

Pasar por referencia frente a Pasar por valor

De forma predeterminada, cuando un tipo de valor se pasa a un método, se pasa una copia en lugar del propio objeto. Por lo tanto, los cambios realizados en el argumento no tienen ningún efecto en la copia original del método de llamada. Puede pasar un tipo de valor por referencia mediante la palabra clave `ref`.

Cuando se pasa un objeto de un tipo de referencia a un método, se pasa una referencia al objeto. Es decir, el método no recibe el objeto concreto, recibe un argumento que indica la ubicación del objeto. Si cambia un miembro del objeto mediante esta referencia, el cambio se refleja en el argumento del método de llamada, incluso si pasa el objeto por valor.

Crea un tipo de referencia mediante la palabra clave `class`, como se muestra en el siguiente ejemplo.

```
public class SampleRefType
{
    public int value;
}
```

Ahora, si se pasa un objeto basado en este tipo a un método, también se pasa una referencia al objeto. En el ejemplo siguiente se pasa un objeto de tipo `SampleRefType` al método `ModifyObject`.

```
public static void TestRefType()
{
    SampleRefType rt = new SampleRefType();
    rt.value = 44;
    ModifyObject(rt);
    Console.WriteLine(rt.value);
}
static void ModifyObject(SampleRefType obj)
{
    obj.value = 33;
}
```

Fundamentalmente, el ejemplo hace lo mismo que el ejemplo anterior en el que se pasa un argumento por valor a un método. Pero, debido a que se utiliza un tipo de referencia, el resultado es diferente. La modificación que se lleva a cabo en `ModifyObject` al campo `value` del parámetro, `obj`, también cambia el campo `value` del argumento, `rt`, en el método `TestRefType`. El método `TestRefType` muestra 33 como salida.

