

石化盈科
信息技术有限责任公司



ABAP培训

韩郁飞



ABAP程序分类

< 报表程序 (Report)

- 1、结构简单，
- 2、由事件块 (Event Blocks) 组成程序架构；
- 3、执行方式多样；

对话程序 (Dialog Program / Transaction Program)

- 1、结构复杂；
- 2、包括ABAP程序代码、用户对话、图形用户接口和事务代码；
- 3、图形用户接口包括菜单栏、标准工具栏、应用程序工具栏和标题栏
- 4、程序由事务代码触发运行；

报表程序分类

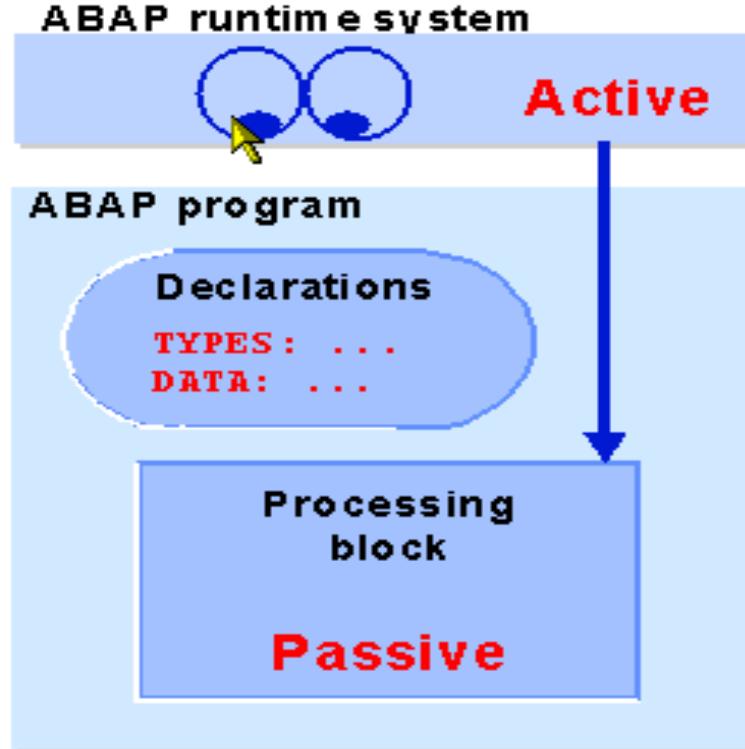
< 普通报表程序

输出结果格式固定，不能对其进行交互式操作；

< 交互式报表程序

通过特定的触发事件，比如双击、菜单、按钮和功能键等，可以对报表结果进行操作；

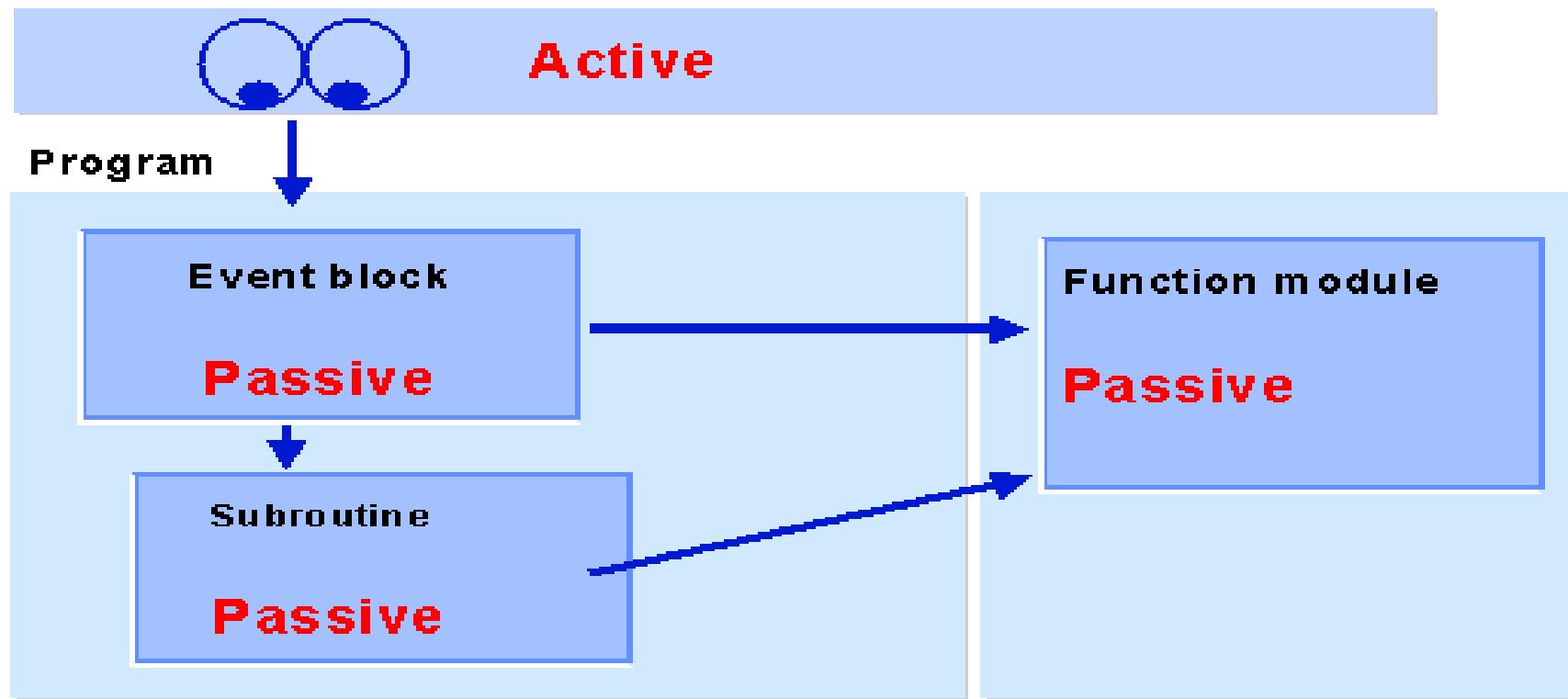
报表程序结构



```
REPORT ...  
DATA: wa_scarr LIKE scarr.  
PARAMETERS:  
      pa_carr LIKE scarr-carrid.  
  
SELECT SINGLE carrname  
      FROM scarr  
      INTO wa_scarr-carrname  
      WHERE carrid = pa_carr.  
  
WRITE: / pa_carr,  
      wa_scarr-carrname.
```

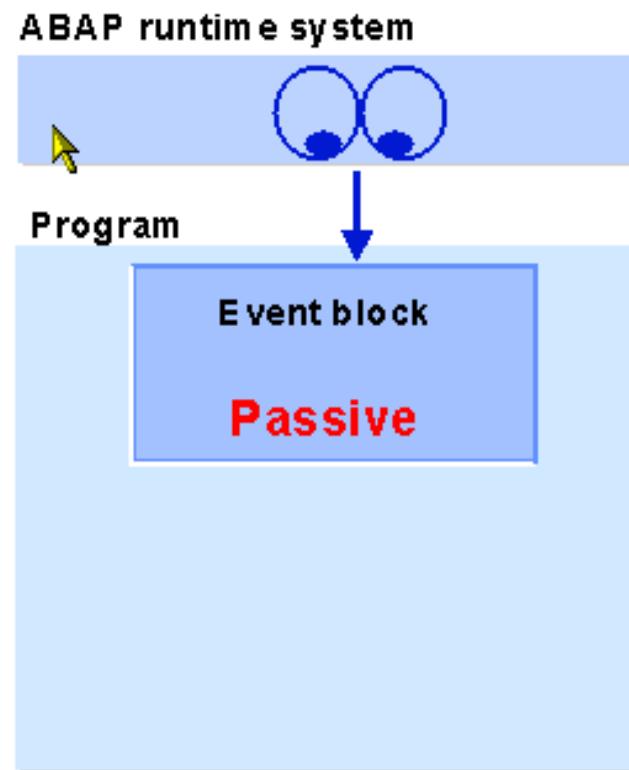
处理块的类型

ABAP runtime system



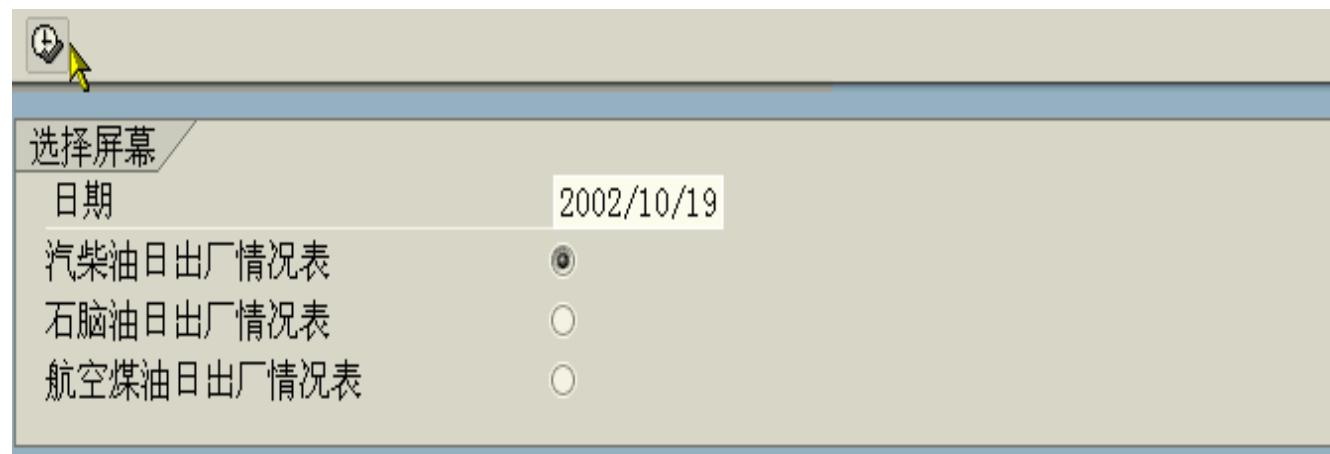
事件块

- < 组成报表程序的基本单位；
- < 原则上每个事件块由各自的关键字为起始，不同的事件块执行不同的任务；
- < 事件块的结束标志为下一个事件块的关键字；
- < 事件块由ABAP运行时环境调用，其调用顺序固定，与程序编码顺序无关；
- < 缺省的事件块START-OF-SELECTION



实例讲解

- < 汽柴油、石脑油、航空煤油日出厂情况表
- < 目的：对配置计划（销售计划）中汽柴油、石脑油、航空煤油的日销售完成情况进行统计，数据来源于系统





程序名

- < REPORT rep.
- < Additions:

1. ... NO STANDARD PAGE HEADING

2. ... LINE-SIZE col

3. ... LINE-COUNT n(m)

表单输出每页由n行，其中的m行作为页脚；

4. ... MESSAGE-ID mid

消息对象

5. ... DEFINING DATABASE ldb

使用逻辑数据库，自动产生





Tables 语句

- < 基本格式
TABLES dbtab.
- < Dbtab是指ABAP字典的一个数据库对象，可以是透明表格、视图和结构；
- < 该语句的作用是声明了一个表工作区，类似于C语言里面的STRUCTURE和PASCAL语言里面的RECORD；
- < 这个表工作区由多个字段组成，其名称、类型和dbtab的字段完全相同；
- < ABAP字典对象中字段的类型自动的转换为ABAP程序的类型；
- < 表工作区通常是一个全局变量。



TYPES语句

- < 定义一个简单的数据类型;
- < 定义一个结构化的数据类型;
- < 定义一个内表形式的数据类型;

Type	Description	DL	Initial value
C	Character	1	Space
N	Numeric text	1	'00...0'
D	Date YYYYMMDD	8	'00000000'
T	Time HHMMSS	6	'000000'
X	Byte (hexadecimal)	1	X'00'
I	Integer	4	0
P	Packed number	8	0
F	Floating point number	8	'0.0'
STRING	<u>String</u>	variable	empty string
XSTRING	<u>Byte sequence</u> (X string)	variable	empty X string



TYPES语句格式

- < 1. TYPES type.
- 2. TYPES type(len).
- 3. TYPES: BEGIN OF structype ...
END OF structype.
- 4. TYPES itabtype {TYPE tabkind OF linetype|
LIKE tabkind OF lineobj}
[WITH [UNIQUE|NON-UNIQUE] keydef] [INITIAL SIZE n].
- 5. TYPES itabtype TYPE RANGE OF type.
TYPES itabtype LIKE RANGE OF f.
- 6. TYPES itabtype {TYPE linetype|LIKE lineobj} OCCURS n.





DATA语句

- < 定义一个简单类型的变量
- < 定义一个结构化的数据对象
- < 定义一个内表
- < 定义一个共享的数据区域
用于外部的子程序调用



DATA语句

1. DATA f.

```
DATA: BEGIN OF struc,      ...
      END   OF struc.
```

2. DATA f(len).

Additions:

1. ... TYPE type

2. ... LIKE f1

3. ... TYPE LINE OF itabtype

4. ... LIKE LINE OF itab

5. ... TYPE REF TO cif

6. ... TYPE REF TO DATA

7. ... VALUE lit

8. ... DECIMALS n

9. ... READ-ONLY

DATA语句定义内表

```
1. DATA itab TYPE itabtype [WITH HEADER LINE].  
  
2. DATA itab {TYPE tabkind OF linetype|  
             LIKE tabkind OF lineobj}  
             WITH [UNIQUE|NON-UNIQUE] keydef  
             [INITIAL SIZE n] [WITH HEADER LINE].  
  
3. DATA itab {TYPE TABLE OF linetype|LIKE TABLE OF lineobj}.  
  
4. DATA itab TYPE RANGE OF type.  
DATA itab LIKE RANGE OF f.  
  
5. DATA itab [TYPE linetype|LIKE lineobj] OCCURS n  
             [WITH HEADER LINE].  
  
6. DATA: BEGIN OF itab OCCURS n,  
             ...  
             END   OF itab [VALID BETWEEN f1 AND f2].
```

定义输入屏幕

< 用SELECTION-SCREEN语句来定义输入屏幕

汽柴油、石脑油、航空煤油日出厂情况表

选择屏幕

日期	2003/06/19
汽柴油日出厂情况表	<input checked="" type="radio"/>
石脑油日出厂情况表	<input type="radio"/>
航空煤油日出厂情况表	<input type="radio"/>

SELECTION-SCREEN语法

- 1a. SELECTION-SCREEN BEGIN OF SCREEN scr.
- 1b. SELECTION-SCREEN END OF SCREEN scr.
2. SELECTION-SCREEN BEGIN OF SCREEN scr AS SUBSCREEN.

特殊效果

1. SELECTION-SCREEN BEGIN OF BLOCK block.
2. SELECTION-SCREEN BEGIN OF TABBED BLOCK block FOR n LINES.
3. SELECTION-SCREEN END OF BLOCK block.
4. SELECTION-SCREEN TAB (len) tabname USER-COMMAND ucom.
5. SELECTION-SCREEN BEGIN OF LINE.
6. SELECTION-SCREEN END OF LINE.
7. SELECTION-SCREEN POSITION pos.
8. SELECTION-SCREEN ULINE.
9. SELECTION-SCREEN COMMENT fmt name.
10. SELECTION-SCREEN PUSHBUTTON fmt name USER-COMMAND ucom.
11. SELECTION-SCREEN SKIP n.
12. SELECTION-SCREEN FUNCTION KEY n.

输入字段设计

< PARAMETERS 语句

日期 	2003/06/19
--	------------

< SELECT-OPTIONS语句

日期			到	
----	--	---	---	---

PARAMETERS语法

PARAMETERS p.

Additions:

1. **DEFAULT** f
2. **TYPE** type
3. **DECIMALS** dec
4. **LIKE** g
5. **MEMORY ID** pid
6. **MATCHCODE OBJECT** mobj
7. **MODIF ID** modid
8. **NO-DISPLAY**
9. **LOWER CASE**
10. **OBLIGATORY**
11. **AS CHECKBOX**
12. **RADIOBUTTON GROUP** radi
13. **FOR TABLE** dbtab
14. **FOR NODE** node
15. **AS SEARCH PATTERN**
16. **VALUE-REQUEST**
17. **HELP-REQUEST**
18. **VISIBLE LENGTH** vlen
19. **VALUE CHECK**
20. **LIKE (g)**
21. **USER-COMMAND** ucom

Select-options语法

SELECT-OPTIONS sel FOR f.

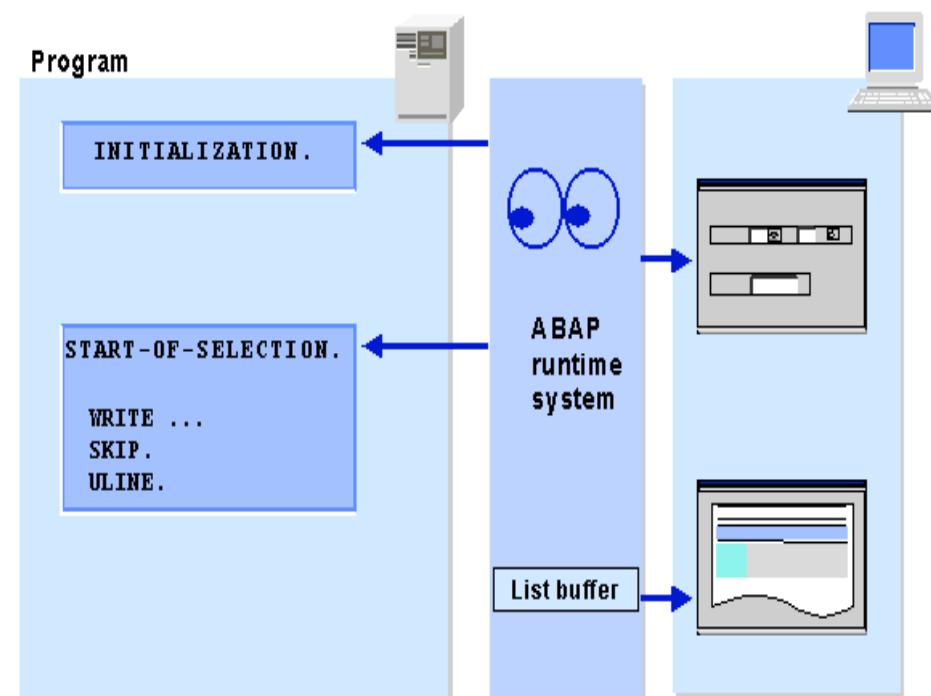
1. **DEFAULT** g
2. **DEFAULT** g **OPTION** xx **SIGN** s
3. **DEFAULT** g **TO** h
4. **DEFAULT** g **TO** h **OPTION** op **SIGN** s
5. **MEMORY** ID pid
6. **MATCHCODE** OBJECT mobj
7. **MODIF** ID key
8. **NO-DISPLAY**
9. **LOWER CASE**
10. **OBLIGATORY**
11. **NO-EXTENSION**
12. **NO INTERVALS**
13. **NO DATABASE SELECTION**
14. **VALUE-REQUEST**
15. **VALUE-REQUEST FOR** LOW/HIGH
16. **HELP-REQUEST**
17. **HELP-REQUEST FOR** LOW/HIGH
18. **VISIBLE LENGTH** vlen

事件块

- < INITIALIZATION
- < AT SELECTION-SCREEN
- < START-OF-SELECTION
- < GET
- < END-OF-SELECTION
- < TOP-OF-PAGE
- < END-OF-PAGE

事件块详细讲解

- < 事件块的简单处理过程:
- < ABAP程序运行的时候，INITIALIZATION首先被调用；
- < 经过初始化的输入屏幕会显示在表示服务器；
- < 用户离开输入屏幕的时候，START-OF-SELCTION事件会被自动调用；
- < 结果数据会以列表的形式显示在第二个屏幕上；



事件块的语法

源代码中的事件块顺序不影响它们的执行顺序

```
REPORT ...  
  
PARAMETERS: pa_date LIKE sy-datum default sy-datum.  
  
INITIALIZATION. " Default values for selection screen  
pa_date = pa_date - 7.  
  
START-OF-SELECTION. " Beginning of data processing  
write pa_date.
```



事件块编码规则 (1)

< INITIALIZATION

通常在此事件块中设定输入屏幕字段的初始值

例如： INITIALIZATION .

 P_DATE = '20030201' .

效果：

 输入屏幕的日期字段初始显示值为2003年2月1日

事件块编码规则（2）

< AT SELECTION-SCREEN

通常在此事件块中进行用户输入数据的合法性检查，发现错误则以消息的形式给出警示，直到用户输入正确的数值

例如： AT SELECTION-SCREEN .

```
IF P_DATE = SPACE .  
MESSAGE E001 .  
ENDIF.
```

效果：如果字段P_DATE为空，则程序会用消息001“日期字段不能为空！”来提示用户必须输入一个日期。而且输入屏幕会等待用户输入，知道该字段数值合法

事件块编码规则（3）

< START-OF-SELCTION

通常在此事件中针对业务需求进行系统数据的查询

例如：

```
start-of-selection .
```

```
    perform get_data_for_oil .
```

效果：

在输入屏幕用户按下执行按钮后，子程序get_data_for_oil被执行，在其中获得业务相关的数据存放到内表或者其它变量，这些数据在END-OF-SELECTION事件块中被输出

事件块编码规则（4）

< END-OF-SELCTION

通常在此事件中进行结果清单的输出

例如：

```
end-of-selection .  
    write : 23(1) sy-vline,  
          24(20) tab-gas_plan right-justified ,
```

效果：

数据以清单的形式输出

事件块编码规则 (5)

< GET

从逻辑数据库中得到数据 (较少用)

< TOP-OF-PAGE

在此事件块中设计输出清单的页头

例如： TOP-OF-PAGE

```
write : /1(240) '汽柴油日出厂情况表' centered .  
write : /20(8) '日期: ',  
      29(10) s_date ,  
      180(6) '单位: ',  
      190(10) '吨' .
```

事件块编码规则 (6)

< END-OF-PAGE

在此事件中设定输出清单的页脚

例如：

END-OF-PAGE .

Write : ‘制作人’ , p_name .

WRITE语句

- < 按照指定的格式把变量里面的数据输出到屏幕
- < 语法:

```
WRITE f.
```

Additions:

1. . . . **AT f** (position and length specification, before the field)
2. . . . **option** (formatting option)
3. . . . **fmt** (output format by field)
4. . . . **AS CHECKBOX** (output as checkbox)
5. . . . **AS SYMBOL** (output as symbol)
6. . . . **AS ICON** (output as icon)
7. . . . **AS LINE** (output as line)
8. . . . **QUICKINFO g** (output with quick info)



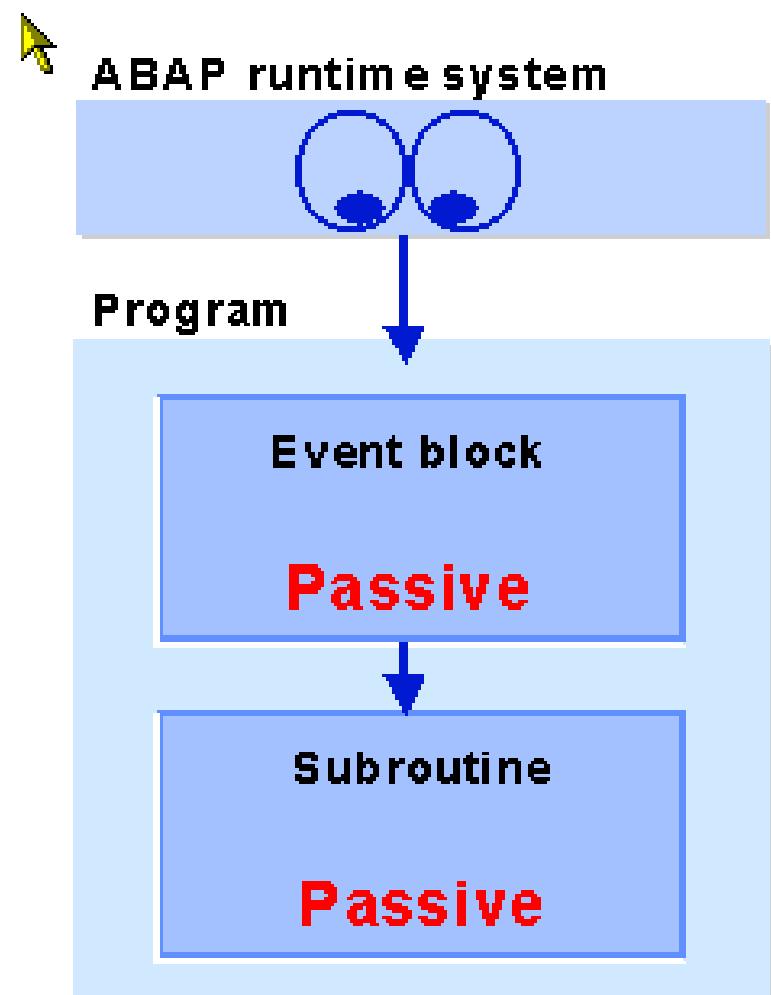
WRITE语句的选项

- < 根据选项的不同，WRITE语句可以输出到屏幕、输出到变量、以复选框的形式输出、以图标的格式输出

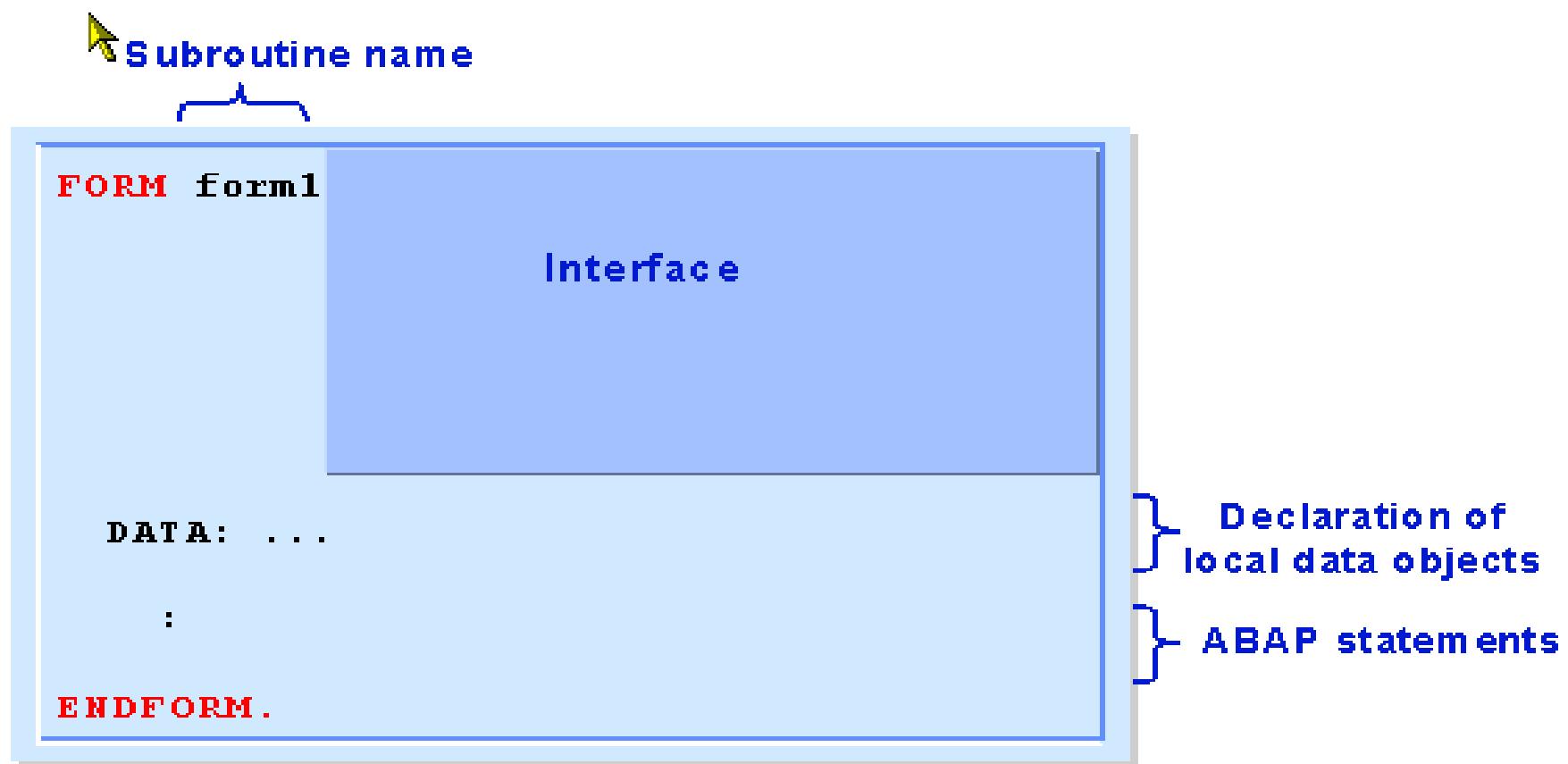


子程序 (1)

- < 子程序类似于C、PASCAL等语言的函数/过程
- < 子程序在事件块中被调用



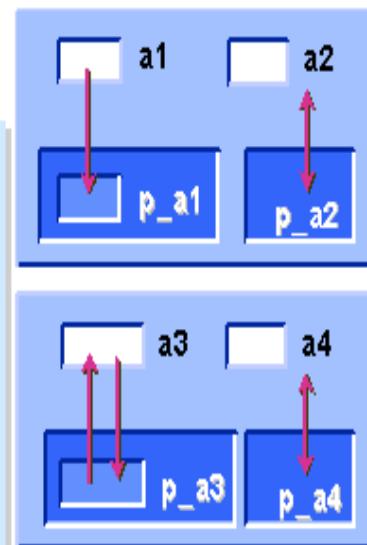
子程序 (2) — 语法



子程序—接口定义

Parameter typing

```
FORM form1 USING VALUE(p_a1) TYPE ...
          p_a2 TYPE ...
          ...
          CHANGING VALUE(p_a3) TYPE ...
          p_a4 TYPE ...
          ...
          .
          .
          .
          DATA: ...
          .
          .
          .
ENDFORM.
```



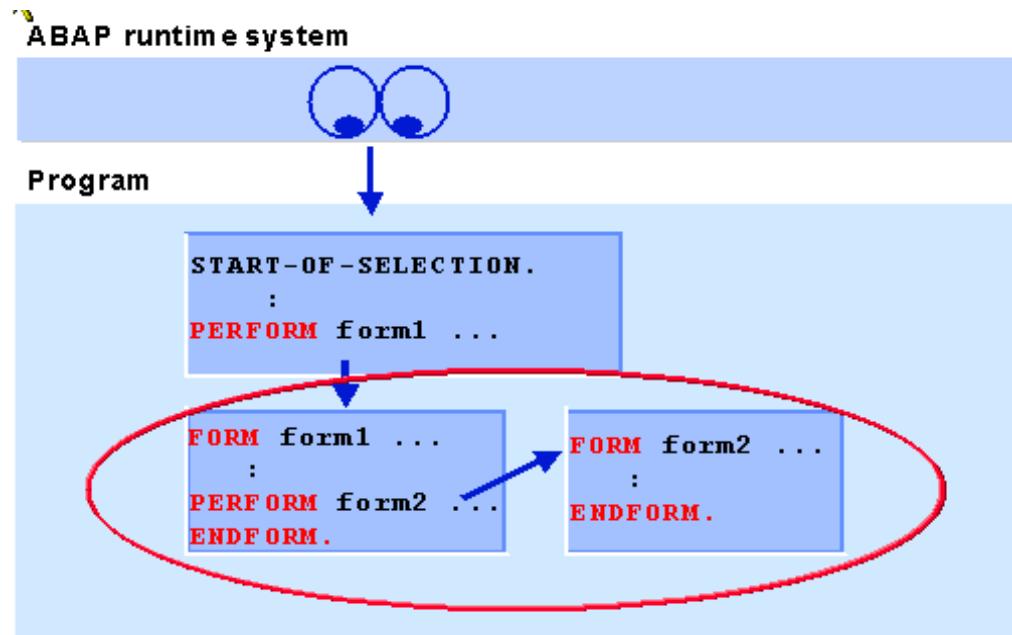
形参—实参传递方式

- 值调用
- 地址调用

子程序调用 (1)

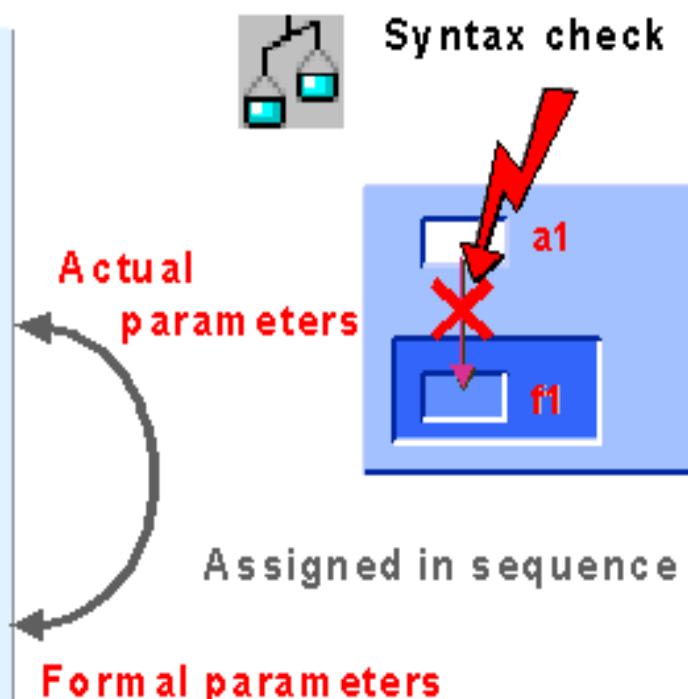
语法:

PERFORM <subroutine name> <interface>.



子程序调用 (2)

```
REPORT ...  
DATA: REVENUE ... ,  
      a1 type c.  
      .  
      PERFORM <name> USING  
            a1 a2 a3  
            CHANGING a4.  
      .  
      FORM <name> USING  
            VALUE(f1) TYPE p  
            VALUE(f2) TYPE i  
            f3 LIKE revenue  
            CHANGING VALUE(f4) TYPE p.  
      <Statements>  
      ENDFORM .
```



结构作为子程序参数

Typing is necessary for structures

内表作为子程序参数

```
REPORT .
TYPES: BEGIN OF wa_flight_type,
         carrid LIKE spfli-carrid,
         connid LIKE spfli-connid,
         .
         .
         END OF wa_flight_type.
TYPES: it_flight_type TYPE STANDARD TABLE OF wa_flight_type.

DATA: wa_flight TYPE wa_flight_type,
      it_flight TYPE TABLE OF wa_flight_type.
      .
      .
      .
PERFORM display_itab USING it_flight.
      .
      .
      .
FORM display_itab USING p_it_flight TYPE it_flight_type.
  DATA: ld_wa_flight TYPE wa_flight_type.
  LOOP AT p_it_flight INTO ld_wa_flight.
    WRITE: / ld_wa_flight-carrid,
           ld_wa_flight-connid,
           .
           .
  ENDLOOP.
ENDFORM.
```

Typing is necessary
for internal tables

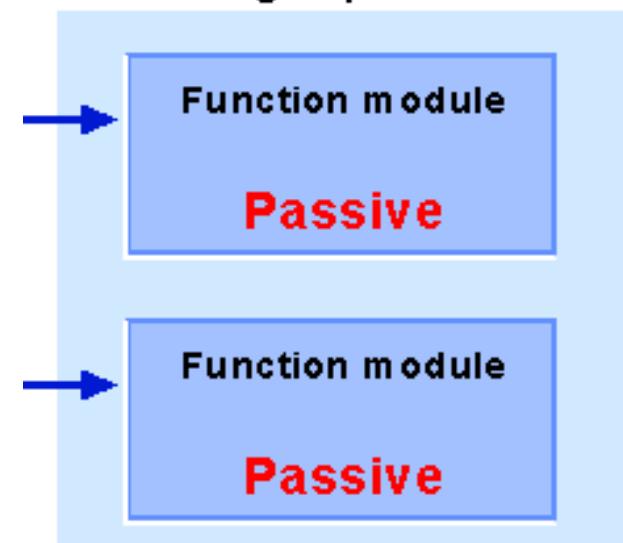
函数模块

- < 与子程序(form)相比，函数模块是在程序外部定义的，而且能够在全系统范围内被调用的子程序；而子程序 (form) 一般来说只在本程序内定义和调用。
- < 函数模块具有透明的接口，它的接口与代码分开定义。
- < 函数模块都隶属于某一个函数组
- < 函数模块由函数制作器 (Function Builder) 创建
- < 函数模块可以被本系统的程序调用，而且可以被其它的R/3系统、R/2系统甚至是非R/3系统调用 (RFC)

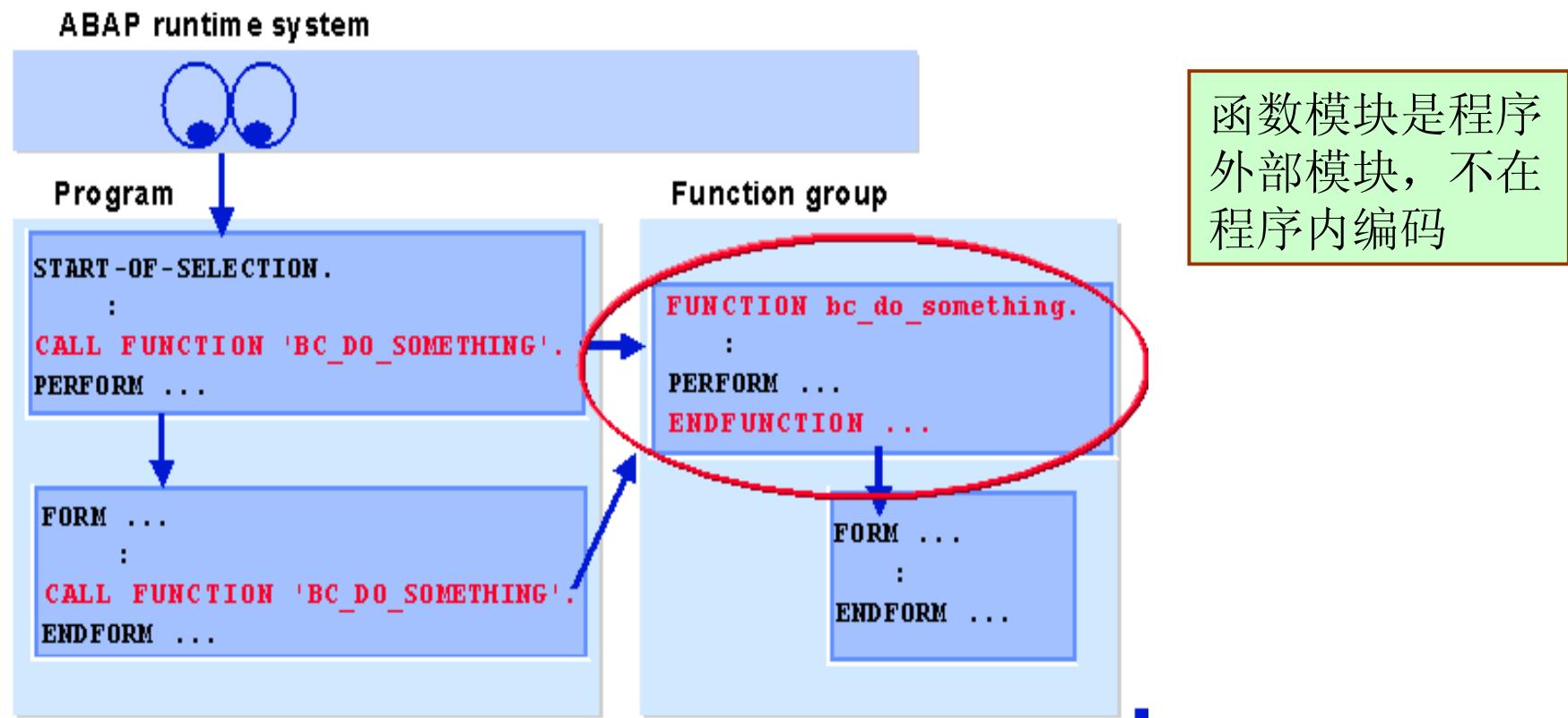
ABAP runtime system



Function group

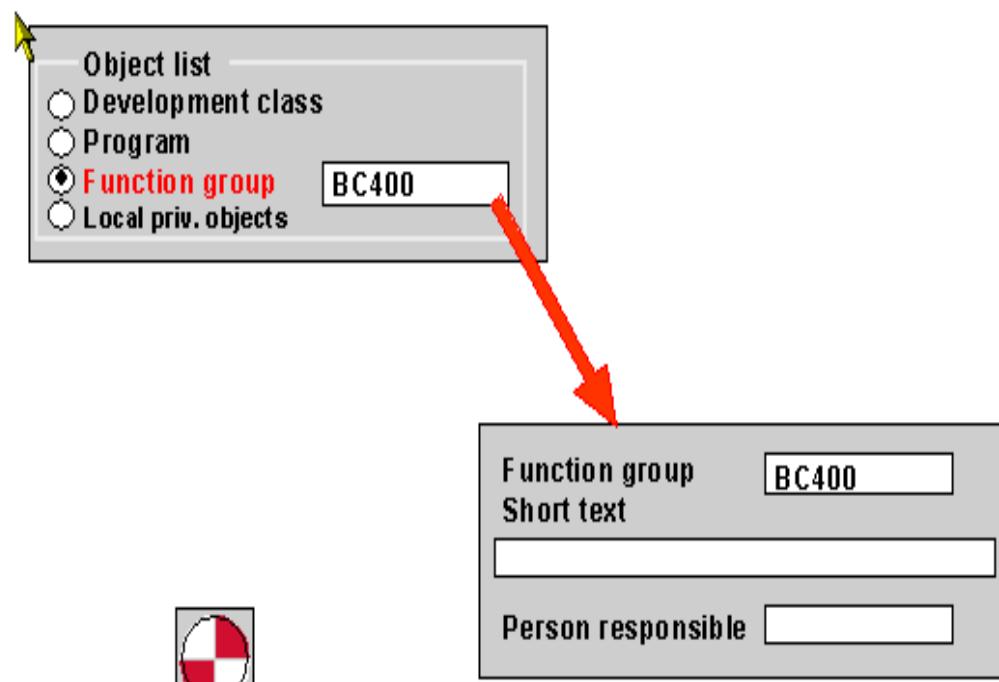


函数模块的调用



函数组

- < 每个函数模块都属于一个函数组
- < 每个函数组里面的函数模块都有相同的主题，实现的功能相互联系
- < 函数组实际上是一个类型为F的ABAP程序，它本身不能够被执行
- < 当一个函数模块被调用的时候，与它同一个函数组的函数都被调到内存区域里面；



函数模块的组成

Administration

Interface

Source code

Documentation

Import parameters

Export parameters

Changing parameters

Tables

Exceptions

Interface
documentation

函数模块的调用语法

```
REPORT .
DATA:      free_seats LIKE sflight-seatsmax.
PARAMETERS: pa_occ LIKE sflight-seatsocc,
            pa_max LIKE sflight-seatsmax.
START-OF-SELECTION.
  CALL FUNCTION 'BC400_FREE_SEATS'
    EXPORTING
      seatsmax      = pa_max
      seatsocc      = pa_occ
    IMPORTING
      seatsfree     = free_seats
  EXCEPTIONS
    occ_gt_max     = 1
    max_eq_0       = 2
    others         = 3.
CASE sy-subrc.
  WHEN 1.
    WRITE text-ex1.
  WHEN 2.
    WRITE text-ex2.
  WHEN 3.
    WRITE text-oth.
ENDCASE.
```

函数模块可以设置对异常的处理，以防出现运行时错误