SAP_ABAP_OO面向对象入门实例

```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A                                                    *
*&---------------------------------------------------------------------*
*& 简单ABAP对象                                                        *
*&---------------------------------------------------------------------*

REPORT  y_test_a                                .
*&---------------------------------------------------------------------*
*&       Class simpleobj
*&---------------------------------------------------------------------*
*        Text
*----------------------------------------------------------------------*
CLASS simpleobj DEFINITION.
  PUBLIC SECTION.
    METHODS: show_text.
  PRIVATE SECTION.
    DATA text(100) TYPE c VALUE 'This is my first ABAP object.'.
ENDCLASS.               "simpleobj
*&---------------------------------------------------------------------*
*&       Class (Implementation)  SIMPLEOBJ
*&---------------------------------------------------------------------*
*        Text
*----------------------------------------------------------------------*
CLASS simpleobj IMPLEMENTATION.
  METHOD show_text.
    WRITE text.
  ENDMETHOD.                    "show_text
ENDCLASS.               "SIMPLEOBJ

* Global Data Declaratioin
DATA ob_app TYPE REF TO simpleobj.

START-OF-SELECTION.
  CREATE OBJECT ob_app.
  CALL METHOD ob_app->show_text.
```

```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A_1                                                  *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&    类的静态属性                                                     *
```

```
*&---------------------------------------------------------------------*

REPORT  y_test_a_1                                .
*&---------------------------------------------------------------------*
*&       Class vehicle
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS vehicle DEFINITION.
  PUBLIC SECTION.
    CLASS-DATA class_name(10) VALUE 'Vehicle'.
    METHODS:accelerate,show_speed.
  PROTECTED SECTION.
    DATA speed TYPE i.
    CONSTANTS: pi TYPE p DECIMALS 2 VALUE '3.14'.
ENDCLASS.               "vehicle
*&---------------------------------------------------------------------*
*&       Class (Implementation)  vehicle
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS vehicle IMPLEMENTATION.
  METHOD accelerate.
    speed = speed + 1.
  ENDMETHOD.                    "accelerate
  METHOD show_speed.
    WRITE: / 'Speed:' ,speed.
  ENDMETHOD.                    "show_speed
ENDCLASS.               "vehicle

DATA ob_app1 TYPE REF TO vehicle.
DATA ob_app2 TYPE REF TO vehicle.
DATA ob_app3 TYPE REF TO vehicle.
DATA o_vehicle TYPE REF TO z_cl_vehicle. "引用全局类

START-OF-SELECTION.
  CREATE OBJECT ob_app1.
  CREATE OBJECT ob_app2.

  WRITE: 'ob_app1',ob_app1->class_name.
  WRITE: / 'ob_app2',ob_app2->class_name.
  WRITE: / 'vehicle',vehicle=>class_name.

  SKIP.

  ob_app1->class_name = 'Ship'.
  WRITE: / 'ob_app1',ob_app1->class_name.
  WRITE: / 'ob_app2',ob_app2->class_name.
  WRITE: / 'vehicle',vehicle=>class_name.

  SKIP.
  vehicle=>class_name = 'Bus'.
  CREATE OBJECT ob_app3.
```

```
  WRITE: / 'ob_app3',ob_app3->class_name.

  CALL METHOD ob_app3->accelerate.
  CALL METHOD ob_app3->show_speed.

  CREATE OBJECT o_vehicle.

  DO 5 TIMES.
    CALL METHOD o_vehicle->accelerate.
  ENDDO.
  CALL METHOD o_vehicle->show_speed.

  DATA o_vehicle2 LIKE o_vehicle.

  o_vehicle2 = o_vehicle.

  CLEAR o_vehicle.

  CALL METHOD o_vehicle2->show_speed.
  WRITE '对象只要有被引用的变量，就是活动的。'.
```

```
ob_app1 Vehicle
ob_app2 Vehicle
vehicle Vehicle

ob_app1 Ship
ob_app2 Ship
vehicle Ship

ob_app3 Bus
Speed:          1
speed:          5
speed:          5  对象只要有被引用的变量，就是活动的。
```

```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A_2                                                  *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&   类方法参数调用                                                    *
*&---------------------------------------------------------------------*

REPORT  y_test_a_2                                .
*&---------------------------------------------------------------------*
*&       Class vehicle
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS vehicle DEFINITION.
  PUBLIC SECTION.
    METHODS:exp_speed IMPORTING cname TYPE string
                      EXPORTING ispeed TYPE i,
            accelerate IMPORTING rate TYPE i,
            add CHANGING addone TYPE i.

  PRIVATE SECTION.
    DATA speed TYPE i VALUE 0.
ENDCLASS.                "vehicle
*&---------------------------------------------------------------------*
*&       Class (Implementation)  vehicle
```

```
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS vehicle IMPLEMENTATION.
  METHOD accelerate.
    speed = speed + rate.
  ENDMETHOD.                    "accelerate
  METHOD exp_speed.
    ispeed = speed.
    WRITE cname.
  ENDMETHOD.                    "show_speed
  METHOD add.
    addone = addone + 1.
  ENDMETHOD.                    "add
ENDCLASS.               "vehicle

DATA o_vehicle TYPE REF TO vehicle.
DATA int TYPE i VALUE 3.
DATA fname TYPE string VALUE 'Speed = '.

START-OF-SELECTION.
  CREATE OBJECT o_vehicle.
  CALL METHOD o_vehicle->accelerate
    EXPORTING
      rate = int.

  CALL METHOD o_vehicle->accelerate
    EXPORTING
      rate = int.

  CALL METHOD o_vehicle->exp_speed
    EXPORTING
      cname  = fname
    IMPORTING
      ispeed = int.
  WRITE:int.

  CALL METHOD o_vehicle->add
    CHANGING
      addone = int.
  WRITE: / int.
```

```
2011.02.13
Speed =         7         6
```

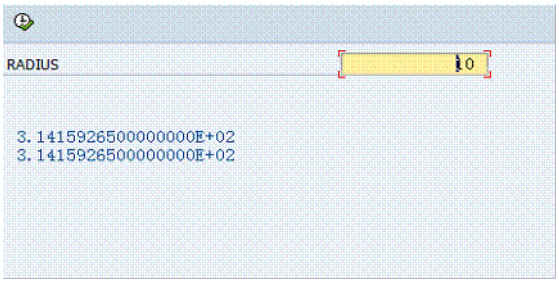
```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A_3                                                  *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&       类函数方法                                                    *
*&---------------------------------------------------------------------*
```

```
REPORT  y_test_a_3                                .
*&---------------------------------------------------------------------*
*&       Class circle
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS circle DEFINITION.
  PUBLIC SECTION.
    METHODS get_area IMPORTING value(i_radius) TYPE i
                     RETURNING value(r_size) TYPE f.
  PRIVATE SECTION.
    CONSTANTS pi TYPE f VALUE '3.14159265'.
ENDCLASS.                "circle
*&---------------------------------------------------------------------*
*&       Class (Implementation)  circle
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS circle IMPLEMENTATION.
  METHOD get_area.
    r_size = i_radius ** 2 * pi.
  ENDMETHOD.                    "get_area
ENDCLASS.                "circle

PARAMETERS radius TYPE i.
DATA: o_circle TYPE REF TO circle,
      area TYPE f.

START-OF-SELECTION.
  CREATE OBJECT o_circle.
  CALL METHOD o_circle->get_area
    EXPORTING
      i_radius = radius
    RECEIVING
      r_size   = area.
  WRITE: / area.

  area = o_circle->get_area( radius ).
  WRITE: / area.
```




```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A_4                                                  *
*&                                                                     *
*&---------------------------------------------------------------------*
```

```
*&                                                                     *
*&    类的构造方法                                                     *
*&---------------------------------------------------------------------*

REPORT  y_test_a_4                              .
*---------------------------------------------------------------------*
*       CLASS vehicle DEFINITION
*---------------------------------------------------------------------*
*
*---------------------------------------------------------------------*
CLASS vehicle DEFINITION.
  PUBLIC SECTION.
    METHODS: accelerate IMPORTING rate TYPE i,
             constructor IMPORTING i_speed TYPE i,
             show_speed.
  PRIVATE SECTION.
    DATA speed TYPE i VALUE 0.
ENDCLASS.                    "vehicle DEFINITION
*&---------------------------------------------------------------------*
*&       Class (Implementation)  vehicle
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS vehicle IMPLEMENTATION.
  METHOD accelerate.
    speed = speed + rate.
  ENDMETHOD.                    "accelertate
  METHOD show_speed.
    WRITE / speed.
  ENDMETHOD.                    "show_speed
  METHOD constructor.
    speed = i_speed.
    WRITE: 'constructor i_speed = ',speed.
  ENDMETHOD.                    "constructor
ENDCLASS.               "vehicle

DATA o_vehicle TYPE REF TO  vehicle.

START-OF-SELECTION.
  CREATE OBJECT o_vehicle EXPORTING i_speed = 4.
  CALL METHOD o_vehicle->accelerate
    EXPORTING
      rate = 2.
  CALL METHOD o_vehicle->show_speed.
```

```
2011.02.13
constructor i_speed =           4
          6
```

```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A_5                                                  *
```

```
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&     类的继承 多态                                                   *
*&---------------------------------------------------------------------*
REPORT  y_test_a_5                                .


*&---------------------------------------------------------------------*
*&       Class superclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS superclass DEFINITION.
  PUBLIC SECTION.
    METHODS write_first.
    METHODS write_second.
ENDCLASS.                "superclass


*&---------------------------------------------------------------------*
*&       Class subclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS subclass DEFINITION INHERITING FROM superclass.
  PUBLIC SECTION.
    METHODS write_third.
ENDCLASS.                "subclass

*&---------------------------------------------------------------------*
*&       Class redefclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS redefclass DEFINITION INHERITING FROM superclass.
  PUBLIC SECTION.
    METHODS:write_me,write_first REDEFINITION.
ENDCLASS.                "redefclass

*&---------------------------------------------------------------------*
*&       Class (Implementation)  superclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS superclass IMPLEMENTATION.
  METHOD write_first.
    WRITE: / 'The first method'.
  ENDMETHOD.                    "write_first

  METHOD write_second.
    WRITE: / 'The second method'.
  ENDMETHOD.                    "write_second
```

```
ENDCLASS.                "superclass
*&---------------------------------------------------------------------*
*&       Class (Implementation)  subclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS subclass IMPLEMENTATION.
  METHOD write_third.
    WRITE: / 'The third method'.
  ENDMETHOD.                    "write_third
ENDCLASS.                "subclass

*&---------------------------------------------------------------------*
*&       Class (Implementation)  REDEFCLASS
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS redefclass IMPLEMENTATION.
  METHOD write_me.
    CALL METHOD me->write_first.
  ENDMETHOD.                    "write_super
  METHOD write_first.
    WRITE: / 'The redefinition method'.
    CALL METHOD super->write_first.
  ENDMETHOD.                    "write_first
ENDCLASS.                "REDEFCLASS



DATA: inher_obj TYPE REF TO subclass.
DATA: redef_obj TYPE REF TO redefclass.

START-OF-SELECTION.
  CREATE OBJECT inher_obj.
  CREATE OBJECT redef_obj.
  CALL METHOD:inher_obj->write_first,
              inher_obj->write_second,
              inher_obj->write_third.
  SKIP.
  CALL METHOD:redef_obj->write_first.
  CALL METHOD:redef_obj->write_me.
```

```
The first method
The second method
The third method

The redefinition method
The first method
The redefinition method
The first method
```

```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A_6                                                  *
```

```
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&   类的抽象和抽象方法 最终和最终方法                                 *
*&---------------------------------------------------------------------*

REPORT  y_test_a_6                              .
*&---------------------------------------------------------------------*
*&       Class superclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS superclass DEFINITION ABSTRACT.
  PUBLIC SECTION.
    DATA:para(30) TYPE c VALUE 'The super abstract method'.
    METHODS write_first ABSTRACT.
ENDCLASS.               "superclass

*&---------------------------------------------------------------------*
*&       Class subclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS subclass DEFINITION INHERITING FROM superclass.
  PUBLIC SECTION.
    METHODS write_first REDEFINITION.
    METHODS write_finalmethod FINAL.
ENDCLASS.               "subclass

*&---------------------------------------------------------------------*
*&       Class finalclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS finalclass DEFINITION FINAL.
  PUBLIC SECTION.
    METHODS:write_finalclass.
ENDCLASS.               "finalclass


*&---------------------------------------------------------------------*
*&       Class (Implementation)  subclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS subclass IMPLEMENTATION .
  METHOD write_first.
    WRITE:para.
  ENDMETHOD.                    "write_first

  METHOD write_finalmethod .
    WRITE: / 'The final method'.
  ENDMETHOD.                    "write_sub
```

```
ENDCLASS.                    "subclass

*&---------------------------------------------------------------------*
*&       Class (Implementation)  finalclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS finalclass IMPLEMENTATION.
  METHOD write_finalclass.
    WRITE: / 'The final class'.
  ENDMETHOD.                    "write_finalclass
ENDCLASS.                    "finalclass

DATA inher_obj TYPE REF TO subclass.
DATA final_obj TYPE REF TO finalclass.

START-OF-SELECTION.
  CREATE OBJECT inher_obj.
  CREATE OBJECT final_obj.
  CALL METHOD inher_obj->write_first.
  CALL METHOD inher_obj->write_finalmethod.
  CALL METHOD final_obj->write_finalclass.
```

```
2011.02.13
The super abstract method
The final method
The final class
```

```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A_7                                                  *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&     类的接口                                                        *
*&---------------------------------------------------------------------*

REPORT  y_test_a_7                              .
*---------------------------------------------------------------------*
*       INTERFACE output
*---------------------------------------------------------------------*
*
*---------------------------------------------------------------------*
INTERFACE output.
  METHODS write.
ENDINTERFACE.                    "output

*---------------------------------------------------------------------*
*       INTERFACE status
*---------------------------------------------------------------------*
*
*---------------------------------------------------------------------*
```

```
INTERFACE status.
  DATA int TYPE i .
  CLASS-DATA cint TYPE i.
  METHODS write.
  CONSTANTS const TYPE i VALUE 30.
ENDINTERFACE.                    "status

*&---------------------------------------------------------------------*
*&       Class superclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS superclass DEFINITION.
  PUBLIC SECTION.
    INTERFACES :output,status,zintest.
    METHODS increment.
  PRIVATE SECTION.
    DATA count TYPE i.
ENDCLASS.               "superclass
*&---------------------------------------------------------------------*
*&       Class (Implementation)  superclass
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS superclass IMPLEMENTATION.
  METHOD output~write.
    WRITE / 'Hello SAP'.
  ENDMETHOD.                    "output~write
  METHOD status~write.
    WRITE: / 'Count in count is ',count.
  ENDMETHOD.                    "status~write
  METHOD zintest~test.
    WRITE: / 'it is the BADI test interface'.
  ENDMETHOD.                    "zintest~test
  METHOD increment.
    ADD 1 TO count.
  ENDMETHOD.                    "increment

ENDCLASS.               "superclass

DATA:super_obj TYPE REF TO superclass,
     super_object TYPE REF TO superclass.

DATA:intf_obj TYPE REF TO output,  "引用接口output
     intf_table TYPE TABLE OF REF TO output. "引用接口创建内表

DATA:interface_obj TYPE REF TO status,  "引用接口status
     interface_table TYPE TABLE OF REF TO status. "引用接口创建内表

DATA:badi_obj TYPE REF TO zintest.  "引用接口zintest

START-OF-SELECTION.
*实例调用接口方法,类引用
```

```
  CREATE OBJECT : super_obj,super_object.
  CALL METHOD: super_obj->output~write.
  SKIP.
* 直接调用接口方法，需要通过内表实现，接口引用
  APPEND super_obj TO intf_table.
  LOOP AT intf_table INTO intf_obj.
    CALL METHOD intf_obj->write.
  ENDLOOP.

  APPEND super_obj TO interface_table.
  LOOP AT interface_table INTO interface_obj.
    CALL METHOD interface_obj->write.
  ENDLOOP.

  SKIP.

  CALL METHOD super_obj->increment.

  APPEND super_obj TO interface_table.
  LOOP AT interface_table INTO interface_obj.
    CALL METHOD interface_obj->write.
  ENDLOOP.

  SKIP.



*接口引用赋值类引用
  interface_obj = super_object.
  CALL METHOD interface_obj->write.

  badi_obj = super_object.
  CALL METHOD super_object->zintest~test.
  CALL METHOD badi_obj->test.

  SKIP.

*类实例访问变量
  super_obj->status~int = 5.
  WRITE / super_obj->status~int.
*类实例访问静态变量
  super_obj->status~cint = 10.
  WRITE / super_obj->status~cint.
*类名访问静态变量
  superclass=>status~cint = 20.
  WRITE / superclass=>status~cint .
*接口实例访问变量
  interface_obj->int = 5.
  WRITE / interface_obj->int.
*接口实例访问静态变量
  ->cint = 10.
  WRITE / interface_obj->cint.
*接口名访问常量
```

```
  WRITE / status=>const.
```

```
Hello SAP

Hello SAP
Count in count is          0

Count in count is          1
Count in count is          1

Count in count is          0
it is the BADI test interface
it is the BADI test interface

         5
        10
        20
         5
        10
        30
```

```
*&---------------------------------------------------------------------*
*& Report  Y_TEST_A_8                                                  *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&     类的事件                                                        *
*&---------------------------------------------------------------------*

REPORT  y_test_a_8                                .
*&---------------------------------------------------------------------*
*&       Class vehicle
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS vehicle DEFINITION.
  PUBLIC SECTION.
*    EVENTS: too_fast.
    EVENTS: too_fast EXPORTING value(p1) TYPE i.
    METHODS:accelerate,show_speed.
    CLASS-DATA speed TYPE i.
ENDCLASS.                    "vehicle
*&---------------------------------------------------------------------*
*&       Class (Implementation)  vehicle
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS vehicle IMPLEMENTATION.
  METHOD accelerate.
    speed = speed + 1.
    IF speed > 5 .
*      RAISE EVENT too_fast.
      RAISE EVENT too_fast EXPORTING p1 = speed.
    ENDIF.
  ENDMETHOD.                    "accelerate
  METHOD show_speed.
    WRITE: / 'Speed:',speed.
  ENDMETHOD.                    "show_speed
ENDCLASS.                    "vehicle

*&---------------------------------------------------------------------*
```

```
*&       Class handler
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS handler DEFINITION.
  PUBLIC SECTION.
*    METHODS handle_excess FOR EVENT too_fast OF vehicle.
    METHODS handle_excess FOR EVENT too_fast OF vehicle IMPORTING p1.
ENDCLASS.                    "handler
*&---------------------------------------------------------------------*
*&       Class (Implementation)  handler
*&---------------------------------------------------------------------*
*        Text
*---------------------------------------------------------------------*
CLASS handler IMPLEMENTATION.
  METHOD handle_excess.
    WRITE: / 'Speed can not be too fast.The speed is ', p1
LEFT-JUSTIFIED.
    vehicle=>speed = 1.
  ENDMETHOD.                    "handle_excess
ENDCLASS.                    "handler

DATA: o_vehicle TYPE REF TO vehicle,
      o_handle TYPE REF TO handler.
```


```
START-OF-SELECTION.
  CREATE OBJECT: o_vehicle,o_handle.
  SET HANDLER o_handle->handle_excess FOR ALL INSTANCES.
  DO 11 TIMES.
    CALL METHOD o_vehicle->accelerate.
    CALL METHOD o_vehicle->show_speed.
  ENDDO.
```

```
Speed:          1
Speed:          2
Speed:          3
Speed:          4
Speed:          5
Speed can not be too fast.The speed is  6
Speed:          1
Speed:          2
Speed:          3
Speed:          4
Speed:          5
Speed can not be too fast.The speed is  6
Speed:          1
```