**Cycle#2: Working With Private Methods**

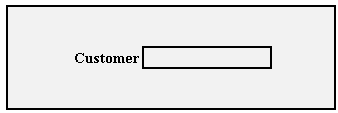
Private methods defined in the class to modularize the logic inside the class. Private method can be called with in the other methods of the Class. Private methods cannot be called from Outside.

**Following examples demonstrates how to**

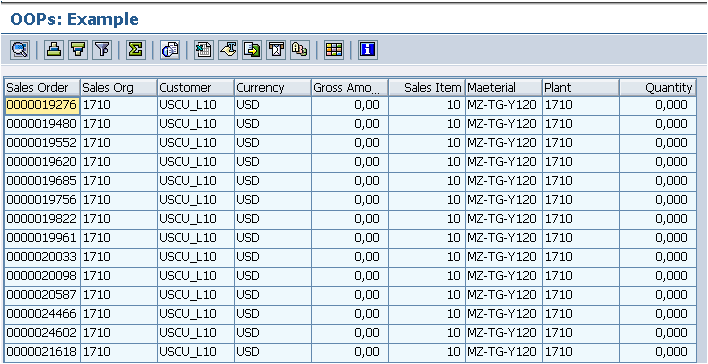
* How to define and Implement Private Methods
* How to call Private Methods from other Method of the Class

**Example: Display list of Sales orders for the Customer**

**Selection Screen:**



**Output:**



**Steps to be done Class**

* 1. **Class Definition**
     1. Private Section
        1. Declare Variable G\_KUNNR of Type KUNNR
        2. Declare Internal table IT\_ORDERS to hold Sales Order Details
        3. Declare Method GET\_ORDERS without any parameters
     2. Public Section
        1. Declare Constructor with Importing Parameter KUNNR
        2. Declare Method DISPLAY without any parameters
  2. **Class Implementation**
     1. CONSTRUCTOR Method

Set the G\_KUNNR with importing value

* + 1. GET\_ORDERS Method

Get Data from VBAK and VBAP Using Inner Join

* + 1. DISPLAY

Select the Sales Order details by calling the Private Method GET\_ORDERS

Either use LOOP … END LOOP to Print the Sales Order or use ALV

In the following Example ALV Used

**Consume Class from the Report**

1. Declare Parameter for the Customer
2. Declare reference to Class
3. Create Instance for the Class and pass the Customer Number
4. Call DISPLAY Method using Instance

**Logic:**

*\*&---------------------------------------------------------------------\**  
*\*& Report ZRTEST\_EXAMPLE\_04*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zrtest\_example\_06.  
TYPES:  
  BEGIN OF ty\_sales,  
    vbeln TYPE vbeln,  
    vkorg TYPE vkorg,  
    kunnr TYPE kunnr,  
    waerk TYPE waerk,  
    netwr TYPE netwr,  
    posnr TYPE posnr,  
    matnr TYPE matnr,  
    werks TYPE werks\_d,  
    zmeng TYPE dzmeng,  
  END OF ty\_sales,  
  ty\_sales\_tab TYPE TABLE OF ty\_sales.  
  
CLASS lcl\_example\_01 DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
      constructor IMPORTING iv\_kunnr TYPE kunnr,  
      display.  
  PROTECTED SECTION.  
  PRIVATE SECTION.  
    DATA:  
      g\_kunnr  TYPE kunnr,  
      it\_sales TYPE ty\_sales\_tab.  
    *" Private method can be called other method of the class only*  
    *" We can't call from outside using Object reference*  
    METHODS get\_orders.  
ENDCLASS.  
  
CLASS lcl\_example\_01 IMPLEMENTATION.  
*\*--------------------------------------------------------------------\**  
*\*  Constructor*  
*\*--------------------------------------------------------------------\**  
  METHOD constructor.  
    g\_kunnr = iv\_kunnr.  
  ENDMETHOD.  
*\*--------------------------------------------------------------------\**  
*\*  Get Sales Orders for the Given Customer*  
*\*--------------------------------------------------------------------\**  
  METHOD get\_orders.  
    SELECT  
      vbak~vbeln  
      vkorg  
      kunnr  
      vbak~waerk  
      vbak~netwr  
      posnr  
      matnr  
      werks  
      zmeng  
      INTO TABLE it\_sales  
      FROM vbak INNER JOIN vbap  
      ON vbak~vbeln = vbap~vbeln  
      WHERE kunnr = g\_kunnr.  
  ENDMETHOD.  
*\*--------------------------------------------------------------------\**  
*\*  Display Sales Orders*  
*\*---------------- ----------------------------------------------------\**  
  METHOD display.  
    *" Call Get Data method*  
    CALL METHOD get\_orders.  
    *" Fill Field Catalog*  
    DATA(lt\_fcat) = VALUE slis\_t\_fieldcat\_alv(  
      ( fieldname = 'VBELN' tabname = 'VBAK' seltext\_m = 'Sales Order' )  
      ( fieldname = 'VKORG' tabname = 'VBAK' seltext\_m = 'Sales Org' )  
      ( fieldname = 'KUNNR' tabname = 'VBAK' seltext\_m = 'Customer' )  
      ( fieldname = 'WAERK' tabname = 'VBAK' seltext\_m = 'Currency' )  
      ( fieldname = 'NETWR' tabname = 'VBAK' seltext\_m = 'Gross Amount' )  
      ( fieldname = 'POSNR' tabname = 'VBAP' seltext\_m = 'Sales Item' )  
      ( fieldname = 'MATNR' tabname = 'VBAP' seltext\_m = 'Maeterial' )  
      ( fieldname = 'WERKS' tabname = 'VBAP' seltext\_m = 'Plant' )  
      ( fieldname = 'ZMENG' tabname = 'VBAP' seltext\_m = 'Quantity' ) ).  
    CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
      EXPORTING  
        i\_callback\_program = sy-repid  
        it\_fieldcat        = lt\_fcat  
      TABLES  
        t\_outtab           = it\_sales  
      EXCEPTIONS  
        program\_error      = 1  
        OTHERS             = 2.  
    IF sy-subrc <> 0.  
*\* Implement suitable error handling here*  
    ENDIF.  
  
  ENDMETHOD.  
ENDCLASS.  
  
PARAMETERS : p\_kunnr TYPE kunnr.  
  
START-OF-SELECTION.  
  DATA: o\_ref TYPE REF TO lcl\_example\_01.  
  CREATE OBJECT o\_ref EXPORTING iv\_kunnr = p\_kunnr.  
  CALL METHOD o\_ref->display.