

Erstellen eines kurzen Konzepts zur folgenden Problemstellung:

1. Einführung

Ein Kunde möchte einen „Programm“. Auf dem Bildschirm soll es drei Eingabefelder geben. In einem Eingabefeld soll man „Datum“ eintragen können, in dem anderen sollen „Payment“ und in dem letzten „Currency“ eintragen können. Außerdem gibt es auch eine Checkbox, mit der die Currency ausgeblendet oder eingeblendet werden. Nachdem erfolgreichen Eingaben soll der User Ergebnis „Durchführen“ Button anklicken.

Anschließend soll es eine Ausgabe wie im Folgenden angezeigt werden.

| Order Number | Order Date | PM | TA | Currency |
|--------------|------------|----|----|----------|
| | | | | |
| | | | | |

2. Ist-Stand

Zurzeit gibt es keine Dasselbe oder Ähnliches.

3. Projektziel

Dieses Konzept bietet ein Konzept, das angezeigt wird, wie viele Käufe gemacht werden.

Das Programm soll dem Kunden helfen, in kurzer Zeit eine bestimmte Eingabe, Datum und Payment und Currency auflisten zu lassen und produktiv zu arbeiten.

Das Programm soll die entsprechende Tabelle als ALV-GRID erstellen. Damit können die Ergebnisse visuell schöner aufgelistet werden.

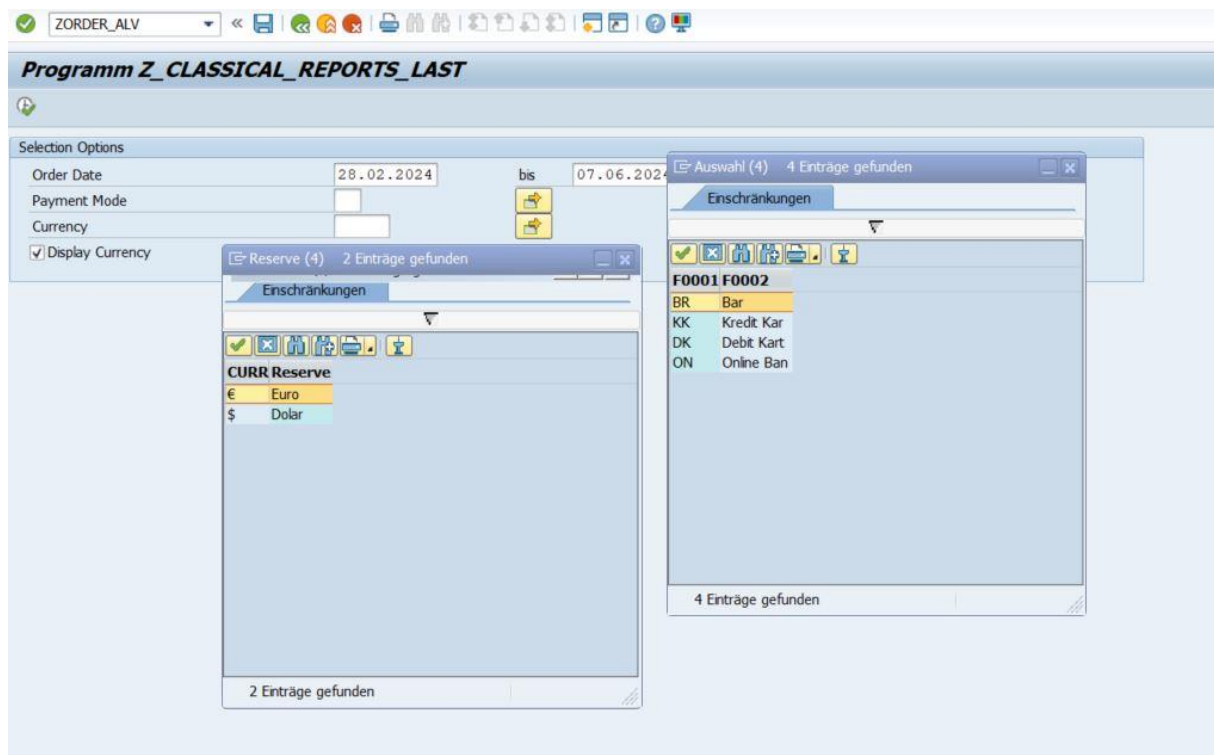
4. Beschreibung des Geschäftsprozess

Das Programm hilft dabei, dass die Firma damit Zeit sparen kann. Mit Hilfe des Programms kann die Bilanz auch erstellt werden.

Die aufgeführten Ergebnisse können die Vertraulichkeit des Unternehmens verletzen. Zu diesem Zweck ist eine Autorisierung vorgesehen. Das Programm ist vorgesehen, dass es von nur Admin benutzt wird.

5. Anforderungen an die Funktionalität

Auf der GUI soll es drei Eingabe und eine Checkbox sein. Die Checkbox hilft es dabei, die Currency einzublenden oder auszublenden. Nicht nur Payment_Mode sondern auch Currency soll es eine F-4 Hilfe erstellt werden.



Hier ist ein Beispiel von Ergebnissen.

| Order Number | Order Date | PM | TA | CURR |
|--------------|------------|----|------|-------|
| 94 | 16.05.2024 | BR | 9400 | USD |
| 95 | 16.05.2024 | BR | 9500 | USD |
| 96 | 16.05.2024 | BR | 9600 | USD |
| 97 | 16.05.2024 | KK | 9700 | € |
| 98 | 16.05.2024 | KK | 9800 | € |
| 99 | 16.05.2024 | KK | 9900 | € |
| 77 | 16.05.2024 | DK | 7700 | Real |
| 78 | 16.05.2024 | DK | 7800 | Real |
| 79 | 16.05.2024 | DK | 7900 | Real |
| 80 | 16.05.2024 | DK | 8000 | Real |
| 81 | 16.05.2024 | DK | 8100 | Real |
| 72 | 20.05.2024 | ON | 7200 | Franc |
| 73 | 20.05.2024 | ON | 7300 | Franc |
| 74 | 20.05.2024 | ON | 7400 | Franc |
| 75 | 20.05.2024 | ON | 7500 | Franc |
| 76 | 20.05.2024 | ON | 7600 | Franc |
| 2 | 24.05.2024 | BR | 0200 | Euro |
| 3 | 24.05.2024 | BR | 0300 | Euro |
| 4 | 24.05.2024 | BR | 0400 | Euro |
| 5 | 24.05.2024 | BR | 0500 | Euro |
| 6 | 24.05.2024 | BR | 0600 | Euro |
| 7 | 24.05.2024 | BR | 0700 | Euro |

6. Technische Lösung

Die sogenannten Codes sind im Folgende aufgelistet.

```

*&-----*
*& Report Z_CLASSICAL_REPORTS_LAST
*&-----*
*& Sefa Alioglu
*& 07.06.2024 eine Transaktion für Auftragsnummer XXXXXX.
*&-----*
REPORT Z_CLASSICAL_REPORTS_last.  " LINE-COUNT 25(2).

INCLUDE z_classical_reports_last_TOP.
INCLUDE z_classical_reports_last_SEL.
INCLUDE z_classical_reports_last_E01.
INCLUDE z_classical_reports_last_P01.

*&-----*
*& Include          Z_CLASSICAL_REPORTS_LAST_TOP
*&-----*

DATA: lv_odate TYPE zdeodate_28,
      lv_pm    TYPE zdepm_28,
      lv_curr  TYPE zdecrr_28.
DATA: gv_option_f4_hilfe(1) TYPE c.

CLASS z_classical_reports_last DEFINITION.

  PUBLIC SECTION.

    METHODS: f4_hilfe IMPORTING lv_option_f4_hilfe TYPE c,
              data_fetch IMPORTING lv_odate TYPE zdeodate_28
                               lv_pm    TYPE zdepm_28
                               lv_curr  TYPE zdecrr_28,
              f4_hilfe_display IMPORTING lv_option_f4_hilfe TYPE c,
              selection_screen,
              alv_grid.

  PROTECTED SECTION.

  PRIVATE SECTION.

    DATA: lt_fieldcat TYPE slis_t_fieldcat_alv,
           ls_fieldcat TYPE slis_fieldcat_alv.

    DATA: BEGIN OF ms_payment,
            pm    TYPE char2, "zdepm_28,
            desc TYPE char10,
          END OF ms_payment,
          mt_payment LIKE TABLE OF ms_payment.

    DATA: BEGIN OF ms_curr,
            curr TYPE zdecrr_28,
            desc TYPE char6,
          END OF ms_curr,

```

```
mt_curr LIKE TABLE OF ms_curr.

DATA: BEGIN OF ms_order,
      lt_ono   TYPE zdeono_28,
      lt_odate TYPE zdeodate_28,
      lt_pm    TYPE zdepm_28,
      lt_ta    TYPE zdepm_28,
      lt_curr  TYPE zdecurr_28,
END OF ms_order,
mt_order LIKE TABLE OF ms_order.

ENDCLASS.
```

```

*&-----*
*& Include          Z_CLASSICAL_REPORTS_LAST_SEL
*&-----*
*****
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-001.
SELECT-OPTIONS: s_odate FOR lv_odate NO-EXTENSION, "Ohne Mehrfachselektion
                s_pm FOR lv_pm NO INTERVALS,      "Ohne max selektin
                s_curr FOR lv_curr NO INTERVALS MODIF ID cu1.  "Ohne max s
elektion  "MODIF ID => name of group of s_curr

PARAMETERS : p_chck TYPE c AS CHECKBOX USER-
COMMAND display. "w/o enter only selecting the chckbox.

SELECTION-SCREEN END OF BLOCK b1.

```

```

*&-----*
*& Include          Z_CLASSICAL_REPORTS_LAST_P01
*&-----*
CLASS z_classical_reports_last IMPLEMENTATION.

METHOD f4_hilfe.
  CASE lv_option_f4_hilfe.
    WHEN 'P'.
      IF mt_payment IS INITIAL.
        ms_payment-pm = 'BR'.
        ms_payment-desc = 'Bar'.
        APPEND ms_payment TO mt_payment.
        CLEAR ms_payment.

        ms_payment-pm = 'KK'.
        ms_payment-desc = 'Kredit Karte'.
        APPEND ms_payment TO mt_payment.
        CLEAR ms_payment.

        ms_payment-pm = 'DK'.
        ms_payment-desc = 'Debit Karte'.
        APPEND ms_payment TO mt_payment.
        CLEAR ms_payment.

        ms_payment-pm = 'ON'.
        ms_payment-desc = 'Online Banking'.
        APPEND ms_payment TO mt_payment.
        CLEAR ms_payment.
      ENDIF.

    WHEN 'C'.
      IF mt_curr IS INITIAL.
        ms_curr-curr = '€'.
        ms_curr-desc = 'Euro' .
        APPEND ms_curr TO mt_curr.
        CLEAR ms_curr.

        ms_curr-curr = '$'.
        ms_curr-desc = 'Dolar' .
        APPEND ms_curr TO mt_curr.
        CLEAR ms_curr.

      ENDIF.

    WHEN OTHERS.
  ENDCASE.

ENDMETHOD.

```

```

METHOD data_fetch.

    SELECT ono odate pm ta curr
    FROM zordh_28
    INTO TABLE mt_order
        WHERE odate IN s_odate
        AND pm IN s_pm
        AND curr IN s_curr
        ORDER BY odate.

ENDMETHOD.

METHOD alv_grid.

data: lo_alv type ref to CL_SALV_TABLE.

TRY.
    cl_salv_table=>factory(
        IMPORTING r_salv_table = lo_alv
        CHANGING  t_table      = mt_order ).

    " Optional: Set ALV display settings
    DATA: lo_columns TYPE REF TO cl_salv_columns_table,
           lo_column  TYPE REF TO cl_salv_column_table,
           lt_columns TYPE salv_t_column_ref,
           ls_column  TYPE salv_s_column_ref.

    lo_columns = lo_alv->get_columns( ).
    lt_columns = lo_columns->get( ).

    lo_alv->display( ).

    CATCH cx_salv_msg INTO DATA(lx_msg).
        MESSAGE lx_msg->get_text( ) TYPE 'S'.
    ENDTRY.

ENDMETHOD.

METHOD f4_hilfe_display.

    CASE lv_option_f4_hilfe.
        WHEN 'P'.
            CALL FUNCTION 'F4IF_INT_TABLE_VALUE_REQUEST'
                EXPORTING
                    retfield      = 'PM'                " Name des Rückgabefeldes
s in FIELD_TAB
                    dynpprog     = sy-repid              " Aktuelles Programm
                    dynpnr       = sy-dynnr              " Dynpro-Nummer
                    dynprofield   = 'S_PM-LOW'           " Name des Dynpro-
Feldes für Werterückgabe
                    value_org    = 'S'                  " Werteübergabe: C: zellenwe

```

```

ise, S: strukturiert
    window_title      = 'Auswahl'
    TABLES
        value_tab      = mt_payment          " Tabelle der Werte
; Einträge zellenweise
*         field_tab    = lt_field_tab
*         return_tab   = it_return
    EXCEPTIONS
        parameter_error = 1          " Fehlerhafte Parameter
        no_values_found = 2          " Es wurden keine Werte gefu
nden
        OTHERS          = 3.

```

```

    WHEN 'C'.
        CALL FUNCTION 'F4IF_INT_TABLE_VALUE_REQUEST'
            EXPORTING
                retfield      = 'DESC'          " Name des Rückgabefel
des in FIELD_TAB
                dynpprog      = sy-repid        " Aktuelles Programm
                dynpnr        = sy-dynnr        " Dynpro-Nummer
                dynprofield    = 'S_CURR-LOW'    " Name des Dynpro-
Feldes für Werterückgabe
                value_org     = 'S'            " Werteübergabe: C: zellenwe
ise, S: strukturiert
            TABLES
                value_tab     = mt_curr          " Tabelle der Werte; E
inträge zellenweise
            EXCEPTIONS
                parameter_error = 1          " Fehlerhafte Parameter
                no_values_found = 2          " Es wurden keine Werte gefu
nden
                OTHERS        = 3.

```

```

    WHEN OTHERS.
    ENDCASE.

```

```

ENDMETHOD.

```

```

METHOD selection_screen.
    lr_obj->f4_hilfe( lv_option_f4_hilfe = lv_option_f4_hilfe ).

```

```

DATA: counter(1) TYPE n.

```

```

IF s_pm IS NOT INITIAL.
    LOOP AT mt_payment INTO ms_payment.
        IF s_pm-low NE ms_payment-pm.

```



```
        counter = counter + 1 .
    ENDIF.
    IF counter EQ 4.
        MESSAGE: 'It is not valid, try again s_pm' TYPE 'E'.
    ENDIF.
ENDLOOP.
ENDIF.

ENDMETHOD.

ENDCLASS.
```

```

*&-----*
*& Include          Z_CLASSICAL_REPORTS_LAST_E01
*&-----*

```

```

DATA: lr_obj TYPE REF TO z_classical_reports_last.
DATA: lv_option_f4_hilfe TYPE c.

```

```

*****
*****

```

```

INITIALIZATION.  "Trigger Selection Screen before display
  CREATE OBJECT lr_obj.
  s_odate-sign = 'I'.  "I Include E Exclude
  s_odate-option = 'BT'. "Between
  s_odate-low = sy-datum - 100 . "System date minus 100 days
  s_odate-high = sy-datum .
  APPEND s_odate .

```

```

*****
*****

```

```

* At Selection Screen the purpose of this event is to validate the input.

```

```

AT SELECTION-SCREEN.
  lr_obj->selection_screen( ).

```

```

*****
*****

```

```

* This event call when user clicks only EXECUTE button on the selection scr
een. The selection logic is teh part of the this event.

```

```

START-OF-SELECTION.

```

```

  lr_obj->data_fetch(
    EXPORTING
      lv_odate = lv_odate
      lv_pm    = lv_pm
      lv_curr  = lv_curr ).

```

```

  lr_obj->alv_grid( ).

```

```

*****
*****

```

```

* This event calls when selection process ends. This event helps to identif
y the end of data/records. SAP call automatically this event.

```

```

END-OF-SELECTION.

```

```

* WRITE: /, / TEXT-000.

```

```

*****
*****

```

```

* event calls when the first WRIE statemant occurs in a program. The purpos
e of this event is to provide title/header at the beginning of a new page.

```

TOP-OF-PAGE.

```
* WRITE: / 'Page Number', sy-pagno.  
* WRITE: / 'Order Nummer' , 15 'Order Date' , 27 'Payment Mode', 40 'Currency'.
```

```
*****  
*****  
* calls when the last WRITE statement occurs in a program for a page. LINE  
COUNT 5(2) the total number of lines on a page in 5 and out of that 2 are  
reserved for end of page.
```

END-OF-PAGE.

```
* WRITE: 'End of Page Number ', sy-pagno.
```

```
*****  
*****  
* calls before displaying the selection screen/Input screen. Purpose is to  
modify the selection screen.  
* INITIALIZATION vs AT SELECTION-SCREEN OUTPUT.  
* INITIALIZATION event calls only first time before displaying the selecti  
on screen  
* AT Selection-Screen-  
Output event calls every time before displaying a selection screen.  
* INITIALIZATION => DEFAULT Values  
* AT SELECTION-SCREEN-OUTPUT => modify  
* Initialization erstmal, AT SELECTION-SCREEN-OUTPUT jedes mal.  
*****  
* AT SELECTION-SCREEN vs AT SELECTION-SCREEN-OUTOUT  
* enter/ Click before displaying  
* validate modify  
* PAI PBO
```

AT SELECTION-SCREEN OUTPUT.

```
IF p_chck EQ ' '.  
  LOOP AT SCREEN.  
    IF screen-group1 EQ 'CU1'.  
      screen-active = 0.  
*      screen-invisible = 1. "low is not invisible also  
    MODIFY SCREEN.  
  ENDIF.  
ENDLOOP.  
ENDIF.
```

```
*****  
* AT SELECTION-SCREEN ON VALUE REQUEST FOR <FIELD>  
* F4 => Value Help / Search Help  
* F1 => Technical Information Help  
* calls when user clicks F4 on field of Selection Screen to provide value h  
elp for a input field.
```

AT SELECTION-SCREEN ON VALUE-REQUEST FOR s_pm-low.

```
lv_option_f4_hilfe = 'P'.
```

```
IF lr_obj IS INITIAL.  
    CREATE OBJECT lr_obj.  
ENDIF.  
lr_obj->f4_hilfe( lv_option_f4_hilfe = lv_option_f4_hilfe ).  
lr_obj->f4_hilfe_display( lv_option_f4_hilfe = lv_option_f4_hilfe ).  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR s_curr-low.  
    lv_option_f4_hilfe = 'C'.  
    IF lr_obj IS INITIAL.  
        CREATE OBJECT lr_obj.  
    ENDIF.  
    lr_obj->f4_hilfe( lv_option_f4_hilfe = lv_option_f4_hilfe ).  
    lr_obj->f4_hilfe_display( lv_option_f4_hilfe = lv_option_f4_hilfe ).
```