

# Clases

## Tipos de referencia

Un tipo que se define como una clase, es un tipo de referencia. Al declarar una variable de un tipo de referencia en tiempo de ejecución, esta contendrá el valor null hasta que se cree expresamente una instancia de la clase mediante el operador new o se le asigne un objeto de un tipo compatible que se ha creado en otro lugar, tal y como se muestra en el ejemplo siguiente:

```
//Declaring an object of type MyClass.  
MyClass mc = new MyClass();  
  
//Declaring another object of the same type, assigning it the value of the  
first object.  
MyClass mc2 = mc;
```

Cuando se crea el objeto, se asigna suficiente memoria en el montón administrado para ese objeto específico y la variable solo contiene una referencia a la ubicación de dicho objeto. Los tipos del montón administrado producen sobrecarga cuando se asignan y cuando los reclama la función de administración de memoria automática de CLR, conocida como *recolección de elementos no utilizados*. En cambio, la recolección de elementos no utilizados también está muy optimizada y no crea problemas de rendimiento en la mayoría de los escenarios.

## Declarar clases

Las clases se declaran mediante la palabra clave class seguida por un identificador único, como se muestra en el siguiente ejemplo:

```
//[access modifier] - [class] - [identifier]  
public class Customer  
{  
    // Fields, properties, methods and events go here...  
}
```

La palabra clave class va precedida del nivel de acceso. Como en este caso se usa **public**, cualquier usuario puede crear instancias de esta clase. El nombre de la clase sigue a la palabra clave class. El nombre de la clase debe ser un nombre de identificador de C# válido. El resto de la definición es el cuerpo de la clase, donde se definen los datos y el comportamiento. Los campos, las propiedades, los métodos y los eventos de una clase se denominan de manera colectiva *miembros de clase*.

# Creación de objetos

Aunque a veces se usan indistintamente, una clase y un objeto son cosas diferentes. Una clase define un tipo de objeto, pero no es un objeto en sí. Un objeto es una entidad concreta basada en una clase y, a veces, se conoce como una instancia de una clase.

Los objetos se pueden crear usando la palabra clave [new](#), seguida del nombre de la clase en la que se basará el objeto, como en este ejemplo:

```
Customer object1 = new Customer();
```

Cuando se crea una instancia de una clase, se vuelve a pasar al programador una referencia al objeto. En el ejemplo anterior, `object1` es una referencia a un objeto que se basa en `Customer`. Esta referencia apunta al objeto nuevo, pero no contiene los datos del objeto. De hecho, puede crear una referencia de objeto sin tener que crear ningún objeto:

```
Customer object2;
```

No se recomienda crear referencias de objeto como esta, que no hace referencia a ningún objeto, ya que, si se intenta obtener acceso a un objeto a través de este tipo de referencia, se producirá un error en tiempo de ejecución. Pero dicha referencia puede haberse creado para hacer referencia a un objeto, ya sea creando un nuevo objeto o asignándola a un objeto existente, como en el siguiente ejemplo:

C#Copiar

```
Customer object3 = new Customer();  
Customer object4 = object3;
```

Este código crea dos referencias de objeto que hacen referencia al mismo objeto. Por lo tanto, los cambios efectuados en el objeto mediante `object3` se reflejan en los usos posteriores de `object4`. Dado que los objetos basados en clases se tratan por referencia, las clases se denominan "tipos de referencia".