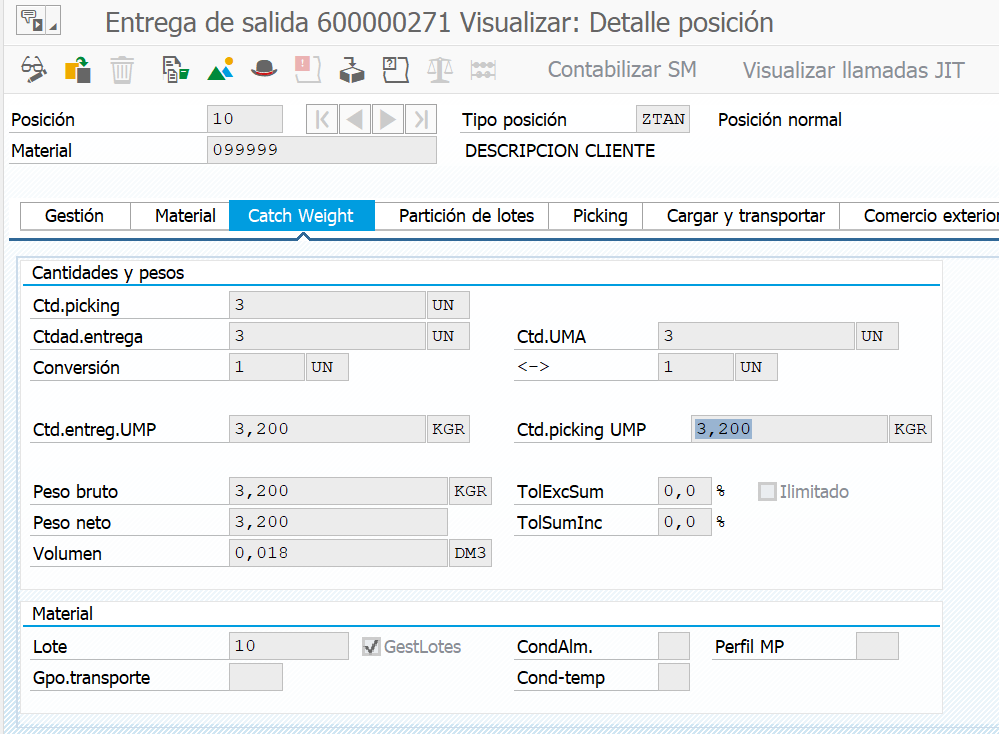
## DT - Cálculo de condiciones

## de precio con picking

El objetivo de este proceso es obtener los valores e importe de las condiciones de compra de una entrega. Usualmente sería tan sencillo como extraer dichos datos de la tabla **PRCD\_ELEMENTS** con elpedido de ventas asociado a la entrega (para cada posición).

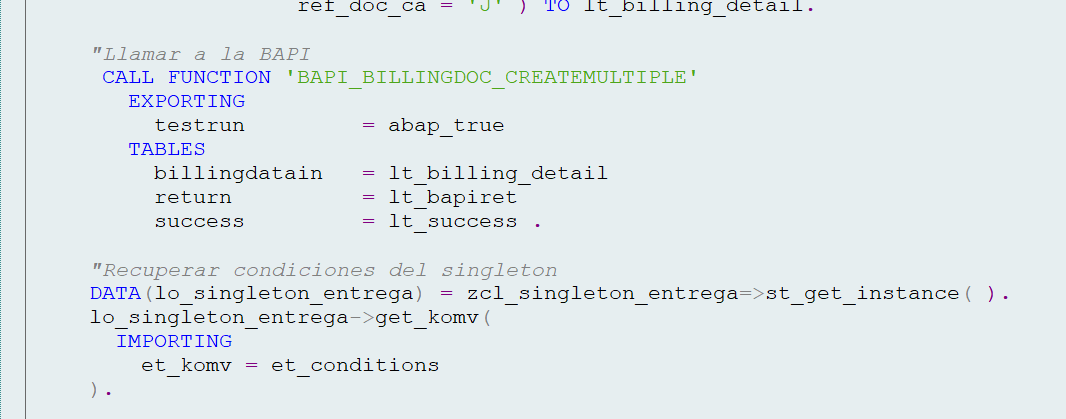
El problema surge que la tabla **PRCD\_ELEMENTS** nocalcula bien los importes cuando hay una cantidad de picking.



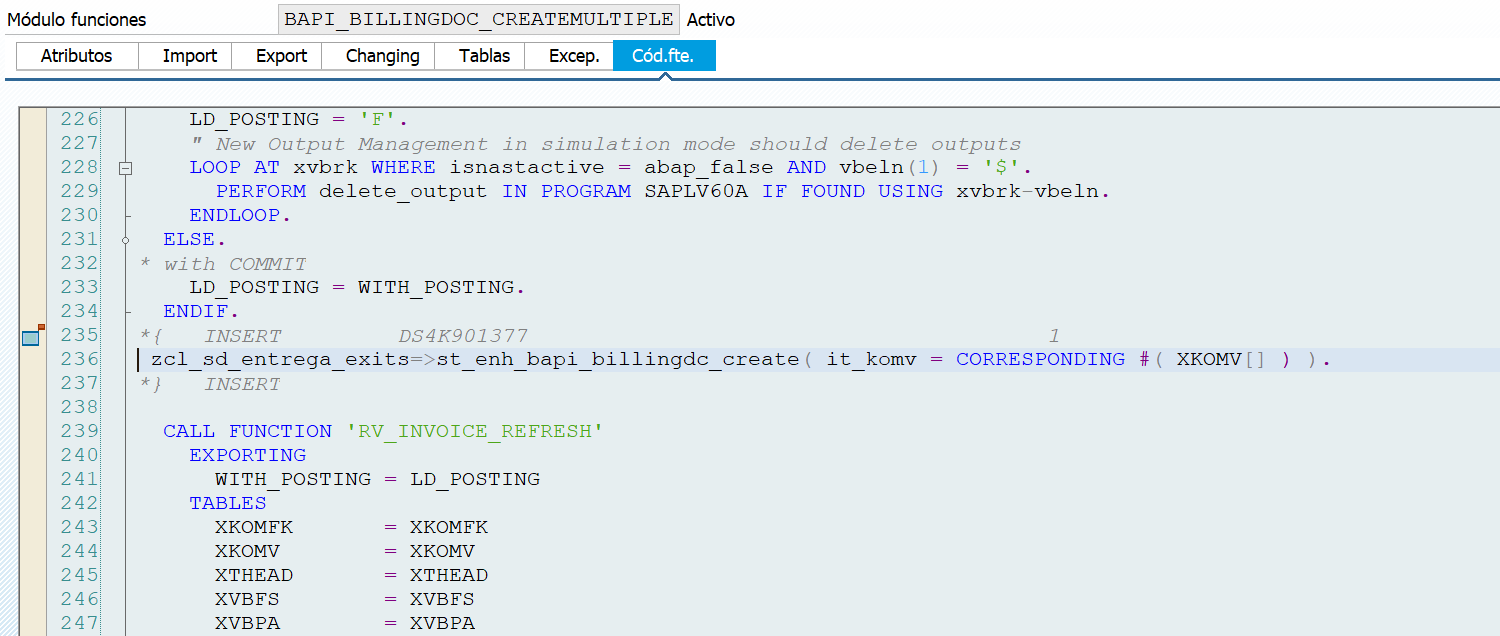
Una de las soluciones a este asunto consiste en hacer crear un pedido ficticio con la entrega, usando la BAPI **BAPI\_BILLINGDOC\_CREATEMULTIPLE.** No obstante, esa solución trae un inconveniente, si bien la BAPI calcula correctamente los importes y los recoge en una tabla, al final de la ejecución borra esa tabla. Es decir, necesitamos recuperar esos datos antes de que la BAPI los borre.

Para la solución de este inconveniente se han realizado los siguientes pasos

Primero se realiza la llamada a la BAPI desde nuestro programa o clase de inicio



Se ha añadido un enhancement a la BAPI **BAPI\_BILLINGDOC\_CREATEMULTIPLE** que extrae los datos de la tabla XKOMV, antes que borre el contenido. Es importante remarcar, que no se ha realizado ninguna modificación del funcionamiento de la BAPI,

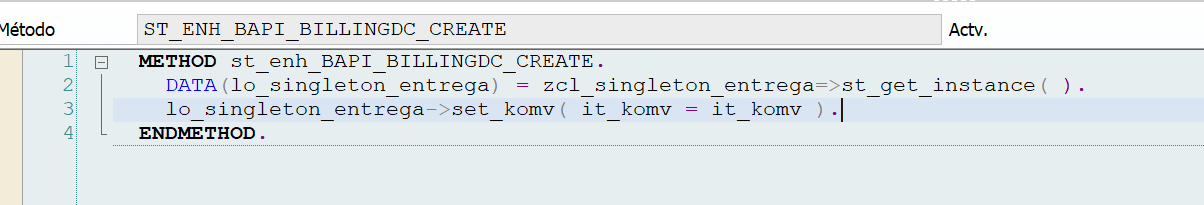


En el enhacement, se ha realizado una llamada al método **ST\_ENH\_BAPI\_BILLINGDC\_CREATE** de la clase **ZCL\_SD\_ENTREGA\_EXITS.**

Todos los procesos que involucren BAPIs, BADIs o Exits están agrupados en una clase Z (en nuestro caso, **ZCL\_SD\_ENTREGA\_EXITS**) con el fin de tener una trazabilidad en el caso que se requiriera realizar más ajustes en un futuro.

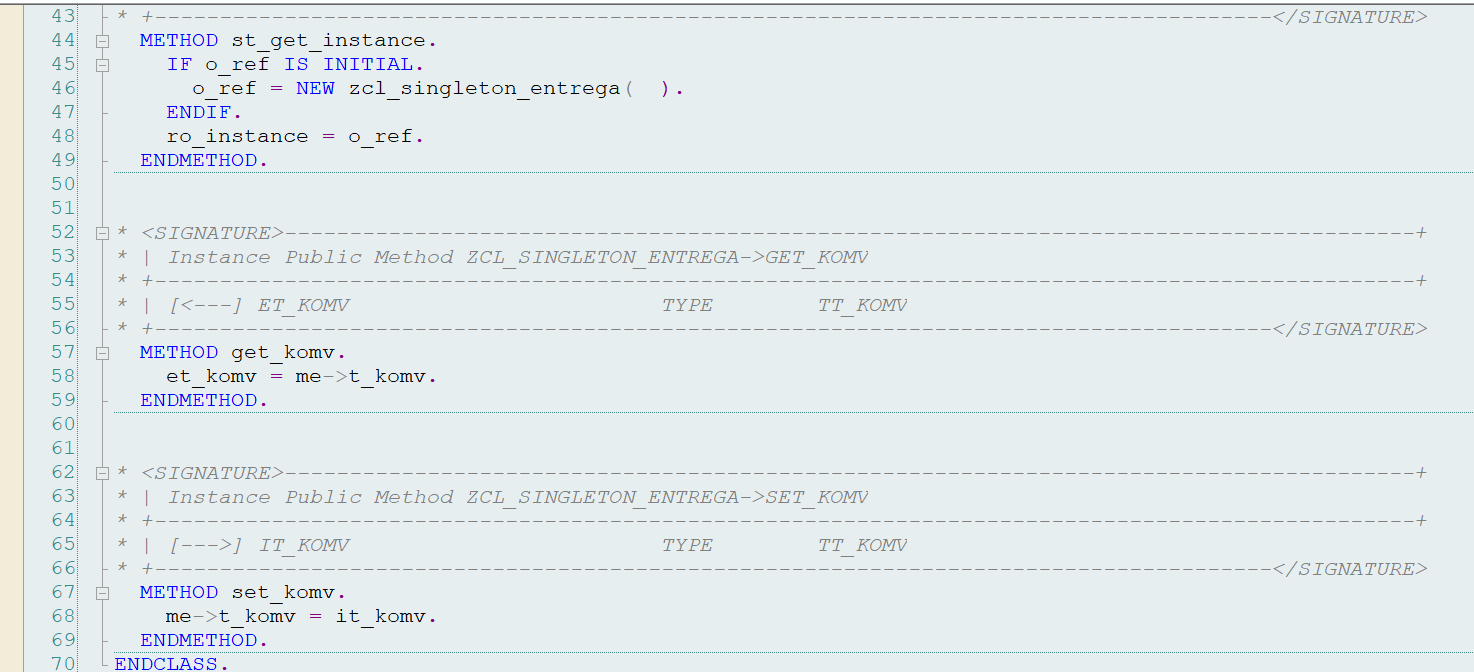
La funcionalidad del método **ST\_ENH\_BAPI\_BILLINGDC\_CREATE** es la de guardar los datos en una memoria interna para, en otra llamada a la clase, recuperarlos y procesar dichos datos.

Para ello, se crea una instancia de su clase Singleton correspondiente (**ZCL\_SINGLETON\_ENTREGA)** y se llama al método para guardar la tabla **(SET\_KOMV)**



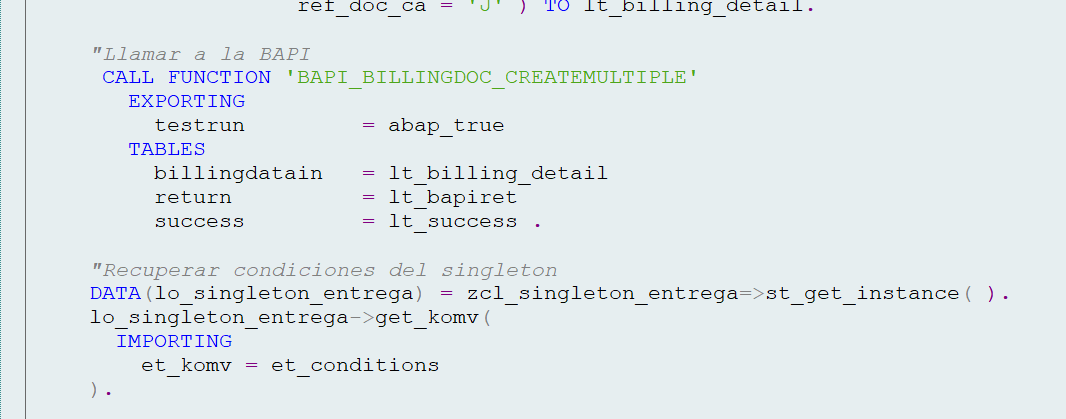
Las clases singleton son un tipo de clases cuyo diseño restringe la creación a una única instancia. Mediante esta particularidad es posible recibir y guardar datos de diferentes procesos sin tener que recurrir a la memoria interna.

Las clases singleton tienen la misma estructura. Un método estático GET\_INSTANCE y un par de métodos SET\_\*, donde se guardan los datos y GET\_\* donde recibe los datos. En nuestro caso, la clase **ZCL\_SINGLETON\_ENTREGA,** con los métodos **GET\_KONV y SET\_KOMV** recoge y envía datos de las condiciones de precio de la entrega. Teniendo la siguiente forma



Ahora que tenemos el contenido de la tabla KOMV guardado en nuestro singleton, podemos llamarlo desde nuestro programa original

Para ello se instancia la clase **ZCL\_SINGLETON\_ENTREGA**, y llamando al método **GET\_KOMV** obtenemos los datosde precio de la entrega que habíamos guardado anteriormente



Y una vez obtenido podemos seguir con el proceso.