

ABAP Editor: Change Report ZOOPS_ABAP

Report Active

```

4  * &
5  * &-----*
6  REPORT ZOOPS_ABAP.
7
8  "#Inheritance
9  "#Creating sub class from Main Class.
10 "#Instance method - Create Object
11 "#Static method - No Need to Create Object
12
13
14 DATA:lv_erdat  TYPE erdat,
15        lv_erzet  TYPE erzet,
16        lv_ernam  TYPE ernam,
17        lv_vbtyp1 TYPE vbtyp1.
18
19
20
21 PARAMETERS:p_vbeln TYPE vbeln_va.
22 PARAMETERS:r1 TYPE c RADIOBUTTON GROUP rad,
23            r2 TYPE c RADIOBUTTON GROUP rad.
24 CLASS class1 DEFINITION.
25     PUBLIC SECTION.
26         METHODS display IMPORTING pvbeln TYPE vbeln_va
27                        EXPORTING perdat TYPE erdat
28                        perzet TYPE erzet
29                        pernam TYPE ernam
30                        pvbtyp TYPE vbtyp1.
31 ENDCLASS.
32
33 CLASS class1 IMPLEMENTATION.
34     METHOD display.
35         SELECT SINGLE erdat erzet ernam vbtyp
36           FROM vbak
37           INTO ( perdat,perzet,pernam,pvbtyp )
38           WHERE vbeln = p_vbeln.
39     ENDMETHOD.
40 ENDCLASS.
41

```

```
41
42
43 CLASS class2 DEFINITION INHERITING FROM class1.
44
45     ENDCCLASS.
```

```
46
47
48 START-OF-SELECTION.
49     DATA:obj1 TYPE REF TO class2.
50     CREATE OBJECT obj1.
51
52     CALL METHOD obj1->display
53     EXPORTING
54         pvbeln = p_vbeln
55     IMPORTING
56         perdat = lv_erdat
57         perzet = lv_erzet
58         pernam = lv_ernam
59         pvbtyp = lv_vbtyp1.
60     WRITE: / lv_erdat,
61             lv_erzet,
62             lv_ernam,
63             lv_vbtyp1.
64
65
66
```

ABAP Editor: Change Report ZOOPS_ABAP

Report ZOOPS_ABAP Active

```
66
67  "#Abstract Class
68  "#Abstract Class should have atleast One abstract Method
69  "#Abstract method have only definition No Implementation
70  "#you can implement abstract method by creating subclass
71  "#In subclass you have to Inherite and Redefine that method to implement
72
73  CLASS zabstract DEFINITION ABSTRACT.
74    PUBLIC SECTION.
75      METHODS display ABSTRACT IMPORTING pvbeln TYPE vbeln_va
76                                EXPORTING perdat TYPE erdat
77                                perzet TYPE erzet
78                                pernam TYPE ernam
79                                pvbtyp TYPE vbtyp1.
80  ENDCCLASS.
81
82
83  CLASS zsales DEFINITION INHERITING FROM zabstract.
84    PUBLIC SECTION.
85      METHODS display REDEFINITION.
86  ENDCCLASS.
87
88
89  CLASS zsales IMPLEMENTATION.
90  METHOD display.
91    SELECT SINGLE erdat erzet ernam vbtyp
92    FROM vbak
93    INTO ( perdat,perzet,pernam,pvbtyp )
94    WHERE vbeln = p_vbeln.
95  ENDMETHOD.
96  ENDCCLASS.
97
98
```

```
97
98
99 □ CLASS zbilling DEFINITION INHERITING FROM zabstract.
100     PUBLIC SECTION.
101         METHODS display REDEFINITION.
102     ENDClass.
```

```
103
104
105 □ CLASS zbilling IMPLEMENTATION.
106     METHOD display.
107         SELECT SINGLE vbtyp ernam erzet erdat
108             FROM vbrk
109             INTO ( pvbtyp,pernam,perzet,perdat )
110             WHERE vbeln = p_vbeln.
111     ENDMETHOD.
112 ENDClass.
```

```
113
114 START-OF-SELECTION.
115     DATA:obj01 TYPE REF TO zsales.
116     DATA:obj02 TYPE REF TO zbilling.
117     CREATE OBJECT obj01.
118     CREATE OBJECT obj02.
119     IF r1 = 'X'.
120         CALL METHOD obj01->display
121             EXPORTING
122                 pvbeln = p_vbeln
123             IMPORTING
124                 perdat = lv_erdat
125                 perzet = lv_erzet
126                 pernam = lv_ernam
127                 pvbtyp = lv_vbtyp1.
128         WRITE: / lv_erdat,
129                lv_erzet,
130                lv_ernam,
131                lv_vbtyp1.
132
133
134
```

```
132  
133  
134  
135 ELSEIF r2 = 'X'.  
136     CALL METHOD obj02->display  
137     EXPORTING  
138         pvbeln = p_vbeln  
139     IMPORTING  
140         pvbtyp = lv_vbtyp1  
141         pernam = lv_ernam  
142         perzet = lv_erzet  
143         perdat = lv_erdat.  
144  
145  
146     WRITE: / lv_vbtyp1,  
147             lv_ernam,  
148             lv_erzet,  
149             lv_erdat.  
150 ENDIF.  
151  
152  
153
```

ABAP Editor: Change Report ZOOPS_ABAP

Report Active

```

153
154   "#Interface
155   "#In Interface All method are Public By default
156   "#In Interface All method are Abstract ( There is only Definition , no implementation ).
157   "#Polymorphism Archive Via Interface
158   "#Multiple Inheritance is Possible
159
160   "#Polymorphism
161   INTERFACE interface_name.
162       METHODS display IMPORTING pvbeln TYPE vbeln_va
163                       EXPORTING perdat TYPE erdat
164                               perzet TYPE erzet
165                               pernam TYPE ernam
166                               pvbtyp TYPE vbtyp1.
167   ENDINTERFACE.
168
169
170   CLASS int_class1 DEFINITION.
171       PUBLIC SECTION.
172           INTERFACES interface_name.
173   ENDCLASS.
174
175   CLASS int_class1 IMPLEMENTATION.
176   METHOD interface_name~display.
177       SELECT SINGLE erdat erzet ernam vbtyp
178       FROM vbak
179       INTO ( perdat,perzet,pernam,pvbtyp )
180       WHERE vbeln = p_vbeln.
181   ENDMETHOD.
182   ENDCLASS.
183
184
185
186

```


ABAP Editor: Change Report ZOOPS_ABAP

Report Active

```

168
169
170 CLASS int_class1 DEFINITION.
171     PUBLIC SECTION.
172         INTERFACES interface_name.
173     ENDCLASS.
174
175 CLASS int_class1 IMPLEMENTATION.
176     METHOD interface_name~display.
177         SELECT SINGLE erdat erzet ernam vbtyp
178         FROM vbak
179         INTO ( perdat,perzet,pernam,pvbtyp )
180         WHERE vbeln = p_vbeln.
181     ENDMETHOD.
182 ENDCLASS.
183
184
185
186
187 CLASS int_class2 DEFINITION.
188     PUBLIC SECTION.
189         INTERFACES interface_name.
190     ENDCLASS.
191
192
193 CLASS int_class2 IMPLEMENTATION.
194     METHOD interface_name~display.
195         SELECT SINGLE vbtyp ernam erzet erdat
196         FROM vbrk
197         INTO ( pvbtyp,pernam,perzet,perdat )
198         WHERE vbeln = p_vbeln.
199     ENDMETHOD.
200 ENDCLASS.
201
202

```

```

202
203 START-OF-SELECTION.
204 DATA:obji1 TYPE REF TO int_class1.
205 DATA:obji2 TYPE REF TO int_class2.
206 CREATE OBJECT obji1.
207 CREATE OBJECT obji2.
208
209 IF r1 = 'X'.
210
211     CALL METHOD obji1->interface_name~display
212     EXPORTING
213         pvbeln = p_vbeln
214     IMPORTING
215         perdat = lv_erdat
216         perzet = lv_erzet
217         pernam = lv_ernam
218         pvbtyp = lv_vbtyp1.
219
220
221     WRITE:'The Sales Order Details are follows',
222         lv_erdat,
223         lv_erzet,
224         lv_ernam,
225         lv_vbtyp1.
226
227 ELSEIF r2 = 'X'.
228
229     CALL METHOD obji2->interface_name~display
230     EXPORTING
231         pvbeln = p_vbeln
232     IMPORTING
233         pvbtyp = lv_vbtyp1
234         pernam = lv_ernam
235         perzet = lv_erzet
236         perdat = lv_erdat.
237
238
239     WRITE:'The Billing Order Details are follows',

```



```
238
239     WRITE: 'The Billing Order Details are follows',
240           lv_vbtyp1,
241           lv_ernam,
242           lv_erzet,
243           lv_erdat.
244
245 ENDIF.
246 "#Multiple Inheritance is Possible in Interface
247 □ INTERFACE interface_name1.
248     METHODS sales IMPORTING pvbeln TYPE vbeln_va
249                   EXPORTING perdat TYPE erdat
250                           perzet TYPE erzet
251                           pernam TYPE ernam
252                           pvbtyp TYPE vbtyp1.
253 ENDINTERFACE.
254
```

```

245     ENDF.
246     "#Multiple Inheritance is Possible in Interface
247 □ INTERFACE interface_name1.
248     METHODS sales IMPORTING pvbeln TYPE vbeln_va
249                 EXPORTING perdat TYPE erdat
250                        perzet TYPE erzet
251                        pernam TYPE ernam
252                        pvbtyp TYPE vbtyp1.
253 □ ENDINTERFACE.
254
255 □ INTERFACE interface_name2.
256     METHODS billing IMPORTING pvbeln TYPE vbeln_va
257                 EXPORTING pvbtyp TYPE vbtyp1
258                        pernam TYPE ernam
259                        perzet TYPE erzet
260                        perdat TYPE erdat.
261 □ ENDINTERFACE.
262
263 □ CLASS multi DEFINITION.
264     PUBLIC SECTION.
265         INTERFACES interface_name1.
266         INTERFACES interface_name2.
267 □ ENDClass.
268
269 □ CLASS multi IMPLEMENTATION.
270 □ METHOD interface_name1-sales.
271     SELECT SINGLE erdat erzet ernam vbtyp
272     FROM vbak
273     INTO ( perdat,perzet,pernam,pvbtyp )
274     WHERE vbeln = p_vbeln.
275 □ ENDMETHOD.
276
277 □ METHOD interface_name2-billing.
278     SELECT SINGLE vbtyp ernam erzet erdat
279     FROM vbrk
280     INTO ( pvbtyp,pernam,perzet,perdat )
281     WHERE vbeln = p_vbeln.

```

```
262
263 CLASS multi DEFINITION.
264     PUBLIC SECTION.
265         INTERFACES interface_name1.
266         INTERFACES interface_name2.
267     ENDClass.
268
269 CLASS multi IMPLEMENTATION.
270     METHOD interface_name1~sales.
271         SELECT SINGLE erdat erzet ernam vbtyp
272         FROM vbak
273         INTO ( perdat,perzet,pernam,pvbtyp )
274         WHERE vbeln = p_vbeln.
275     ENDMETHOD.
276
277     METHOD interface_name2~billing.
278         SELECT SINGLE vbtyp ernam erzet erdat
279         FROM vbrk
280         INTO ( pvbtyp,pernam,perzet,perdat )
281         WHERE vbeln = p_vbeln.
282     ENDMETHOD.
283
284 ENDClass.
```

286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320

START-OF-SELECTION.

DATA:object1 TYPE REF TO multi.

CREATE OBJECT object1.

IF r1 = 'X'.

CALL METHOD object1->interface_name1~sales

EXPORTING

pvbeln = p_vbeln

IMPORTING

perdat = lv_erdat

perzet = lv_erzet

pernam = lv_ernam

pvbtyp = lv_vbtyp1.

WRITE:'The Sales Order Details are follows',

lv_erdat,

lv_erzet,

lv_ernam,

lv_vbtyp1.

ELSEIF r2 = 'X'.

CALL METHOD object1->interface_name2~billing

EXPORTING

pvbeln = p_vbeln

IMPORTING

pvbtyp = lv_vbtyp1

pernam = lv_ernam

perzet = lv_erzet

perdat = lv_erdat.

```
308
309 ELSEIF r2 = 'X'.
310
311     CALL METHOD object1->interface_name2~billing
312     EXPORTING
313         pvbeln = p_vbeln
314     IMPORTING
315         pvbtyp = lv_vbtyp1
316         pernam = lv_ernam
317         perzet = lv_erzet
318         perdat = lv_erdat.
319
320
321     WRITE: 'The Billing Order Details are follows',
322           lv_vbtyp1,
323           lv_ernam,
324           lv_erzet,
325           lv_erdat.
326
327 ENDIF.
328
329
```

ABAP Editor: Change Report ZOOPS_ABAP

Report Active

```

329
330     "#OOPSEVENTS
331
332     CLASS cl_event DEFINITION.
333     PUBLIC SECTION.
334         METHODS display IMPORTING pvbeln TYPE vbeln_va
335                          EXPORTING perdat TYPE erdat
336                               perzet TYPE erzet
337                               pernam TYPE ernam
338                               pvbtyp TYPE vbtyp1.
339         EVENTS: no_inputs, wrong_inputs.
340     ENDClass.
341
342     CLASS cl_event IMPLEMENTATION.
343     METHOD display.                                "Triggering Event
344     IF pvbeln IS INITIAL.
345         RAISE EVENT no_inputs.
346     ELSE.
347         SELECT SINGLE erdat erzet ernam vbtyp
348         FROM vbak
349         INTO ( perdat, perzet, pernam, pvbtyp )
350         WHERE vbeln = p_vbeln.
351     IF sy-subrc = 0.
352         RAISE EVENT wrong_inputs.
353     ENDIF.
354     ENDIF.
355     ENDMETHOD.
356
357     ENDClass.
358

```



```
358
359 CLASS event_1 DEFINITION.
360     PUBLIC SECTION.                                "Event Handler
361     METHODS message FOR EVENT no_inputs OF cl_event.
362     METHODS msg FOR EVENT wrong_inputs OF cl_event.
363
364 ENDCCLASS.
```

```
365
366 CLASS event_1 IMPLEMENTATION.
367     METHOD message.
368         WRITE:'Please Enter Inputs'.
369     ENDMETHOD.
370
371     METHOD msg.
372         WRITE:'Enter Correct Inputs'.
373     ENDMETHOD.
374
375 ENDCCLASS.
```

```
376
```

```

376
377 START-OF-SELECTION.
378
379 DATA: eve_obj1 TYPE REF TO cl_event.
380 DATA: eve_obj2 TYPE REF TO event_1.
381 CREATE OBJECT eve_obj1.
382 CREATE OBJECT eve_obj2.
383
384 "registering Event Handler.
385 IF p_vbeln IS INITIAL.
386     SET HANDLER eve_obj2->message FOR eve_obj1.
387 ELSE.
388     SET HANDLER eve_obj2->msg FOR eve_obj1.
389 ENDIF.
390
391 CALL METHOD eve_obj1->display
392     EXPORTING
393         pvbeln = p_vbeln
394     IMPORTING
395         perdat = lv_erdat
396         perzet = lv_erzet
397         pernam = lv_ernam
398         pvbtyp = lv_vbtyp1.
399
400 IF lv_erdat IS NOT INITIAL AND lv_erzet IS NOT INITIAL AND lv_ernam IS NOT INITIAL AND lv_erdat IS NOT INITIAL AND lv_vbtyp1 IS NOT INITIAL.
401     WRITE:/ lv_erdat,
402             lv_erzet,
403             lv_ernam,
404             lv_vbtyp1.
405 ENDIF.

```