Práctica.B: Conversión de datos

Objetivos

Al final de esta práctica, usted será capaz de:

- Convertir valores de un tipo referencia a otro.
- Probar si una variable referencia es compatible con una interfaz determinada.

Requisitos previos

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

- Conceptos de programación orientada a objetos
- Creación de clases
- Definición de métodos

Ejercicio 1
Prueba de la implementación de una interfaz

En este ejercicio añadirá un método estático llamado **IsItFormattable** a la clase **Utils** creada en la Práctica 5. Si no terminó esa práctica, puede conseguir una copia de la clase en la carpeta Starter dentro del fichero lab08.zip.

El método **IsItFormattable** recibe un parámetro de tipo **object** y prueba si ese parámetro implementa la interfaz **System.IFormattable**. El método devuelve **true** si el objeto tiene esta interfaz y **false** en caso contrario.

Una clase implementa la interfaz **System.IFormattable** para devolver una representación en cadena de caracteres de una instancia de esa clase. Los tipos básicos, como **int** y **ulong**, implementan esta interfaz (después de que se aplique boxing al valor), pero no así muchos tipos referencia, como por ejemplo **string**. Los tipos definidos por el usuario pueden implementar la interfaz si así lo quiere el desarrollador. Para más información sobre esta interfaz, consulte la documentación de ayuda del SDK de .NET Framework.

Escribirá código de prueba que llame al método **Utils**. **IsItFormattable** con argumentos de distintos tipos y muestre los resultados en la pantalla.

Cómo crear el método IsItFormattable

1. Abra el proyecto InterfaceTest.sln en la carpeta Starter\InterfaceTest dentro del fichero lab08.zip.
2. Edite la clase **Utils** de la siguiente manera:
   a. Cree un método público estático llamado **IsItFormattable** en la clase **Utils**.
   b. Este método recibe un parámetro llamado *x* de tipo **object**, que se pasa por valor. El método devuelve un **bool**.
   c. Use el operador **is** para determinar si el objeto pasado es compatible con la interfaz **System.IFormattable**. Devolverá **true** si es así, y **false** en caso contrario.

El método finalizado debería ser como éste:

```
using System;

...

class Utils
{
        public static bool IsItFormattable(object x)
        {
                // Usa el operador is para probar si el
                // objeto tiene la interfaz IFormattable

                if (x is IFormattable)
                        return true;
                else
                        return false;

        }
}
```

Cómo probar el método IsItFormattable

1. Edite la clase **Test** del archivo.
2. En el método **Main**, declare e inicialice variables de tipos **int**, **ulong** y **string**.
3. Pase cada variable a **Utils. IsItFormattable()** e imprima el resultado de cada llamada.
4. La clase **Test** podría ser así:

```
using System;
public class Test
{
    static void Main( )
    {
         int   i = 0;
         ulong ul = 0;
         string s = "Test";

         Console.WriteLine("int: {0}",
    ➥          Utils.IsItFormattable(i));
         Console.WriteLine("ulong: {0}",
    ➥          Utils.IsItFormattable(ul));
         Console.WriteLine("String: {0}",
    ➥          Utils.IsItFormattable(s));
    }
}
```

5. Compile y pruebe el código. Debería obtener **true** para los valores **int** y **ulong**, y **false** para el valor **string**.

Ejercicio 2
Uso de interfaces

En este ejercicio escribirá un método **Display** que usará el operador **as** para determinar si el objeto pasado como parámetro es compatible con una interfaz definida por el usuario llamada **IPrintable**, y en caso afirmativo llamará a un método de esa interfaz.

Cómo crear el método Display

1. Abra el proyecto TestDisplay.sln en la carpeta Starter\TestDisplay dentro del fichero lab08.zip.

   El código de inicio incluye la definición de una interfaz llamada **IPrintable**, que contiene un método llamado **Print**. Una clase que implemente esta interfaz podrá usar el método **Print** para mostrar en la consola los valores contenidos en el objeto. En los archivos del código de inicio también está definida una clase llamada **Coordinate**, que implementa la interfaz **IPrintable**.

   Un objeto **Coordinate** contiene un par de números que pueden definir una posición en dos dimensiones. No es necesario que sepa cómo funciona la clase **Coordinate** (aunque puede examinarla), sino sólo que implementa la interfaz **IPrintable** y que puede usar el método **Print** para mostrar sus contenidos.

2. Edite la clase **Utils** de la siguiente manera:

   a. Añada un método **void** público estático llamado **Display** en la clase **Utils**. Este método recibirá un parámetro (un **object** pasado por valor) llamado *item*.

   b. En **Display**, declare una variable de interfaz llamada *ip* de tipo **IPrintable**.

   c. Convierta la referencia en el parámetro *item* en una referencia a la interfaz **IPrintable** que utilice el operador **as**. Asigne el resultado a *ip*.

   d. Si el valor de *ip* es distinto de **null**, use la interfaz **IPrintable** para llamar a **Print**. Si es **null**, el objeto no es compatible con la interfaz. En este último caso, utilice **Console.WriteLine** para mostrar los resultados del método **ToString** en el parámetro.

   El método finalizado debería ser como éste:

```
public static void Display(object item)
{
    IPrintable ip;

    ip = (item as IPrintable);

    if (ip != null)
            ip.Print( );
    else
            Console.WriteLine(item.ToString( ));
}
```

Cómo probar el método Display

1. Dentro del método **Main** de la clase **Test**, cree una variable de tipo **int**, otra de tipo **string** y otra de tipo **Coordinate**. Para inicializar la variable **Coordinate** puede usar el siguiente constructor de dos parámetros:

   ```
   Coordinate c = new Coordinate(21.0, 68.0);
   ```

2. Pase cada una de estas res variables a **Utils.Display** para ver el resultado.
3. El código debería ser el siguiente:

   ```
   public class Test
   {
       static void Main( )
       {
           int    num = 65;
           string msg = "A String";
           Coordinate c = new Coordinate(21.0,68.0);

           Utils.Display(num);
           Utils.Display(msg);
           Utils.Display(c);
       }
   }
   ```

4. Compile y pruebe la aplicación.