## 1. Simple class

This example shows how to create a simple employee class. The constructor method is used to initialize number and name of the employee when the object is created. A display\_employee method can be called to show the attributes of the employee, and CLASS-METHOD display\_no\_of\_employees can be called to show the total number of employees (Number of instances of the employee class).

REPORT zbc404\_hf\_events\_1.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* L C L \_ E M P L O Y E

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*---- LCL Employee - Definition

CLASS lcl\_employee DEFINITION.

PUBLIC SECTION.

\*--------------------------------------------------------------------

\* The public section is accessible from outside

\*--------------------------------------------------------------------

TYPES:

BEGIN OF t\_employee,

no TYPE i,

name TYPE string,

END OF t\_employee.

METHODS:

constructor

IMPORTING im\_employee\_no TYPE i

im\_employee\_name TYPE string,

display\_employee.

\* Class methods are global for all instances

CLASS-METHODS: display\_no\_of\_employees.

PROTECTED SECTION.

\*--------------------------------------------------------------------

\* The protected section is accessible from the class and its subclasses

\*--------------------------------------------------------------------

\* Class data are global for all instances

CLASS-DATA: g\_no\_of\_employees TYPE i.

PRIVATE SECTION.

\*--------------------------------------------------------------------

\* The private section is only accessible from within the class

\*--------------------------------------------------------------------

DATA: g\_employee TYPE t\_employee.

ENDCLASS.

\*--- LCL Employee - Implementation

CLASS lcl\_employee IMPLEMENTATION.

METHOD constructor.

g\_employee-no = im\_employee\_no.

g\_employee-name = im\_employee\_name.

g\_no\_of\_employees = g\_no\_of\_employees + 1.

ENDMETHOD.

METHOD display\_employee.

WRITE:/ 'Employee', g\_employee-no, g\_employee-name.

ENDMETHOD.

METHOD display\_no\_of\_employees.

WRITE: / 'Number of employees is:', g\_no\_of\_employees.

ENDMETHOD.

ENDCLASS.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* R E P O R T

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA: g\_employee1 TYPE REF TO lcl\_employee,

g\_employee2 TYPE REF TO lcl\_employee.

START-OF-SELECTION.

CREATE OBJECT g\_employee1

EXPORTING im\_employee\_no = 1

im\_employee\_name = 'John Jones'.

CREATE OBJECT g\_employee2

EXPORTING im\_employee\_no = 2

im\_employee\_name = 'Sally Summer'.

CALL METHOD g\_employee1->display\_employee.

CALL METHOD g\_employee2->display\_employee.

CALL METHOD g\_employee2->display\_no\_of\_employees.

## 2. Inheritance and polymorphism

This example uses a superclass lcl\_company\_employees and two subclasses lcl\_bluecollar\_employee and lcl\_whitecollar\_employee to add employees to a list and then display a list of employees and there wages. The wages are calculated in the method add\_employee, but as the wages are calculated differently for blue collar employees and white collar emplyees, the superclass method add\_employee is redefined in the subclasses.

Principles:

Create super class LCL\_CompanyEmployees.

The class has the methods:

Constructor

Add\_Employee - Adds a new employee to the list of employees

Display\_Employee\_List - Displays all employees and there wage

Display\_no\_of\_employees - Displays total number of employees

Note the use of CLASS-DATA to keep the list of employees and number of employees the same from instance to instance.

Create subclasses lcl\_bluecollar\_employee and lcl\_whitecollar\_employee. The classes are identical, except for the redefinition of the add\_employee method, where the calculation of wage is different.

Methodes:

Constructor. The constructor is used to initialize the attributes of the employee. Note that the constructor in the supclasss has to be called from within the constructor of the subclass.

Add\_Employee. This is a redinition of the same method in the superclass. In the redefined class the wage is calculated, and the superclass method is called to add the employees to the employee list.

REPORT zbc404\_hf\_events\_2 .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Super class LCL\_CompanyEmployees

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_company\_employees DEFINITION.**

PUBLIC SECTION.

TYPES:

BEGIN OF t\_employee,

no TYPE i,

name TYPE string,

wage TYPE i,

END OF t\_employee.

METHODS:

constructor,

add\_employee

IMPORTING im\_no TYPE i

im\_name TYPE string

im\_wage TYPE i,

display\_employee\_list,

display\_no\_of\_employees.

PRIVATE SECTION.

CLASS-DATA: i\_employee\_list TYPE TABLE OF t\_employee,

no\_of\_employees TYPE i.

**ENDCLASS.**

\*-- CLASS LCL\_CompanyEmployees IMPLEMENTATION

**CLASS lcl\_company\_employees IMPLEMENTATION.**

METHOD constructor.

no\_of\_employees = no\_of\_employees + 1.

ENDMETHOD.

METHOD add\_employee.

\* Adds a new employee to the list of employees

DATA: l\_employee TYPE t\_employee.

l\_employee-no = im\_no.

l\_employee-name = im\_name.

l\_employee-wage = im\_wage.

APPEND l\_employee TO i\_employee\_list.

ENDMETHOD.

METHOD display\_employee\_list.

\* Displays all employees and there wage

DATA: l\_employee TYPE t\_employee.

WRITE: / 'List of Employees'.

LOOP AT i\_employee\_list INTO l\_employee.

WRITE: / l\_employee-no, l\_employee-name, l\_employee-wage.

ENDLOOP.

ENDMETHOD.

METHOD display\_no\_of\_employees.

\* Displays total number of employees

SKIP 3.

WRITE: / 'Total number of employees:', no\_of\_employees.

ENDMETHOD.

**ENDCLASS.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Sub class LCL\_BlueCollar\_Employee

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_bluecollar\_employee DEFINITION**

**INHERITING FROM lcl\_company\_employees.**

PUBLIC SECTION.

METHODS:

constructor

IMPORTING im\_no TYPE i

im\_name TYPE string

im\_hours TYPE i

im\_hourly\_payment TYPE i,

add\_employee REDEFINITION.

PRIVATE SECTION.

DATA:no TYPE i,

name TYPE string,

hours TYPE i,

hourly\_payment TYPE i.

**ENDCLASS.**

\*---- CLASS LCL\_BlueCollar\_Employee IMPLEMENTATION

**CLASS lcl\_bluecollar\_employee IMPLEMENTATION.**

METHOD constructor.

\* The superclass constructor method must be called from the subclass

\* constructor method

CALL METHOD super->constructor.

no = im\_no.

name = im\_name.

hours = im\_hours.

hourly\_payment = im\_hourly\_payment.

ENDMETHOD.

METHOD add\_employee.

\* Calculate wage an call the superclass method add\_employee to add

\* the employee to the employee list

DATA: l\_wage TYPE i.

l\_wage = hours \* hourly\_payment.

CALL METHOD super->add\_employee

EXPORTING im\_no = no

im\_name = name

im\_wage = l\_wage.

ENDMETHOD.

ENDCLASS.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Sub class LCL\_WhiteCollar\_Employee

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_whitecollar\_employee DEFINITION**

**INHERITING FROM lcl\_company\_employees.**

PUBLIC SECTION.

METHODS:

constructor

IMPORTING im\_no TYPE i

im\_name TYPE string

im\_monthly\_salary TYPE i

im\_monthly\_deductions TYPE i,

add\_employee REDEFINITION.

PRIVATE SECTION.

DATA:

no TYPE i,

name TYPE string,

monthly\_salary TYPE i,

monthly\_deductions TYPE i.

**ENDCLASS.**

\*---- CLASS LCL\_WhiteCollar\_Employee IMPLEMENTATION

**CLASS lcl\_whitecollar\_employee IMPLEMENTATION.**

METHOD constructor.

\* The superclass constructor method must be called from the subclass

\* constructor method

CALL METHOD super->constructor.

no = im\_no.

name = im\_name.

monthly\_salary = im\_monthly\_salary.

monthly\_deductions = im\_monthly\_deductions.

ENDMETHOD.

METHOD add\_employee.

\* Calculate wage an call the superclass method add\_employee to add

\* the employee to the employee list

DATA: l\_wage TYPE i.

l\_wage = monthly\_salary - monthly\_deductions.

CALL METHOD super->add\_employee

EXPORTING im\_no = no

im\_name = name

im\_wage = l\_wage.

ENDMETHOD.

**ENDCLASS.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* R E P O R T

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA:

\* Object references

o\_bluecollar\_employee1 TYPE REF TO lcl\_bluecollar\_employee,

o\_whitecollar\_employee1 TYPE REF TO lcl\_whitecollar\_employee.

START-OF-SELECTION.

\* Create blue-collar employee obeject

CREATE OBJECT o\_bluecollar\_employee1

EXPORTING im\_no = 1

im\_name = ‘Karen Johnson’

im\_hours = 38

im\_hourly\_payment = 75.

\* Add blue-collar employee to employee list

CALL METHOD o\_bluecollar\_employee1->add\_employee

EXPORTING im\_no = 1

im\_name = ‘Karen Johnson’

im\_wage = 0.

\* Create whitecollar employee obeject

CREATE OBJECT o\_whitecollar\_employee1

EXPORTING im\_no = 2

im\_name = 'John Dickens'

im\_monthly\_salary = 10000

im\_monthly\_deductions = 2500.

\* Add bluecollar employee to employee list

CALL METHOD o\_whitecollar\_employee1->add\_employee

EXPORTING im\_no = 1

im\_name = 'Karen Johnson'

im\_wage = 0.

\* Display employee list and number of employees. Note that the result

\* will be the same when called from o\_whitecollar\_employee1 or

\* o\_bluecolarcollar\_employee1, because the methods are defined

\* as static (CLASS-METHODS)

CALL METHOD o\_whitecollar\_employee1->display\_employee\_list.

CALL METHOD o\_whitecollar\_employee1->display\_no\_of\_employees.

The resulting report

List of Employees   
1 Karen Johnson 2.850   
2 John Dickens 7.500   
  
  
Total number of employees: 2

## 3. Interfaces

This example is similar to the previous example; however, an interface is implemented with the method add\_employee. Note that the interface is only implemented in the superclass ( The INTERFACE statement), but also used in the subclasses.

The interface in the example only contains a method, but an interface can also contain attributes, constants, types and alias names.

The output from example 3 is similar to the output in example 2.

All changes in the program compared to example 2 are marked with red.

REPORT zbc404\_hf\_events\_3 .

\*---------------------------------------------------------------------\*

\* INTERFACE lif\_employee

\*---------------------------------------------------------------------\*

INTERFACE lif\_employee.

METHODS:

add\_employee

IMPORTING im\_no TYPE i

im\_name TYPE string

im\_wage TYPE i.

ENDINTERFACE.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Super class LCL\_CompanyEmployees

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_company\_employees DEFINITION.**

PUBLIC SECTION.

INTERFACES lif\_employee.

TYPES:

BEGIN OF t\_employee,

no TYPE i,

name TYPE string,

wage TYPE i,

END OF t\_employee.

METHODS:

constructor,

\* add\_employee "Removed

IMPORTING im\_no TYPE i

im\_name TYPE string

im\_wage TYPE i,

display\_employee\_list,

display\_no\_of\_employees.

PRIVATE SECTION.

CLASS-DATA: i\_employee\_list TYPE TABLE OF t\_employee,

no\_of\_employees TYPE i.

**ENDCLASS.**

\*-- CLASS LCL\_CompanyEmployees IMPLEMENTATION

**CLASS lcl\_company\_employees IMPLEMENTATION.**

METHOD constructor.

no\_of\_employees = no\_of\_employees + 1.

ENDMETHOD.

METHOD lif\_employee~add\_employee.

\* Adds a new employee to the list of employees

DATA: l\_employee TYPE t\_employee.

l\_employee-no = im\_no.

l\_employee-name = im\_name.

l\_employee-wage = im\_wage.

APPEND l\_employee TO i\_employee\_list.

ENDMETHOD.

METHOD display\_employee\_list.

\* Displays all employees and there wage

DATA: l\_employee TYPE t\_employee.

WRITE: / 'List of Employees'.

LOOP AT i\_employee\_list INTO l\_employee.

WRITE: / l\_employee-no, l\_employee-name, l\_employee-wage.

ENDLOOP.

ENDMETHOD.

METHOD display\_no\_of\_employees.

\* Displays total number of employees

SKIP 3.

WRITE: / 'Total number of employees:', no\_of\_employees.

ENDMETHOD.

**ENDCLASS.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Sub class LCL\_BlueCollar\_Employee

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_bluecollar\_employee DEFINITION**

**INHERITING FROM lcl\_company\_employees.**

PUBLIC SECTION.

METHODS:

constructor

IMPORTING im\_no TYPE i

im\_name TYPE string

im\_hours TYPE i

im\_hourly\_payment TYPE i,

lif\_employee~add\_employee REDEFINITION..

PRIVATE SECTION.

DATA:no TYPE i,

name TYPE string,

hours TYPE i,

hourly\_payment TYPE i.

**ENDCLASS.**

\*---- CLASS LCL\_BlueCollar\_Employee IMPLEMENTATION

**CLASS lcl\_bluecollar\_employee IMPLEMENTATION.**

METHOD constructor.

\* The superclass constructor method must be called from the subclass

\* constructor method

CALL METHOD super->constructor.

no = im\_no.

name = im\_name.

hours = im\_hours.

hourly\_payment = im\_hourly\_payment.

ENDMETHOD.

METHOD lif\_employee~add\_employee.

\* Calculate wage an call the superclass method add\_employee to add

\* the employee to the employee list

DATA: l\_wage TYPE i.

l\_wage = hours \* hourly\_payment.

CALL METHOD super->lif\_employee~add\_employee

EXPORTING im\_no = no

im\_name = name

im\_wage = l\_wage.

ENDMETHOD.

ENDCLASS.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Sub class LCL\_WhiteCollar\_Employee

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_whitecollar\_employee DEFINITION**

**INHERITING FROM lcl\_company\_employees.**

PUBLIC SECTION.

METHODS:

constructor

IMPORTING im\_no TYPE i

im\_name TYPE string

im\_monthly\_salary TYPE i

im\_monthly\_deductions TYPE i,

lif\_employee~add\_employee REDEFINITION.

PRIVATE SECTION.

DATA:

no TYPE i,

name TYPE string,

monthly\_salary TYPE i,

monthly\_deductions TYPE i.

**ENDCLASS.**

\*---- CLASS LCL\_WhiteCollar\_Employee IMPLEMENTATION

**CLASS lcl\_whitecollar\_employee IMPLEMENTATION.**

METHOD constructor.

\* The superclass constructor method must be called from the subclass

\* constructor method

CALL METHOD super->constructor.

no = im\_no.

name = im\_name.

monthly\_salary = im\_monthly\_salary.

monthly\_deductions = im\_monthly\_deductions.

ENDMETHOD.

METHOD lif\_employee~add\_employee.

\* Calculate wage an call the superclass method add\_employee to add

\* the employee to the employee list

DATA: l\_wage TYPE i.

l\_wage = monthly\_salary - monthly\_deductions.

CALL METHOD super->lif\_employee~add\_employee

EXPORTING im\_no = no

im\_name = name

im\_wage = l\_wage.

ENDMETHOD.

**ENDCLASS.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* R E P O R T

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA:

\* Object references

o\_bluecollar\_employee1 TYPE REF TO lcl\_bluecollar\_employee,

o\_whitecollar\_employee1 TYPE REF TO lcl\_whitecollar\_employee.

START-OF-SELECTION.

\* Create bluecollar employee obeject

CREATE OBJECT o\_bluecollar\_employee1

EXPORTING im\_no = 1

im\_name = ‘Karen Johnson’

im\_hours = 38

im\_hourly\_payment = 75.

\* Add bluecollar employee to employee list

CALL METHOD o\_bluecollar\_employee1->lif\_employee~add\_employee

EXPORTING im\_no = 1

im\_name = 'Karen Johnson'

im\_wage = 0.

\* Create whitecollar employee obeject

CREATE OBJECT o\_whitecollar\_employee1

EXPORTING im\_no = 2

im\_name = 'John Dickens'

im\_monthly\_salary = 10000

im\_monthly\_deductions = 2500.

\* Add bluecollar employee to employee list

CALL METHOD o\_whitecollar\_employee1->lif\_employee~add\_employee

EXPORTING im\_no = 1

im\_name = ‘Karen Johnson’

im\_wage = 0.

\* Display employee list and number of employees. Note that the result

\* will be the same when called from o\_whitecollar\_employee1 or

\* o\_bluecolarcollar\_employee1, because the methods are defined

\* as static (CLASS-METHODS)

CALL METHOD o\_whitecollar\_employee1->display\_employee\_list.

CALL METHOD o\_whitecollar\_employee1->display\_no\_of\_employees.

## 4. Events

This is the same example as example 4. All changes are marked with red. There have been no changes to the subclasses, only to the superclass and the report, subclasses are not shown.

For a simple example refer to Events in the examples.

REPORT zbc404\_hf\_events\_4.

\*---------------------------------------------------------------------\*

\* INTERFACE lif\_employee

\*---------------------------------------------------------------------\*

INTERFACE lif\_employee.

METHODS:

add\_employee

IMPORTING im\_no TYPE i

im\_name TYPE string

im\_wage TYPE i.

ENDINTERFACE.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Super class LCL\_CompanyEmployees

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_company\_employees DEFINITION.**

PUBLIC SECTION.

TYPES:

BEGIN OF t\_employee,

no TYPE i,

name TYPE string,

wage TYPE i,

END OF t\_employee.

\* Declare event. Note that declaration could also be placed in the

\* interface

EVENTS: employee\_added\_to\_list

EXPORTING value(ex\_employee\_name) TYPE string.

\* CLASS-EVENTS: Events can also be defined as class-events

INTERFACES lif\_employee.

METHODS:

constructor,

display\_employee\_list,

display\_no\_of\_employees,

\* Declare event method

on\_employee\_added\_to\_list FOR EVENT employee\_added\_to\_list OF lcl\_company\_employees

IMPORTING ex\_employee\_name sender.

PRIVATE SECTION.

CLASS-DATA:

i\_employee\_list TYPE TABLE OF t\_employee,

no\_of\_employees TYPE i.

**ENDCLASS.**

\*-- CLASS LCL\_CompanyEmployees IMPLEMENTATION

**CLASS lcl\_company\_employees IMPLEMENTATION.**

METHOD constructor.

no\_of\_employees = no\_of\_employees + 1.

ENDMETHOD.

METHOD lif\_employee~add\_employee.

\* Adds a new employee to the list of employees

DATA: l\_employee TYPE t\_employee.

l\_employee-no = im\_no.

l\_employee-name = im\_name.

l\_employee-wage = im\_wage.

APPEND l\_employee TO i\_employee\_list.

\* Raise event employee\_added\_to\_list

RAISE EVENT employee\_added\_to\_list

EXPORTING ex\_employee\_name = l\_employee-name.

ENDMETHOD.

METHOD display\_employee\_list.

\* Displays all employees and there wage

DATA: l\_employee TYPE t\_employee.

WRITE: / 'List of Employees'.

LOOP AT i\_employee\_list INTO l\_employee.

WRITE: / l\_employee-no, l\_employee-name, l\_employee-wage.

ENDLOOP.

ENDMETHOD.

METHOD display\_no\_of\_employees.

\* Displays total number of employees

SKIP 3.

WRITE: / 'Total number of employees:', no\_of\_employees.

ENDMETHOD.

METHOD on\_employee\_added\_to\_list.

\* Event method

WRITE: / 'Employee added to list', ex\_employee\_name.

ENDMETHOD.

**ENDCLASS.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Sub class LCL\_BlueCollar\_Employee

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_bluecollar\_employee DEFINITION**

**INHERITING FROM lcl\_company\_employees.**

See code in example 3...

**ENDCLASS.**

**CLASS lcl\_bluecollar\_employee IMPLEMENTATION.**

See code in example 3...

**ENDCLASS.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Sub class LCL\_WhiteCollar\_Employee

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CLASS lcl\_whitecollar\_employee DEFINITION**

See code in example 3...

**ENDCLASS.**

**CLASS lcl\_whitecollar\_employee IMPLEMENTATION.**

See code in example 3...

**ENDCLASS.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* R E P O R T

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATA:

\* Object references

o\_bluecollar\_employee1 TYPE REF TO lcl\_bluecollar\_employee,

o\_whitecollar\_employee1 TYPE REF TO lcl\_whitecollar\_employee.

START-OF-SELECTION.

\* Create bluecollar employee obeject

CREATE OBJECT o\_bluecollar\_employee1

EXPORTING im\_no = 1

im\_name = 'Karen Johnson'

im\_hours = 38

im\_hourly\_payment = 75.

\* Register event for o\_bluecollar\_employee1

SET HANDLER o\_bluecollar\_employee1->on\_employee\_added\_to\_list

FOR o\_bluecollar\_employee1.

\* Add bluecollar employee to employee list

CALL METHOD o\_bluecollar\_employee1->lif\_employee~add\_employee

EXPORTING im\_no = 1

im\_name = ‘Karen Johnson’

im\_wage = 0.

\* Create whitecollar employee obeject

CREATE OBJECT o\_whitecollar\_employee1

EXPORTING im\_no = 2

im\_name = 'John Dickens'

im\_monthly\_salary = 10000

im\_monthly\_deductions = 2500.

\* Register event for o\_whitecollar\_employee1

SET HANDLER o\_whitecollar\_employee1->on\_employee\_added\_to\_list

FOR o\_whitecollar\_employee1.´

\* Add bluecollar employee to employee list

CALL METHOD o\_whitecollar\_employee1->lif\_employee~add\_employee

EXPORTING im\_no = 1

im\_name = ‘Karen Johnson’

im\_wage = 0.

\* Display employee list and number of employees. Note that the result

\* will be the same when called from o\_whitecollar\_employee1 or

\* o\_bluecolarcollar\_employee1, because the methods are defined

\* as static (CLASS-METHODS)

CALL METHOD o\_whitecollar\_employee1->display\_employee\_list.

CALL METHOD o\_whitecollar\_employee1->display\_no\_of\_employees.

Result:

Employee added to list Karen Johnson   
Employee added to list John Dickens   
List of Employees   
1 Karen Johnson 2.850   
2 John Dickens 7.500   
  
  
Total number of employees: 2