ZAD2 Raport **ZMJ\_MANAGE\_CARS** plik ( cars ),

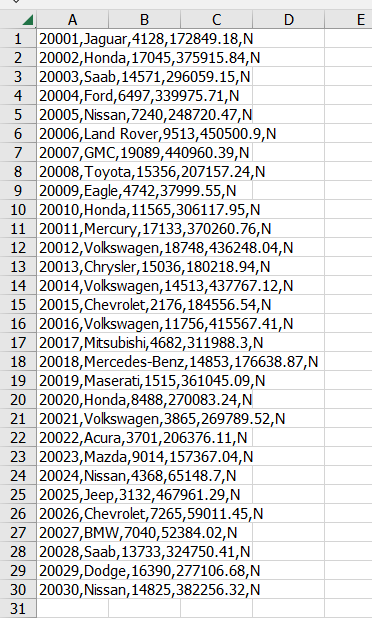
1. Radiobutton służący do dodania nowych samochodów do bazy danych

ma za zadanie operując na pliku wejściowym typu CSV wyciągnać z niego dane i umieścic w tabelii z samochodami, w pliku znajdują się kolumny ID, marka, przebieg, cena, status.

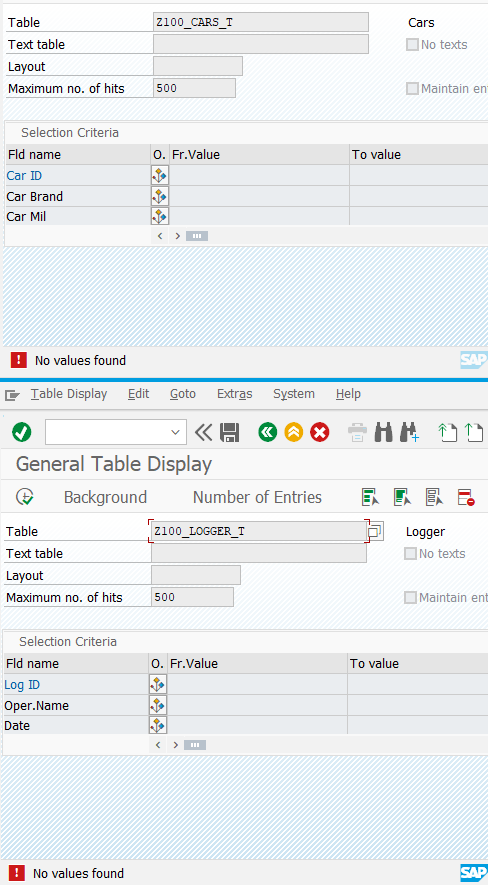
( kazda aktualizacje zapisz w tabeli

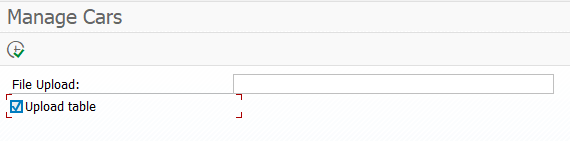
ZMJ\_LOGGER\_T )

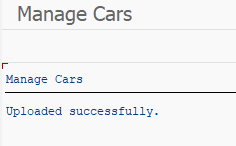
Plik do załadowania:



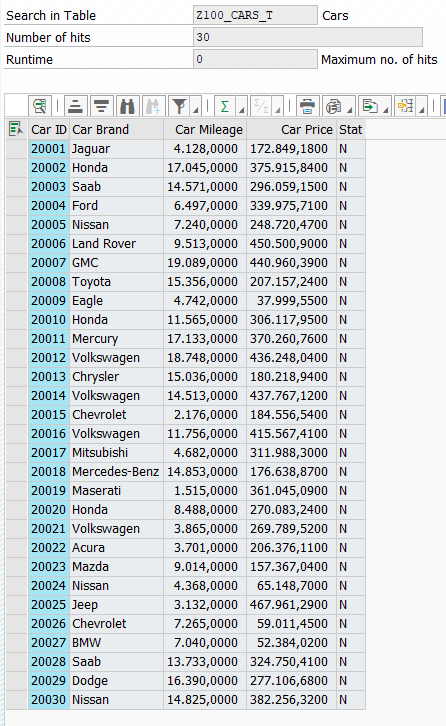
Zawartość tabeli przed załadowaniem: (puste)



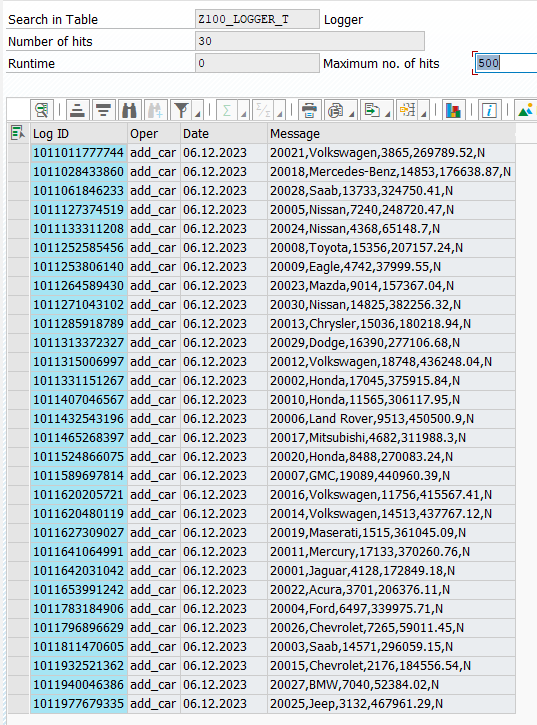




Po załadowaniu:



Logger:



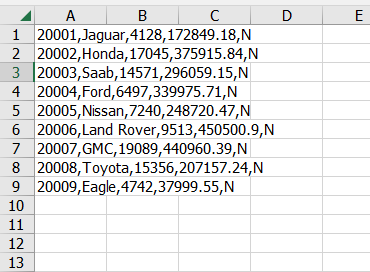
2. Radiobutton służący do usuwania wpisów z tabeli z samochodami na podstawie pliku CSV

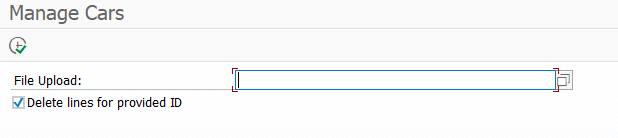
po kolumnie ID podanej w owym pliku

( kazda aktualizacje zapisz w tabeli

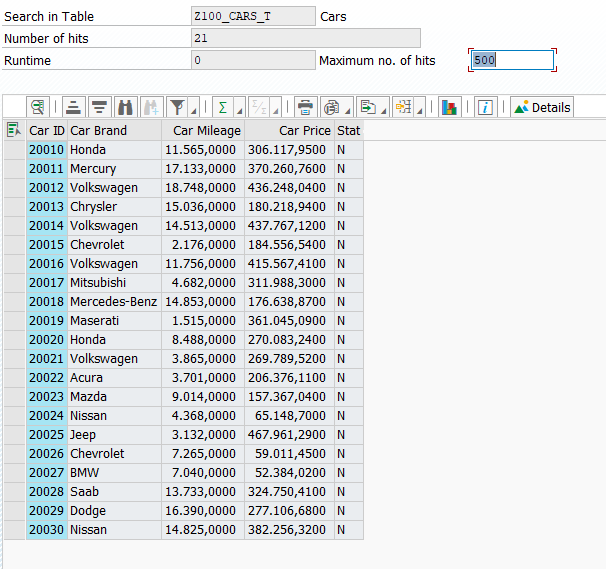
ZMJ\_LOGGER\_T )

Plik do załadowania (do usuniecia te od 20001 do 20009)

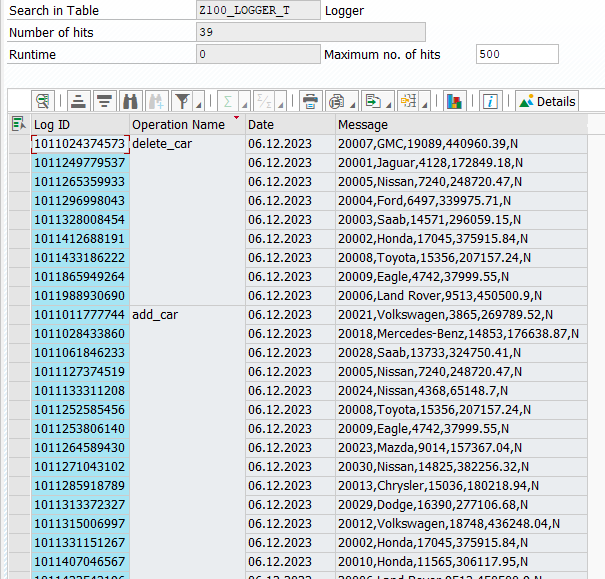




Po usunieciu: (nie ma 20001-9)



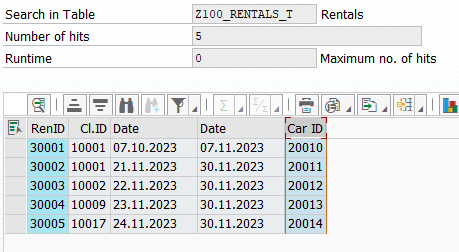
Logger:

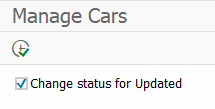


3. Radiobutton sluzacy do aktualizacji statusu -> ma sprawdzić czy jakiekolwiek auto,

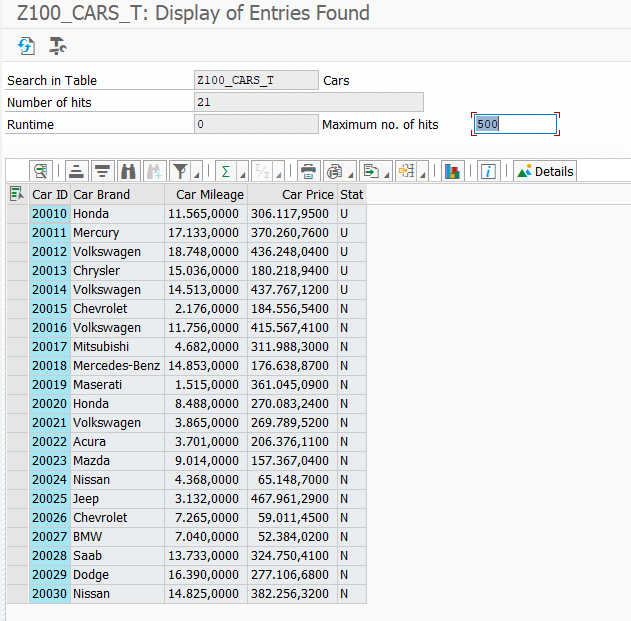
ze statusem ( N ) istnieje w tabeli z wypożeczeniami jeśli tak to zmień status tego auta na U

Zawartość Rentals (program powinien zmienic status dla 20010-20014)

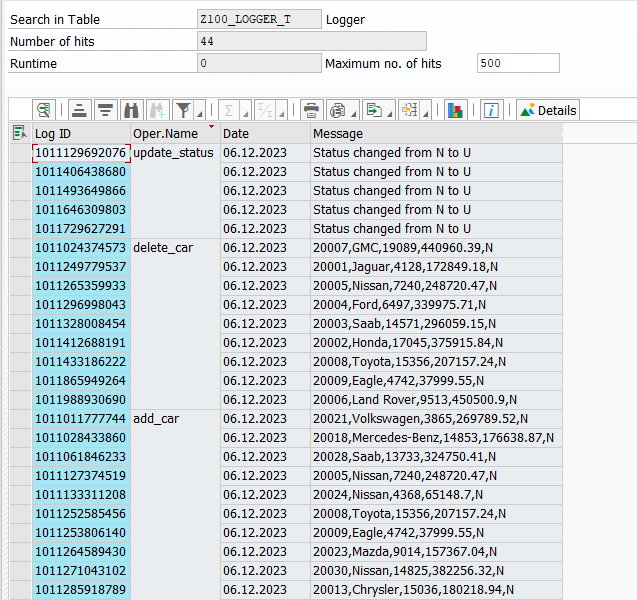




Po załadowaniu - status dla 20010-20014 zmieniony



Logger:



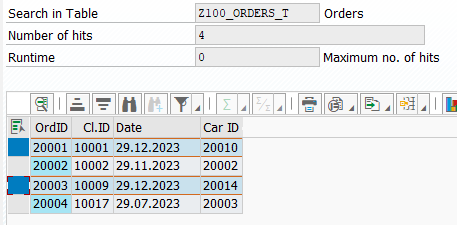
4. Radiobutton służący do czyszczenia wpisów, jeśli jakiekolwiek auto znajduje się

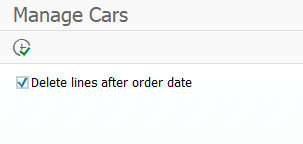
w tabeli z zamówieniami ( i obecny dzień jest dniem PO dacie zamówienia ) usuń ten wpis

z tabeli z samochodami. ( kazda aktualizacje zapisz w tabeli

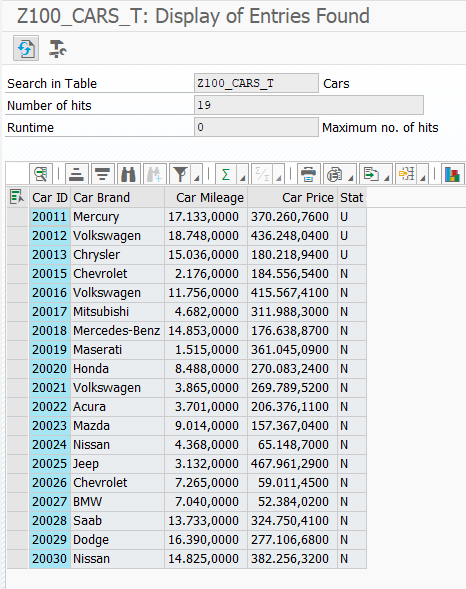
ZMJ\_LOGGER\_T )

Zawartość tabeli Orders - dla 20010 oraz 20014 data jest PO dniu dzisiejszym wiec powinny te dwa byc usuniete

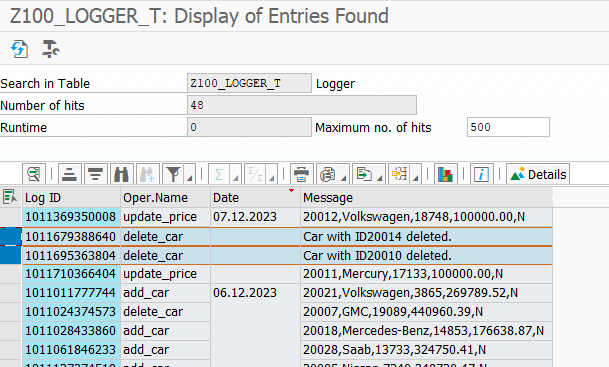




Po uruchomieniu programu - linie dla 20010 i 20014 usuniete



Logger:

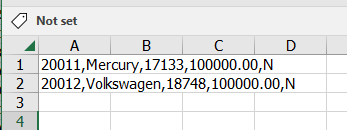


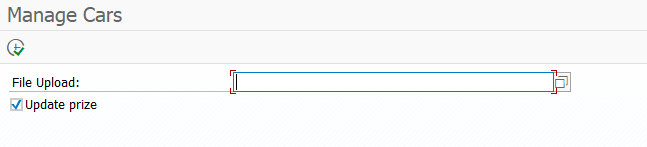
5. Radiobutton do aktualizacji cenowej również na podstawie pliku wejściowego opierający się

o ID. ( kazda aktualizacje zapisz w tabeli

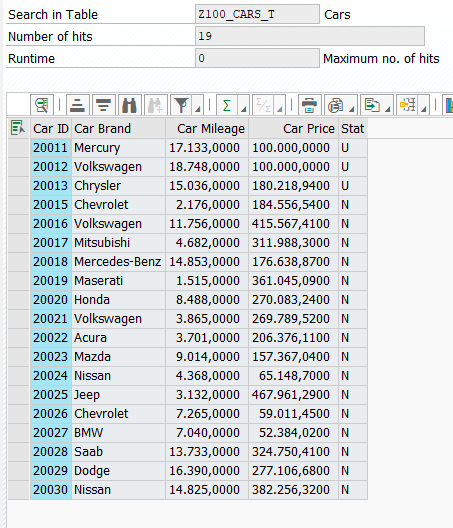
ZMJ\_LOGGER\_T )

Ładujemy plik z kwotami 100tys dla 20011 oraz 20012:

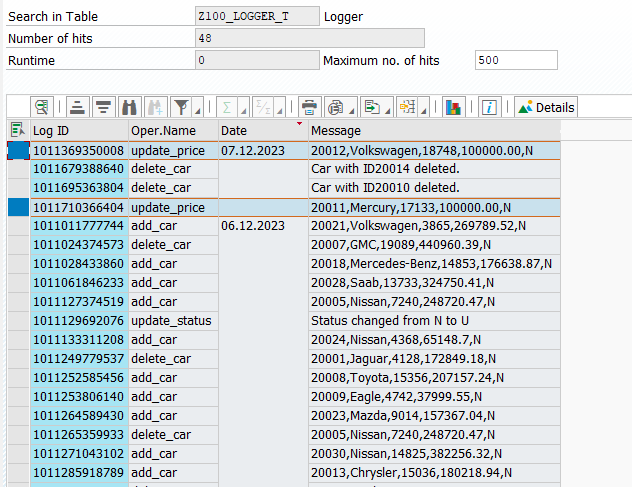




Po załadowaniu, ceny są nowe:

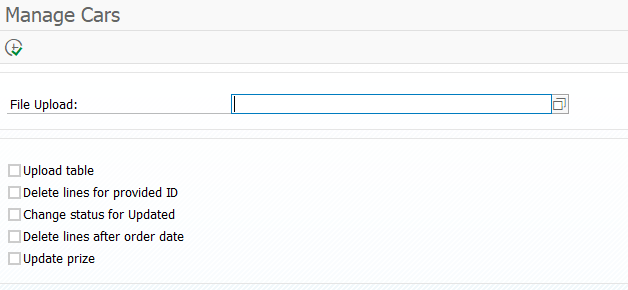


Logger:

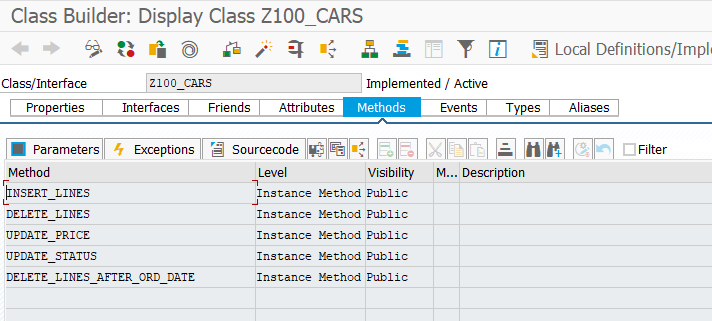


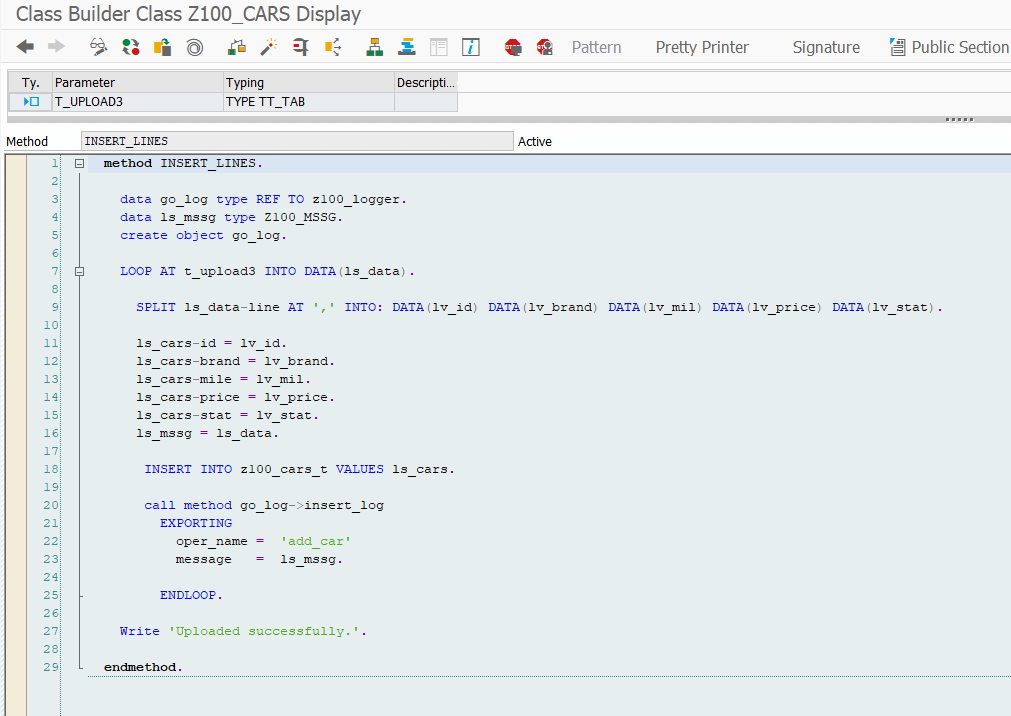
KOD RAPORTU

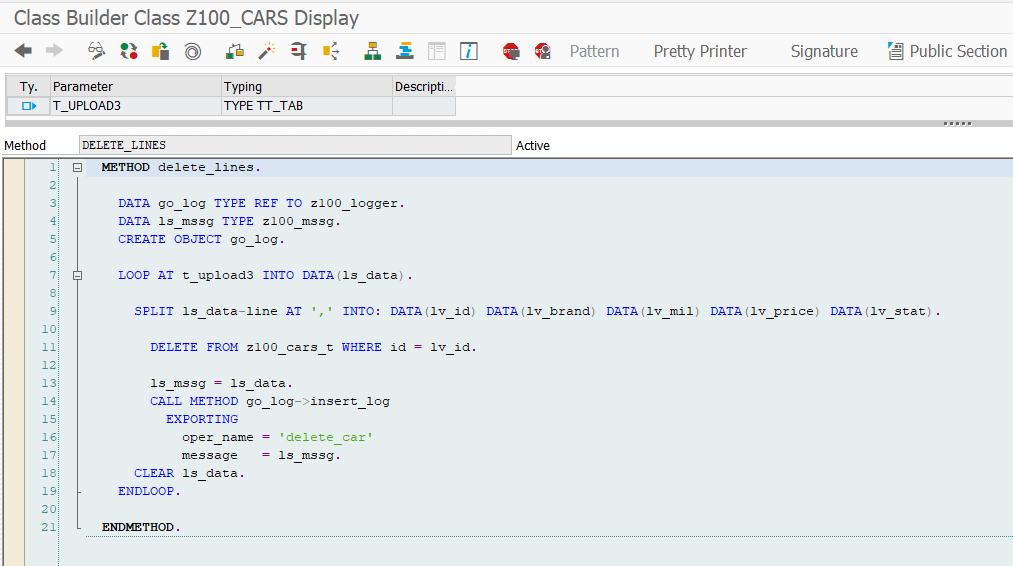
Report Z100\_MANAGE\_CARS.  
  
TYPE-POOLS truxs.  
TABLES z100\_clients\_t.  
  
TYPES: BEGIN OF t\_tab1,  
         line TYPE char255,  
       END OF t\_tab1.  
  
DATA: t\_upload1  TYPE STANDARD TABLE OF t\_tab1,  
      ls\_upload  TYPE t\_tab1,  
      lt\_filetab TYPE filetable,  
      lv\_rc      TYPE i.  
  
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-000.  
PARAMETERS: p\_file  TYPE   rlgrap-filename MODIF ID eid.  
SELECTION-SCREEN END OF BLOCK b1.  
  
SELECTION-SCREEN BEGIN OF BLOCK b2 WITH FRAME TITLE TEXT-001.  
PARAMETERS:  
  p\_chx1 AS CHECKBOX USER-COMMAND pay MODIF ID sc1,  
  p\_chx2 AS CHECKBOX USER-COMMAND sur MODIF ID sc2,  
  p\_chx3 AS CHECKBOX USER-COMMAND sur MODIF ID sc3,  
  p\_chx4 AS CHECKBOX USER-COMMAND sur MODIF ID sc4,  
  p\_chx5 AS CHECKBOX USER-COMMAND sta MODIF ID sc5.  
SELECTION-SCREEN END OF BLOCK b2.  
  
AT SELECTION-SCREEN OUTPUT.  
  LOOP AT SCREEN.  
    IF p\_chx1 EQ abap\_true.  
      p\_chx2 = abap\_false.  
      p\_chx3 = abap\_false.  
      p\_chx4 = abap\_false.  
      p\_chx5 = abap\_false.  
      IF screen-group1 EQ 'SC1' OR screen-group1 EQ 'EID'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
  
    IF p\_chx2 EQ abap\_true.  
      p\_chx1 = abap\_false.  
      p\_chx3 = abap\_false.  
      p\_chx4 = abap\_false.  
      p\_chx5 = abap\_false.  
      IF screen-group1 EQ 'SC2' OR screen-group1 EQ 'EID'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
  
    IF p\_chx3 EQ abap\_true.  
      p\_chx1 = abap\_false.  
      p\_chx2 = abap\_false.  
      p\_chx4 = abap\_false.  
      p\_chx5 = abap\_false.  
      IF screen-group1 EQ 'SC3'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
  
    IF p\_chx4 EQ abap\_true.  
      p\_chx1 = abap\_false.  
      p\_chx2 = abap\_false.  
      p\_chx3 = abap\_false.  
      p\_chx5 = abap\_false.  
      IF screen-group1 EQ 'SC4'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
  
     IF p\_chx5 EQ abap\_true.  
      p\_chx1 = abap\_false.  
      p\_chx2 = abap\_false.  
      p\_chx3 = abap\_false.  
      p\_chx4 = abap\_false.  
      IF screen-group1 EQ 'SC5' OR screen-group1 EQ 'EID'.  
        screen-active = 1.  
      ELSE.  
        screen-active = 0.  
      ENDIF.  
      MODIFY SCREEN.  
    ENDIF.  
  ENDLOOP.  
  
  AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_file.  
  
  CALL METHOD cl\_gui\_frontend\_services=>file\_open\_dialog  
    CHANGING  
      file\_table              = lt\_filetab  
      rc                      = lv\_rc  
    EXCEPTIONS  
      file\_open\_dialog\_failed = 1  
      cntl\_error              = 2  
      error\_no\_gui            = 3  
      not\_supported\_by\_gui    = 4  
      OTHERS                  = 5.  
  
  p\_file = lt\_filetab[ 1 ]-filename.  
  
START-OF-SELECTION.  
  
  DATA: go\_cars TYPE REF TO z100\_cars,  
        go\_table type REF TO z100\_tables.  
  
  CREATE OBJECT go\_cars.  
  CREATE OBJECT go\_table.  
  
 IF p\_chx1 IS NOT INITIAL.  
  
    CALL METHOD go\_table->upload\_table  
      EXPORTING  
        p\_file1  = p\_file  
      IMPORTING  
        t\_upload = t\_upload1.  
  
    CALL METHOD go\_cars->insert\_lines  
      EXPORTING  
        t\_upload3 = t\_upload1.  
  
ENDIF.  
  
IF p\_chx2 IS NOT INITIAL.  
  
    CALL METHOD go\_table->upload\_table  
      EXPORTING  
        p\_file1  = p\_file  
      IMPORTING  
        t\_upload = t\_upload1.  
  
    CALL METHOD go\_cars->delete\_lines  
      IMPORTING  
        t\_upload3 = t\_upload1.  
  
ENDIF.  
  
 IF p\_chx3 IS NOT INITIAL.  
  
    CALL METHOD go\_table->upload\_table  
      EXPORTING  
        p\_file1  = p\_file  
      IMPORTING  
        t\_upload = t\_upload1.  
  
    CALL METHOD go\_cars->update\_status.  
  
ENDIF.  
  
IF p\_chx4 IS NOT INITIAL.  
  
    CALL METHOD go\_cars->delete\_lines\_after\_ord\_date.  
  
ENDIF.  
  
 IF p\_chx5 IS NOT INITIAL.  
  
    CALL METHOD go\_table->upload\_table  
      EXPORTING  
        p\_file1  = p\_file  
      IMPORTING  
        t\_upload = t\_upload1.  
  
    CALL METHOD go\_cars->update\_price  
      IMPORTING  
        t\_upload3 = t\_upload1.  
  
ENDIF.

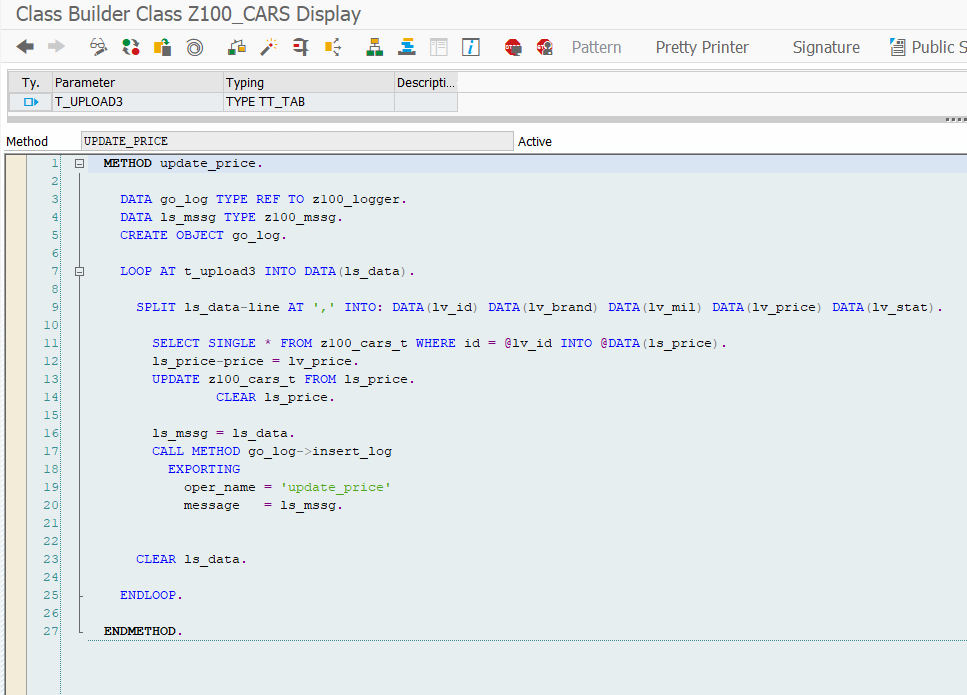


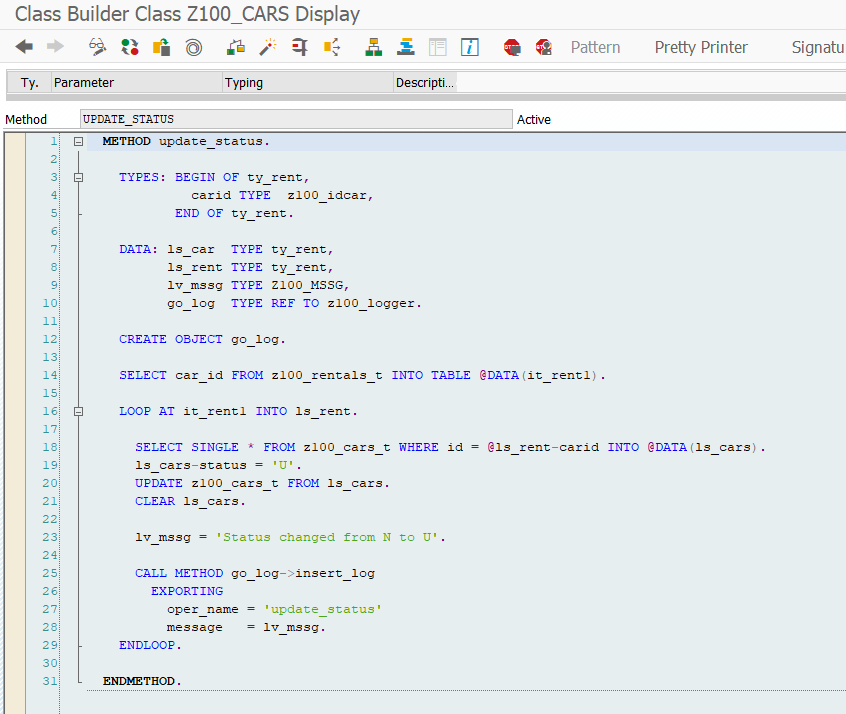
KLASY:

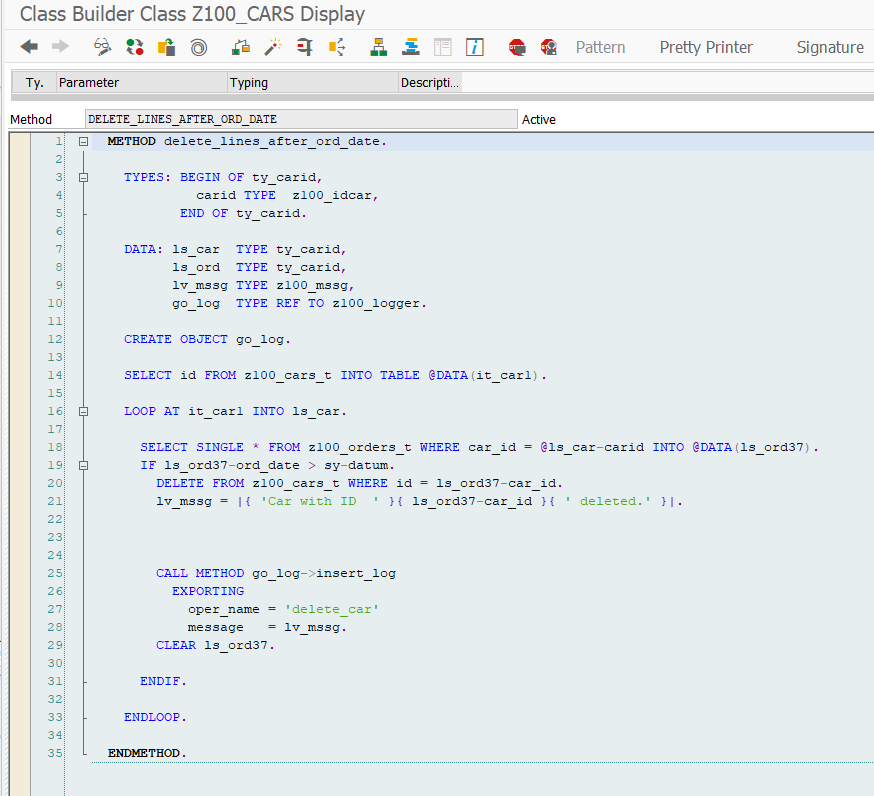












Zrobiłem tez osobne klasy dla Upload Table oraz Logger (to poprawione po Zad1)

