el valor de una variable de tipo enumeración, se utiliza el método **ToString()**. Este método mostrará una cadena con el valor correspondiente, y se podrá utilizar dentro de la lógica de la aplicación para desplegar valores.

El método **ToString()** regresa una cadena y esa cadena mostrará el valor correspondiente. Por ejemplo, si en la enumeración listamos un valor como **CajaAhorroPesos**, entonces la cadena regresada será "CajaAhorroPesos".

```
String mensaje = "La cuenta es de tipo";  
... ..  
mensaje = mensaje + cuenta.ToString();  
... ..  
Console.WriteLine("La cuenta es de tipo {0}", cuenta);
```

Iniciar la enumeración con un valor definido

Tal como mencionamos anteriormente, los valores numéricos se pueden personalizar:

```
enum tipo_cuenta{  
    CajaAhorroPesos = 10,  
    CajaAhorroDolares,  
    CuentaCorriente,  
    CuentaSueldo};
```

De esta forma, CajaAhorroPesos tendrá un valor de 10, CajaAhorroDolares de 11 y así sucesivamente.



Ejemplo de aplicación

Nuestra aplicación puede quedar de la siguiente manera:

```
// Declaramos las enumeraciones
enum tipo_cuenta{ CajaAhorro, CuentaCorriente, CuentaSueldo };
enum moneda{ Pesos = 1, Dolares, Euros, Reales };

static void Main(string[] args)
{
    int numero = 0;
    // Creamos una variable de tipo cuenta
    tipo_cuenta cuenta;

    // Asignamos un valor
    cuenta = tipo_cuenta.CuentaCorriente;

    // Pasamos de enumeración a entero
    numero = (int)cuenta;
```

...

```
// Mostramos la información
Console.WriteLine($"La cuenta es una {cuenta} con valor {numero}");

// Creamos una variable de tipo moneda
moneda moneda_cuenta = moneda.Pesos;

// Mostramos la información
Console.WriteLine($"La cuenta es una {cuenta} en {moneda_cuenta} con valor {numero}");

Console.ReadKey();
}
```

```
La cuenta es una CuentaCorriente con valor 1
La cuenta es una CuentaCorriente en Pesos con valor 1
```



**¡Sigamos
trabajando!**