REPORT ZNYN\_OOP\_DEMO\_27.  
  
\* Class with functional method  
CLASS LCL\_AIRPLANE DEFINITION.  
  PUBLIC SECTION.  
    CONSTANTS POS\_1 TYPE I VALUE 30.  
    DATA PUB\_ATTR  TYPE I.  
    METHODS: SET\_ATTRIBUTES IMPORTING  
                              IM\_NAME      TYPE STRING  
                              IM\_PLANETYPE TYPE SAPLANE-PLANETYPE,  
      DISPLAY\_ATTRIBUTES,  
      GET\_NO\_AIRPLANES RETURNING VALUE(RE\_COUNT) TYPE I.  
  
    CLASS-METHODS DISPLAY\_NO\_AIRPLANES.  
  
  PRIVATE SECTION.  
    DATA: NAME      TYPE STRING,  
          PLANETYPE TYPE SAPLANE-PLANETYPE.  
    CLASS-DATA NO\_AIRPLANES TYPE I.  
  
ENDCLASS.                    "lcl\_airplane DEFINITION  
  
\*------------------------------------------------------------------\*  
\*       CLASS lcl\_airplane IMPLEMENTATION                          \*  
\*------------------------------------------------------------------\*  
CLASS LCL\_AIRPLANE IMPLEMENTATION.  
  
  METHOD SET\_ATTRIBUTES.  
    NAME          = IM\_NAME.  
    PLANETYPE     = IM\_PLANETYPE.  
    NO\_AIRPLANES = NO\_AIRPLANES + 1.  
  ENDMETHOD.                    "set\_attributes  
  
  METHOD DISPLAY\_ATTRIBUTES.  
    WRITE: / 'Name of airplane:', AT POS\_1 NAME,  
           / 'Type of airplane', AT POS\_1 PLANETYPE.  
  ENDMETHOD.                    "display\_attributes  
  
  METHOD DISPLAY\_NO\_AIRPLANES.  
    WRITE: / 'Total number of airplanes',  
           AT POS\_1 NO\_AIRPLANES LEFT-JUSTIFIED.  
  ENDMETHOD.                    "display\_no\_airplanes  
  
  METHOD GET\_NO\_AIRPLANES.  
    RE\_COUNT = NO\_AIRPLANES.  
  ENDMETHOD.  
ENDCLASS.                    "lcl\_airplane IMPLEMENTATION  
  
DATA: R\_PLANE1 TYPE REF TO LCL\_AIRPLANE,  
      R\_PLANE2 TYPE REF TO LCL\_AIRPLANE,  
      R\_PLANE3 TYPE REF TO LCL\_AIRPLANE,  
      COUNT    TYPE I.  
  
START-OF-SELECTION.  
  
  LCL\_AIRPLANE=>DISPLAY\_NO\_AIRPLANES( ).  
  
  CREATE OBJECT R\_PLANE1.  
  R\_PLANE1->SET\_ATTRIBUTES( IM\_NAME = 'LH Berlin'  
                           IM\_PLANETYPE = 'A321' ).  
  
  CREATE OBJECT R\_PLANE2.  
  R\_PLANE2->SET\_ATTRIBUTES( IM\_NAME = 'AA New York'  
                           IM\_PLANETYPE = '747-400' ).  
  
  CREATE OBJECT R\_PLANE3.  
  R\_PLANE3->SET\_ATTRIBUTES( IM\_NAME = 'US Hercules'  
                          IM\_PLANETYPE = '747-500' ).  
  
\* long syntax for functional call:  
  CALL METHOD r\_plane1->get\_no\_airplanes  
   RECEIVING  
     re\_count = count.  
  WRITE: / 'The count is', COUNT.  
  
  " a little bit shorter:  
  r\_plane1->get\_no\_airplanes( RECEIVING re\_count = count ).  
  WRITE: / 'The count is', COUNT.  
  
\* the shortest syntax for functional call:  
  COUNT = R\_PLANE1->GET\_NO\_AIRPLANES( ).  
\*  SKIP 2.  
  WRITE: / 'The count is', COUNT. 