

www.PragimTech.com

Training +Placements = Our success

Pragim@PragimTech.com

► **Parte 29 – Diferencias entre clases y estructuras en C#**

Cristina Carrasco

PRAGIM Technologies

Cristina@cristinacarrasco.com

En esta lección veremos:

- ▶ Diferencias entre clases y estructuras

www.PragimTech.com

Training +Placements = Our success
Pragim@PragimTech.com

Estructuras(Struct)

En la parte 28 de esta serie de video comprendimos que las estructuras son muy similares a las clases. Utilizamos la palabra reservada **struct** para crear una estructura.

Al igual que las clases las estructuras pueden tener campo, propiedades, constructores y métodos.

Sin embargo, existen varias diferencias entre las clases y las estructuras de las cuales hablaremos en esta lección.

Clases vs Estructuras

Una estructura es **value type** o tipo de valor mientras que una clase es **reference type** o tipo de referencia.

Todas las diferencias que son aplicables a los **value types** o **reference types** también aplican para las clases y estructuras.

Las estructuras se almacenan en el **stack(la pila)**, mientras que las clases se almacenan en el **heap(el montón)**.

Los tipos de valor mantienen su valor en memoria cuando son declarados mientras que los tipos de referencia mantienen una referencia a un objeto en la memoria.

Los tipos de valor son destruidos inmediatamente después de que ya nadie los utiliza o de que pierden el scope, mientras que en los tipos de referencia solo la variable de referencia es destruida después de que se pierde el scope, el objeto se destruirá después por el **garbage collector**. (Hablaemos del garbage collector o recolector de basura en próximas lecciones.)

Cuando pasamos el valor de una estructura a otra estructura, se crea una nueva copia de la estructura y si modificamos algo en alguna de las estructuras esos cambios no se verán reflejados en la otra.

Cuando copiamos una clase a otra clase solo se genera una copia de la variable de referencia. Y ambas variables de referencia apuntan al mismo objeto en memoria que se encuentra almacenado en el **heap**(el montón). Así que cualquier cambio hecho en una de las clases afectará al valor contenido por la otra variable de referencia porque ambas apuntan al mismo valor.

Clases vs Estructuras

Una estructura no puede tener destructores mientras que una clase si puede tener destructores.

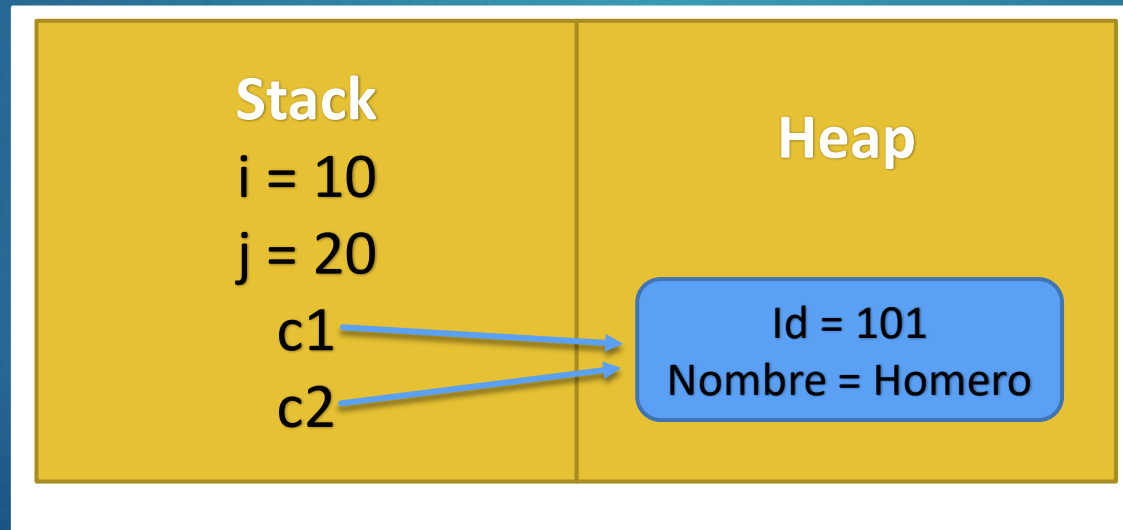
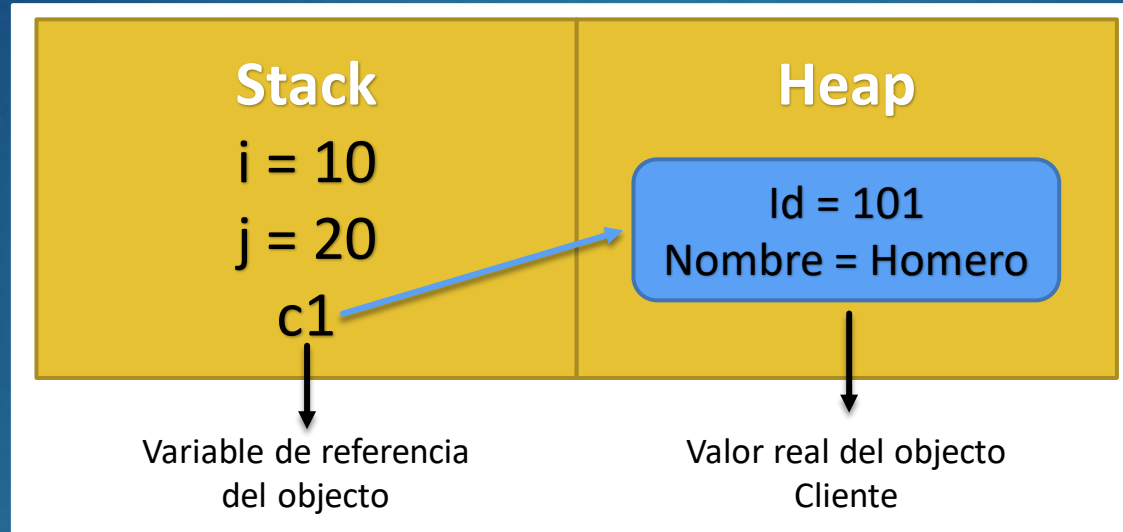
Las estructuras no pueden tener un constructor explicito sin parámetros mientras que las clases si.

Ejemplos de estructuras en el .NET framework: `int(System.Int32)`, `double(System.Double)`, etc.

Note 1: Una clase o una estructura no puede heredar de otra estructura. Las estructuras son sealed types (tipos sellados).

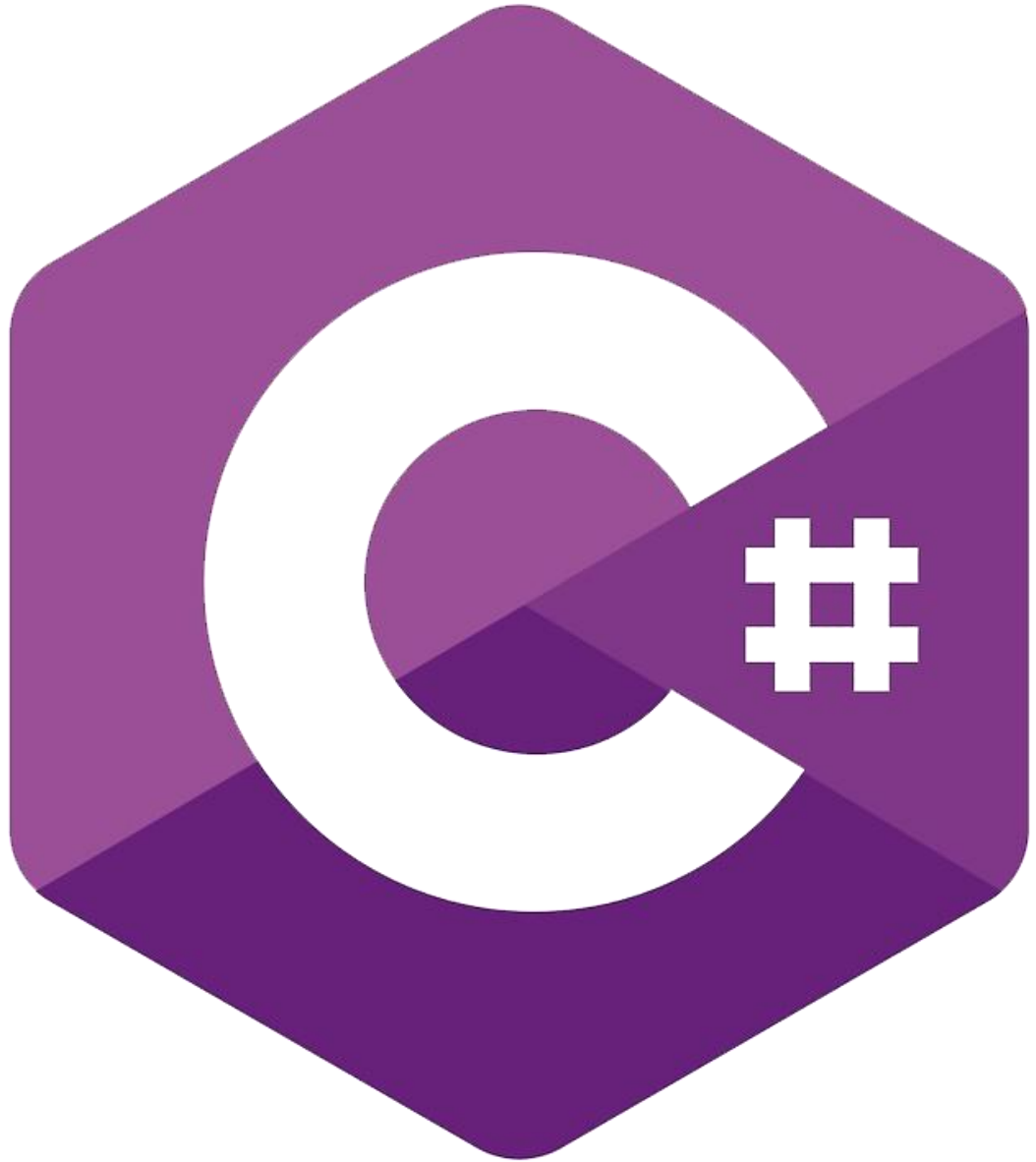
Note 2: ¿Cómo prevenir que una clase pueda ser heredada? O ¿Cuál es el significado de la palabra reservada `sealed`?

Stack(la pila) y Heap(el montón)



Recursos adicionales

- ▶ PRAGIM Pagina principal
 - ▶ <http://www.PragimTech.com>



[29]

Diferencias
entre Clases y
Estructuras