MÓDULOS Y PROCEDIMIENTOS

Cuando se desarrollan programas robustos y complejos es probable que existan muchos procesos comunes. En estos casos es necesario pensar en subprogramas que realicen una tarea específica denominados procedimientos o funciones. Y para los casos que se necesite compartir variables globales y procedimientos en varios formularios se trabaja con módulos.

Módulos

Es un contenedor que se visualiza de forma independiente en el explorador de soluciones y solamente se le puede incluir código y no tendrá interfaz gráfica. Puede contener variables globales o públicas, procedimientos sub y function y estos pueden ser compartidos por varios formularios.

```
1   Module ModuloPrueba
2       'Variables globales o públicas.
3       'Procedimientos Sub y Function.
4   End Module
5
6
```

Para crear un módulo dentro de un proyecto o aplicación se puede realizar de varias maneras:

♣ Desde el menú Proyecto.

♣ Desde la barra de herramientas.

♣ Desde el explorador de soluciones.

Para declarar una variable pública en un módulo es necesario escribir al inicio del módulo la palabra public seguido del nombre de la variable y el tipo de dato. De esta manera se podrá leer, modificar o visualizar la misma.

Procedimientos y funciones

En VB.NET existen dos tipos de procedimiento:

♣ Function: pueden ser invocados por su nombre desde otros procedimientos, pueden recibir argumentos y siempre devuelven un valor con el nombre de la función.

Sintaxis:

[Modificador de acceso] Function nombre_función(argumentos) As tipo

Instrucciones

Nombre_función = valor a retornar

End Function

♣ Sub: también pueden ser invocados por su nombre desde otros procedimientos, pueden recibir argumentos, pero no devuelven valores.

Sintaxis:

[Modificador de acceso] Sub nombre_sub(argumentos)

Instrucciones

End Sub

Se recomienda definirlos dentro de un módulo para que todos los elementos de un proyecto puedan utilizarlos.

Función fecha y hora

El objeto Date permite la manipulación de la fecha y hora. Dentro de las funciones básicas se pueden encontrar: Now: muestra en formato numérico la fecha y la hora del sistema: 03/11/2016 01:50:16 y Today: muestra en formato numérico la fecha del sistema: 03/11/2016. Se pueden manejar diferentes formatos de presentación de fecha y hora utilizando la función Format.

Para formatos de fechas:

Carácter	Descripción
(/)	Separador de fecha.
D	Muestra una fecha con el formato de fecha larga. Por ejemplo: martes, 31 de agosto de 2010.
d	Muestra una fecha con el formato de fecha corta. Por ejemplo: 31/10/2010.
dd	Muestra el día como un número con cero a la izquierda (por ejemplo: 01).
ddd	Muestra el día de forma abreviada (por ejemplo: mar).
dddd	Muestra el día de forma completa (por ejemplo: martes).
M	Muestra el día y el mes de una fecha (por ejemplo: 31 agosto).
MM	Muestra el mes como un número con cero a la izquierda (por ejemplo, 09).
MMM	Muestra el mes en forma abreviada (por ejemplo, ago).
MMMM	Muestra el mes en forma completa (por ejemplo: agosto).
y	Muestra el mes y el año (por ejemplo: agosto de 2010).
yy	Muestra el año en formato numérico de dos dígitos.
yyy	Muestra el año en formato numérico de cuatro dígitos.
F,f	Muestra la fecha larga y la hora corta. Por ejemplo: martes, 31 de agosto de 2010 11:07.

Para formatos de hora:

Carácter	Descripción
(:)	Separador de hora.
hh	Muestra la hora como un número con ceros a la izquierda y en formato de 12 horas.
HH	Muestra la hora como un número con ceros a la izquierda y en formato de 24 horas.
mm	Muestra los minutos como un número con ceros a la izquierda.
ss	Muestra los segundos como un número con ceros a la izquierda.
t	Muestra la hora y los minutos en formato de 24 horas.

Ejemplos de uso:

```
4
5       Format(Date.Now, "d/MM/y")
6       Format(Date.Now, "D")
7       Format(Date.Now, "d-MMMM-YYY")
8       Format(Date.Now, "h:m:s")
9       Format(Date.Now, "HH")
10      Format(Date.Now, "HH:mm")
11
```

Funciones matemáticas

La clase Math de VB.NET contiene una serie de funciones trigonométricas, logarítmicas y otras funciones matemáticas útiles para realizar cálculos aritméticos

Método de Visual Basic .NET	Descripción
Math.Ceiling(double)	Devuelve el número entero más pequeño mayor o igual que el número especificado. Math.ceiling(1.6)=2 Math.ceiling(0.8)=1
Math.Floor(double)	Devuelve el número entero más grande menor o igual que el número especificado. Math.Floor(1.6)=1 Math.Floor(0.8)=0
Math.sqrt(n)	Devuelve la raíz cuadrada de un número.
Math.pow(n,p)	Devuelve un número especificado elevado a la potencia especificada.
Math.Abs(n)	Sobrecargado. Devuelve el valor absoluto de un número especificado.
Math.sin(n)	Devuelve el seno del ángulo especificado.
Math.cos(n)	Devuelve el coseno del ángulo especificado.
Math.tan(n)	Devuelve la tangente del ángulo especificado.
Math.Max(n1,n2)	Devuelve el mayor de dos números.
Math.min(n1,n2)	Devuelve el menor de dos números.
Math.BigMul(n1,n2)	Calcula el producto de dos números.
Math.round(double)	Devuelve el número más próximo al valor especificado. Math.round(10.5)=10 Math.round(10.51)=11

Ejemplos de uso:

```
13          Math.Floor(8.5)
14          Math.Round(10.5)
15          Math.Sqrt(3)
16          Math.Max(7, 5)
17          Math.Min(7, 5)
```

Funciones para cadenas.

Chars(n): permite obtener un carácter específico de una cadena de caracteres.

Length: permite obtener la longitud de una cadena de caracteres.

Concat(texto1, texto2,…., texton): permite unir dos o más cadenas de caracteres.

ToUpper: convierte una cadena de caracteres de minúscula a mayúscula.

ToLower: convierte una cadena de caracteres de mayúsculas a minúsculas.

Remove(posición inicial, numero de caracteres): permite eliminar una cantidad determinada de caracteres en una posición específica de una cadena de caracteres.

Insert(posición de inserción, “cadena de caracteres”): permite insertar una cadena cantidad determinada de caracteres en una posición específica de una cadena de caracteres.
SubString(posición inicial, número de caracteres): permite obtener una subcadena de una cadena de caracteres.

Replace(cadena de caracteres): devuelve una cadena de caracteres invertida según el orden de los caracteres de la cadena especificada.

Len(cadena de caracteres): devuelve un entero que contiene el número de caracteres de una cadena de caracteres.

ToCharArray: convierte una cadena de caracteres en un arregla de caracteres.

Split(cadena de caracteres, delimitador): devuelve una arreglo unidimensional que contiene un número específico de subcadenas.

Trim(cadena): elimina los espacios de ambos lados de una cadena de caracteres.
String.Copy(cadena): copia una cadena de caracteres.

IndexOf(carácter a buscar): devuelve la posición de un carácter específico.