Article
Alexey Maslov · Mar 14, 2019 3m read

Side Effects of Quit and Return Commands with $Increment, or "I.E.Repin. Unexpected..."

In recent discussion on CachéQuality I was (friendly) blamed for old syntax promotion and deliberate obfuscation of the code. Therefore I decided to clarify my point and shed some light on one of possible source of side effects that may unexpectedly occur with RETURN command with an argument.

Disclaimer. While the specific of RETURN command is not discussed in this text, it can be replaced with QUIT everywhere without loosing the meaning.

Honestly, there are not so many constructs in ObjectScript prone to side effects. One of them is built-in $increment function. As everybody knows, it performs an atomic operation on its arguments, not only returning the calculated sum, but also affecting the first argument. E.g.

```
USER> set a=1 set b=$increment(a,2) zwrite a,b
a=3
b=3
```

The default value for the second argument is 1. Further details on $increment can be found in docs.

Now, *revenons à nos moutons.* After restyling, my sample from CachéQuality discussion looks like this:

```
/// write ##class(z.Scratch).t1()
ClassMethod t1() As %Integer [ ProcedureBlock = 1 ]
{
  do ..t2(.a,.b)
  do ..t3(.a,.b)
  return $increment(a,-1)*$increment(b,-1)
}

ClassMethod t2(ByRef pA, ByRef pB) As %Integer [ ProcedureBlock = 1 ]
{
  set pA=2, pB=3
  return pA*pB
}

ClassMethod t3(ByRef pA, ByRef pB) As %Integer [ ProcedureBlock = 1 ]
{
  set pA=4, pB=5
  return $increment(pA)*$increment(pB)
}
```

Let's try to guess the return value of method t1().

First of all, method t2() sets private variables **a** and **b** (passed by reference using ".var" syntax) to 2 and 3 respectively. Returned value of t2() is discarded by caller as t2() is called using DO command rather than using "SET x=..t2(.a,.b) ". Off with it.

Secondly, method t3() does the similar job, setting the same variables to 4 and 5. But on return it calculates the product of their $incremented values. Again, the returned value is discarded by caller.

At last, t1() returns $increment(a,-1)*$increment(b,-1).

What value will it return? One can expect that as $increment(pA)*$increment(pB) result is dropped, the calculation does not occur at all, so 3*4=12 will be returned, and will be wrong! In reality, the RETURN command argument comprises the expression $increment(pA)*$increment(pB) is calculated anyway, therefore both variables are incremented, so **a**=5 and **b**=6 on t3() exit. That's why RETURNing $increment(**a**,-1)*$increment(**b**,-1) we get (5-1)*(6-1)=20.

To be honest, these species of RETURN command are well documented. To avoid possible side effects of that kind, we could modify the last line of t3(), getting the following:

```
ClassMethod t3(ByRef pA, ByRef pB) As %Integer [ ProcedureBlock = 1 ]
{
  set pA=4, pB=5
  return:$quit $increment(pA)*$increment(pB) return
}
```

By doing so, we introduce the check up of calling sequence type: as a function ($QUIT=1) or not ($QUIT=0), calculating the result in the first case only. Our case is second one, so **a** and **b** would keep 4 and 5 values on return, and t1() would return (4-1)*(5-1)=12. Whether it is desirable, depends on algorithm used. My sample looks rather silly, but its purpose was no more than demonstrate RETURN + $increment possible combined side effect.

Hope this help somebody.

P.S. Ilya Repin's "Unexpected [Visitor]" famous painting image can be found here.

#ObjectScript #Caché #InterSystems IRIS

Source URL:https://community.intersystems.com/post/side-effects-quit-and-return-commands-increment-or-ierepin-unexpected