

Pregunta

[Arturo Masero](#) · 17 feb, 2022

Ayuda con indirección y ámbito de variables

Hola compañeros/as, necesito de vuestra sabiduría.

En mi compañía programamos en ObjectScript y .int todo el día. Cuando entré nadie sabía apenas nada de clases y su uso era prácticamente sólo para definición de almacenamiento. Durante mi autoaprendizaje en caché descubrí que la programación orientada a objetos y el desarrollo de clases era posible en caché y comencé a codificar en .cls . Al ser algo autodidacta, puede que tenga algunas dudas básicas que se me hayan pasado en mis lecturas de documentación.

Con este escenario en mente, es común que utilicemos el operador @ para utilizar comodines para almacenar las mismas estructuras de datos en tablas con nombre diferente, o para jugar con los índices (para desglosar y agrupar información en distintos niveles, por ejemplo). Y en relación a esto he encontrado un problema en las clases que no sé cómo resolver.

Intentaré poner un ejemplo tan genérico y entendible como me sea posible:

En OS .int haríamos esto:

```
Indirection()  
N (evenTotal)  
  
set TABLES(0)="EVEN"  
set TABLES(1)="ODD"  
  
for i=1:1:100 {  
    set table = TABLES((i#2)) set @(table_"_i_")=i  
    // Xecute alternative, same issue  
    ; set cmd = "set "_table_"("_i_")="_i_  
    ; x cmd  
}  
  
set evenTotal=0  
  
set i="" for { set i=$O(EVEN(i)) Q:i=""  
    set evenTotal = evenTotal+EVEN(i)  
}  
zw evenTotal  
Q
```

Funciona y cuando termina no hay variables en memoria (salvo evenTotal, que está incluido en el NEW).

Pero cuando hago esto en .cls:

```
ClassMethod Indirection()  
{  
set TABLES(0)="EVEN"
```

```
set TABLES(1)="ODD"

for i=1:1:100 {
    set table = TABLES((i#2))
    set @(table_"("_i_)")=i

    // Xecute alternative, same issue
    ; set cmd = "set "_table_"("_i_)="_i
    ; x cmd
}

set evenTotal=0
set i="" for { set i=$ORDER(EVEN(i)) QUIT:i=""
    set evenTotal = evenTotal+EVEN(i)
}
zwrite evenTotal

}
```

Obtengo evenTotal igual a cero. Dentro del método puedo hacer zwrite EVEN y ver que contiene los valores, pero el ORDER no obtiene nada y si hago 'write EVEN(2)' obtengo UNDEFINED, aunque veo el valor si hago un zwrite. Es muy confuso.

Un efecto secundario que obtengo usando @ es que las variables EVEN y ODD aparecen cargados en memoria después de la ejecución del método. No los quiero en memoria.

Jugar con la keyword PublicList en la definición del método no me ha funcionado.

¿Cómo puedo hacer que el ejemplo de 'ObjectScript .int' funcione correctamente en .cls?

Actualmente trato de evadir esta práctica en cls, y algunas raras veces agrego las variables a la publicList y trato de MATARLAS antes de que termine el método. Hasta ahora ha funcionado.

[#ObjectScript](#) [#Caché](#)

Product version: Caché 2018.1

\$ZV: Cache for Windows (x86-64) 2018.1.2 (Build 309521269) Fri Nov 26 2021 11:37:39 EST

URL de
fuente: <https://es.community.intersystems.com/post/ayuda-con-indirecci%C3%B3n-y-%C3%A1mbito-de-variables>