\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_HELLO\_WORLD  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_hello\_world.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_HELLO\_WORLD                                    \*  
\* Report Title : Hello World Programm                                  \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/08/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S.                 \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.08.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
PARAMETERS:  
p\_nicknm(15) TYPE c.  
  
WRITE: /  'Hello World', p\_nicknm, '!'.  
  
WRITE: /  'Hello World', '!'.  
WRITE  /  'This is my first program.'.  
WRITE     'And not my last one.'.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_SYSTEM\_INFORMATION  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_system\_information.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_SYSTEM\_INFORMATION                            \*  
\* Report Title : System Information                                    \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/08/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 113                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.08.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
DATA:  
\* Date & time information variables  
l\_timestamp  TYPE timestampl,  
l\_string(30) TYPE c.  
  
\*Refresh all internal date and time fields like  
\* sy-uhzeit  
GET TIME.  
  
\*get exact timestamp  
GET TIME STAMP FIELD l\_timestamp.  
WRITE l\_timestamp TIME ZONE sy-zonlo TO l\_string.  
  
\* write info to screen  
WRITE: /, / 'Date & Time Information',  
       / 'System Date: ',             30 sy-datum,  
       / 'System Time: ',             30 sy-uzeit,  
       / 'Exact System Time Stamp:',  30 l\_string,  
       / 'Daylight Saving time: ',    30 sy-dayst AS  
                            CHECKBOX INPUT OFF,  
       / 'Local Date: ',              30 sy-datlo,  
       / 'Local Time: ',              30 sy-timlo,  
       / 'Local Time Zone: ',         30 sy-zonlo.  
WRITE: /, / 'Logical Information',  
       / 'Logical SAP System: ',      30 sy-sysid,  
       / 'Client: ',                  30 sy-mandt,  
       / 'user\_name: ',               30 sy-uname.  
WRITE: /,/ 'System & Program Information',  
       / 'SAP WebAS Name: ',          30 sy-host,  
       / 'Database Name: ',           30 sy-dbsys,  
       / 'Operating System Name: ',   30 sy-opsys,  
       / 'Program Transaction: ',     30 sy-tcode,  
       / ' Program Name: ',           30 sy-repid.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_SALES\_ORDER  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_sales\_order.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_SALES\_ORDER                                                 \*  
\* Report Title : Create outgoing Invoice Sales Order                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/08/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 127                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.09.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
PARAMETERS:  
      p\_ano(10) TYPE n OBLIGATORY,  
      p\_aname(40) TYPE c OBLIGATORY,  
      p\_aprice TYPE p DECIMALS 2 OBLIGATORY,  
      p\_curr TYPE currencysap OBLIGATORY DEFAULT 'EUR',  
      p\_aquant TYPE i OBLIGATORY DEFAULT 1,  
\*Tax  
      p\_tax TYPE p DECIMALS 2 DEFAULT '19' OBLIGATORY,  
\*Terms of Payment  
      p\_cash TYPE c RADIOBUTTON GROUP 0001 DEFAULT 'X',  
      p\_credit TYPE c RADIOBUTTON GROUP 0001,  
      p\_month TYPE i OBLIGATORY DEFAULT '24'.  
  
CONSTANTS:  
\*Interest per year in percent  
con\_annual\_interest TYPE p DECIMALS 2 VALUE '6.75'.   " Jahreszinssatz  
  
DATA:  
\*Temporary Data  
      l\_net\_amount TYPE p DECIMALS 2,         " Netto Betrag  
      l\_tax\_factor TYPE f,                    " Steuerfaktor  
      l\_credit\_amount TYPE p DECIMALS 2,      " Kreditbetrag  
      l\_monthly\_interest\_factor TYPE f,       " Monatlicher Zinssatz  
\*Result data  
      l\_monthly\_vat\_amount TYPE p DECIMALS 2, " Monatlicher Mehrwertsteuer Betrag  
      l\_monthly\_amount TYPE p DECIMALS 2,     " Monatliche Menge  
      l\_vat\_amount TYPE p DECIMALS 2,         " Mehrwertsteuerbetrag  
      l\_total\_amount LIKE l\_net\_amount.       " Gesamtmenge  
  
\* Temporary calculations  
l\_net\_amount = p\_aprice \* p\_aquant.  
l\_tax\_factor = p\_tax / 100.  
  
\*Write article information to screen  
WRITE: /, / 'Artikel Information',  
          / 'Artikle Number: ', 30 p\_ano,  
          / 'Artikle Name: ', 30 p\_aname,  
          / 'Artikle net price: ', 30 p\_aprice, p\_curr,  
          / 'Quantity: ', 30 p\_aquant.  
  
WRITE: /, / 'Result: '.  
IF p\_cash = 'X'.  
  
\* Calculate cash results  
  l\_vat\_amount = l\_net\_amount \* l\_tax\_factor.  
  l\_total\_amount = l\_net\_amount + l\_vat\_amount.  
  
\*Write results to screen  
  WRITE: / 'Total vat amount: ', 30 l\_vat\_amount,  
          p\_curr,  
         / 'Total amount: ', 30 l\_total\_amount, p\_curr.  
ELSE.  
  
\*Calculate interest result  
  l\_monthly\_interest\_factor = con\_annual\_interest / 100 / 12.  
  l\_credit\_amount = l\_net\_amount + l\_net\_amount \*  
                  l\_monthly\_interest\_factor \* p\_month.  
  
  l\_vat\_amount = l\_credit\_amount \* l\_tax\_factor.  
  l\_total\_amount = l\_credit\_amount + l\_vat\_amount.  
  l\_monthly\_vat\_amount = l\_vat\_amount / p\_month.  
  l\_monthly\_amount = l\_total\_amount / p\_month.  
\* Write results to screen  
  WRITE: / 'Month: ', 30 p\_month,  
         / 'Monthly vat amount: ', 30 l\_monthly\_vat\_amount, p\_curr,  
         / 'Monthlyamount: ', 30 l\_monthly\_amount, p\_curr,  
         / '(VAT amount: ', 30 l\_vat\_amount, p\_curr, ')',  
         / '(Total amount: ', 30 l\_total\_amount, p\_curr, ')'.  
ENDIF.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_SALES\_ORDER\_DYNAMIC  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_sales\_order\_dynamic.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_SALES\_ORDER\_DYNAMIC                                                 \*  
\* Report Title : Create outgoing Invoice Sales Order                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/08/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 133                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.09.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
PARAMETERS:  
      p\_ano(10)   TYPE n OBLIGATORY,  
      p\_aname(40) TYPE c OBLIGATORY,  
      p\_aprice    TYPE p DECIMALS 2 OBLIGATORY,  
      p\_curr      TYPE currencysap OBLIGATORY DEFAULT 'EUR',  
      p\_aquant    TYPE i OBLIGATORY DEFAULT 1,  
\*Tax  
      p\_tax       TYPE p DECIMALS 2 DEFAULT '19' OBLIGATORY,  
\*Terms of Payment  
      p\_cash      TYPE c RADIOBUTTON GROUP 0001 DEFAULT 'X',  
      p\_credit    TYPE c RADIOBUTTON GROUP 0001,  
      p\_month     TYPE i OBLIGATORY DEFAULT '24'.  
  
CONSTANTS:  
\*Interest per year in percent  
con\_annual\_interest TYPE p DECIMALS 2 VALUE '6.75'.   " Jahreszinssatz  
  
TYPES: currency     TYPE p DECIMALS 2.  
DATA:  
\*Temporary Data  
      l\_net\_amount TYPE p DECIMALS 2,                 " Netto Betrag  
      l\_tax\_factor TYPE f,                            " Steuerfaktor  
  
\*Result data  
      l\_vat\_amount TYPE p DECIMALS 2,                   " Mehrwertsteuerbetrag  
      l\_total\_amount LIKE l\_net\_amount,                 " Gesamtmenge  
  
\*Dynamic variables  
l\_rda\_credit\_amount TYPE REF TO currency,               " Kreditbetrag  
l\_rda\_monthly\_interest\_factor TYPE REF TO f,            " Monatlicher Zinssatz  
l\_rda\_monthly\_vat\_amount TYPE REF TO currency,          " Monatlicher Mehrwertsteuer Betrag  
l\_rda\_monthly\_amount TYPE REF TO currency.              " Monatliche Menge  
  
FIELD-SYMBOLS:  
\*A Access to reference variables  
<l\_credit\_amount> TYPE currency,  
<l\_monthly\_interest\_factor> TYPE f,  
<l\_monthly\_vat\_amount> TYPE ANY,  
<l\_monthly\_amount> TYPE currency.  
\*Temporary Calculations  
l\_net\_amount = p\_aprice \* p\_aquant.  
l\_tax\_factor = p\_tax / 100.  
  
  
\*Write article information to screen  
WRITE: /, / 'Artikel Information',  
          / 'Artikle Number: ', 30 p\_ano,  
          / 'Artikle Name: ', 30 p\_aname,  
          / 'Artikle net price: ', 30 p\_aprice, p\_curr,  
          / 'Quantity: ', 30 p\_aquant.  
\* Write conditions to screen  
WRITE: /, / 'Conditions',  
          / 'Tax rate: ', 30 p\_tax,  
          / 'Quantity: ', 30 p\_aquant.  
WRITE: /, / 'Result: '.  
IF p\_cash = 'X'.  
\* Calculate cash results  
  l\_vat\_amount = l\_net\_amount \* l\_tax\_factor.  
  l\_total\_amount = l\_net\_amount + l\_vat\_amount.  
  
\*Write results to screen  
  WRITE: / 'Total vat amount: ', 30 l\_vat\_amount,  
          p\_curr,  
         / 'Total amount: ', 30 l\_total\_amount, p\_curr.  
ELSE.  
\* Calculate Results  
  CREATE DATA l\_rda\_monthly\_interest\_factor.  
  ASSIGN l\_rda\_monthly\_interest\_factor->\* TO  
              <l\_monthly\_interest\_factor>.  
  
  CREATE DATA l\_rda\_credit\_amount.  
  ASSIGN l\_rda\_credit\_amount->\* TO <l\_credit\_amount>.  
  
  CREATE DATA l\_rda\_monthly\_vat\_amount.  
  
  ASSIGN l\_rda\_monthly\_vat\_amount->\* TO <l\_monthly\_vat\_amount>.  
  
  CREATE DATA l\_rda\_monthly\_amount.  
  ASSIGN l\_rda\_monthly\_amount->\* TO <l\_monthly\_amount>.  
  
  <l\_monthly\_interest\_factor> = con\_annual\_interest / 100 / 12.  
  <l\_credit\_amount> = l\_net\_amount + l\_net\_amount \*  
                          <l\_monthly\_interest\_factor> \* p\_month.  
  <l\_monthly\_amount> = l\_total\_amount / p\_month.  
  
\* Write results to screen  
  WRITE: / 'Month: ', 30 p\_month,  
         / 'Monthly vat amount: ', 30 <l\_monthly\_vat\_amount>, p\_curr,  
         / 'Monthlyamount: ', 30 <l\_monthly\_amount>, p\_curr,  
         / '(VAT amount: ', 30 l\_vat\_amount, p\_curr, ')',  
         / '(Total amount: ', 30 l\_total\_amount, p\_curr, ')'.  
ENDIF.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_INTERNAL\_TABLE\_JUGGLER  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_internal\_table\_juggler.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_INTERNAL\_TABLE\_JUGGLER                                                 \*  
\* Report Title : Interanl Table Juggler                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/08/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 147                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.09.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
TYPES: BEGIN OF str\_address,  
        street\_and\_number(40) TYPE c,  
        zipcode\_and\_city(40)  TYPE c,  
       END OF str\_address,  
       BEGIN OF str\_customer,  
         name(80) TYPE c,  
         address  TYPE str\_address,  
       END OF str\_customer.  
  
DATA:  
\*customer  
   l\_tab\_customer TYPE STANDARD TABLE OF str\_customer,  
   l\_str\_customer TYPE str\_customer.  
\* Fill some Data  
l\_str\_customer-name                      = 'John Miller'.  
l\_str\_customer-address-street\_and\_number = 'Yorkstr. 3'.  
l\_str\_customer-address-zipcode\_and\_city  = '69120 Heidelberg'.  
APPEND l\_str\_customer TO l\_tab\_customer.  
l\_str\_customer-name                      = 'Linda Evens'.  
l\_str\_customer-address-street\_and\_number = 'Yorkstr. 4'.  
APPEND l\_str\_customer TO l\_tab\_customer.  
l\_str\_customer-name                      = 'Robbie Busch'.  
l\_str\_customer-address-street\_and\_number = 'Yorkstr. 5'.  
APPEND l\_str\_customer TO l\_tab\_customer.  
  
\*Use Loop to write all data to screen  
LOOP AT l\_tab\_customer INTO l\_str\_customer.  
  WRITE: l\_str\_customer-name, l\_str\_customer-address-street\_and\_number.  
ENDLOOP.  
  
\*Read data of first customer into structure  
READ TABLE l\_tab\_customer INDEX 1 INTO l\_str\_customer.  
  
\* Modify name  
l\_str\_customer-name = 'Zeta Jones'.  
  
\*Write it back to first line of internal Table  
MODIFY l\_tab\_customer INDEX 1 FROM l\_str\_customer.  
  
\* Sort the table by the content of name column  
SORT l\_tab\_customer BY name.  
  
\* Just to Train the read statement  
WRITE sy-uline AS LINE.  
READ TABLE l\_tab\_customer INDEX 1 INTO l\_str\_customer.  
WRITE:  l\_str\_customer-name, l\_str\_customer-address-street\_and\_number.  
  
READ TABLE l\_tab\_customer INDEX 2 INTO l\_str\_customer.  
WRITE:  l\_str\_customer-name, l\_str\_customer-address-street\_and\_number.  
  
READ TABLE l\_tab\_customer INDEX 3 INTO l\_str\_customer.  
WRITE:  l\_str\_customer-name, l\_str\_customer-address-street\_and\_number.  
  
\*Actualy, clearing l\_tab\_customer would be better  
DELETE l\_tab\_customer FROM 1 TO 3.

FUNCTION ZPTB00\_CALCULATE\_TRAVEL\_EXPENS.  
\*"--------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     VALUE(I\_KM) TYPE  I  
\*"  EXPORTING  
\*"     VALUE(E\_AMOUNT) TYPE  F  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"--------------------------------------------------------------------  
\*ABAP Basics S. 162  
  e\_amount = i\_km \* '0.3'.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
ENDFUNCTION.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PERFORMANCE\_TESTER  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_performance\_tester.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PERFORMANCE\_TESTER                                                 \*  
\* Report Title : Trip Costs                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/08/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 158                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.09.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
PARAMETERS:  
\* KM of last Travel  
p\_km TYPE i.  
DATA:  
\* Amount and runtime for form and function  
l\_amount\_form TYPE f,         "Betrag Formular  
l\_runtime\_form TYPE i,        "Laufzeitformular  
l\_amount\_function TYPE f,     "Betragsfunction  
l\_runtime\_function TYPE i.    "Laufzeitfunction  
  
\*First initialize l\_runtime\_form, than measure performance  
GET RUN TIME FIELD l\_runtime\_form.  
PERFORM calculate\_travel\_expense USING p\_km CHANGING l\_amount\_form.  
PERFORM calculate\_travel\_expense USING p\_km CHANGING l\_amount\_form.  
PERFORM calculate\_travel\_expense USING p\_km CHANGING l\_amount\_form.  
PERFORM calculate\_travel\_expense USING p\_km CHANGING l\_amount\_form.  
GET RUN TIME FIELD l\_runtime\_form.  
\*First initialize l\_runtime\_function, then measure performance  
GET RUN TIME FIELD l\_runtime\_function.  
GET RUN TIME FIELD l\_runtime\_function.  
  
CALL FUNCTION 'ZPTB00\_CALCULATE\_TRAVEL\_EXPENS'  
  EXPORTING  
    i\_km     = p\_km  
  IMPORTING  
    e\_amount = l\_amount\_function  
  EXCEPTIONS  
    failed   = 1  
    OTHERS   = 2.  
IF sy-subrc <> 0.  
  RETURN.  
ENDIF.  
  
  
\*Write performance results to screen  
WRITE: / 'Runtime of form: ', l\_runtime\_form,  
       / 'Runtime of Function: ', l\_runtime\_function.  
  
WRITE: / 'Runtime of form: ', l\_amount\_form,  
       / 'Runtime of Function: ', l\_amount\_function.  
\*&---------------------------------------------------------------------\*  
\*&      Form  calculate\_travel\_expense  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      -->I\_KM       text  
\*      -->E\_AMOUNT   text  
\*----------------------------------------------------------------------\*  
FORM calculate\_travel\_expense   "Reisekosten berechnen  
  USING  
        i\_km TYPE i  
  CHANGING  
    e\_amount TYPE f.  
  
  e\_amount = i\_km \* '0.3'.  
ENDFORM.                    "calculate\_travel\_expense

,

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PROVISION\_CALCULATOR  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_provision\_calculator.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PROVISION\_CALCULATOR                                                 \*  
\* Report Title : <description>                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 99/99/9999 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.# | Date:99.99.99 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
PARAMETERS:  
\* Start price and Sales price  
p\_astart TYPE p DECIMALS 2,  
p\_asales TYPE p DECIMALS 2.  
DATA:  
\* Net free, net provision, cost and total cost  
l\_fee         TYPE p DECIMALS 2,  
l\_provision   TYPE p DECIMALS 2,  
l\_net\_cost    TYPE p DECIMALS 2,  
l\_total\_cost  TYPE p DECIMALS 2.  
\* Calculate fee by start price  
IF p\_astart <= 0.  
  WRITE: / 'Please enter a valid start price.'.  
  RETURN.  
ELSEIF p\_astart <= 1.  
  l\_fee = '0.25'.  
ELSEIF p\_astart <=  '9.99'.  
  l\_fee = '0.40'.  
ELSEIF p\_astart <= '24.99'.  
  l\_fee = '0.60'.  
ELSEIF p\_astart <= '99.99'.  
  l\_fee = '1.20'.  
ELSE.  
  l\_fee = '2.40'.  
ENDIF.  
\* Calculate provision by sales price  
IF p\_asales < p\_astart.  
  WRITE: / 'Please enter a valid sales price.'.  
  RETURN.  
ELSEIF p\_asales <= 50.  
  l\_provision = p\_asales \* '0.40'.  
ELSEIF p\_asales <= 500.  
  l\_provision = 2 + ( p\_asales - 50 ) \* '0.03'.  
ELSE.  
  l\_provision = '15.50' + ( p\_asales - 500 ) \* '0.015'.  
ENDIF.  
\* Calculate total cost and write them to screen  
l\_net\_cost = l\_fee + l\_provision.  
l\_total\_cost = l\_net\_cost + ( l\_net\_cost \* '0.16' ).  
WRITE: / 'Start price : ', p\_astart,  
       / 'Sales price : ', p\_asales,  
       / 'Net cost : ', l\_net\_cost,  
       / 'Total cost : ', l\_total\_cost.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_SAVINGS\_CALCULATOR  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_savings\_calculator.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_SAVINGS\_CALCULATOR                                                 \*  
\* Report Title : <description>                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 99/99/9999 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.# | Date:99.99.99 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
PARAMETERS:  
\* Savings target, monthley savings and yearly interest rate  
p\_atargt  TYPE p DECIMALS 2,      "Einsparungsziel  
p\_mrate   TYPE p DECIMALS 2,      "Monatlicher Sparbetrag  
p\_iperc   TYPE p DECIMALS 2.      "Bankzinssatz  
  
DATA:  
\* Monthly interest rate, current savings and month  
l\_monthly\_interest\_rate TYPE f,             "Monatlicher Zinssatz  
l\_savings               TYPE p DECIMALS 2,  "Ersparnisse  
l\_month                 TYPE i.             "Monat  
  
\* Calculate Month and savings  
l\_monthly\_interest\_rate = p\_iperc / 12 / 100.  
  
WRITE: / 'Savings target: ',  p\_atargt,  
       / 'Monthly Amount: ',  p\_mrate,  
       / 'Interest Rate %: ', p\_iperc.  
  
WRITE sy-uline AS LINE.  
\* while Version  
WHILE l\_savings < p\_atargt.  
  l\_savings = l\_savings + ( l\_savings \*  
                  l\_monthly\_interest\_rate ) + p\_mrate.  
  l\_month = l\_month + 1.  
ENDWHILE.  
WRITE: / 'While Statement',  
       / 'Month needet: ', l\_month,  
       / 'Saved amount: ', l\_savings.  
\* Do Version  
CLEAR l\_savings.  
CLEAR l\_month.  
DO.  
  l\_savings = l\_savings + ( l\_savings \*  
              l\_monthly\_interest\_rate ) + p\_mrate.  
  l\_month = l\_month + 1.  
  IF l\_savings >= p\_atargt.  
    EXIT.  
  ENDIF.  
ENDDO.  
WRITE: / 'DO Statement',  
       / 'Month needet:', l\_month,  
       / 'Saved amount:', l\_savings.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_ROOM\_CHECKER  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_room\_checker.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_ROOM\_CHECKER                                                 \*  
\* Report Title : <description>                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 99/99/9999 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 203                 \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.09.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
PARAMETERS:  
p\_room    TYPE zptb00\_room\_number OBLIGATORY VALUE CHECK,  
p\_chkin   TYPE d OBLIGATORY,  
p\_chkout  TYPE d OBLIGATORY.  
  
WRITE: / 'Room Reservation',  
       / 'Check in day:', p\_chkin,  
       / 'Check out day: ', p\_chkout.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_HOTEL\_RESERVATION  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_hotel\_reservation.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_HOTEL\_RESERVATION                                                 \*  
\* Report Title : Hotel Reservierung                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/08/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 219                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.09.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
PARAMETERS:  
p\_room    TYPE zptb00\_room\_number OBLIGATORY VALUE CHECK,  
p\_chkin   TYPE zptb00\_checkin,  
p\_chkout  TYPE zptb00\_checkout,  
p\_name    TYPE zptb00\_customer\_name,  
\* Application menue  
p\_add TYPE c RADIOBUTTON GROUP grp1 DEFAULT 'X',  
p\_delete TYPE c RADIOBUTTON GROUP grp1,  
p\_show TYPE c RADIOBUTTON GROUP grp1.  
DATA:  
\* For working with table zptb\_hreservat  
l\_str\_reservation TYPE zptb00\_hreservat,  
l\_tab\_reservation TYPE STANDARD TABLE OF  
          zptb00\_hreservat.  
IF p\_add = 'X'.  
\* Check whether period is free  
  SELECT \* FROM zptb00\_hreservat INTO TABLE l\_tab\_reservation  
    WHERE ( room\_number = p\_room )  
  and ( ( checkin between p\_chkin and p\_chkout )  
        or ( checkout between p\_chkin and p\_chkout ) ).  
  IF sy-dbcnt > 0.  
    WRITE: / 'Period already reserved'.  
  ELSE.  
\* Make Reservation  
    CALL FUNCTION 'GUID\_CREATE'  
     IMPORTING  
\*   EV\_GUID\_16       =  
\*   EV\_GUID\_22       =  
       ev\_guid\_32       = l\_str\_reservation-id  
              .  
    l\_str\_reservation-room\_number = p\_room.  
    l\_str\_reservation-checkin = p\_chkin.  
    l\_str\_reservation-checkout = p\_chkout.  
    l\_str\_reservation-customer\_name = p\_name.  
    INSERT zptb00\_hreservat FROM l\_str\_reservation.  
    WRITE: / 'Reservation made.'.  
  ENDIF.  
ELSEIF p\_delete = 'X'.  
\* Delete reservation  
  DELETE FROM zptb00\_hreservat WHERE room\_number = p\_room  
  AND checkin = p\_chkin AND checkout = p\_chkout.  
  WRITE: / 'Reservation deleted'.  
ELSEIF p\_show = 'X'.  
\*Show Resevations  
  SELECT \* FROM ZPTB00\_HRESERVAT INTO TABLE l\_tab\_reservation  
    ORDER BY room\_number checkin customer\_name.  
  WRITE: / 'Room Reservations'.  
  LOOP AT l\_tab\_reservation INTO l\_str\_reservation.  
  
    WRITE: / 'Room:', l\_str\_reservation-room\_number,  
           / 'Check in: ', l\_str\_reservation-checkin,  
           / 'Check out: ', l\_str\_reservation-checkout,  
           / 'Customer name: ', l\_str\_reservation-customer\_name.  
  ENDLOOP.  
ENDIF.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_HOTEL\_RESERVATION\_COOL  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_hotel\_reservation\_cool.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_HOTEL\_RESERVATION\_COOL                                                 \*  
\* Report Title : Hotel Reservation Cool                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/08/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 236                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:02.09.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
\*Contains ALV Grid structures, needet for variables  
TYPE-POOLS slis.  
\*Main Menue as selection screen  
SELECTION-SCREEN BEGIN OF SCREEN 110.  
SELECTION-SCREEN BEGIN OF BLOCK b110 WITH FRAME TITLE text-001.  
PARAMETERS:  
p\_add TYPE c RADIOBUTTON GROUP grp1 DEFAULT 'X',  
p\_delete TYPE c RADIOBUTTON GROUP grp1,  
p\_show TYPE c RADIOBUTTON GROUP grp1.  
SELECTION-SCREEN END OF BLOCK b110.  
SELECTION-SCREEN END OF SCREEN 110.  
  
SELECTION-SCREEN BEGIN OF SCREEN 120.  
SELECTION-SCREEN  BEGIN OF BLOCK b120 WITH FRAME TITLE text-002.  
PARAMETERS:  
p\_room TYPE zptb00\_room\_number VALUE CHECK,  
p\_chkin TYPE zptb00\_checkin,  
p\_chkout TYPE zptb00\_checkout,  
p\_name TYPE zptb00\_customer\_name.  
SELECTION-SCREEN END OF BLOCK b120.  
SELECTION-SCREEN END OF SCREEN 120.  
  
  
DO.  
  CALL SELECTION-SCREEN 110.  
  IF sy-subrc <> 0.  
    EXIT.  
  ELSEIF p\_add = 'X'.  
    PERFORM zptb00\_hrc\_add.  
  ELSEIF p\_delete = 'X'.  
    PERFORM zptb00\_hrc\_delete.  
  ELSEIF p\_show = 'X'.  
    PERFORM zptb00\_hrc\_show.  
  ENDIF.  
ENDDO.  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  zptb00\_hrc\_add  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
FORM zptb00\_hrc\_add.  
  DATA:  
\*For working with table zptb00\_hreservat.  
  l\_str\_reservation TYPE zptb00\_hreservat,  
  l\_tab\_reservation TYPE STANDARD TABLE OF zptb00\_hreservat.  
  
\* call input dialog  
  CALL SELECTION-SCREEN 120 STARTING AT 5 5.  
  IF sy-subrc = 0.  
\* Check period is free  
    SELECT \* FROM zptb00\_hreservat INTO TABLE l\_tab\_reservation  
      WHERE ( room\_number = p\_room )  
        AND ( ( checkin BETWEEN p\_chkin AND p\_chkout )  
        OR ( checkout BETWEEN p\_chkin AND p\_chkout ) ).  
    IF sy-dbcnt > 0.  
      MESSAGE s000(zptb00\_messages).  
    ELSE.  
\* Make reservation  
      CALL FUNCTION 'GUID\_CREATE'  
       IMPORTING  
\*   EV\_GUID\_16       =  
\*   EV\_GUID\_22       =  
         ev\_guid\_32       = l\_str\_reservation-id.  
  
      l\_str\_reservation-room\_number = p\_room.  
      l\_str\_reservation-checkin = p\_chkin.  
      l\_str\_reservation-checkout = p\_chkout.  
      l\_str\_reservation-customer\_name = p\_name.  
      INSERT zptb00\_hreservat FROM l\_str\_reservation.  
      MESSAGE s001(zptb00\_messages).  
    ENDIF.  
  ENDIF.  
ENDFORM.                    "zptb00\_hrc\_add  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  zptb00\_hrc\_delete  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
FORM zptb00\_hrc\_delete.  
  TYPES:  
\*Builds a structure named t\_str\_reservationx,  
\* containing of the fields Mark, client, id,  
\*ROOM\_NUMBER, CHEKIN, CHEKOUT, CUSTOMER\_NAME:  
\* Needet for alv-Grid-Display.  
  BEGIN OF t\_mark,  
    mark TYPE c,  
    END OF t\_mark,  
    BEGIN OF t\_str\_reservationx.  
          INCLUDE STRUCTURE zptb00\_hreservat.  
  INCLUDE TYPE t\_mark.  
  TYPES:  
  END OF t\_str\_reservationx.  
  DATA:  
\* For ALV Grid-Display  
l\_dis TYPE disvariant,  
l\_str\_layout TYPE slis\_layout\_alv,  
l\_str\_exit\_by\_user TYPE slis\_exit\_by\_user,  
l\_tab\_reservationx TYPE STANDARD TABLE OF  
t\_str\_reservationx,  
l\_str\_reservationx TYPE t\_str\_reservationx,  
\* For Working with table zptb00\_hreservat  
l\_tab\_reservation TYPE STANDARD TABLE OF  
zptb00\_hreservat,  
l\_count TYPE i.  
  FIELD-SYMBOLS:  
\* For Working with table zptb00\_hreservat  
<l\_str\_reservation> TYPE zptb00\_hreservat,  
<l\_str\_reservationx> TYPE t\_str\_reservationx.  
\* Load content of reservation table into memory table  
\*with additional field mark.  
  SELECT \* FROM zptb00\_hreservat INTO TABLE  
    l\_tab\_reservation.  
  LOOP AT l\_tab\_reservation ASSIGNING  
    <l\_str\_reservation>.  
    MOVE-CORRESPONDING <l\_str\_reservation>  
            TO l\_str\_reservationx.  
    APPEND l\_str\_reservationx TO l\_tab\_reservationx.  
  ENDLOOP.  
\* Make colum width as small as possible  
  l\_str\_layout-colwidth\_optimize = 'X'.  
\* This text should be displayed as the window Title  
  l\_str\_layout-window\_titlebar = text-003.  
\*This is the name of the field in the table, wich  
\* ALV-Grid can use to store, whether the user has  
\*marked it or not  
  l\_str\_layout-box\_fieldname = 'MARK'.  
\*give th name of our report, so that the user can save  
\* Individual configurations of the ALV-Grid-Dispaly  
  l\_dis-report = sy-repid.  
  
\*Call alv grid function  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
   EXPORTING  
\*   I\_INTERFACE\_CHECK                 = ' '  
\*   I\_BYPASSING\_BUFFER                = ' '  
\*   I\_BUFFER\_ACTIVE                   = ' '  
     i\_callback\_program                = l\_dis-report  
     i\_callback\_pf\_status\_set          = 'CB\_ALV\_PF\_SET\_STATUS'  
     i\_callback\_user\_command           = 'CB\_ALV\_USER\_COMMAND'  
\*   I\_CALLBACK\_TOP\_OF\_PAGE            = ' '  
\*   I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '  
\*   I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '  
     i\_structure\_name                  = 'ZPTB00\_HRESERVAT'  
\*   I\_BACKGROUND\_ID                   = ' '  
\*   I\_GRID\_TITLE                      =  
\*   I\_GRID\_SETTINGS                   =  
     is\_layout                         = l\_str\_layout  
\*   IT\_FIELDCAT                       =  
\*   IT\_EXCLUDING                      =  
\*   IT\_SPECIAL\_GROUPS                 =  
\*   IT\_SORT                           =  
\*   IT\_FILTER                         =  
\*   IS\_SEL\_HIDE                       =  
     i\_default                         = 'X'  
     i\_save                            = 'A'  
     is\_variant                        = l\_dis  
\*   IT\_EVENTS                         =  
\*   IT\_EVENT\_EXIT                     =  
\*   IS\_PRINT                          =  
\*   IS\_REPREP\_ID                      =  
\*   I\_SCREEN\_START\_COLUMN             = 0  
\*   I\_SCREEN\_START\_LINE               = 0  
\*   I\_SCREEN\_END\_COLUMN               = 0  
\*   I\_SCREEN\_END\_LINE                 = 0  
\*   I\_HTML\_HEIGHT\_TOP                 = 0  
\*   I\_HTML\_HEIGHT\_END                 = 0  
\*   IT\_ALV\_GRAPHICS                   =  
\*   IT\_HYPERLINK                      =  
\*   IT\_ADD\_FIELDCAT                   =  
\*   IT\_EXCEPT\_QINFO                   =  
\*   IR\_SALV\_FULLSCREEN\_ADAPTER        =  
   IMPORTING  
\*   E\_EXIT\_CAUSED\_BY\_CALLER           =  
     es\_exit\_caused\_by\_user            = l\_str\_exit\_by\_user  
    TABLES  
      t\_outtab                          = l\_tab\_reservationx  
   EXCEPTIONS  
     program\_error                     = 1  
     OTHERS                            = 2  
            .  
  IF sy-subrc <> 0.  
\* Error? -> Show to the user what happend  
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  ENDIF.  
  
\*Leave the whole program, if the user pressed exit  
\* Leave back to the main screen, if the user pressed  
\* back or cancel  
  
  CASE 'X'.  
    WHEN l\_str\_exit\_by\_user-exit.  
      LEAVE PROGRAM.  
    WHEN l\_str\_exit\_by\_user-back OR  
         l\_str\_exit\_by\_user-cancel.  
      RETURN.  
  ENDCASE.  
  
\*Otherwise the user must have pressed our delete  
\*Button, so delete all lines in the reservation table.  
\* Wich are marked with an 'X'.  
  l\_count = 0.  
  LOOP AT l\_tab\_reservationx ASSIGNING  
    <l\_str\_reservationx>.  
    IF <l\_str\_reservationx>-mark = 'X'.  
      DELETE FROM zptb00\_hreservat WHERE id =  
      <l\_str\_reservationx>-id.  
    ENDIF.  
  ENDLOOP.  
  MESSAGE s002(zptb00\_messages) WITH l\_count.  
ENDFORM.                    "zptb00\_hrc\_delete  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  zptb00\_hrc\_show  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
FORM zptb00\_hrc\_show.  
  DATA:  
\* for alv Grid Display  
l\_dis TYPE disvariant,  
l\_rda\_table TYPE REF TO data,  
      l\_str\_layout TYPE slis\_layout\_alv,  
\* For working with table zptb00\_hreservat  
l\_tab\_reservation TYPE STANDARD TABLE OF  
zptb00\_hreservat.  
  
\* Load all data of reservation table into memory table  
  SELECT \* FROM zptb00\_hreservat INTO TABLE  
    l\_tab\_reservation.  
\* Make colum width as small as possible  
  l\_str\_layout-colwidth\_optimize = 'X'.  
\*This text should be displayed as the window title  
  l\_str\_layout-window\_titlebar = text-004.  
\* Give the name of our report, so that the user can  
\*Save individual configurations of the alv grid  
\*Display. You could also provide another text here if,  
\*you want seperate configuration of the show and  
\*deleten display.  
  l\_dis-report = sy-repid.  
  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
   EXPORTING  
\*   I\_INTERFACE\_CHECK                 = ' '  
\*   I\_BYPASSING\_BUFFER                = ' '  
\*   I\_BUFFER\_ACTIVE                   = ' '  
\*   I\_CALLBACK\_PROGRAM                = ' '  
\*   I\_CALLBACK\_PF\_STATUS\_SET          = ' '  
\*   I\_CALLBACK\_USER\_COMMAND           = ' '  
\*   I\_CALLBACK\_TOP\_OF\_PAGE            = ' '  
\*   I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '  
\*   I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '  
   i\_structure\_name                  = 'ZPTB00\_HRESERVAT'  
\*   I\_BACKGROUND\_ID                   = ' '  
\*   I\_GRID\_TITLE                      =  
\*   I\_GRID\_SETTINGS                   =  
     is\_layout                         = l\_str\_layout  
\*   IT\_FIELDCAT                       =  
\*   IT\_EXCLUDING                      =  
\*   IT\_SPECIAL\_GROUPS                 =  
\*   IT\_SORT                           =  
\*   IT\_FILTER                         =  
\*   IS\_SEL\_HIDE                       =  
\*   I\_DEFAULT                         = 'X'  
     i\_save                            = 'X'  
     is\_variant                        = l\_dis  
\*   IT\_EVENTS                         =  
\*   IT\_EVENT\_EXIT                     =  
\*   IS\_PRINT                          =  
\*   IS\_REPREP\_ID                      =  
\*   I\_SCREEN\_START\_COLUMN             = 0  
\*   I\_SCREEN\_START\_LINE               = 0  
\*   I\_SCREEN\_END\_COLUMN               = 0  
\*   I\_SCREEN\_END\_LINE                 = 0  
\*   I\_HTML\_HEIGHT\_TOP                 = 0  
\*   I\_HTML\_HEIGHT\_END                 = 0  
\*   IT\_ALV\_GRAPHICS                   =  
\*   IT\_HYPERLINK                      =  
\*   IT\_ADD\_FIELDCAT                   =  
\*   IT\_EXCEPT\_QINFO                   =  
\*   IR\_SALV\_FULLSCREEN\_ADAPTER        =  
\* IMPORTING  
\*   E\_EXIT\_CAUSED\_BY\_CALLER           =  
\*   ES\_EXIT\_CAUSED\_BY\_USER            =  
    TABLES  
      t\_outtab                          = l\_tab\_reservation  
   EXCEPTIONS  
     program\_error                     = 1  
     OTHERS                            = 2  
            .  
  IF sy-subrc <> 0.  
\* Error? -> Show to the user what happend  
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  ENDIF.  
  
  
  
ENDFORM.                    "zptb00\_hrc\_show  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  cd\_alv\_user\_command  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      -->R\_UCOMM      text  
\*      -->RS\_SELFIELD  text  
\*----------------------------------------------------------------------\*  
FORM cb\_alv\_user\_command  
  USING  
        r\_ucomm LIKE sy-ucomm  
  rs\_selfield TYPE slis\_selfield.                           "#EC called  
  
\*Delete button pressed? -> then force to leave the  
\* grid display now by switching on the exit button  
  IF r\_ucomm = 'EXECUTE'.  
    rs\_selfield-exit = 'X'.  
  ENDIF.  
ENDFORM.                    "cd\_alv\_user\_command  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  cb\_alv\_pf\_set\_status  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      -->RT\_EXTAB   text  
\*----------------------------------------------------------------------\*  
FORM cb\_alv\_pf\_set\_status  
  USING  
  rt\_extab TYPE slis\_t\_extab.     "#ec called  
\* take our pf-status insteed of your own  
  SET PF-STATUS 'STANDARD\_FULLSCREEN' EXCLUDING  
  rt\_extab.  
ENDFORM.                    "cb\_alv\_pf\_set\_status

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*   System-defined Include-files.                                 \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  INCLUDE LZPTB00\_OBJ\_BTATOP.                " Global Data  
  INCLUDE LZPTB00\_OBJ\_BTAUXX.                " Function Modules  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*   User-defined Include-files (if necessary).                    \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* INCLUDE LZPTB00\_OBJ\_BTAF...                " Subprograms  
\* INCLUDE LZPTB00\_OBJ\_BTAO...                " PBO-Modules  
\* INCLUDE LZPTB00\_OBJ\_BTAI...                " PAI-Modules  
  
\*&SPWizard: Include inserted by SP Wizard. DO NOT CHANGE THIS LINE!  
INCLUDE LZPTB00\_OBJ\_BTAO01 .  
INCLUDE LZPTB00\_OBJ\_BTAI01 .  
INCLUDE LZPTB00\_OBJ\_BTAF01 .

FUNCTION-POOL zptb00\_obj\_bta.               "MESSAGE-ID ..  
  
  
  
DATA: g\_str\_bta TYPE zptb00\_str\_bta.  
\*\*\*&SPWIZARD: DATA DECLARATION FOR TABLECONTROL 'POSITION\_TABLE'  
\*&SPWIZARD: DEFINITION OF DDIC-TABLE  
TABLES:   ZPTB00\_BTAITEM.  
  
\*&SPWIZARD: TYPE FOR THE DATA OF TABLECONTROL 'POSITION\_TABLE'  
TYPES: BEGIN OF T\_POSITION\_TABLE,  
         PR\_ID LIKE ZPTB00\_BTAITEM-PR\_ID,  
         PSTYPE LIKE ZPTB00\_BTAITEM-PSTYPE,  
         WH\_ID LIKE ZPTB00\_BTAITEM-WH\_ID,  
         AMOUNT LIKE ZPTB00\_BTAITEM-AMOUNT,  
         CURRENCY LIKE ZPTB00\_BTAITEM-CURRENCY,  
         QUANTITY LIKE ZPTB00\_BTAITEM-QUANTITY,  
         UNIT LIKE ZPTB00\_BTAITEM-UNIT,  
       END OF T\_POSITION\_TABLE.  
  
\*&SPWIZARD: INTERNAL TABLE FOR TABLECONTROL 'POSITION\_TABLE'  
DATA:     G\_POSITION\_TABLE\_ITAB   TYPE T\_POSITION\_TABLE OCCURS 0,  
          G\_POSITION\_TABLE\_WA     TYPE T\_POSITION\_TABLE. "work area  
DATA:     G\_POSITION\_TABLE\_COPIED.           "copy flag  
  
\*&SPWIZARD: DECLARATION OF TABLECONTROL 'POSITION\_TABLE' ITSELF  
CONTROLS: POSITION\_TABLE TYPE TABLEVIEW USING SCREEN 0100.  
  
\*&SPWIZARD: LINES OF TABLECONTROL 'POSITION\_TABLE'  
DATA:     G\_POSITION\_TABLE\_LINES  LIKE SY-LOOPC.  
  
DATA:     OK\_CODE LIKE SY-UCOMM.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*   THIS FILE IS GENERATED BY THE FUNCTION LIBRARY.             \*  
\*   NEVER CHANGE IT MANUALLY, PLEASE!                           \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
INCLUDE LZPTB00\_OBJ\_BTAU01.  
                    "ZPTB00\_OBJ\_BTA\_EDIT  
INCLUDE LZPTB00\_OBJ\_BTAU02.  
                    "ZPTB00\_OBJ\_BTA\_SAVE  
INCLUDE LZPTB00\_OBJ\_BTAU03.  
                    "ZPTB00\_OBJ\_BTA\_ENRICH  
INCLUDE LZPTB00\_OBJ\_BTAU04.  
                    "ZPTB00\_OBJ\_BTA\_LOAD  
INCLUDE LZPTB00\_OBJ\_BTAU05.  
                    "ZPTB00\_OBJ\_BTA\_MARK\_AS\_BOOKED

\*&---------------------------------------------------------------------\*  
\*&  Include           LZPTB00\_OBJ\_BTAO01  
\*&---------------------------------------------------------------------\*  
\*ABAP Basics S. 303  
  
\*&SPWIZARD: OUTPUT MODULE FOR TC 'POSITION\_TABLE'. DO NOT CHANGE THIS LI  
\*&SPWIZARD: COPY DDIC-TABLE TO ITAB  
MODULE POSITION\_TABLE\_INIT OUTPUT.  
  IF G\_POSITION\_TABLE\_COPIED IS INITIAL.  
\*&SPWIZARD: COPY DDIC-TABLE 'ZPTB00\_BTAITEM'  
\*&SPWIZARD: INTO INTERNAL TABLE 'g\_POSITION\_TABLE\_itab'  
    SELECT \* FROM ZPTB00\_BTAITEM  
       INTO CORRESPONDING FIELDS  
       OF TABLE G\_POSITION\_TABLE\_ITAB.  
    G\_POSITION\_TABLE\_COPIED = 'X'.  
    REFRESH CONTROL 'POSITION\_TABLE' FROM SCREEN '0100'.  
  ENDIF.  
ENDMODULE.  
  
\*&SPWIZARD: OUTPUT MODULE FOR TC 'POSITION\_TABLE'. DO NOT CHANGE THIS LI  
\*&SPWIZARD: MOVE ITAB TO DYNPRO  
MODULE POSITION\_TABLE\_MOVE OUTPUT.  
  MOVE-CORRESPONDING G\_POSITION\_TABLE\_WA TO ZPTB00\_BTAITEM.  
ENDMODULE.  
  
\*&SPWIZARD: OUTPUT MODULE FOR TC 'POSITION\_TABLE'. DO NOT CHANGE THIS LI  
\*&SPWIZARD: GET LINES OF TABLECONTROL  
MODULE POSITION\_TABLE\_GET\_LINES OUTPUT.  
  G\_POSITION\_TABLE\_LINES = SY-LOOPC.  
ENDMODULE.  
  
\*&---------------------------------------------------------------------\*  
\*&      Module  STATUS\_0100  OUTPUT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
MODULE status\_0100 OUTPUT.  
  DATA: g\_header\_copied TYPE c.  
  
  SET PF-STATUS 'ZPTB00\_OBJ\_BTA\_STD'.  
  SET TITLEBAR 'ZPTB00\_OBJ\_BTA\_STD'.  
  
  
  IF g\_header\_copied IS INITIAL.  
\* Assign header data to appropriate controls  
    zptb00\_str\_bta-bp\_id  = g\_str\_bta-bp\_id.  
    zptb00\_str\_bta-bttype = g\_str\_bta-bttype.  
    zptb00\_str\_bta-btdate = g\_str\_bta-btdate.  
    g\_header\_copied = 'X'.  
  ENDIF.  
  
  
ENDMODULE.                 " STATUS\_0100  OUTPUT

\*&---------------------------------------------------------------------\*  
\*&  Include           LZPTB00\_OBJ\_BTAI01  
\*&---------------------------------------------------------------------\*  
\*ABAP Basics S. 303  
  
\*&SPWIZARD: INPUT MODULE FOR TC 'POSITION\_TABLE'. DO NOT CHANGE THIS LIN  
\*&SPWIZARD: MODIFY TABLE  
MODULE POSITION\_TABLE\_MODIFY INPUT.  
  MOVE-CORRESPONDING ZPTB00\_BTAITEM TO G\_POSITION\_TABLE\_WA.  
  MODIFY G\_POSITION\_TABLE\_ITAB  
    FROM G\_POSITION\_TABLE\_WA  
    INDEX POSITION\_TABLE-CURRENT\_LINE.  
ENDMODULE.  
  
\*&SPWIZARD: INPUT MODULE FOR TC 'POSITION\_TABLE'. DO NOT CHANGE THIS LIN  
\*&SPWIZARD: PROCESS USER COMMAND  
MODULE POSITION\_TABLE\_USER\_COMMAND INPUT.  
  OK\_CODE = SY-UCOMM.  
  PERFORM USER\_OK\_TC USING    'POSITION\_TABLE'  
                              'G\_POSITION\_TABLE\_ITAB'  
                              'FLAG'  
                     CHANGING OK\_CODE.  
  SY-UCOMM = OK\_CODE.  
ENDMODULE.  
  
\*&---------------------------------------------------------------------\*  
\*&      Module  USER\_COMMAND\_0100  INPUT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
MODULE user\_command\_0100 INPUT.  
  DATA:  
        g\_answer   TYPE c,  
        g\_index    TYPE sy-tabix,  
        g\_str\_item TYPE zptb00\_btaitem.  
\*Save\_ok like ok\_code  
  FIELD-SYMBOLS:  
\*Position checking  
  <position\_str> TYPE t\_position\_table.  
  IF ok\_code = 'BACK' OR ok\_code = 'QUIT'.  
  
    CALL FUNCTION 'POPUP\_TO\_CONFIRM\_STEP'  
      EXPORTING  
\*   DEFAULTOPTION        = 'Y'  
        textline1            = text-003  
\*   TEXTLINE2            = ' '  
        titel                = text-001  
\*   START\_COLUMN         = 25  
\*   START\_ROW            = 6  
\*   CANCEL\_DISPLAY       = 'X'  
     IMPORTING  
       answer               = g\_answer.  
  
    IF g\_answer = 'N'.  
      CLEAR g\_str\_bta.  
      LEAVE TO SCREEN 0.  
    ELSEIF g\_answer = 'J'.  
      ok\_code = 'SAVE'.  
    ENDIF.  
  ENDIF.  
  IF ok\_code = 'SAVE'.  
\* check header data  
    IF zptb00\_str\_bta-bttype IS INITIAL.  
      MESSAGE i006(zptb00\_bta) WITH g\_index.  
      RETURN.  
    ENDIF.  
    IF zptb00\_str\_bta-btdate IS INITIAL.  
      MESSAGE i007(zptb00\_bta) WITH g\_index.  
      RETURN.  
    ENDIF.  
\*Check position data  
    LOOP AT g\_position\_table\_itab ASSIGNING  
      <position\_str>.  
      g\_index = sy-tabix.  
      IF  <position\_str>-pr\_id IS INITIAL.  
        MESSAGE i000(zptb00\_bta) WITH g\_index.  
        RETURN.  
      ENDIF.  
      IF <position\_str>-pstype IS INITIAL.  
        MESSAGE i001(zptb00\_bta) WITH g\_index.  
        RETURN.  
      ENDIF.  
      IF <position\_str>-amount IS INITIAL.  
        MESSAGE i002(zptb00\_bta) WITH g\_index.  
        RETURN.  
      ENDIF.  
      IF <position\_str>-currency IS INITIAL.  
        MESSAGE i003(zptb00\_bta) WITH g\_index.  
        RETURN.  
      ENDIF.  
      IF <position\_str>-quantity IS INITIAL.  
        MESSAGE i004(zptb00\_bta) WITH g\_index.  
        RETURN.  
      ENDIF.  
      IF <position\_str>-unit IS INITIAL.  
        MESSAGE i005(zptb00\_bta) WITH g\_index.  
        RETURN.  
      ENDIF.  
\* put header data into communication structure  
\* g\_str\_bta  
      IF g\_str\_bta-id IS INITIAL.  
        CALL FUNCTION 'GUID\_CREATE'  
          IMPORTING  
            ev\_guid\_32 = g\_str\_bta-id.  
      ENDIF.  
      g\_str\_bta-bp\_id  = zptb00\_btaheader-bp\_id.  
      g\_str\_bta-bttype = zptb00\_btaheader-bttype.  
      g\_str\_bta-btdate = zptb00\_btaheader-btdate.  
\*Put position data into communication structure  
\* g\_str\_bta  
  
      READ TABLE g\_str\_bta-tab\_item INDEX g\_index INTO  
      g\_str\_item.  
      IF sy-subrc <> 0.  
        CALL FUNCTION 'GUID\_CREATE'  
          IMPORTING  
            ev\_guid\_32 = g\_str\_bta-id.  
        g\_str\_item-header\_id = g\_str\_item-id.  
        MOVE-CORRESPONDING <position\_str>  
        TO g\_str\_item.  
        APPEND g\_str\_item TO g\_str\_bta-tab\_item.  
      ELSE.  
        IF g\_str\_item IS INITIAL.  
          CALL FUNCTION 'GUID\_CREATE'  
            IMPORTING  
              ev\_guid\_32 = g\_str\_bta-id.  
          g\_str\_item-header\_id = g\_str\_bta-id.  
        ENDIF.  
        MOVE-CORRESPONDING <position\_str> TO  
        g\_str\_item.  
        MODIFY g\_str\_bta-tab\_item FROM g\_str\_item INDEX  
        g\_index.  
      ENDIF.  
    ENDLOOP.  
    LEAVE TO SCREEN 0.  
  ENDIF.  
  CLEAR ok\_code.  
ENDMODULE.                 " USER\_COMMAND\_0100  INPUT  
\*&---------------------------------------------------------------------\*  
\*&      Module  user\_exit\_0100  INPUT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
MODULE user\_exit\_0100 INPUT.  
\* has the user typed in any data?...  
  IF sy-datar = 'X'.  
\* ...then we have to conf the cancel process  
    CALL FUNCTION 'POPUP\_TO\_CONFIRM\_LOSS\_OF\_DATA'  
      EXPORTING  
        textline1           = text-002  
\*   TEXTLINE2           = ' '  
        titel               = text-001  
\*   START\_COLUMN        = 25  
\*   START\_ROW           = 6  
\*   DEFAULTOPTION       = 'N'  
     IMPORTING  
       answer              = g\_answer  
              .  
  ELSE.  
\*Otherwise we need no confirmation  
    g\_answer = 'J'.  
  ENDIF.  
  IF g\_answer = 'J'.  
    CLEAR g\_str\_bta.  
    LEAVE TO SCREEN 0.  
  ENDIF.  
  
  
ENDMODULE.                 " user\_exit\_0100  INPUT

\*&---------------------------------------------------------------------\*  
\*&  Include           LZPTB00\_OBJ\_BTAF01  
\*&---------------------------------------------------------------------\*  
  
\*----------------------------------------------------------------------\*  
\*   INCLUDE TABLECONTROL\_FORMS                                         \*  
\*----------------------------------------------------------------------\*  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  USER\_OK\_TC                                               \*  
\*&---------------------------------------------------------------------\*  
 FORM USER\_OK\_TC USING    P\_TC\_NAME TYPE DYNFNAM  
                          P\_TABLE\_NAME  
                          P\_MARK\_NAME  
                 CHANGING P\_OK      LIKE SY-UCOMM.  
  
\*&SPWIZARD: BEGIN OF LOCAL DATA----------------------------------------\*  
   DATA: L\_OK              TYPE SY-UCOMM,  
         L\_OFFSET          TYPE I.  
\*&SPWIZARD: END OF LOCAL DATA------------------------------------------\*  
  
\*&SPWIZARD: Table control specific operations                          \*  
\*&SPWIZARD: evaluate TC name and operations                            \*  
   SEARCH P\_OK FOR P\_TC\_NAME.  
   IF SY-SUBRC <> 0.  
     EXIT.  
   ENDIF.  
   L\_OFFSET = STRLEN( P\_TC\_NAME ) + 1.  
   L\_OK = P\_OK+L\_OFFSET.  
\*&SPWIZARD: execute general and TC specific operations                 \*  
   CASE L\_OK.  
     WHEN 'INSR'.                      "insert row  
       PERFORM FCODE\_INSERT\_ROW USING    P\_TC\_NAME  
                                         P\_TABLE\_NAME.  
       CLEAR P\_OK.  
  
     WHEN 'DELE'.                      "delete row  
       PERFORM FCODE\_DELETE\_ROW USING    P\_TC\_NAME  
                                         P\_TABLE\_NAME  
                                         P\_MARK\_NAME.  
       CLEAR P\_OK.  
  
     WHEN 'P--' OR                     "top of list  
          'P-'  OR                     "previous page  
          'P+'  OR                     "next page  
          'P++'.                       "bottom of list  
       PERFORM COMPUTE\_SCROLLING\_IN\_TC USING P\_TC\_NAME  
                                             L\_OK.  
       CLEAR P\_OK.  
\*     WHEN 'L--'.                       "total left  
\*       PERFORM FCODE\_TOTAL\_LEFT USING P\_TC\_NAME.  
\*  
\*     WHEN 'L-'.                        "column left  
\*       PERFORM FCODE\_COLUMN\_LEFT USING P\_TC\_NAME.  
\*  
\*     WHEN 'R+'.                        "column right  
\*       PERFORM FCODE\_COLUMN\_RIGHT USING P\_TC\_NAME.  
\*  
\*     WHEN 'R++'.                       "total right  
\*       PERFORM FCODE\_TOTAL\_RIGHT USING P\_TC\_NAME.  
\*  
     WHEN 'MARK'.                      "mark all filled lines  
       PERFORM FCODE\_TC\_MARK\_LINES USING P\_TC\_NAME  
                                         P\_TABLE\_NAME  
                                         P\_MARK\_NAME   .  
       CLEAR P\_OK.  
  
     WHEN 'DMRK'.                      "demark all filled lines  
       PERFORM FCODE\_TC\_DEMARK\_LINES USING P\_TC\_NAME  
                                           P\_TABLE\_NAME  
                                           P\_MARK\_NAME .  
       CLEAR P\_OK.  
  
\*     WHEN 'SASCEND'   OR  
\*          'SDESCEND'.                  "sort column  
\*       PERFORM FCODE\_SORT\_TC USING P\_TC\_NAME  
\*                                   l\_ok.  
  
   ENDCASE.  
  
 ENDFORM.                              " USER\_OK\_TC  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  FCODE\_INSERT\_ROW                                         \*  
\*&---------------------------------------------------------------------\*  
 FORM fcode\_insert\_row  
               USING    P\_TC\_NAME           TYPE DYNFNAM  
                        P\_TABLE\_NAME             .  
  
\*&SPWIZARD: BEGIN OF LOCAL DATA----------------------------------------\*  
   DATA L\_LINES\_NAME       LIKE FELD-NAME.  
   DATA L\_SELLINE          LIKE SY-STEPL.  
   DATA L\_LASTLINE         TYPE I.  
   DATA L\_LINE             TYPE I.  
   DATA L\_TABLE\_NAME       LIKE FELD-NAME.  
   FIELD-SYMBOLS <TC>                 TYPE CXTAB\_CONTROL.  
   FIELD-SYMBOLS <TABLE>              TYPE STANDARD TABLE.  
   FIELD-SYMBOLS <LINES>              TYPE I.  
\*&SPWIZARD: END OF LOCAL DATA------------------------------------------\*  
  
   ASSIGN (P\_TC\_NAME) TO <TC>.  
  
\*&SPWIZARD: get the table, which belongs to the tc                     \*  
   CONCATENATE P\_TABLE\_NAME '[]' INTO L\_TABLE\_NAME. "table body  
   ASSIGN (L\_TABLE\_NAME) TO <TABLE>.                "not headerline  
  
\*&SPWIZARD: get looplines of TableControl                              \*  
   CONCATENATE 'G\_' P\_TC\_NAME '\_LINES' INTO L\_LINES\_NAME.  
   ASSIGN (L\_LINES\_NAME) TO <LINES>.  
  
\*&SPWIZARD: get current line                                           \*  
   GET CURSOR LINE L\_SELLINE.  
   IF SY-SUBRC <> 0.                   " append line to table  
     L\_SELLINE = <TC>-LINES + 1.  
\*&SPWIZARD: set top line                                               \*  
     IF L\_SELLINE > <LINES>.  
       <TC>-TOP\_LINE = L\_SELLINE - <LINES> + 1 .  
     ELSE.  
       <TC>-TOP\_LINE = 1.  
     ENDIF.  
   ELSE.                               " insert line into table  
     L\_SELLINE = <TC>-TOP\_LINE + L\_SELLINE - 1.  
     L\_LASTLINE = <TC>-TOP\_LINE + <LINES> - 1.  
   ENDIF.  
\*&SPWIZARD: set new cursor line                                        \*  
   L\_LINE = L\_SELLINE - <TC>-TOP\_LINE + 1.  
  
\*&SPWIZARD: insert initial line                                        \*  
   INSERT INITIAL LINE INTO <TABLE> INDEX L\_SELLINE.  
   <TC>-LINES = <TC>-LINES + 1.  
\*&SPWIZARD: set cursor                                                 \*  
   SET CURSOR LINE L\_LINE.  
  
 ENDFORM.                              " FCODE\_INSERT\_ROW  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  FCODE\_DELETE\_ROW                                         \*  
\*&---------------------------------------------------------------------\*  
 FORM fcode\_delete\_row  
               USING    P\_TC\_NAME           TYPE DYNFNAM  
                        P\_TABLE\_NAME  
                        P\_MARK\_NAME   .  
  
\*&SPWIZARD: BEGIN OF LOCAL DATA----------------------------------------\*  
   DATA L\_TABLE\_NAME       LIKE FELD-NAME.  
  
   FIELD-SYMBOLS <TC>         TYPE cxtab\_control.  
   FIELD-SYMBOLS <TABLE>      TYPE STANDARD TABLE.  
   FIELD-SYMBOLS <WA>.  
   FIELD-SYMBOLS <MARK\_FIELD>.  
\*&SPWIZARD: END OF LOCAL DATA------------------------------------------\*  
  
   ASSIGN (P\_TC\_NAME) TO <TC>.  
  
\*&SPWIZARD: get the table, which belongs to the tc                     \*  
   CONCATENATE P\_TABLE\_NAME '[]' INTO L\_TABLE\_NAME. "table body  
   ASSIGN (L\_TABLE\_NAME) TO <TABLE>.                "not headerline  
  
\*&SPWIZARD: delete marked lines                                        \*  
   DESCRIBE TABLE <TABLE> LINES <TC>-LINES.  
  
   LOOP AT <TABLE> ASSIGNING <WA>.  
  
\*&SPWIZARD: access to the component 'FLAG' of the table header         \*  
     ASSIGN COMPONENT P\_MARK\_NAME OF STRUCTURE <WA> TO <MARK\_FIELD>.  
  
     IF <MARK\_FIELD> = 'X'.  
       DELETE <TABLE> INDEX SYST-TABIX.  
       IF SY-SUBRC = 0.  
         <TC>-LINES = <TC>-LINES - 1.  
       ENDIF.  
     ENDIF.  
   ENDLOOP.  
  
 ENDFORM.                              " FCODE\_DELETE\_ROW  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  COMPUTE\_SCROLLING\_IN\_TC  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*      -->P\_TC\_NAME  name of tablecontrol  
\*      -->P\_OK       ok code  
\*----------------------------------------------------------------------\*  
 FORM COMPUTE\_SCROLLING\_IN\_TC USING    P\_TC\_NAME  
                                       P\_OK.  
\*&SPWIZARD: BEGIN OF LOCAL DATA----------------------------------------\*  
   DATA L\_TC\_NEW\_TOP\_LINE     TYPE I.  
   DATA L\_TC\_NAME             LIKE FELD-NAME.  
   DATA L\_TC\_LINES\_NAME       LIKE FELD-NAME.  
   DATA L\_TC\_FIELD\_NAME       LIKE FELD-NAME.  
  
   FIELD-SYMBOLS <TC>         TYPE cxtab\_control.  
   FIELD-SYMBOLS <LINES>      TYPE I.  
\*&SPWIZARD: END OF LOCAL DATA------------------------------------------\*  
  
   ASSIGN (P\_TC\_NAME) TO <TC>.  
\*&SPWIZARD: get looplines of TableControl                              \*  
   CONCATENATE 'G\_' P\_TC\_NAME '\_LINES' INTO L\_TC\_LINES\_NAME.  
   ASSIGN (L\_TC\_LINES\_NAME) TO <LINES>.  
  
  
\*&SPWIZARD: is no line filled?                                         \*  
   IF <TC>-LINES = 0.  
\*&SPWIZARD: yes, ...                                                   \*  
     L\_TC\_NEW\_TOP\_LINE = 1.  
   ELSE.  
\*&SPWIZARD: no, ...                                                    \*  
     CALL FUNCTION 'SCROLLING\_IN\_TABLE'  
          EXPORTING  
               ENTRY\_ACT             = <TC>-TOP\_LINE  
               ENTRY\_FROM            = 1  
               ENTRY\_TO              = <TC>-LINES  
               LAST\_PAGE\_FULL        = 'X'  
               LOOPS                 = <LINES>  
               OK\_CODE               = P\_OK  
               OVERLAPPING           = 'X'  
          IMPORTING  
               ENTRY\_NEW             = L\_TC\_NEW\_TOP\_LINE  
          EXCEPTIONS  
\*              NO\_ENTRY\_OR\_PAGE\_ACT  = 01  
\*              NO\_ENTRY\_TO           = 02  
\*              NO\_OK\_CODE\_OR\_PAGE\_GO = 03  
               OTHERS                = 0.  
   ENDIF.  
  
\*&SPWIZARD: get actual tc and column                                   \*  
   GET CURSOR FIELD L\_TC\_FIELD\_NAME  
              AREA  L\_TC\_NAME.  
  
   IF SYST-SUBRC = 0.  
     IF L\_TC\_NAME = P\_TC\_NAME.  
\*&SPWIZARD: et actual column                                           \*  
       SET CURSOR FIELD L\_TC\_FIELD\_NAME LINE 1.  
     ENDIF.  
   ENDIF.  
  
\*&SPWIZARD: set the new top line                                       \*  
   <TC>-TOP\_LINE = L\_TC\_NEW\_TOP\_LINE.  
  
  
 ENDFORM.                              " COMPUTE\_SCROLLING\_IN\_TC  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  FCODE\_TC\_MARK\_LINES  
\*&---------------------------------------------------------------------\*  
\*       marks all TableControl lines  
\*----------------------------------------------------------------------\*  
\*      -->P\_TC\_NAME  name of tablecontrol  
\*----------------------------------------------------------------------\*  
FORM FCODE\_TC\_MARK\_LINES USING P\_TC\_NAME  
                               P\_TABLE\_NAME  
                               P\_MARK\_NAME.  
\*&SPWIZARD: EGIN OF LOCAL DATA-----------------------------------------\*  
  DATA L\_TABLE\_NAME       LIKE FELD-NAME.  
  
  FIELD-SYMBOLS <TC>         TYPE cxtab\_control.  
  FIELD-SYMBOLS <TABLE>      TYPE STANDARD TABLE.  
  FIELD-SYMBOLS <WA>.  
  FIELD-SYMBOLS <MARK\_FIELD>.  
\*&SPWIZARD: END OF LOCAL DATA------------------------------------------\*  
  
  ASSIGN (P\_TC\_NAME) TO <TC>.  
  
\*&SPWIZARD: get the table, which belongs to the tc                     \*  
   CONCATENATE P\_TABLE\_NAME '[]' INTO L\_TABLE\_NAME. "table body  
   ASSIGN (L\_TABLE\_NAME) TO <TABLE>.                "not headerline  
  
\*&SPWIZARD: mark all filled lines                                      \*  
  LOOP AT <TABLE> ASSIGNING <WA>.  
  
\*&SPWIZARD: access to the component 'FLAG' of the table header         \*  
     ASSIGN COMPONENT P\_MARK\_NAME OF STRUCTURE <WA> TO <MARK\_FIELD>.  
  
     <MARK\_FIELD> = 'X'.  
  ENDLOOP.  
ENDFORM.                                          "fcode\_tc\_mark\_lines  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  FCODE\_TC\_DEMARK\_LINES  
\*&---------------------------------------------------------------------\*  
\*       demarks all TableControl lines  
\*----------------------------------------------------------------------\*  
\*      -->P\_TC\_NAME  name of tablecontrol  
\*----------------------------------------------------------------------\*  
FORM FCODE\_TC\_DEMARK\_LINES USING P\_TC\_NAME  
                                 P\_TABLE\_NAME  
                                 P\_MARK\_NAME .  
\*&SPWIZARD: BEGIN OF LOCAL DATA----------------------------------------\*  
  DATA L\_TABLE\_NAME       LIKE FELD-NAME.  
  
  FIELD-SYMBOLS <TC>         TYPE cxtab\_control.  
  FIELD-SYMBOLS <TABLE>      TYPE STANDARD TABLE.  
  FIELD-SYMBOLS <WA>.  
  FIELD-SYMBOLS <MARK\_FIELD>.  
\*&SPWIZARD: END OF LOCAL DATA------------------------------------------\*  
  
  ASSIGN (P\_TC\_NAME) TO <TC>.  
  
\*&SPWIZARD: get the table, which belongs to the tc                     \*  
   CONCATENATE P\_TABLE\_NAME '[]' INTO L\_TABLE\_NAME. "table body  
   ASSIGN (L\_TABLE\_NAME) TO <TABLE>.                "not headerline  
  
\*&SPWIZARD: demark all filled lines                                    \*  
  LOOP AT <TABLE> ASSIGNING <WA>.  
  
\*&SPWIZARD: access to the component 'FLAG' of the table header         \*  
     ASSIGN COMPONENT P\_MARK\_NAME OF STRUCTURE <WA> TO <MARK\_FIELD>.  
  
     <MARK\_FIELD> = SPACE.  
  ENDLOOP.  
ENDFORM.                                          "fcode\_tc\_mark\_lines

FUNCTION zptb00\_obj\_bta\_edit.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  CHANGING  
\*"     REFERENCE(C\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA OPTIONAL  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 431  
  
  DATA: l\_str\_item TYPE zptb00\_btaitem.  
  
  IF c\_str\_bta IS INITIAL.  
\*Just add one line to the position by default.  
    APPEND l\_str\_item TO c\_str\_bta-tab\_item.  
  ENDIF.  
\*Make manual posting global for dynpro.  
  MOVE-CORRESPONDING c\_str\_bta TO g\_str\_bta.  
\* Call edit screen  
  CALL SCREEN 0100.  
  
  MOVE-CORRESPONDING g\_str\_bta TO c\_str\_bta.  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_enrich.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  CHANGING  
\*"     REFERENCE(C\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 433  
  
  FIELD-SYMBOLS:  
                 <l\_str\_item> TYPE zptb00\_btaitem.  
  
  c\_str\_bta-client = sy-mandt.  
  
  CALL FUNCTION 'GUID\_CREATE'  
   IMPORTING  
\*   EV\_GUID\_16       =  
\*   EV\_GUID\_22       =  
     ev\_guid\_32       = c\_str\_bta-id.  
  c\_str\_bta-booked = '1'.  
  
\*enrich items.  
  
  LOOP AT c\_str\_bta-tab\_item ASSIGNING <l\_str\_item>.  
    <l\_str\_item>-client = sy-mandt.  
  
    CALL FUNCTION 'GUID\_CREATE'  
     IMPORTING  
\*   EV\_GUID\_16       =  
\*   EV\_GUID\_22       =  
       ev\_guid\_32       = <l\_str\_item>-id.  
    <l\_str\_item>-header\_id = c\_str\_bta-id.  
    <l\_str\_item>-mark = ' '.  
  ENDLOOP.  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_load.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_BOOKED) TYPE  ZPTB00\_DTE\_BTA\_BOOKED DEFAULT '1'  
\*"  EXPORTING  
\*"     REFERENCE(E\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 431  
  
  DATA:  
\*db  
        l\_tab\_bta\_header TYPE STANDARD TABLE OF zptb00\_btaheader,  
        l\_tab\_bta\_item TYPE STANDARD TABLE OF zptb00\_btaitem,  
\*local  
        l\_str\_bta TYPE zptb00\_str\_bta.  
  
  FIELD-SYMBOLS:  
\*db  
                 <l\_str\_bta\_header> TYPE zptb00\_btaheader,  
                 <l\_str\_bta\_item> TYPE zptb00\_btaitem.  
\* read the header data  
  SELECT \* FROM zptb00\_btaheader INTO TABLE  
    l\_tab\_bta\_header  
    WHERE booked = i\_booked.  
  
  IF sy-dbcnt = 0.  
    RAISE failed.  
  ENDIF.  
  
\*read the item data. APAB statement "For all entries" only works  
\* up to ca. 1700 entries so we better loop.  
  
  LOOP AT l\_tab\_bta\_header ASSIGNING <l\_str\_bta\_header>.  
    MOVE-CORRESPONDING <l\_str\_bta\_header> TO l\_str\_bta.  
    SELECT \* FROM zptb00\_btaitem INTO TABLE l\_tab\_bta\_item  
      WHERE header\_id = <l\_str\_bta\_header>-id.  
    l\_str\_bta-tab\_item = l\_tab\_bta\_item.  
    APPEND l\_str\_bta TO e\_tab\_bta.  
  ENDLOOP.  
  IF sy-dbcnt = 0.  
    RAISE failed.  
  ENDIF.  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_mark\_as\_booked.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 435  
  
  FIELD-SYMBOLS:  
                 <l\_str\_bta> TYPE zptb00\_str\_bta.  
  
  LOOP AT i\_tab\_bta ASSIGNING <l\_str\_bta>.  
    UPDATE zptb00\_btaheader SET booked = '2'  
    WHERE id = <l\_str\_bta>-id.  
  ENDLOOP.  
  
  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_save.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA OPTIONAL  
\*"     REFERENCE(I\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA OPTIONAL  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 436  
  
  DATA:  
        l\_tab\_bta TYPE zptb00\_tty\_bta,  
        l\_str\_btaheader TYPE zptb00\_btaheader.  
  FIELD-SYMBOLS:  
            <l\_str\_bta> TYPE zptb00\_str\_bta.  
  l\_tab\_bta = i\_tab\_bta.  
  IF NOT i\_str\_bta IS INITIAL.  
    APPEND i\_str\_bta TO l\_tab\_bta.  
  ENDIF.  
  LOOP AT i\_tab\_bta ASSIGNING <l\_str\_bta>.  
  
\*        l\_tab\_btaitem TYPE STANDARD TABLE OF  
\*        zptb00\_btaitem.  
  
    MOVE-CORRESPONDING <l\_str\_bta> TO l\_str\_btaheader.  
    MODIFY zptb00\_btaheader FROM l\_str\_btaheader.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
    DELETE FROM zptb00\_btaitem WHERE header\_id =  
    l\_str\_btaheader-id.  
    MODIFY zptb00\_btaitem FROM TABLE <l\_str\_bta>-tab\_item.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
  ENDLOOP.  
  
ENDFUNCTION.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PRODUCT\_CONFIGURATOR  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  ZPTB00\_PRODUCT\_CONFIGURATOR.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PRODUCT\_CONFIGURATOR                                                 \*  
\* Report Title :   PRODUCT\_CONFIGURATOR                                       \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 335                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
CLASS lcl\_category DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
    constructor  
    IMPORTING  
      i\_rcl\_parent TYPE REF TO lcl\_category  
      OPTIONAL  
      i\_title TYPE string,  
      write\_to\_screen  
      IMPORTING  
        i\_column TYPE i DEFAULT 0.  
    DATA:  
          parent TYPE REF TO lcl\_category,  
          childs TYPE STANDARD TABLE OF REF TO object,  
  
          title TYPE string,  
          description TYPE string,  
          vat TYPE f VALUE '0.16'.  
ENDCLASS.                    "lcl\_category DEFINITION  
  
  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_product DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_product DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
    constructor  
    IMPORTING  
      i\_rcl\_parent TYPE REF TO lcl\_category  
      i\_title TYPE string  
      i\_net\_price TYPE f,  
      write\_to\_screen  
      IMPORTING  
        i\_column TYPE i DEFAULT 0,  
        get\_gross\_price  
        RETURNING  
        value(r\_price) TYPE f.  
    DATA:  
          title TYPE string,  
          description TYPE string,  
          parent TYPE REF TO lcl\_category,  
          net\_price TYPE f.  
ENDCLASS.                    "lcl\_product DEFINITION  
  
\*Main Program start  
DATA: l\_category\_root TYPE REF TO lcl\_category,  
      l\_category TYPE REF TO lcl\_category,  
      l\_product TYPE REF TO lcl\_product.  
  
CREATE OBJECT l\_category\_root  
EXPORTING  
  i\_title = 'Products'.  
  
CREATE OBJECT l\_category  
  EXPORTING  
    i\_rcl\_parent = l\_category\_root  
    i\_title = 'Hardware'.  
  
CREATE OBJECT l\_product  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_product  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'IBM Thinkpad 30p'  
i\_net\_price = '1650'.  
  
CREATE OBJECT l\_category  
EXPORTING  
i\_rcl\_parent = l\_category\_root  
i\_title = 'Software'.  
  
CREATE OBJECT l\_product  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'Microsoft Office SBE'  
i\_net\_price = '600'.  
l\_category\_root->write\_to\_screen( ).  
  
\*Main Program end  
  
\*----------------------------------------------------------------------\*  
\*       CLASS l\_category IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_category IMPLEMENTATION.  
  METHOD constructor.  
\*Super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\* This is my daddy  
      parent = i\_rcl\_parent.  
\*Hi daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
  ENDMETHOD.                    "constructor  
  
  METHOD write\_to\_screen.  
    DATA:  
          l\_rcl\_category TYPE REF TO lcl\_category,  
          l\_rcl\_product TYPE REF TO lcl\_product,  
          l\_rcl\_descr TYPE REF TO cl\_abap\_typedescr,  
          l\_classname TYPE string,  
          l\_column TYPE i.  
    FIELD-SYMBOLS:  
            <l\_object> TYPE REF TO object.  
    WRITE: AT /i\_column 'Title:', title.  
    l\_column = i\_column + 2.  
    LOOP AT childs ASSIGNING <l\_object>.  
      l\_rcl\_descr = cl\_abap\_typedescr=>describe\_by\_object\_ref(  
              <l\_object> ).  
      l\_classname = l\_rcl\_descr->absolute\_name.  
      FIND 'CL\_CATEGORY' IN l\_classname.  
      IF sy-subrc = 0.  
        l\_rcl\_category ?= <l\_object>.  
        l\_rcl\_category->write\_to\_screen( l\_column ).  
      ELSE.  
        l\_rcl\_product ?= <l\_object>.  
        l\_rcl\_product->write\_to\_screen( l\_column ).  
      ENDIF.  
    ENDLOOP.  
  ENDMETHOD.                    "write\_to\_screen  
ENDCLASS.                    "l\_category IMPLEMENTATION  
  
  
  
\* Class Implementation  
CLASS lcl\_product IMPLEMENTATION.  
  METHOD constructor.  
\*Initialize attributes  
\*super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\*This is my daddy.  
      parent = i\_rcl\_parent.  
\*Hi, daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
    net\_price = i\_net\_price.  
  ENDMETHOD.                    "constructor  
  METHOD write\_to\_screen.  
    DATA:  
    l\_gross\_price TYPE p DECIMALS 2.  
    l\_gross\_price = me->get\_gross\_price( ).  
    WRITE: AT /i\_column 'Title :', title, '. Gross Price :',  
              (10) l\_gross\_price.  
  ENDMETHOD.                    "write\_to\_screen  
  METHOD get\_gross\_price.  
    r\_price = net\_price + net\_price \* parent->vat.  
  ENDMETHOD.                    "get\_gross\_price  
ENDCLASS.                    "lcl\_product IMPLEMENTATION

\*&---------------------------------------------------------------------\*  
\*& Report  Z\_SIMPLEPROGRAM  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  ZPTB00\_SIMPLEPROGRAM.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : Z\_SIMPLEPROGRAM                                                 \*  
\* Report Title : <description>                                         \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 99/99/9999 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.# | Date:99.99.99 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
CLASS lcl\_category DEFINITION.  
  PUBLIC SECTION.  
    DATA:  
          title TYPE string.  
    CONSTANTS:  
    vat TYPE f VALUE '0.16'.  
ENDCLASS.                    "lcl\_category DEFINITION  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_product DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_product DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
    constructor IMPORTING \_category TYPE REF TO  
      lcl\_category,  
    get\_price RETURNING value(\_price) TYPE f,  
    set\_price IMPORTING \_price TYPE f.  
    DATA:  
          title TYPE string,  
          description TYPE string,  
          category TYPE REF TO lcl\_category,  
          unit TYPE string.  
  PROTECTED SECTION.  
    DATA:  
          price TYPE f.  
ENDCLASS.                    "lcl\_product DEFINITION  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_product IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_product IMPLEMENTATION.  
  METHOD constructor.  
    super->constructor( ).  
    IF \_category IS BOUND.  
      category = \_category.  
    ELSE.  
      CREATE OBJECT category TYPE lcl\_category.  
    ENDIF.  
    price = 10.  
    title = 'Cherry G80 3000'.  
    description = 'With featured computer keybord with click.'.  
    unit = 'Piece'.  
  ENDMETHOD.                    "constructor  
  
  METHOD get\_price.  
    \_price = price.  
  ENDMETHOD.                    "get\_price  
  
  METHOD set\_price.  
    price = \_price.  
  ENDMETHOD.                    "set\_price  
ENDCLASS.                    "lcl\_product IMPLEMENTATION

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PRODUCT\_CONFIGURATOR  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  ZPTB00\_PRODUCT\_CONFIGURATOR.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PRODUCT\_CONFIGURATOR                                                 \*  
\* Report Title :   PRODUCT\_CONFIGURATOR                                       \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 335                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
CLASS lcl\_category DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
    constructor  
    IMPORTING  
      i\_rcl\_parent TYPE REF TO lcl\_category  
      OPTIONAL  
      i\_title TYPE string,  
      write\_to\_screen  
      IMPORTING  
        i\_column TYPE i DEFAULT 0.  
    DATA:  
          parent TYPE REF TO lcl\_category,  
          childs TYPE STANDARD TABLE OF REF TO object,  
  
          title TYPE string,  
          description TYPE string,  
          vat TYPE f VALUE '0.16'.  
ENDCLASS.                    "lcl\_category DEFINITION  
  
  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_product DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_product DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
    constructor  
    IMPORTING  
      i\_rcl\_parent TYPE REF TO lcl\_category  
      i\_title TYPE string  
      i\_net\_price TYPE f,  
      write\_to\_screen  
      IMPORTING  
        i\_column TYPE i DEFAULT 0,  
        get\_gross\_price  
        RETURNING  
        value(r\_price) TYPE f.  
    DATA:  
          title TYPE string,  
          description TYPE string,  
          parent TYPE REF TO lcl\_category,  
          net\_price TYPE f.  
ENDCLASS.                    "lcl\_product DEFINITION  
  
\*Main Program start  
DATA: l\_category\_root TYPE REF TO lcl\_category,  
      l\_category TYPE REF TO lcl\_category,  
      l\_product TYPE REF TO lcl\_product.  
  
CREATE OBJECT l\_category\_root  
EXPORTING  
  i\_title = 'Products'.  
  
CREATE OBJECT l\_category  
  EXPORTING  
    i\_rcl\_parent = l\_category\_root  
    i\_title = 'Hardware'.  
  
CREATE OBJECT l\_product  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_product  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'IBM Thinkpad 30p'  
i\_net\_price = '1650'.  
  
CREATE OBJECT l\_category  
EXPORTING  
i\_rcl\_parent = l\_category\_root  
i\_title = 'Software'.  
  
CREATE OBJECT l\_product  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'Microsoft Office SBE'  
i\_net\_price = '600'.  
l\_category\_root->write\_to\_screen( ).  
  
\*Main Program end  
  
\*----------------------------------------------------------------------\*  
\*       CLASS l\_category IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_category IMPLEMENTATION.  
  METHOD constructor.  
\*Super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\* This is my daddy  
      parent = i\_rcl\_parent.  
\*Hi daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
  ENDMETHOD.                    "constructor  
  
  METHOD write\_to\_screen.  
    DATA:  
          l\_rcl\_category TYPE REF TO lcl\_category,  
          l\_rcl\_product TYPE REF TO lcl\_product,  
          l\_rcl\_descr TYPE REF TO cl\_abap\_typedescr,  
          l\_classname TYPE string,  
          l\_column TYPE i.  
    FIELD-SYMBOLS:  
            <l\_object> TYPE REF TO object.  
    WRITE: AT /i\_column 'Title:', title.  
    l\_column = i\_column + 2.  
    LOOP AT childs ASSIGNING <l\_object>.  
      l\_rcl\_descr = cl\_abap\_typedescr=>describe\_by\_object\_ref(  
              <l\_object> ).  
      l\_classname = l\_rcl\_descr->absolute\_name.  
      FIND 'CL\_CATEGORY' IN l\_classname.  
      IF sy-subrc = 0.  
        l\_rcl\_category ?= <l\_object>.  
        l\_rcl\_category->write\_to\_screen( l\_column ).  
      ELSE.  
        l\_rcl\_product ?= <l\_object>.  
        l\_rcl\_product->write\_to\_screen( l\_column ).  
      ENDIF.  
    ENDLOOP.  
  ENDMETHOD.                    "write\_to\_screen  
ENDCLASS.                    "l\_category IMPLEMENTATION  
  
  
  
\* Class Implementation  
CLASS lcl\_product IMPLEMENTATION.  
  METHOD constructor.  
\*Initialize attributes  
\*super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\*This is my daddy.  
      parent = i\_rcl\_parent.  
\*Hi, daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
    net\_price = i\_net\_price.  
  ENDMETHOD.                    "constructor  
  METHOD write\_to\_screen.  
    DATA:  
    l\_gross\_price TYPE p DECIMALS 2.  
    l\_gross\_price = me->get\_gross\_price( ).  
    WRITE: AT /i\_column 'Title :', title, '. Gross Price :',  
              (10) l\_gross\_price.  
  ENDMETHOD.                    "write\_to\_screen  
  METHOD get\_gross\_price.  
    r\_price = net\_price + net\_price \* parent->vat.  
  ENDMETHOD.                    "get\_gross\_price  
ENDCLASS.                    "lcl\_product IMPLEMENTATION

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PRODUCT\_CONFIGURATOR\_GL  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_product\_configurator\_gl.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PRODUCT\_CONFIGURATOR\_GL                                                 \*  
\* Report Title :   PRODUCT\_CONFIGURATOR                                       \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 340                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.2 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  
DATA: l\_category\_root TYPE REF TO zptb00\_cl\_category,  
      l\_category TYPE REF TO zptb00\_cl\_category,  
      l\_product TYPE REF TO zptb00\_cl\_product.  
  
CREATE OBJECT l\_category\_root  
EXPORTING  
  i\_title = 'Products'.  
  
CREATE OBJECT l\_category  
  EXPORTING  
    i\_rcl\_parent = l\_category\_root  
    i\_title = 'Hardware'.  
  
CREATE OBJECT l\_product  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_product  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'IBM Thinkpad 30p'  
i\_net\_price = '1650'.  
  
CREATE OBJECT l\_category  
EXPORTING  
i\_rcl\_parent = l\_category\_root  
i\_title = 'Software'.  
  
CREATE OBJECT l\_product  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'Microsoft Office SBE'  
i\_net\_price = '600'.  
l\_category\_root->write\_to\_screen( ).

class ZPTB00\_CL\_CATEGORY definition  
  public  
  create public .  
  
\*"\* public components of class ZPTB00\_CL\_CATEGORY  
\*"\* do not include other source files here!!!  
public section.  
  
  data PARENT type ref to ZPTB00\_CL\_CATEGORY .  
  data CHILDS type ZPTB00\_TTY\_CHILDS .  
  data TITLE type STRING .  
  data DESCRIPTION type STRING .  
  data VAT type F value '0.16'. "#EC NOTEXT .  
  
  methods CONSTRUCTOR  
    importing  
      !I\_RCL\_PARENT type ref to ZPTB00\_CL\_CATEGORY optional  
      !I\_TITLE type STRING  
    preferred parameter I\_RCL\_PARENT .  
  methods WRITE\_TO\_SCREEN  
    importing  
      !I\_COLUMN type I default 0 .

METHOD constructor.  
\*ABAP Basics S. 347  
\*Super->constructor( ).  
  IF i\_rcl\_parent IS BOUND.  
\* This is my daddy  
    parent = i\_rcl\_parent.  
\*Hi daddy, i am your child  
    APPEND me TO parent->childs.  
  ENDIF.  
  title = i\_title.  
  
  
ENDMETHOD.

method WRITE\_TO\_SCREEN.  
\*ABAP Basics S. 348  
    DATA:  
          l\_rcl\_category TYPE REF TO zptb00\_cl\_category,  
          l\_rcl\_product TYPE REF TO zptb00\_cl\_product,  
          l\_rcl\_descr TYPE REF TO cl\_abap\_typedescr,  
          l\_classname TYPE string,  
          l\_column TYPE i.  
    FIELD-SYMBOLS:  
            <l\_object> TYPE REF TO object.  
    WRITE: AT /i\_column 'Title:', title.  
    l\_column = i\_column + 2.  
    LOOP AT childs ASSIGNING <l\_object>.  
      l\_rcl\_descr = cl\_abap\_typedescr=>describe\_by\_object\_ref(  
              <l\_object> ).  
      l\_classname = l\_rcl\_descr->absolute\_name.  
      FIND 'CL\_CATEGORY' IN l\_classname.  
      IF sy-subrc = 0.  
        l\_rcl\_category ?= <l\_object>.  
        l\_rcl\_category->write\_to\_screen( l\_column ).  
      ELSE.  
        l\_rcl\_product ?= <l\_object>.  
        l\_rcl\_product->write\_to\_screen( l\_column ).  
      ENDIF.  
    ENDLOOP.  
endmethod.

class ZPTB00\_CL\_PRODUCT definition  
  public  
  create public .  
  
\*"\* public components of class ZPTB00\_CL\_PRODUCT  
\*"\* do not include other source files here!!!  
public section.  
  
  data TITLE type STRING .  
  data DESCRIPTION type STRING .  
  data PARENT type ref to ZPTB00\_CL\_CATEGORY .  
  data NET\_PRICE type F .  
  
  events ON\_TITLE\_CHANGE  
    exporting  
      value(I\_TITLE) type STRING .  
  
  methods CONSTRUCTOR  
    importing  
      !I\_RCL\_PARENT type ref to ZPTB00\_CL\_CATEGORY optional  
      !I\_TITLE type STRING  
      !I\_NET\_PRICE type STRING  
    preferred parameter I\_RCL\_PARENT .  
  methods WRITE\_TO\_SCREEN  
    importing  
      !I\_COLUMN type I default 0 .  
  methods GET\_GROSS\_PRICE  
    returning  
      value(R\_PRICE) type I .  
  methods SET\_TITLE  
    for event ON\_TITLE\_CHANGE of ZPTB00\_CL\_PRODUCT  
    importing  
      !I\_TITLE .

method CONSTRUCTOR.  
\*ABAP Basics S. 349  
\*Initialize attributes  
\*super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\*This is my daddy.  
      parent = i\_rcl\_parent.  
\*Hi, daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
    net\_price = i\_net\_price.  
endmethod.

METHOD write\_to\_screen.  
\*ABAP Basics S. 349  
  DATA:  
  l\_gross\_price TYPE p DECIMALS 2.  
  l\_gross\_price = me->get\_gross\_price( ).  
  WRITE: AT /i\_column 'Title :', title, '. Gross Price :',  
            (10) l\_gross\_price.  
ENDMETHOD.

method GET\_GROSS\_PRICE.  
\*ABAP Basics S. 349  
  
   r\_price = net\_price + net\_price \* parent->vat.  
endmethod.

METHOD set\_title.  
\*ABAP Basics S. 388  
  RAISE EVENT on\_title\_change  
  EXPORTING  
    i\_title = i\_title.  
  title = i\_title.  
  
ENDMETHOD.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PRODUCT\_CONFIGURATOR  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_product\_enhancer.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PRODUCT\_ENHANCER.                                                 \*  
\* Report Title :   PRODUCT\_Enhancer                                       \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 356                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.2 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
CLASS lcl\_category DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
    constructor  
    IMPORTING  
      i\_rcl\_parent TYPE REF TO lcl\_category  
      OPTIONAL  
      i\_title TYPE string,  
      write\_to\_screen  
      IMPORTING  
        i\_column TYPE i DEFAULT 0.  
    DATA:  
          parent TYPE REF TO lcl\_category,  
          childs TYPE STANDARD TABLE OF REF TO object,  
  
          title TYPE string,  
          description TYPE string,  
          vat TYPE f VALUE '0.16'.  
ENDCLASS.                    "lcl\_category DEFINITION  
  
  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_product DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_product DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
    constructor  
    IMPORTING  
      i\_rcl\_parent TYPE REF TO lcl\_category  
      i\_title TYPE string  
      i\_net\_price TYPE f,  
      write\_to\_screen  
      IMPORTING  
        i\_column TYPE i DEFAULT 0,  
        get\_gross\_price  
        RETURNING  
        value(r\_price) TYPE f.  
    DATA:  
          title TYPE string,  
          description TYPE string,  
          parent TYPE REF TO lcl\_category,  
          net\_price TYPE f.  
ENDCLASS.                    "lcl\_product DEFINITION  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_salesproduct DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_salesproduct DEFINITION INHERITING FROM lcl\_product.  
  PUBLIC SECTION.  
    METHODS:  
    get\_gross\_price REDEFINITION.  
  PROTECTED SECTION.  
    DATA:  
          margin TYPE f VALUE '0.2'.  
  
ENDCLASS.                    "lcl\_salesproduct DEFINITION  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_purchaseproduct DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_purchaseproduct DEFINITION INHERITING FROM lcl\_product.  
  PUBLIC SECTION.  
    METHODS:  
    get\_gross\_price REDEFINITION.  
  PROTECTED SECTION.  
    DATA:  
          discount TYPE f VALUE '0.3'.  
ENDCLASS.                    "lcl\_purchaseproduct DEFINITION  
  
  
  
\*Main Program start  
DATA: l\_category\_root TYPE REF TO lcl\_category,  
      l\_category TYPE REF TO lcl\_category,  
      l\_salesproduct TYPE REF TO lcl\_salesproduct,  
      l\_purchaseproduct TYPE REF TO lcl\_purchaseproduct.  
  
  
CREATE OBJECT l\_category\_root  
EXPORTING  
  i\_title = 'Products'.  
  
CREATE OBJECT l\_category  
  EXPORTING  
    i\_rcl\_parent = l\_category\_root  
    i\_title = 'Hardware'.  
  
CREATE OBJECT l\_purchaseproduct  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_salesproduct  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_purchaseproduct  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'IBM Thinkpad 30p'  
i\_net\_price = '1650'.  
  
CREATE OBJECT l\_salesproduct  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'IBM Thinkpad 30p'  
i\_net\_price = '1650'.  
  
CREATE OBJECT l\_category  
EXPORTING  
i\_rcl\_parent = l\_category\_root  
i\_title = 'Software'.  
  
CREATE OBJECT l\_purchaseproduct  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'Microsoft Office SBE'  
i\_net\_price = '600'.  
  
CREATE OBJECT l\_salesproduct  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'Microsoft Office SBE'  
i\_net\_price = '600'.  
l\_category\_root->write\_to\_screen( ).  
\*Main Program end  
  
\*----------------------------------------------------------------------\*  
\*       CLASS l\_category IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_category IMPLEMENTATION.  
  METHOD constructor.  
\*Super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\* This is my daddy  
      parent = i\_rcl\_parent.  
\*Hi daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
  ENDMETHOD.                    "constructor  
  
  METHOD write\_to\_screen.  
    DATA:  
          l\_rcl\_category TYPE REF TO lcl\_category,  
          l\_rcl\_product TYPE REF TO lcl\_product,  
          l\_rcl\_descr TYPE REF TO cl\_abap\_typedescr,  
          l\_classname TYPE string,  
          l\_column TYPE i.  
    FIELD-SYMBOLS:  
            <l\_object> TYPE REF TO object.  
    WRITE: AT /i\_column 'Title:', title.  
    l\_column = i\_column + 2.  
    LOOP AT childs ASSIGNING <l\_object>.  
      l\_rcl\_descr = cl\_abap\_typedescr=>describe\_by\_object\_ref(  
              <l\_object> ).  
      l\_classname = l\_rcl\_descr->absolute\_name.  
      FIND 'CL\_CATEGORY' IN l\_classname.  
      IF sy-subrc = 0.  
        l\_rcl\_category ?= <l\_object>.  
        l\_rcl\_category->write\_to\_screen( l\_column ).  
      ELSE.  
        l\_rcl\_product ?= <l\_object>.  
        l\_rcl\_product->write\_to\_screen( l\_column ).  
      ENDIF.  
    ENDLOOP.  
  ENDMETHOD.                    "write\_to\_screen  
ENDCLASS.                    "l\_category IMPLEMENTATION  
  
  
  
\* Class Implementation  
CLASS lcl\_product IMPLEMENTATION.  
  METHOD constructor.  
\*Initialize attributes  
\*super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\*This is my daddy.  
      parent = i\_rcl\_parent.  
\*Hi, daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
    net\_price = i\_net\_price.  
  ENDMETHOD.                    "constructor  
  METHOD write\_to\_screen.  
    DATA:  
    l\_gross\_price TYPE p DECIMALS 2.  
    l\_gross\_price = me->get\_gross\_price( ).  
    WRITE: AT /i\_column 'Title :', title, '. Gross Price :',  
              (10) l\_gross\_price.  
  ENDMETHOD.                    "write\_to\_screen  
  METHOD get\_gross\_price.  
    r\_price = net\_price + net\_price \* parent->vat.  
  ENDMETHOD.                    "get\_gross\_price  
ENDCLASS.                    "lcl\_product IMPLEMENTATION  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_salesproduct IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_salesproduct IMPLEMENTATION.  
  METHOD get\_gross\_price.  
    DATA:  
          l\_margin\_net\_price TYPE f.  
    l\_margin\_net\_price = net\_price + net\_price \*  
                            margin.  
    r\_price = l\_margin\_net\_price +  
    l\_margin\_net\_price \* parent->vat.  
  ENDMETHOD.                    "get\_gross\_price  
ENDCLASS.                    "lcl\_salesproduct IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_purchaseproduct IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_purchaseproduct IMPLEMENTATION.  
  METHOD get\_gross\_price.  
    DATA:  
          l\_gross\_price TYPE f.  
    l\_gross\_price = super->get\_gross\_price( ).  
    r\_price = l\_gross\_price - l\_gross\_price \* discount.  
  ENDMETHOD.                    "get\_gross\_price  
ENDCLASS.                    "lcl\_purchaseproduct IMPLEMENTATION

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PRODUCT\_ENHANCER\_GL  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  ZPTB00\_PRODUCT\_ENHANCER\_GL.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PRODUCT\_ENHANCER\_GL                                                 \*  
\* Report Title : ZPTB00\_PRODUCT\_ENHANCER\_GL                                        \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 368                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.2 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
\*Main Program start  
DATA: l\_category\_root TYPE REF TO zptb00\_cl\_category,  
      l\_category TYPE REF TO zptb00\_cl\_category,  
      l\_salesproduct TYPE REF TO zptb00\_cl\_salesproduct,  
      l\_purchaseproduct TYPE REF TO zptb00\_cl\_purchaseproduct.  
  
  
CREATE OBJECT l\_category\_root  
EXPORTING  
  i\_title = 'Products'.  
  
CREATE OBJECT l\_category  
  EXPORTING  
    i\_rcl\_parent = l\_category\_root  
    i\_title = 'Hardware'.  
  
CREATE OBJECT l\_purchaseproduct  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_salesproduct  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_purchaseproduct  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'IBM Thinkpad 30p'  
i\_net\_price = '1650'.  
  
CREATE OBJECT l\_salesproduct  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'IBM Thinkpad 30p'  
i\_net\_price = '1650'.  
  
CREATE OBJECT l\_category  
EXPORTING  
i\_rcl\_parent = l\_category\_root  
i\_title = 'Software'.  
  
CREATE OBJECT l\_purchaseproduct  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'Microsoft Office SBE'  
i\_net\_price = '600'.  
  
CREATE OBJECT l\_salesproduct  
EXPORTING  
i\_rcl\_parent = l\_category  
i\_title = 'Microsoft Office SBE'  
i\_net\_price = '600'.  
l\_category\_root->write\_to\_screen( ).  
\*Main Program end

class ZPTB00\_CL\_SALESPRODUCT definition  
  public  
  inheriting from ZPTB00\_CL\_PRODUCT  
  final  
  create public .  
  
\*"\* public components of class ZPTB00\_CL\_SALESPRODUCT  
\*"\* do not include other source files here!!!  
public section.  
  
  methods GET\_GROSS\_PRICE  
    redefinition .

METHOD get\_gross\_price.  
\*ABAP Basics S. 366  
\*CALL METHOD SUPER->GET\_GROSS\_PRICE  
\*  RECEIVING  
\*    R\_PRICE =  
\*    .  
  
  DATA:  
        l\_margin\_net\_price TYPE f.  
  l\_margin\_net\_price = net\_price + net\_price \*  
                          margin.  
  r\_price = l\_margin\_net\_price +  
  l\_margin\_net\_price \* parent->vat.  
ENDMETHOD.

METHOD write\_to\_screen.  
\*CALL METHOD SUPER->WRITE\_TO\_SCREEN  
\*\*  EXPORTING  
\*\*    I\_COLUMN = 0  
\*    .  
\*ABAP Basics S. 366  
  DATA:  
  l\_gross\_price TYPE p DECIMALS 2.  
  l\_gross\_price = me->get\_gross\_price( ).  
  WRITE: AT /i\_column 'Title :', title, '. Gross Price :',  
            (10) l\_gross\_price.  
  
ENDMETHOD.

METHOD write\_to\_screen.  
\*CALL METHOD SUPER->WRITE\_TO\_SCREEN  
\*\*  EXPORTING  
\*\*    I\_COLUMN = 0  
\*    .  
\*ABAP Basics S. 366  
  DATA:  
  l\_gross\_price TYPE p DECIMALS 2.  
  l\_gross\_price = me->get\_gross\_price( ).  
  WRITE: AT /i\_column 'Title :', title, '. Gross Price :',  
            (10) l\_gross\_price.  
  
ENDMETHOD.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PRODUCT\_EVENT  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_product\_event.  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PRODUCT\_EVENT                                               \*  
\* Report Title :   PRODUCT\_\_EVENT                                        \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 376                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
CLASS lcl\_category DEFINITION.  
  PUBLIC SECTION.  
    METHODS:  
    constructor  
    IMPORTING  
      i\_rcl\_parent TYPE REF TO lcl\_category  
      OPTIONAL  
      i\_title TYPE string,  
      write\_to\_screen  
      IMPORTING  
        i\_column TYPE i DEFAULT 0.  
    DATA:  
          parent TYPE REF TO lcl\_category,  
          childs TYPE STANDARD TABLE OF REF TO object,  
  
          title TYPE string,  
          description TYPE string,  
          vat TYPE f VALUE '0.16'.  
ENDCLASS.                    "lcl\_category DEFINITION  
  
  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_product DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_product DEFINITION.  
  PUBLIC SECTION.  
    EVENTS:on\_title\_change  
    EXPORTING  
      value(i\_title) TYPE string.  
    METHODS:  
    constructor  
    IMPORTING  
      i\_rcl\_parent TYPE REF TO lcl\_category  
      i\_title TYPE string  
      i\_net\_price TYPE f,  
      write\_to\_screen  
      IMPORTING  
        i\_column TYPE i DEFAULT 0,  
        get\_gross\_price  
        RETURNING  
        value(r\_price) TYPE f,  
        set\_title for event on\_title\_change of  
        lcl\_product  
        importing i\_title.  
    DATA:  
          title TYPE string,  
          description TYPE string,  
          parent TYPE REF TO lcl\_category,  
          net\_price TYPE f.  
ENDCLASS.                    "lcl\_product DEFINITION  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_salesproduct DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_salesproduct DEFINITION INHERITING FROM lcl\_product.  
  PUBLIC SECTION.  
    METHODS:  
    get\_gross\_price REDEFINITION.  
  PROTECTED SECTION.  
    DATA:  
          margin TYPE f VALUE '0.2'.  
  
ENDCLASS.                    "lcl\_salesproduct DEFINITION  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_purchaseproduct DEFINITION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_purchaseproduct DEFINITION INHERITING FROM lcl\_product.  
  PUBLIC SECTION.  
    METHODS:  
    get\_gross\_price REDEFINITION.  
  PROTECTED SECTION.  
    DATA:  
          discount TYPE f VALUE '0.3'.  
ENDCLASS.                    "lcl\_purchaseproduct DEFINITION  
  
  
  
\*Main Program start \*\*\*  
DATA: l\_category\_root TYPE REF TO lcl\_category,  
      l\_category TYPE REF TO lcl\_category,  
      l\_salesproduct TYPE REF TO lcl\_salesproduct,  
      l\_purchaseproduct TYPE REF TO lcl\_purchaseproduct.  
  
  
CREATE OBJECT l\_category\_root  
EXPORTING  
  i\_title = 'Products'.  
  
CREATE OBJECT l\_category  
  EXPORTING  
    i\_rcl\_parent = l\_category\_root  
    i\_title = 'Hardware'.  
  
CREATE OBJECT l\_purchaseproduct  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_salesproduct  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
l\_category\_root->write\_to\_screen( ).  
set handler l\_salesproduct->set\_title for  
                            l\_purchaseproduct.  
call method l\_purchaseproduct->set\_title  
exporting  
  i\_title = 'Toschiba Portege 120s'.  
l\_category\_root->write\_to\_screen( ).  
  
\*End of Main Program \*\*\*\*\*  
  
\*----------------------------------------------------------------------\*  
\*       CLASS l\_category IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_category IMPLEMENTATION.  
  METHOD constructor.  
\*Super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\* This is my daddy  
      parent = i\_rcl\_parent.  
\*Hi daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
  ENDMETHOD.                    "constructor  
  
  METHOD write\_to\_screen.  
    DATA:  
          l\_rcl\_category TYPE REF TO lcl\_category,  
          l\_rcl\_product TYPE REF TO lcl\_product,  
          l\_rcl\_descr TYPE REF TO cl\_abap\_typedescr,  
          l\_classname TYPE string,  
          l\_column TYPE i.  
    FIELD-SYMBOLS:  
            <l\_object> TYPE REF TO object.  
    WRITE: AT /i\_column 'Title:', title.  
    l\_column = i\_column + 2.  
    LOOP AT childs ASSIGNING <l\_object>.  
      l\_rcl\_descr = cl\_abap\_typedescr=>describe\_by\_object\_ref(  
              <l\_object> ).  
      l\_classname = l\_rcl\_descr->absolute\_name.  
      FIND 'CL\_CATEGORY' IN l\_classname.  
      IF sy-subrc = 0.  
        l\_rcl\_category ?= <l\_object>.  
        l\_rcl\_category->write\_to\_screen( l\_column ).  
      ELSE.  
        l\_rcl\_product ?= <l\_object>.  
        l\_rcl\_product->write\_to\_screen( l\_column ).  
      ENDIF.  
    ENDLOOP.  
  ENDMETHOD.                    "write\_to\_screen  
ENDCLASS.                    "l\_category IMPLEMENTATION  
  
  
  
\* Class Implementation  
CLASS lcl\_product IMPLEMENTATION.  
  METHOD constructor.  
\*Initialize attributes  
\*super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\*This is my daddy.  
      parent = i\_rcl\_parent.  
\*Hi, daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
    net\_price = i\_net\_price.  
  ENDMETHOD.                    "constructor  
  METHOD write\_to\_screen.  
    DATA:  
    l\_gross\_price TYPE p DECIMALS 2.  
    l\_gross\_price = me->get\_gross\_price( ).  
    WRITE: AT /i\_column 'Title :', title, '. Gross Price :',  
              (10) l\_gross\_price.  
  ENDMETHOD.                    "write\_to\_screen  
  METHOD get\_gross\_price.  
    r\_price = net\_price + net\_price \* parent->vat.  
  ENDMETHOD.                    "get\_gross\_price  
  METHOD set\_title.  
    RAISE EVENT on\_title\_change  
    EXPORTING  
      i\_title = i\_title.  
    title = i\_title.  
    endmethod.  
  ENDCLASS.                    "lcl\_product IMPLEMENTATION  
  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_salesproduct IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_salesproduct IMPLEMENTATION.  
  METHOD get\_gross\_price.  
    DATA:  
          l\_margin\_net\_price TYPE f.  
    l\_margin\_net\_price = net\_price + net\_price \*  
                            margin.  
    r\_price = l\_margin\_net\_price +  
    l\_margin\_net\_price \* parent->vat.  
  ENDMETHOD.                    "get\_gross\_price  
ENDCLASS.                    "lcl\_salesproduct IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_purchaseproduct IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*  
\*----------------------------------------------------------------------\*  
CLASS lcl\_purchaseproduct IMPLEMENTATION.  
  METHOD get\_gross\_price.  
    DATA:  
          l\_gross\_price TYPE f.  
    l\_gross\_price = super->get\_gross\_price( ).  
    r\_price = l\_gross\_price - l\_gross\_price \* discount.  
  ENDMETHOD.                    "get\_gross\_price  
ENDCLASS.                    "lcl\_purchaseproduct IMPLEMENTATION

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_PRODUCT\_EVENT  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_product\_event\_GL.  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_PRODUCT\_EVENT                                               \*  
\* Report Title :   PRODUCT\_\_EVENT                                        \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 376                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
\*Main Program start \*\*\*  
DATA: l\_category\_root TYPE REF TO zptb00\_cl\_category,  
      l\_category TYPE REF TO zptb00\_cl\_category,  
      l\_salesproduct TYPE REF TO zptb00\_cl\_salesproduct,  
      l\_purchaseproduct TYPE REF TO zptb00\_cl\_purchaseproduct.  
  
CREATE OBJECT l\_category\_root  
EXPORTING  
  i\_title = 'Products'.  
  
CREATE OBJECT l\_category  
  EXPORTING  
    i\_rcl\_parent = l\_category\_root  
    i\_title = 'Hardware'.  
  
CREATE OBJECT l\_purchaseproduct  
  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
  
CREATE OBJECT l\_salesproduct  
EXPORTING  
  i\_rcl\_parent = l\_category  
  i\_title = 'Toshiba Satelite 430s'  
  i\_net\_price = '1500'.  
l\_category\_root->write\_to\_screen( ).  
SET HANDLER l\_salesproduct->set\_title FOR  
                            l\_purchaseproduct.  
CALL METHOD l\_purchaseproduct->set\_title  
  EXPORTING  
    i\_title = 'Toschiba Portege 120s'.  
l\_category\_root->write\_to\_screen( ).  
  
\*End of Main Program \*\*\*\*\*

class ZPTB00\_CL\_PRODUCT definition  
  public  
  create public .  
  
\*"\* public components of class ZPTB00\_CL\_PRODUCT  
\*"\* do not include other source files here!!!  
public section.  
  
  data TITLE type STRING .  
  data DESCRIPTION type STRING .  
  data PARENT type ref to ZPTB00\_CL\_CATEGORY .  
  data NET\_PRICE type F .  
  
  events ON\_TITLE\_CHANGE  
    exporting  
      value(I\_TITLE) type STRING .  
  
  methods CONSTRUCTOR  
    importing  
      !I\_RCL\_PARENT type ref to ZPTB00\_CL\_CATEGORY optional  
      !I\_TITLE type STRING  
      !I\_NET\_PRICE type STRING  
    preferred parameter I\_RCL\_PARENT .  
  methods WRITE\_TO\_SCREEN  
    importing  
      !I\_COLUMN type I default 0 .  
  methods GET\_GROSS\_PRICE  
    returning  
      value(R\_PRICE) type I .  
  methods SET\_TITLE  
    for event ON\_TITLE\_CHANGE of ZPTB00\_CL\_PRODUCT  
    importing  
      !I\_TITLE .

method CONSTRUCTOR.  
\*ABAP Basics S. 349  
\*Initialize attributes  
\*super->constructor( ).  
    IF i\_rcl\_parent IS BOUND.  
\*This is my daddy.  
      parent = i\_rcl\_parent.  
\*Hi, daddy, i am your child  
      APPEND me TO parent->childs.  
    ENDIF.  
    title = i\_title.  
    net\_price = i\_net\_price.  
endmethod.

METHOD write\_to\_screen.  
\*ABAP Basics S. 349  
  DATA:  
  l\_gross\_price TYPE p DECIMALS 2.  
  l\_gross\_price = me->get\_gross\_price( ).  
  WRITE: AT /i\_column 'Title :', title, '. Gross Price :',  
            (10) l\_gross\_price.  
ENDMETHOD.,

method GET\_GROSS\_PRICE.  
\*ABAP Basics S. 349  
  
   r\_price = net\_price + net\_price \* parent->vat.  
endmethod.

METHOD set\_title.  
\*ABAP Basics S. 388  
  RAISE EVENT on\_title\_change  
  EXPORTING  
    i\_title = i\_title.  
  title = i\_title.  
  
ENDMETHOD.

----------------------------------------------------------------------------------------------------------------------

FUNCTION zptb00\_map\_bta\_set.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 431  
  
  CALL FUNCTION 'ZPTB00\_API\_BTA\_SET'  
    EXPORTING  
      i\_tab\_bta = i\_str\_bta  
    EXCEPTIONS  
      failed    = 1  
      OTHERS    = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
ENDFUNCTION.

FUNCTION zptb00\_map\_rep\_get.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_BOOKING\_DATE) TYPE  DDTRANGE  
\*"  EXPORTING  
\*"     REFERENCE(E\_TAB\_DOC) TYPE  ZPTB00\_TTY\_DOC  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 432  
  
  CALL FUNCTION 'ZPTB00\_API\_DOC\_LOAD'  
    EXPORTING  
      i\_tab\_booking\_date = i\_tab\_booking\_date  
    IMPORTING  
      e\_tab\_doc          = e\_tab\_doc  
    EXCEPTIONS  
      failed             = 1  
      OTHERS             = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
ENDFUNCTION.

FUNCTION zptb00\_api\_bta\_load.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_BOOKED) TYPE  ZPTB00\_DTE\_BTA\_BOOKED DEFAULT '1'  
\*"  EXPORTING  
\*"     REFERENCE(E\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 425  
  
  CALL FUNCTION 'ZPTB00\_OBJ\_BTA\_LOAD'  
\* EXPORTING  
\*   I\_BOOKED        = '1'  
   IMPORTING  
     e\_tab\_bta       = e\_tab\_bta  
  EXCEPTIONS  
    failed = 1  
    OTHERS = 2.  
  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  
  
ENDFUNCTION.

FUNCTION ZPTB00\_API\_BTA\_MARK\_AS\_BOOKED.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 426  
  
CALL FUNCTION 'ZPTB00\_OBJ\_BTA\_MARK\_AS\_BOOKED'  
  EXPORTING  
    i\_tab\_bta       = i\_tab\_bta  
 EXCEPTIONS  
   FAILED          = 1  
   OTHERS          = 2  
          .  
IF sy-subrc <> 0.  
raise failed.  
ENDIF.  
  
  
  
  
ENDFUNCTION.

FUNCTION zptb00\_api\_bta\_set.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 426  
  
  
  DATA: l\_str\_bta TYPE ZPTB00\_STR\_BTA.  
  
\*assign to local variable, because we want to enrich the structure  
  l\_str\_bta = i\_str\_bta.  
  
\* Enrich the structure by technical fields  
  CALL FUNCTION 'ZPTB00\_OBJ\_BTA\_ENRICH'  
    CHANGING  
      c\_str\_bta = l\_str\_bta  
    EXCEPTIONS  
      failed    = 1  
      OTHERS    = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  CALL FUNCTION 'ZPTB00\_API\_BTC\_CHECK'  
    EXPORTING  
      i\_str\_bta = l\_str\_bta  
    EXCEPTIONS  
      failed    = 1  
      OTHERS    = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  CALL FUNCTION 'ZPTB00\_OBJ\_BTA\_SAVE'  
   EXPORTING  
     i\_str\_bta       = l\_str\_bta  
\*   I\_TAB\_BTA       =  
   EXCEPTIONS  
     failed          = 1  
     OTHERS          = 2  
            .  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
ENDFUNCTION.

FUNCTION zptb00\_api\_btc\_check.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 427  
  
  CALL FUNCTION 'ZPTB00\_API\_BTC\_CHECK'  
    EXPORTING  
      i\_str\_bta = i\_str\_bta  
    EXCEPTIONS  
      failed    = 1  
      OTHERS    = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
ENDFUNCTION.

FUNCTION zptb00\_api\_doc\_load.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_BOOKING\_DATE) TYPE  DDTRANGE  
\*"  EXPORTING  
\*"     REFERENCE(E\_TAB\_DOC) TYPE  ZPTB00\_TTY\_DOC  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 428  
  
  CALL FUNCTION 'ZPTB00\_API\_DOC\_LOAD'  
    EXPORTING  
      i\_tab\_booking\_date = i\_tab\_booking\_date  
    IMPORTING  
      e\_tab\_doc          = e\_tab\_doc  
    EXCEPTIONS  
      failed             = 1  
      OTHERS             = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
ENDFUNCTION.

FUNCTION zptb00\_api\_doc\_run.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 429  
  
  DATA: l\_tab\_bta TYPE zptb00\_tty\_bta,  
        l\_tab\_doc TYPE zptb00\_tty\_doc.  
  
  CALL FUNCTION 'ZPTB00\_API\_BTA\_LOAD'  
\* EXPORTING  
\*   I\_BOOKED        = '1'  
   IMPORTING  
     e\_tab\_bta       = l\_tab\_bta  
   EXCEPTIONS  
     failed          = 1  
     OTHERS          = 2  
            .  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  
  CALL FUNCTION 'ZPTB00\_OBJ\_DOC\_BOOK'  
    EXPORTING  
      i\_tab\_bta = l\_tab\_bta  
    IMPORTING  
      e\_tab\_doc = l\_tab\_doc  
    EXCEPTIONS  
      failed    = 1  
      OTHERS    = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  CALL FUNCTION 'ZPTB00\_OBJ\_DOC\_SAVE'  
   EXPORTING  
\*   I\_STR\_DOC       =  
     i\_tab\_doc       = l\_tab\_doc  
   EXCEPTIONS  
     failed          = 1  
     OTHERS          = 2  
            .  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  CALL FUNCTION 'ZPTB00\_API\_BTA\_MARK\_AS\_BOOKED'  
    EXPORTING  
      i\_tab\_bta = l\_tab\_bta  
    EXCEPTIONS  
      failed    = 1  
      OTHERS    = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  
ENDFUNCTION.

FUNCTION zptb00\_api\_rep\_display.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_BOOKING\_DATE) TYPE  DDTRANGE  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 430  
  
  DATA: l\_tab\_doc TYPE zptb00\_tty\_doc.  
  
  CALL FUNCTION 'ZPTB00\_OBJ\_DOC\_LOAD'  
    EXPORTING  
      i\_tab\_booking\_date = i\_tab\_booking\_date  
    IMPORTING  
      e\_tab\_doc          = l\_tab\_doc  
    EXCEPTIONS  
      failed             = 1  
      OTHERS             = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  
  CALL FUNCTION 'ZPTB00\_OBJ\_REP\_SHOW'  
    EXPORTING  
      i\_tab\_doc = l\_tab\_doc  
    EXCEPTIONS  
      failed    = 1  
      OTHERS    = 2.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_edit.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  CHANGING  
\*"     REFERENCE(C\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA OPTIONAL  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 431  
  
  DATA: l\_str\_item TYPE zptb00\_btaitem.  
  
  IF c\_str\_bta IS INITIAL.  
\*Just add one line to the position by default.  
    APPEND l\_str\_item TO c\_str\_bta-tab\_item.  
  ENDIF.  
\*Make manual posting global for dynpro.  
  MOVE-CORRESPONDING c\_str\_bta TO g\_str\_bta.  
\* Call edit screen  
  CALL SCREEN 0100.  
  
  MOVE-CORRESPONDING g\_str\_bta TO c\_str\_bta.  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_enrich.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  CHANGING  
\*"     REFERENCE(C\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 433  
  
  FIELD-SYMBOLS:  
                 <l\_str\_item> TYPE zptb00\_btaitem.  
  
  c\_str\_bta-client = sy-mandt.  
  
  CALL FUNCTION 'GUID\_CREATE'  
   IMPORTING  
\*   EV\_GUID\_16       =  
\*   EV\_GUID\_22       =  
     ev\_guid\_32       = c\_str\_bta-id.  
  c\_str\_bta-booked = '1'.  
  
\*enrich items.  
  
  LOOP AT c\_str\_bta-tab\_item ASSIGNING <l\_str\_item>.  
    <l\_str\_item>-client = sy-mandt.  
  
    CALL FUNCTION 'GUID\_CREATE'  
     IMPORTING  
\*   EV\_GUID\_16       =  
\*   EV\_GUID\_22       =  
       ev\_guid\_32       = <l\_str\_item>-id.  
    <l\_str\_item>-header\_id = c\_str\_bta-id.  
    <l\_str\_item>-mark = ' '.  
  ENDLOOP.  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_load.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_BOOKED) TYPE  ZPTB00\_DTE\_BTA\_BOOKED DEFAULT '1'  
\*"  EXPORTING  
\*"     REFERENCE(E\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 431  
  
  DATA:  
\*db  
        l\_tab\_bta\_header TYPE STANDARD TABLE OF zptb00\_btaheader,  
        l\_tab\_bta\_item TYPE STANDARD TABLE OF zptb00\_btaitem,  
\*local  
        l\_str\_bta TYPE zptb00\_str\_bta.  
  
  FIELD-SYMBOLS:  
\*db  
                 <l\_str\_bta\_header> TYPE zptb00\_btaheader,  
                 <l\_str\_bta\_item> TYPE zptb00\_btaitem.  
\* read the header data  
  SELECT \* FROM zptb00\_btaheader INTO TABLE  
    l\_tab\_bta\_header  
    WHERE booked = i\_booked.  
  
  IF sy-dbcnt = 0.  
    RAISE failed.  
  ENDIF.  
  
\*read the item data. APAB statement "For all entries" only works  
\* up to ca. 1700 entries so we better loop.  
  
  LOOP AT l\_tab\_bta\_header ASSIGNING <l\_str\_bta\_header>.  
    MOVE-CORRESPONDING <l\_str\_bta\_header> TO l\_str\_bta.  
    SELECT \* FROM zptb00\_btaitem INTO TABLE l\_tab\_bta\_item  
      WHERE header\_id = <l\_str\_bta\_header>-id.  
    l\_str\_bta-tab\_item = l\_tab\_bta\_item.  
    APPEND l\_str\_bta TO e\_tab\_bta.  
  ENDLOOP.  
  IF sy-dbcnt = 0.  
    RAISE failed.  
  ENDIF.  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_mark\_as\_booked.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 435  
  
  FIELD-SYMBOLS:  
                 <l\_str\_bta> TYPE zptb00\_str\_bta.  
  
  LOOP AT i\_tab\_bta ASSIGNING <l\_str\_bta>.  
    UPDATE zptb00\_btaheader SET booked = '2'  
    WHERE id = <l\_str\_bta>-id.  
  ENDLOOP.  
  
  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_bta\_save.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA OPTIONAL  
\*"     REFERENCE(I\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA OPTIONAL  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 436  
  
  DATA:  
        l\_tab\_bta TYPE zptb00\_tty\_bta,  
        l\_str\_btaheader TYPE zptb00\_btaheader.  
  FIELD-SYMBOLS:  
            <l\_str\_bta> TYPE zptb00\_str\_bta.  
  l\_tab\_bta = i\_tab\_bta.  
  IF NOT i\_str\_bta IS INITIAL.  
    APPEND i\_str\_bta TO l\_tab\_bta.  
  ENDIF.  
  LOOP AT i\_tab\_bta ASSIGNING <l\_str\_bta>.  
  
\*        l\_tab\_btaitem TYPE STANDARD TABLE OF  
\*        zptb00\_btaitem.  
  
    MOVE-CORRESPONDING <l\_str\_bta> TO l\_str\_btaheader.  
    MODIFY zptb00\_btaheader FROM l\_str\_btaheader.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
    DELETE FROM zptb00\_btaitem WHERE header\_id =  
    l\_str\_btaheader-id.  
    MODIFY zptb00\_btaitem FROM TABLE <l\_str\_bta>-tab\_item.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
  ENDLOOP.  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_btc\_check.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_STR\_BTA) TYPE  ZPTB00\_STR\_BTA  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 437  
  
  DATA: l\_tab\_dd07v LIKE STANDARD TABLE OF dd07v,  
        l\_str\_currency TYPE tcurc,  
        l\_str\_quantity TYPE t006.  
  FIELD-SYMBOLS:  
                 <l\_str\_item> TYPE zptb00\_btaitem.  
  
\*\*\*\*\*\*\*\*\*\*\* check header  
\* check bp\_id  
  IF i\_str\_bta-bp\_id IS INITIAL.  
    RAISE failed.  
  ENDIF.  
  
\* check bttype  
  CALL FUNCTION 'DD\_DOMVALUES\_GET'  
    EXPORTING  
      domname              = 'ZPTB00\_DOM\_BTA\_BTTYPE'  
\*   TEXT                 = ' '  
\*   LANGU                = ' '  
\*   BYPASS\_BUFFER        = ' '  
\* IMPORTING  
\*   RC                   =  
    TABLES  
      dd07v\_tab            = l\_tab\_dd07v  
   EXCEPTIONS  
     wrong\_textflag       = 1  
     OTHERS               = 2  
            .  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  READ TABLE l\_tab\_dd07v WITH KEY domvalue\_l =  
  i\_str\_bta-bttype TRANSPORTING NO FIELDS.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
  
  
\* check item  
\* check tab\_item  
  IF i\_str\_bta-tab\_item IS INITIAL.  
    RAISE failed.  
  ENDIF.  
  LOOP AT i\_str\_bta-tab\_item ASSIGNING <l\_str\_item>.  
  
  
\* check pr\_id  
  
    CALL FUNCTION 'DD\_DOMVALUES\_GET'  
      EXPORTING  
        domname              = 'ZPTB00\_DOM\_BTA\_PR\_ID'  
\*   TEXT                 = ' '  
\*   LANGU                = ' '  
\*   BYPASS\_BUFFER        = ' '  
\* IMPORTING  
\*   RC                   =  
      TABLES  
        dd07v\_tab            = l\_tab\_dd07v  
     EXCEPTIONS  
       wrong\_textflag       = 1  
       OTHERS               = 2  
              .  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
    READ TABLE l\_tab\_dd07v WITH KEY domvalue\_l =  
    <l\_str\_item>-pr\_id TRANSPORTING NO FIELDS.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
  
\*check pstype  
    CALL FUNCTION 'DD\_DOMVALUES\_GET'  
      EXPORTING  
        domname              = 'ZPTB00\_DOM\_BTA\_PSTYPE'  
\*   TEXT                 = ' '  
\*   LANGU                = ' '  
\*   BYPASS\_BUFFER        = ' '  
\* IMPORTING  
\*   RC                   =  
      TABLES  
        dd07v\_tab            = l\_tab\_dd07v  
     EXCEPTIONS  
       wrong\_textflag       = 1  
       OTHERS               = 2  
              .  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
    READ TABLE l\_tab\_dd07v WITH KEY domvalue\_l =  
    <l\_str\_item>-pstype TRANSPORTING NO FIELDS.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
  
\*check wh\_id  
  
    CALL FUNCTION 'DD\_DOMVALUES\_GET'  
      EXPORTING  
        domname              = 'ZPTB00\_DOM\_BTA\_WH\_ID'  
\*   TEXT                 = ' '  
\*   LANGU                = ' '  
\*   BYPASS\_BUFFER        = ' '  
\* IMPORTING  
\*   RC                   =  
      TABLES  
        dd07v\_tab            = l\_tab\_dd07v  
     EXCEPTIONS  
       wrong\_textflag       = 1  
       OTHERS               = 2  
              .  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
    READ TABLE l\_tab\_dd07v WITH KEY domvalue\_l =  
    <l\_str\_item>-pr\_id TRANSPORTING NO FIELDS.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
\* check currency  
  
    SELECT SINGLE \* FROM tcurc INTO l\_str\_currency  
      WHERE waers = <l\_str\_item>-currency.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
  
\* check unit  
  SELECT SINGLE \* FROM t006 INTO l\_str\_quantity WHERE  
    msehi = <l\_str\_item>-unit.  
  IF sy-subrc <> 0.  
    RAISE failed.  
  ENDIF.  
ENDLOOP.  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_doc\_book.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_BTA) TYPE  ZPTB00\_TTY\_BTA  
\*"  EXPORTING  
\*"     REFERENCE(E\_TAB\_DOC) TYPE  ZPTB00\_TTY\_DOC  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 440  
  
  DATA:  
\*accounting document  
  l\_str\_doc TYPE  zptb00\_str\_doc,  
  l\_str\_doc\_item TYPE zptb00\_docitem,  
\* business transaction  
  l\_str\_bta TYPE zptb00\_btaheader.  
  
  FIELD-SYMBOLS:  
\* business transaction  
  <l\_str\_bta> TYPE zptb00\_str\_bta,  
  <l\_str\_bta\_item> TYPE zptb00\_btaitem.  
  
  CONSTANTS:  
\* BUSINESS TRANSACTION field constants  
  con\_bta\_bttype\_purchase TYPE zptb00\_dte\_bta\_bttype  
        VALUE 'PURCHASE',  
  con\_bta\_bttype\_sale TYPE zptb00\_dte\_bta\_bttype  
        VALUE 'SALE',  
  con\_bta\_pstype\_service TYPE zptb00\_dte\_bta\_pstype  
        VALUE 'SERVICE',  
  con\_bta\_pstype\_product TYPE zptb00\_dte\_bta\_pstype  
        VALUE 'PRODUCT',  
\*document field constants  
   con\_doc\_debit TYPE zptb00\_dte\_doc\_debitcredit  
      VALUE 'D',  
   con\_doc\_credit TYPE zptb00\_dte\_doc\_debitcredit  
   VALUE 'C',  
   con\_doc\_account\_bga  TYPE zptb00\_dte\_doc\_account  
   VALUE '001000',  
   con\_doc\_account\_cash TYPE zptb00\_dte\_doc\_account  
   VALUE '002000'.  
  
\*loop through every business transaction  
  LOOP AT i\_tab\_bta ASSIGNING <l\_str\_bta>.  
\*direct move-corresponding  
    MOVE-CORRESPONDING <l\_str\_bta> TO l\_str\_bta.  
    MOVE-CORRESPONDING l\_str\_bta TO l\_str\_doc.  
  
\*loop through every business transaction position  
    LOOP AT <l\_str\_bta>-tab\_item ASSIGNING  
      <l\_str\_bta\_item>.  
      MOVE-CORRESPONDING <l\_str\_bta\_item> TO  
            l\_str\_doc\_item.  
  
\* this is the accounting brain  
\*book every business transaction as an accounting  
\*document  
      IF <l\_str\_bta>-bttype = con\_bta\_bttype\_purchase.  
        IF <l\_str\_bta\_item>-pstype =  
          con\_bta\_pstype\_service.  
        ELSEIF <l\_str\_bta\_item>-pstype =  
          con\_bta\_pstype\_product.  
  
\*For our project we implement only one example,  
\* but more to come  
          l\_str\_doc\_item-debitcredit = con\_doc\_debit.  
          l\_str\_doc\_item-account = con\_doc\_account\_bga.  
          APPEND l\_str\_doc\_item TO l\_str\_doc-tab\_item.  
  
          CALL FUNCTION 'GUID\_CREATE'  
           IMPORTING  
\*         EV\_GUID\_16       =  
\*         EV\_GUID\_22       =  
             ev\_guid\_32       = l\_str\_doc\_item-id.  
  
          l\_str\_doc\_item-debitcredit = con\_doc\_credit.  
          l\_str\_doc\_item-account = con\_doc\_account\_cash.  
          APPEND l\_str\_doc\_item TO l\_str\_doc-tab\_item.  
        ENDIF.  
      ENDIF.  
    ENDLOOP.  
    APPEND l\_str\_doc TO e\_tab\_doc.  
  ENDLOOP.  
  .  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_doc\_load.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_BOOKING\_DATE) TYPE  DDTRANGE  
\*"  EXPORTING  
\*"     REFERENCE(E\_TAB\_DOC) TYPE  ZPTB00\_TTY\_DOC  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 443  
  
  
  DATA:  
\*db  
        l\_tab\_doc\_header TYPE STANDARD TABLE OF zptb00\_docheader,  
        l\_tab\_doc\_item TYPE STANDARD TABLE OF zptb00\_docitem,  
\*local  
        l\_str\_doc TYPE zptb00\_str\_doc.  
  
  FIELD-SYMBOLS:  
\*db  
                 <l\_str\_doc\_header> TYPE zptb00\_docheader,  
                 <l\_str\_doc\_item> TYPE zptb00\_docitem.  
\* read the header data  
  SELECT \* FROM zptb00\_docheader INTO TABLE  
    l\_tab\_doc\_header  
    WHERE btdate IN i\_tab\_booking\_date.  
  
  IF sy-dbcnt = 0.  
    RAISE failed.  
  ENDIF.  
  
\*read the item data. APAB statement "For all entries" only works  
\* up to ca. 1700 entries so we better loop.  
  
  LOOP AT l\_tab\_doc\_header ASSIGNING <l\_str\_doc\_header>.  
    MOVE-CORRESPONDING <l\_str\_doc\_header> TO l\_str\_doc.  
    SELECT \* FROM zptb00\_docitem INTO TABLE l\_tab\_doc\_item  
      WHERE header\_id = <l\_str\_doc\_header>-id.  
    l\_str\_doc-tab\_item = l\_tab\_doc\_item.  
    APPEND l\_str\_doc TO e\_tab\_doc.  
  ENDLOOP.  
  IF sy-dbcnt = 0.  
    RAISE failed.  
  ENDIF.  
  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_doc\_save.  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_STR\_DOC) TYPE  ZPTB00\_STR\_DOC OPTIONAL  
\*"     REFERENCE(I\_TAB\_DOC) TYPE  ZPTB00\_TTY\_DOC OPTIONAL  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 444  
  
  DATA:  
        l\_tab\_doc TYPE zptb00\_tty\_doc,  
        l\_str\_docheader TYPE zptb00\_docheader.  
  FIELD-SYMBOLS:  
            <l\_str\_doc> TYPE zptb00\_str\_doc.  
  l\_tab\_doc = i\_tab\_doc.  
  IF NOT i\_str\_doc IS INITIAL.  
    APPEND i\_str\_doc TO l\_tab\_doc.  
  ENDIF.  
  
  LOOP AT i\_tab\_doc ASSIGNING <l\_str\_doc>.  
  
\*        l\_tab\_btaitem TYPE STANDARD TABLE OF  
\*        zptb00\_btaitem.  
  
    MOVE-CORRESPONDING <l\_str\_doc> TO l\_str\_docheader.  
    MODIFY zptb00\_docheader FROM l\_str\_docheader.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
    DELETE FROM zptb00\_docitem WHERE header\_id =  
    l\_str\_docheader-id.  
    MODIFY zptb00\_docitem FROM TABLE <l\_str\_doc>-tab\_item.  
    IF sy-subrc <> 0.  
      RAISE failed.  
    ENDIF.  
  
  ENDLOOP.  
  
  
  
  
ENDFUNCTION.

FUNCTION zptb00\_obj\_rep\_show .  
\*"----------------------------------------------------------------------  
\*"\*"Lokale Schnittstelle:  
\*"  IMPORTING  
\*"     REFERENCE(I\_TAB\_DOC) TYPE  ZPTB00\_TTY\_DOC  
\*"  EXCEPTIONS  
\*"      FAILED  
\*"----------------------------------------------------------------------  
\*ABAP Basics S. 444  
  
\* Contains ALV Grid structures needet for variables  
  TYPE-POOLS slis.  
  
  DATA:  
        "for data handling  
              l\_str\_doc\_flat TYPE zptb00\_str\_doc\_flat,  
              l\_tab\_doc\_flat TYPE STANDARD TABLE OF  
              zptb00\_str\_doc\_flat,  
        " for ALV-Grid display  
        l\_dis TYPE disvariant,  
        l\_rda\_table TYPE REF TO data,  
              l\_str\_layout TYPE slis\_layout\_alv.  
  
  FIELD-SYMBOLS:  
        "for data Handling  
              <l\_str\_doc> TYPE zptb00\_str\_doc,  
              <l\_str\_doc\_item> TYPE zptb00\_docitem.  
  
\*convert data to flast structure  
  LOOP AT i\_tab\_doc ASSIGNING <l\_str\_doc>.  
    MOVE-CORRESPONDING <l\_str\_doc> TO l\_str\_doc\_flat.  
    LOOP AT <l\_str\_doc>-tab\_item ASSIGNING <l\_str\_doc\_item>.  
      MOVE-CORRESPONDING <l\_str\_doc\_item> TO l\_str\_doc\_flat.  
      APPEND l\_str\_doc\_flat TO l\_tab\_doc\_flat.  
    ENDLOOP.  
  ENDLOOP.  
  
  
\*Make column width as small as possible  
  l\_str\_layout-colwidth\_optimize = 'X'.  
\*Window title  
  l\_str\_layout-window\_titlebar = text-001.  
  l\_dis-report = sy-repid.  
  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
   EXPORTING  
\*   I\_INTERFACE\_CHECK                 = ' '  
\*   I\_BYPASSING\_BUFFER                = ' '  
\*   I\_BUFFER\_ACTIVE                   = ' '  
\*   I\_CALLBACK\_PROGRAM                = ' '  
\*   I\_CALLBACK\_PF\_STATUS\_SET          = ' '  
\*   I\_CALLBACK\_USER\_COMMAND           = ' '  
\*   I\_CALLBACK\_TOP\_OF\_PAGE            = ' '  
\*   I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '  
\*   I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '  
     i\_structure\_name                  = 'ZPTB00\_STR\_DOC\_FLAT'  
\*   I\_BACKGROUND\_ID                   = ' '  
\*   I\_GRID\_TITLE                      =  
\*   I\_GRID\_SETTINGS                   =  
     is\_layout                         = l\_str\_layout  
\*   IT\_FIELDCAT                       =  
\*   IT\_EXCLUDING                      =  
\*   IT\_SPECIAL\_GROUPS                 =  
\*   IT\_SORT                           =  
\*   IT\_FILTER                         =  
\*   IS\_SEL\_HIDE                       =  
\*   I\_DEFAULT                         = 'X'  
     i\_save                            = 'A'  
     is\_variant                        = l\_dis  
\*   IT\_EVENTS                         =  
\*   IT\_EVENT\_EXIT                     =  
\*   IS\_PRINT                          =  
\*   IS\_REPREP\_ID                      =  
\*   I\_SCREEN\_START\_COLUMN             = 0  
\*   I\_SCREEN\_START\_LINE               = 0  
\*   I\_SCREEN\_END\_COLUMN               = 0  
\*   I\_SCREEN\_END\_LINE                 = 0  
\*   I\_HTML\_HEIGHT\_TOP                 = 0  
\*   I\_HTML\_HEIGHT\_END                 = 0  
\*   IT\_ALV\_GRAPHICS                   =  
\*   IT\_HYPERLINK                      =  
\*   IT\_ADD\_FIELDCAT                   =  
\*   IT\_EXCEPT\_QINFO                   =  
\*   IR\_SALV\_FULLSCREEN\_ADAPTER        =  
\* IMPORTING  
\*   E\_EXIT\_CAUSED\_BY\_CALLER           =  
\*   ES\_EXIT\_CAUSED\_BY\_USER            =  
    TABLES  
      t\_outtab                          = l\_tab\_doc\_flat  
   EXCEPTIONS  
     program\_error                     = 1  
     OTHERS                            = 2.  
  
\*Error? ->show why          .  
  IF sy-subrc <> 0.  
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  ENDIF.  
  
  
  
ENDFUNCTION.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_BUSINESS\_TRANSACTION  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_business\_transaction.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_BUSINESS\_TRANSACTION                         \*  
\* Report Title : Enter a Business Transaction manually                 \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 99/99/9999 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : ABAP Basics S. 74 / 422                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.# | Date:99.99.99 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
DATA: l\_str\_bta TYPE zptb00\_str\_bta.  
  
DO.  
  CALL FUNCTION 'ZPTB00\_OBJ\_BTA\_EDIT'  
    CHANGING  
      c\_str\_bta = l\_str\_bta.  
  
  IF NOT l\_str\_bta IS INITIAL.  
    CALL FUNCTION 'ZPTB00\_OBJ\_BTA\_SAVE'  
      EXPORTING  
        i\_str\_bta = l\_str\_bta  
      EXCEPTIONS  
        failed    = 1  
        OTHERS    = 2.  
    CLEAR l\_str\_bta.  
  ELSE.  
    RETURN.  
  ENDIF.  
ENDDO.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_DOCUMENT\_BOOKING  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_document\_booking.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_DOCUMENT\_BOOKING                                                 \*  
\* Report Title :   DOCUMENT\_BOOKING                                      \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 423                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
PARAMETERS:  
p\_book TYPE boolean.  
  
CALL FUNCTION 'ZPTB00\_API\_DOC\_RUN'  
  EXCEPTIONS  
    failed = 1  
    OTHERS = 2.  
IF sy-subrc <> 0.  
  MESSAGE e000(zptb00\_doc).  
ENDIF.

\*&---------------------------------------------------------------------\*  
\*& Report  ZPTB00\_DOCUMENT\_VIEWER  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
  
REPORT  zptb00\_document\_viewer.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Report Name  : ZPTB00\_DOCUMENT\_VIEWER                      \*  
\* Report Title : <description>                                         \*  
\* Report Title :   PRODUCT\_CONFIGURATOR                                \*  
\* SAP Release 6.0                                                      \*  
\* Author´s Name: BUESCHERM                                         \*  
\* Create Date  : 02/01/2019 (MM/DD/YYYY)                               \*  
\* Called from  : <Report where used Information>                       \*  
\*                                                                      \*  
\* Description  : <ABAP Basics S. 423                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*              : <DESCRIPTION EXPLAINING FUNCTIONALITY>                \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Development / Modification History                                   \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Ver.1 | Date:2.01.19 | Author: M.Büscher | Destrcription/Transport  \*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
DATA: l\_btdate TYPE zptb00\_dte\_bta\_date,  
      l\_tab\_btdate TYPE ddtrange,  
      l\_str\_btdate TYPE ddrange.  
  
FIELD-SYMBOLS:  
               <l\_str\_btdate> TYPE ANY.  
SELECTION-SCREEN BEGIN OF SCREEN 110.  
SELECTION-SCREEN BEGIN OF BLOCK b110 WITH FRAME TITLE text-005.  
SELECT-OPTIONS:  
p\_btdate FOR l\_btdate.  
SELECTION-SCREEN END OF BLOCK b110.  
SELECTION-SCREEN END OF SCREEN 110.  
  
\* Main Program  
DO.  
  CALL SELECTION-SCREEN 110.  
  IF sy-subrc <> 0.  
    EXIT.  
  ELSE.  
    LOOP AT p\_btdate ASSIGNING <l\_str\_btdate>.  
      MOVE-CORRESPONDING <l\_str\_btdate> TO l\_str\_btdate.  
    ENDLOOP.  
    CALL FUNCTION 'ZPTB00\_API\_REP\_DISPLAY'  
      EXPORTING  
        i\_tab\_booking\_date = l\_tab\_btdate  
      EXCEPTIONS  
        failed             = 1  
        OTHERS             = 2.  
    IF sy-subrc <> 0.  
      MESSAGE e000(zptb00\_rep).  
    ENDIF.  
  
  ENDIF.  
ENDDO.