

Práctica : Creación y uso de tipos

Objetivos

Al final de esta práctica, usted será capaz de:

- Crear nuevos tipos de datos.
- Definir y utilizar variables.

Requisitos previos

Antes de realizar la práctica debe estar familiarizado con los siguientes temas:

- El sistema de tipos comunes (Common Type System, CTS).
- Variables de tipo valor en C#.

Sumario

En el Ejercicio 1 escribirá un programa que crea un tipo **enum** simple y a continuación asigna e imprime los valores utilizando la instrucción **Console.WriteLine**.

En el Ejercicio 2 escribirá un programa que utiliza en una **struct** el tipo **enum** declarado en el Ejercicio 1.

Si le queda tiempo, también añadirá funcionalidad de entrada/salida al programa escrito en el Ejercicio 2.

Archivos de inicio y solución

Existen archivos de inicio y solución para esta práctica. Los archivos de inicio están en la carpeta Starter, mientras que los archivos de solución se encuentran en Solution. Estas dos carpetas se encuentran dentro del archivo lab03.zip

Ejercicio 1

Creación de un tipo enum

En este ejercicio creará un tipo enumerado para representar distintos tipos de cuentas bancarias (cuentas corrientes y de ahorro). Utilizará este tipo **enum** para crear dos variables a las que asignará los valores Corriente y Ahorro. Finalmente empleará la función **System.Console.WriteLine** para imprimir los valores de las variables.

Cómo crear un tipo enum

1. Abra el proyecto BankAccount.sln en la carpeta Starter\BankAccount dentro del fichero lab03.zip.
2. Abra el archivo Enum.cs y añada un **enum** llamado **TipoCuenta** antes de la definición de clase:

```
public enum TipoCuenta { Corriente, Ahorro }
```

Este **enum** contendrá los tipos Corriente y Ahorro.

3. Declare en **Main** dos variables de tipo **TipoCuenta**:

```
TipoCuenta goldCuenta;  
TipoCuenta platinumCuenta;
```

4. Asigne a la primera variable el valor Corriente y a la segunda el valor Ahorro:

```
goldCuenta = TipoCuenta.Corriente;  
platinumCuenta = TipoCuenta.Ahorro;
```

5. Añada dos instrucciones **Console.WriteLine** que impriman el valor de cada variable:

```
Console.WriteLine("El tipo de cuenta del cliente es {0}"  
    ↵, goldCuenta);  
Console.WriteLine("El tipo de cuenta del cliente es {0}"  
    ↵, platinumCuenta);
```

6. Compile y ejecute el programa.

Ejercicio 2

Creación y uso de un tipo struct

En este ejercicio definirá una **struct** que se puede usar para representar una cuenta bancaria. Empleará variables para el número de cuenta (**long**), el saldo de la cuenta (**decimal**) y el tipo de cuenta (el **enum** creado en el Ejercicio 1). Creará una variable de tipo **struct**, le asignará algunos datos de ejemplo e imprimirá el resultado.

Cómo crear un tipo struct

1. Abra el proyecto StructType.sln en la carpeta Starter\StructType dentro del fichero lab03.zip.
2. Abra el archivo Struct.cs y añada una **public struct** llamada **CuentaBancaria** que contenga los siguientes campos.

Tipo	Variable
public long	<i>cuentaNum</i>
public decimal	<i>cuentaSaldo</i>
public TipoCuenta	<i>cuentaTipo</i>

3. Declare en **Main** una variable *goldCuenta* de tipo **TipoCuenta**.
`CuentaBancaria goldCuenta;`
4. Asigne valores a los campos *cuentaTipo*, *cuentaSaldo* y *cuentaNum* de la variable *goldCuenta*.
`goldCuenta.cuentaTipo = TipoCuenta.Corriente;
goldCuenta.CuentaSaldo = (decimal)3200.00;
goldCuenta.cuentaNum = 123;`
5. Añada instrucciones **Console.WriteLine** que impriman el valor de cada elemento en la variable struct.
`Console.WriteLine("Número de cuenta {0}",
goldCuenta.cuentaNum);
Console.WriteLine("Tipo de cuenta {0}",
goldCuenta.cuentaTipo);
Console.WriteLine("Saldo de la cuenta ${0}",
goldCuenta.CuentaSaldo);`
6. Compile y ejecute el programa.

Si el tiempo lo permite

Adición de funcionalidad de entrada/salida

En este ejercicio tendrá que modificar el código escrito en el Ejercicio 2. En lugar de usar el número de cuenta 123, pedirá que sea el usuario quien lo introduzca el número y usará ese número para imprimir el resumen de la cuenta.

Cómo añadir funcionalidad de entrada/salida

1. Abra el proyecto StructType.sln en la carpeta Starter\Optional dentro del fichero lab03.zip.
2. Abra el archivo Struct.cs y cambie la línea siguiente:
`goldCuenta.cuentaNum = 123; //borre esta línea y escriba el siguiente código`

por una instrucción `Console.Write` que pregunte el número de cuenta al usuario:

```
Console.Write("Introduzca el número de cuenta: ");
```

3. Utilice una instrucción **Console.ReadLine** para leer el número de cuenta y asigne este valor a `goldCuenta.cuentaNum`.
`goldCuenta.cuentaNum = long.Parse(Console.ReadLine());`

Nota Hay que utilizar el método **long.Parse** para convertir la cadena leída por la instrucción **Console.ReadLine** en un valor decimal antes de asignarlo a `goldCuenta.cuentaNum`.

4. Compile y ejecute el programa. Escriba un número de cuenta cuando se lo pida el programa.