\*&---------------------------------------------------------------------\*

\*& Include ZZ\_3325\_COMPANY\_5\_CLASSES

\*&---------------------------------------------------------------------\*

CLASS lcl\_employee DEFINITION ABSTRACT.

PUBLIC SECTION.

EVENTS employee\_created.

METHODS: constructor IMPORTING im\_name TYPE string im\_firstname TYPE string im\_base\_salary TYPE i im\_dob TYPE d,

set\_name IMPORTING im\_name TYPE string,

set\_firstname IMPORTING im\_firstname TYPE string,

set\_adress IMPORTING im\_adress TYPE string,

set\_dob IMPORTING im\_dob TYPE d,

get\_name RETURNING VALUE(re\_name) TYPE string,

get\_firstname RETURNING VALUE(re\_firstname) TYPE string,

get\_adress RETURNING VALUE(re\_adress) TYPE string,

get\_dob RETURNING VALUE(re\_dob) TYPE d,

get\_base\_salary RETURNING VALUE(re\_base\_salary) TYPE i,

get\_salary RETURNING VALUE(re\_salary) TYPE i,

get\_full\_name RETURNING VALUE(re\_full\_name) TYPE string,

change\_base\_salary IMPORTING im\_amount TYPE i,

print,

print\_type ABSTRACT.

CLASS-METHODS get\_n\_o\_employee EXPORTING ex\_count TYPE i.

PRIVATE SECTION.

DATA: name TYPE string,

firstname TYPE string,

adress TYPE string,

base\_salary TYPE i,

dob TYPE d.

CLASS-DATA n\_o\_employee VALUE 0.

METHODS init\_base\_salary IMPORTING VALUE(im\_base\_salary) TYPE i.

ENDCLASS.

CLASS lcl\_office\_employee DEFINITION INHERITING FROM lcl\_employee.

PUBLIC SECTION.

METHODS: set\_office IMPORTING im\_office TYPE string,

get\_office RETURNING VALUE(re\_office) TYPE string,

constructor IMPORTING im\_office TYPE string

im\_name TYPE string

im\_firstname TYPE string

im\_base\_salary TYPE i

im\_dob TYPE d,

print REDEFINITION,

print\_type REDEFINITION.

PRIVATE SECTION.

DATA office TYPE string.

ENDCLASS.

CLASS lcl\_field\_staff DEFINITION INHERITING FROM lcl\_employee.

PUBLIC SECTION.

METHODS: constructor IMPORTING im\_percentage TYPE i im\_sales TYPE i

im\_name TYPE string

im\_firstname TYPE string

im\_base\_salary TYPE i

im\_dob TYPE d,

set\_percentage IMPORTING im\_percentage TYPE p,

get\_percentage RETURNING VALUE(re\_percentage) TYPE i,

set\_sales IMPORTING im\_sales TYPE p,

get\_sales RETURNING VALUE(re\_sales) type i,

get\_salary REDEFINITION,

print REDEFINITION,

print\_type REDEFINITION.

PRIVATE SECTION.

DATA percentage TYPE i.

DATA sales TYPE i.

ENDCLASS.

CLASS lcl\_department DEFINITION.

PUBLIC SECTION.

METHODS: constructor IMPORTING im\_department\_name TYPE string,

set\_department\_name IMPORTING im\_department\_name TYPE string,

get\_department\_name RETURNING VALUE(re\_department\_name) TYPE string,

add\_employee IMPORTING im\_add\_employee TYPE REF TO lcl\_employee,

avg\_salary RETURNING VALUE(re\_avg\_salary) TYPE i,

get\_avg\_percentage RETURNING VALUE(re\_avg\_percentage) TYPE i,

print\_department.

METHODS on\_employee\_created FOR EVENT employee\_created of lcl\_employee

IMPORTING sender.

PRIVATE SECTION.

DATA department\_name TYPE string.

DATA it\_employees TYPE TABLE OF REF TO lcl\_employee.

ENDCLASS.

CLASS lcl\_department IMPLEMENTATION.

METHOD constructor.

department\_name = im\_department\_name.

ENDMETHOD.

METHOD set\_department\_name.

department\_name = im\_department\_name.

ENDMETHOD.

METHOD get\_department\_name.

re\_department\_name = department\_name.

ENDMETHOD.

METHOD add\_employee.

APPEND im\_add\_employee TO it\_employees.

ENDMETHOD.

METHOD avg\_salary.

DATA: n\_employee TYPE i, "Anzahl Mitarbeiter in der Abteilung.

sum\_salary TYPE i VALUE 0. "Gehaltsumme alle Mitarbeiter in der Abteilung.

DATA r\_employee TYPE REF TO lcl\_employee.

LOOP AT it\_employees INTO r\_employee.

sum\_salary = sum\_salary + r\_employee->get\_salary( ).

WRITE: / sum\_salary.

ENDLOOP.

n\_employee = lines( it\_employees ).

re\_avg\_salary = sum\_salary / n\_employee.

WRITE: / 'Durchschnittsgehalt = ', re\_avg\_salary, '€'.

ENDMETHOD.

METHOD get\_avg\_percentage.

DATA: n\_field\_staff TYPE i VALUE 0, "Anzahl der Außendienstmitarbeiter.

sum\_provisionsatz TYPE i VALUE 0. "summe Provision alle Außenmitarbeiter.

DATA r\_employee TYPE REF TO lcl\_employee.

DATA r\_field\_staff TYPE REF TO lcl\_field\_staff.

LOOP AT it\_employees INTO r\_employee.

TRY.

\* ich schreibe die Referenz r\_employee in die Referenz r\_field\_staff.

\* Anweisungen, die nach der Zuweisung ?= stehen nur für r\_field\_staff ausgefürt werden.

r\_field\_staff ?= r\_employee.

n\_field\_staff = n\_field\_staff + 1.

sum\_provisionsatz = sum\_provisionsatz + r\_field\_staff->get\_percentage( ).

CATCH cx\_sy\_move\_cast\_error.

ENDTRY.

ENDLOOP.

re\_avg\_percentage = sum\_provisionsatz / n\_field\_staff.

WRITE: / 'Durchschnittlicher provisionssatz = ', re\_avg\_percentage.

ENDMETHOD.

METHOD on\_employee\_created.

APPEND sender TO it\_employees.

ENDMETHOD.

METHOD print\_department.

DATA r\_employee TYPE REF TO lcl\_employee.

WRITE: / 'Abteilung: ', get\_department\_name( ).

LOOP AT it\_employees INTO r\_employee.

r\_employee->print( ).

ENDLOOP.

ENDMETHOD.

ENDCLASS.

CLASS lcl\_employee IMPLEMENTATION.

METHOD constructor.

add 1 to n\_o\_employee.

name = im\_name.

firstname = im\_firstname.

base\_salary = im\_base\_salary.

dob = im\_dob.

RAISE EVENT employee\_created.

ENDMETHOD.

METHOD set\_name.

name = im\_name.

ENDMETHOD.

METHOD set\_firstname.

firstname = im\_firstname.

ENDMETHOD.

METHOD set\_adress.

adress = im\_adress.

ENDMETHOD.

METHOD set\_dob.

dob = im\_dob.

ENDMETHOD.

METHOD get\_name.

re\_name = name.

ENDMETHOD.

METHOD get\_firstname.

re\_firstname = firstname.

ENDMETHOD.

METHOD get\_adress.

re\_adress = adress.

ENDMETHOD.

METHOD get\_dob.

re\_dob = dob.

ENDMETHOD.

METHOD get\_full\_name.

CONCATENATE firstname name INTO re\_full\_name SEPARATED BY ' '.

ENDMETHOD.

METHOD init\_base\_salary.

base\_salary = 2000.

ENDMETHOD.

METHOD get\_base\_salary.

re\_base\_salary = base\_salary.

ENDMETHOD.

METHOD change\_base\_salary.

IF base\_salary IS INITIAL.

init\_base\_salary( base\_salary ).

ELSE.

base\_salary = 0.

ENDIF.

ENDMETHOD.

METHOD get\_salary.

DATA age TYPE i.

age = ( sy-datum - dob ) / 360.

IF age > 40 AND age < 50.

re\_salary = ( get\_base\_salary( ) \* 5 ) / 100 + get\_base\_salary( ).

ELSEIF age > 50.

re\_salary = ( get\_base\_salary( ) \* 10 ) / 100 + get\_base\_salary( ).

ELSE.

re\_salary = get\_base\_salary( ).

ENDIF.

ENDMETHOD.

METHOD get\_n\_o\_employee.

ex\_count = n\_o\_employee.

WRITE: /, / 'Es Wurden ', ex\_count, ' employee gefunden.'.

ENDMETHOD.

METHOD print.

print\_type( ).

WRITE: / 'Name: ', get\_full\_name( ), 30 'Adresse: ', adress, 70 'Gehalt: ', get\_salary( ), '€', 100 'Geburtsdatum: ', dob DD/MM/YYYY.

ENDMETHOD.

ENDCLASS.

CLASS lcl\_office\_employee IMPLEMENTATION.

METHOD constructor.

super->constructor( im\_name = im\_name

im\_firstname = im\_firstname

im\_base\_salary = im\_base\_salary

im\_dob = im\_dob ).

office = im\_office.

ENDMETHOD.

METHOD set\_office.

office = im\_office.

ENDMETHOD.

METHOD get\_office.

re\_office = office.

ENDMETHOD.

METHOD print\_type.

WRITE: / 'Innendienstmitarbeiter'.

ENDMETHOD.

METHOD print.

super->print( ).

WRITE: office.

ENDMETHOD.

ENDCLASS.

CLASS lcl\_field\_staff IMPLEMENTATION.

METHOD constructor.

super->constructor( im\_name = im\_name

im\_firstname = im\_firstname

im\_base\_salary = im\_base\_salary

im\_dob = im\_dob ).

percentage = im\_percentage.

sales = im\_sales.

ENDMETHOD.

METHOD set\_sales.

sales = im\_sales.

ENDMETHOD.

METHOD get\_sales.

re\_sales = sales.

ENDMETHOD.

METHOD set\_percentage.

percentage = im\_percentage.

ENDMETHOD.

METHOD get\_percentage.

re\_percentage = percentage.

ENDMETHOD.

METHOD get\_salary.

re\_salary = get\_base\_salary( ) + sales \* percentage / 100.

ENDMETHOD.

METHOD print\_type.

WRITE: / 'Außendienstmitarbeiter'.

ENDMETHOD.

METHOD print.

super->print( ).

write: 'verkäufe: ', sales, 'Prov. ', percentage.

ENDMETHOD.

ENDCLASS.