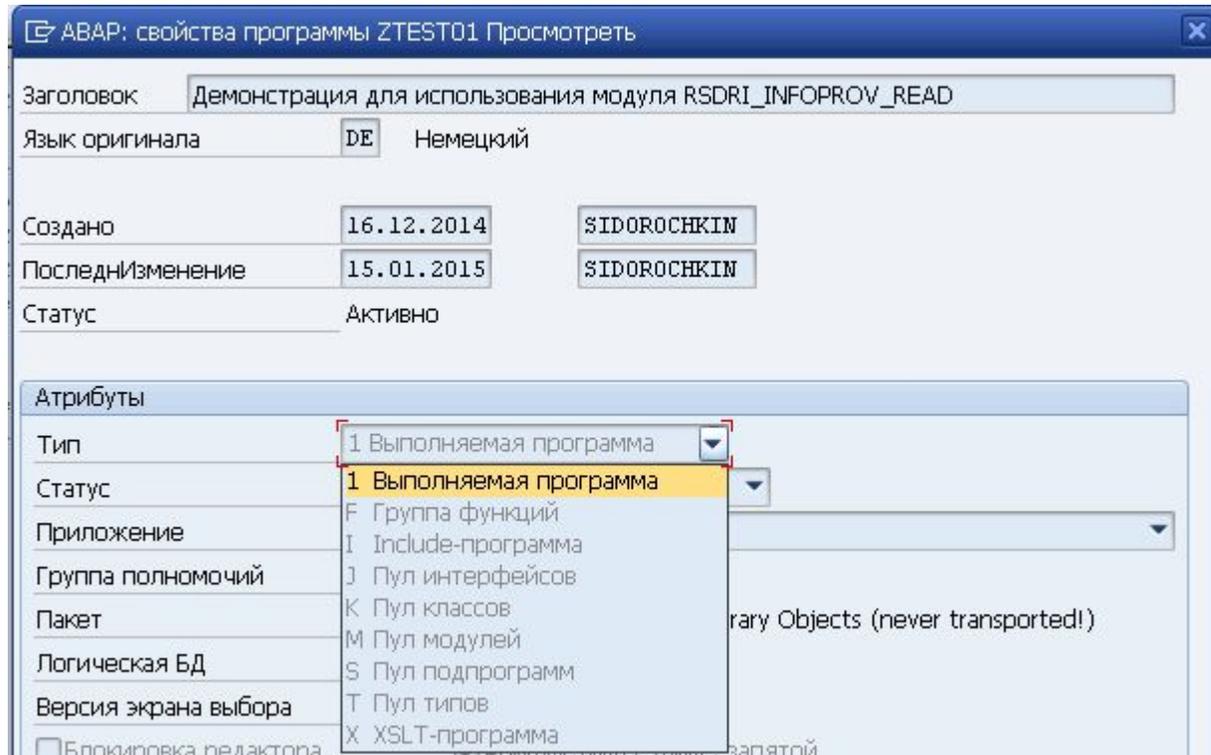


Отчёт написанный на языке АВАР представляет собой программу, считывающую и анализирующую данные из базы данных, обычно без модификации, но при необходимости предоставляющую и такие возможности. Как правило, такая программа генерирует список в виде последовательностей текстов, который далее может быть отправлен на принтер (спул). Выборка данных из БД обычно представляет объединение нескольких таблиц (JOIN).

С технической точки зрения АВАР отчёт это программа имеющая технический тип 1 – выполняемая программа:

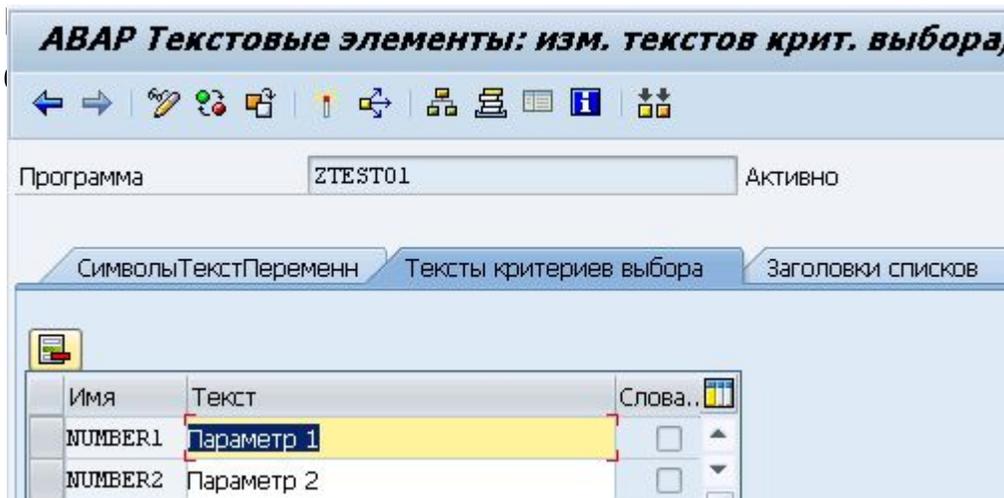


В зависимости от типов программ, среда выполнения по-разному производит их загрузку и выполнение. Так например, выполняемая программа управляется с помощью вызова определенных событий внутри её кода, а пул модулей (тип M) через вызов экранов и логики их обработки. Однако при необходимости из выполняемой программы так же могут быть вызваны дополнительные экраны.

Что такое экраны выбора?

Экран выбора это часть программы, создаваемая с целью получить от пользователей отчёта необходимые для работы программы данные. (пример: за какой отчётный день необходимо получить отчёт о запасах). Данные, которые вводит пользователи так же называют критериями выбора. Экран выбора может состоять из полей ввода, переключателей, радиокнопок, табуляторов и пр.

Экран выбора запускается сразу после старта выполняемой программы. После чего пользователь может заполнить на экране необходимые критерии выбора. Благодаря встроенной системе выбора можно задавать для разных языков. В зависимости от языка, пользователь увидит тот или иной перевод:



Для создания экрана выбора в ABAP встроены следующие команды:

- PARAMTERS – для определения параметров, одиночное поле ввода.
- SELECT-OPTIONS – для определения критериев выбора, с возможностью множественного ввода
- SELECTION-SCREEN – для форматирования вывода, например задания рамки.

PARAMETERS

Команда для определения параметра и его типа. Используется следующий синтаксис:

PARAMETERS <p>[(<length>)] <type> [<decimals>]. Данный синтаксис накладывает следующее ограничение: в качестве типа для параметров не может быть использован тип f (FLOAT), встроенный в ABAP тип данных, используемый для хранения значений с запятой, в качестве альтернативы может быть использован тип P (упакованное число).

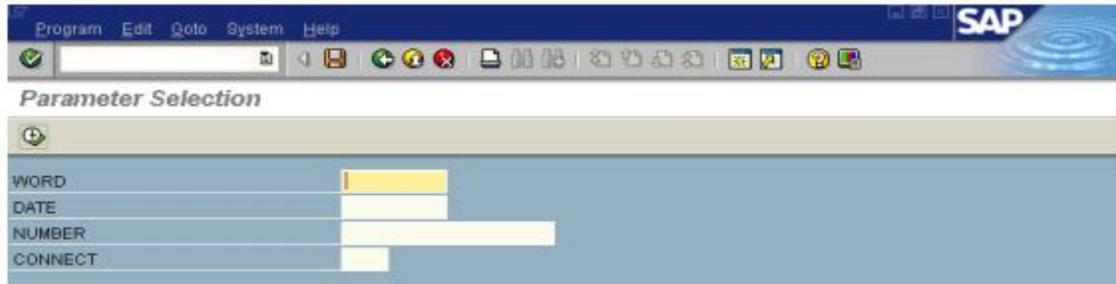
Пример ABAP программы с использованием оператора PARAMETERS:

```
REPORT SAPMZTST.  
  
TABLES SPFLI.  
  
PARAMETERS : WORD(10) TYPE C,  
  
DATE TYPE D,  
  
NUMBER TYPE P DECIMALS 2,  
  
CONNECT Like SPFLI-CONNID.
```

В данном примере создаются 4 параметра (поля ввода):

- word – символьное поле длиной 10 символов;
- date – поле ввода с типом хранения даты;
- числовое поле number с возможностью ввода двух знаков после запятой;
- поле connect описание которого хранится в АВАР словаре в таблице SPFLI поле CONNID;

После запуска программы пользователь увидит следующий экран:



Вы можете использовать параметры как АВАР переменные, для того чтобы контролировать выборку из БД данных с помощью оператора SELECT:

```
TABLES SPFLI.  
PARAMETERS : LOW LIKE SPFLI-CARRID,  
HIGH LIKE SPFLI-CARRID.  
SELECT * FROM SPFLI WHERE CARRIED BETWEEN LOW AND HIGH.  
-----  
ENDSELECT.
```

В данном примере после того как пользователь заполнит параметры LOW и HIGH выполняется оператор SELECT который выбирает из таблицы SPFLI все строки поле CARRIER у которых находится в диапазоне указанном пользователем.

У оператора PARAMETERS могут быть следующие дополнения:

PARAMETERS <p> DEFAULT <f>

PARAMETERS <p> NO-DISPLAY

PARAMETERS <p> LOWER CASE

PARAMETERS <p> OBLIGATORY

PARAMETERS <p> AS CHECKBOX

PARAMETERS <p> RADIOBUTTON GROUP <radi>.....

PARAMETERS <p> MEMORY ID <pid>.....

PARAMETERS <p> MATCHCODE OBJECT <obj>

PARAMETERS <p> MODIF ID <key>

Присвоение начальных значений для параметров:

```
REPORT ZTEST2 .
```

```
TABLES SPFLI.
```

```
PARAMETERS : value type i default 100,
```

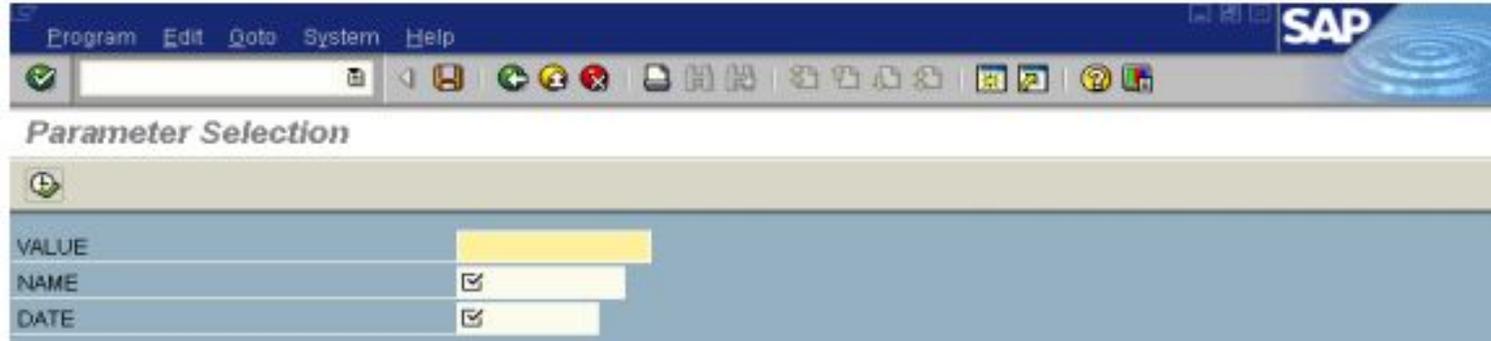
```
name like sy-uname default sy-uname,
```

```
date like sy-datum default '20020110'.
```



После запуска отчёта параметры будут заполнены автоматически

Обязательность ввода параметра: Parameters <p> Obligatory. Когда данное свойство выставлено для параметра, пользователь не сможет запустить отчёт не указав значение для поля. Внешне это выглядит следующим образом:



Вывод параметра как флажка. Parameters <p> ...as CHECKBOX... Пример программы:

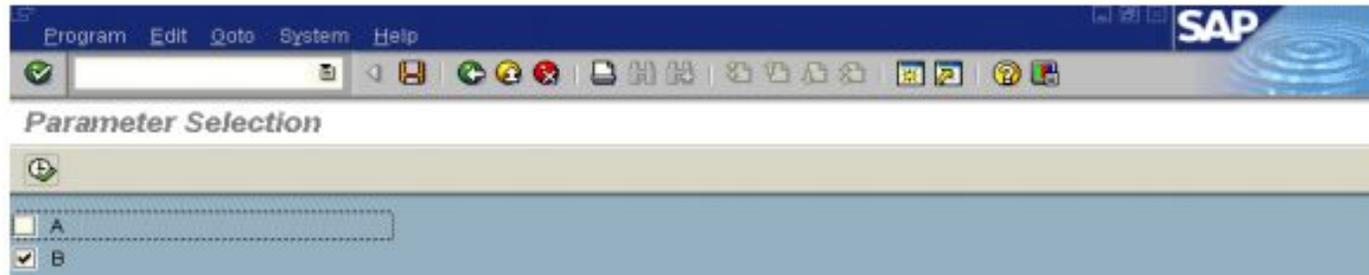
```
REPORT ZTEST2 .
```

```
TABLES SPFLI.
```

```
PARAMETERS : A as Checkbox,
```

```
B as Checkbox Default 'X'.
```

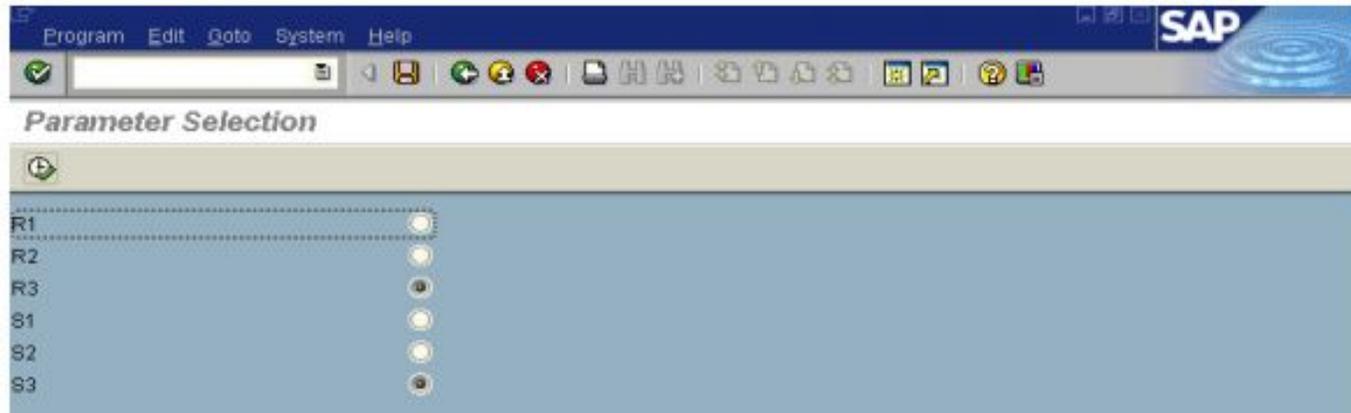
В результате на экран будет выведено следующее:



Вывод параметра как набора радиокнопок. Parameters <p>....RADIOBUTTON GRUP <radi>..... Пример программы:

```
REPORT ZTEST2 .  
  
TABLES SPFLI .  
  
PARAMETERS : R1 RADIOBUTTON GROUP RAD1,  
  
R2 RADIOBUTTON GROUP RAD1,  
  
R3 RADIOBUTTON GROUP RAD1 DEFAULT 'X',  
  
S1 RADIOBUTTON GROUP RAD2,  
  
S2 RADIOBUTTON GROUP RAD2,  
  
S3 RADIOBUTTON GROUP RAD2 DEFAULT 'X' .
```

В данной программе заданы две радио-группы RAD1 и RAD2, экран будет таким:



Критерии выбора

Критерии выбора задаются с помощью оператора SELECT-OPTIONS.

Синтаксис оператора для объявления критерия выбора следующий:

SELECT-OPTIONS <seltab> FOR <f>. Пример:

```
PROGRAM ZTEST.
```

```
TABLES: SPFLI.
```

```
SELECT-OPTIONS: so_carrid FOR spfli-carrid.
```



При указании критериев выбора в качестве типа ссылаются на поле в таблице АВАР словаря. Критерии, в отличие от параметров, позволяют получать комплексные запросы с экрана выбора (множественные значения). С помощью критериев можно задать диапазон дат, указать определенные условия <> какой-либо даты, исключить из диапазона перечень не нужных значений и т.п. Основное предназначение критериев выбора это передача их значений в операторы выборки SELECT в условии WHERE. Если параметры можно использовать как обычные АВАР переменные, то критерии внутри АВАР программы представляют собой внутреннюю таблицу особой структуры с заголовком. Структура таблицы состоит из 4-х полей:

- SIGN – Имеет символьный тип данных длиной 1. В качестве значения может быть I – обозначает необходимость вхождения значения в выборку. E – исключение выбранного значения из выборки.
- OPTION – Имеет символьный тип данных длиной 2. Содержит оператор сравнения если HIGH пустое, может быть заполнен следующими значениями: EQ, NE, GT, LE, LT, CP и NP каждое из которых задает определенную логику сравнения. EQ – значение равно выбранному, NE – значение не равно и т.д. Если HIGH заполнено, вы можете использовать BT или NB что означают вхождение значения в диапазон от LOW до HIGH или наоборот исключение значения из этого диапазона.
- LOW и HIGH – сами значения указанные пользователями, в качестве типа данных будет использован тип из поля в таблице АВАР словаря к которому привязан критерий выбора.

У критериев выбора могут быть заданы следующие атрибуты:

- SELECT-OPTIONS <seltab> FOR <f> DEFAULT <g> [TO <h>]
- SELECT-OPTIONS <seltab> FOR <f> ... NO-EXTENSION
- SELECT-OPTIONS <seltab> FOR <f> ... NO INTERVALS
- SELECT-OPTIONS <seltab> FOR <f> .. NO DATABASE
- SELECTION.....
- SELECT-OPTIONS <seltab> FOR <f> ... NO-DISPLAY
- SELECT-OPTIONS <seltab> FOR <f> ... LOWER CASE
- SELECT-OPTIONS <seltab> FOR <f> ... OBLIGATORY
- SELECT-OPTIONS <seltab> FOR <f> ... MEMORY ID <pid>.....
- SELECT-OPTIONS <seltab> FOR <f> ... MODIF ID <key>.....
- SELECT-OPTIONS <seltab> FOR <f> ... MATCHCODE OBJECT <obj>...

Следующий пример показывает как заполняется таблица критерия вывода после ввода пользователя, в примере делается цикл по таблице и выводятся все ее поля:

```
REPORT SAPMZTST.
```

```
TABLES SPFLI.
```

```
SELECT-POTIONS AIRLINE FOR SPFLI-CARRID.
```

```
LOOP AT AIRLINE.
```

```
  WRITE : / 'SIGN:', AIRLINE-SIGN,  
          'OPTION:', AIRLINE-OPTION,  
          'LOW:', AIRLINE-LOW,  
          'HIGH:', AIRLINE-HIGH.
```

```
ENDLOOP.
```

Использование критерия выбора в условиях выбора.

Критерии выбора можно использовать для ограничения выборки в операторах Open SQL таких как SELECT, UPDATE, DELETE, делается это при указании условия:

```
..... WHERE <f> IN <seltab>.
```

Пример:

```
REPORT SAPMZTST.
```

```
TABLES SPFLI.
```

```
SELECT-OPTIONS AIRLINE FOR SPFLI-CARRID.
```

```
SELECT * FROM SPFLI WHERE CARRID IN AIRLINE.
```

```
WRITE : SPFLI-CARRID.
```

```
ENDSELECT.
```

В данном примере из базы данных будут выбраны только те записи, которые удовлетворяют запросу пользователя. Если критерий выбора не будет задан, таблица будет выбрана целиком. Предположим что в таблице поле CARRID содержит следующие значения: **DL DL SQ UA UA UA**, тогда если пользователь укажет в критерии условие:

```
SIGN      OPTION LOW HIGH
```

```
E      EQ      LH
```

Будут выбраны все записи кроме LH.

Критерии выбора так же могут быть использованы в логических выражениях, например когда необходимо проверить вхождение значения в критерий выбора:

```
... <f> IN <seltab> ....
```

Логическое выражение будет истинным, если <f> удовлетворяет всем условиям в таблице критерия выбора <seltab>. Пример:

```
REPORT SAPMZTST.  
TABLES SPFLI.  
SELECT-OPTIONS AIRLINE FOR SPFLI-CARRID.  
WRITE : 'Inside', 'Outside'.  
SELECT * FROM SPFLI.  
  IF SPFLI-CARRID IN AIRLINE.  
    WRITE : / SPFLI-CARRID UNDER 'Inside'.  
  ELSE.  
    WRITE : / SPFLI-CARRID UNDER 'Outside'.  
  ENDIF.  
ENDSELECT.
```

Если относительно пользовательского критерия выбора будут выбраны данные из БД, выведется значение CARRID под словом INSIDE, иначе под словом OUTSIDE.

В SQL циклах SELECT ... ENDSELECT так же доступна короткая форма проверки критерия выбора:

...<seltab>...

```
SELECT * FROM SPFLI.
```

```
  IF AIRLINE.
```

```
    WRITE : / SPFLI-CARRID UNDER 'Inside'.
```

```
  ELSE.
```

```
    WRITE : / SPFLI-CARRID UNDER 'Outside'.
```

```
  ENDIF.
```

```
ENDSELECT.
```

Используя критерии выбора и параметры в программе, при ее запуске автоматически генерируется экран выбора, где критерии и параметры идут друг за другом последовательно. Для того чтобы иметь возможность отформатировать экран выбора (добавить тексты, пропуски, группы и т.п.) используется оператор SELECTION SCREEN. Данный оператор используется только для отчётов с экранами выбора.

Вставка разрывов (пустых строк) между элементами экрана выбора.

```
SELECTION-SCREEN SKIP [<n>].
```

Где n количество пустых строк.

Вставка подчеркиваний на экране.

```
SELECTION-SCREEN ULINE [[/]<pos(len)>] [MODIF ID <key>].
```

Где pos – позиция на экране, len – длинна подчеркивания.

Вставка текста.

```
SELECTION-SCREEN COMMENT [[/]<pos(len)> <name>] [MODIF ID <key>].
```

Где name – текст из хранилища текстов АВАР программы.

Пример использования оператора SELECTION-SCREEN:

```
SELECTION-SCREEN COMMENT /2(5) TEXT-001 MODIF ID SC1.
```

```
SELECTION-SCREEN SKIP 2.
```

```
SELECTION-SCREEN COMMENT /10(30) COMM1.
```

```
SELECTION-SCREEN ULINE.
```

```
PARAMETERS : R1 RADIOBUTTON GROUP RAD1,
```

```
R2 RADIOBUTTON GROUP RAD1,
```

```
R3 RADIOBUTTON GROUP RAD1.
```

```
SELECTION-SCREEN ULINE /1(50).
```

```
SELECTION-SCREEN COMMENT /10(30) COMM2.
```

```
SELECTION-SCREEN ULINE.
```

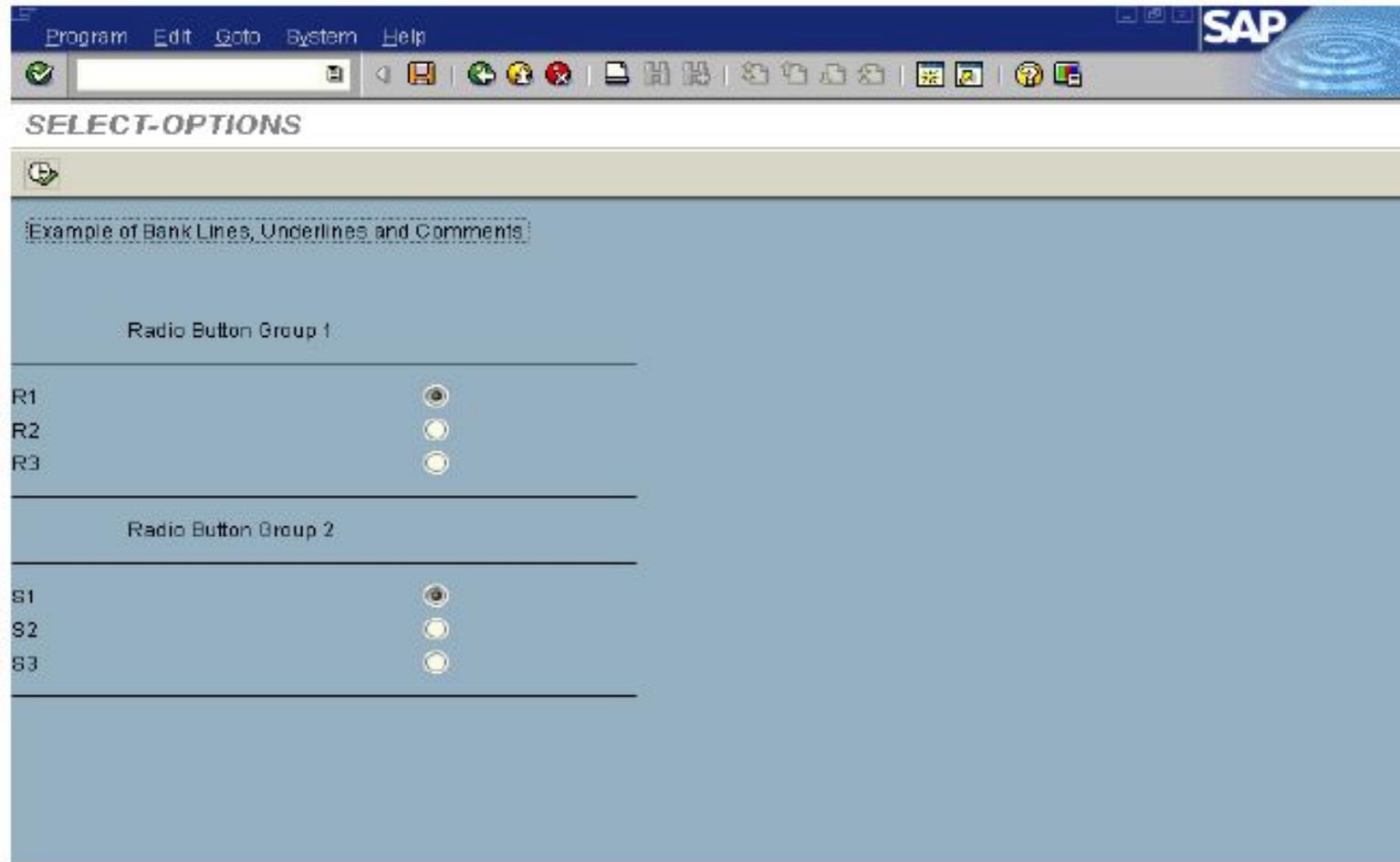
```
PARAMETERS : S1 RADIOBUTTON GROUP RAD2,
```

```
S2 RADIOBUTTON GROUP RAD2,
```

```
S3 RADIOBUTTON GROUP RAD2.
```

```
SELECTION-SCREEN ULINE /1(50).
```

Результат:



Расположение элементов в одну линию. По умолчанию каждый новый элемент экрана выбора начинается с новой линии, чтобы расположить элементы в одну линию необходимо воспользоваться дополнением BEGIN LINE:

```
SELECTION-SCREEN BEGIN OF LINE.
```

...

```
SELECTION-SCREEN END OF LINE.
```

Следует запомнить что тексты к критериям выбора и параметрам в таком случае выводится не будут, но для вывода текста можно воспользоваться ранее рассмотренным способом вывода комментариев SELECTION-SCREEN COMMENT...

Пример:

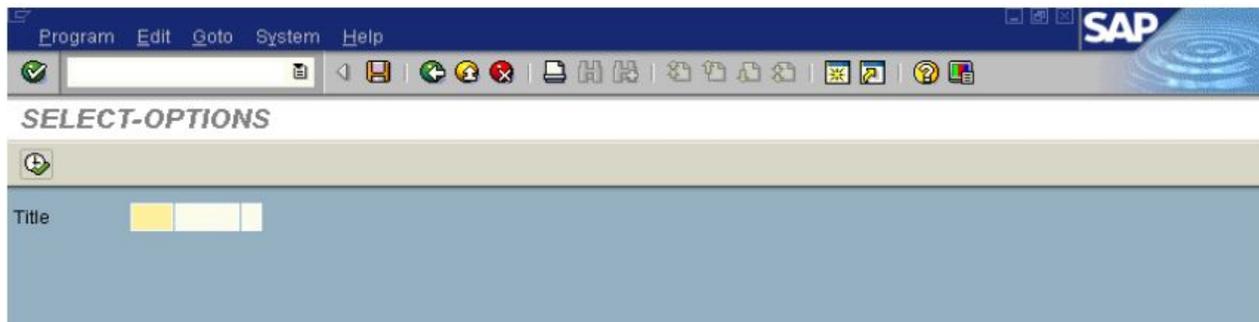
```
SELECTION-SCREEN BEGIN OF LINE.
```

```
SELECTION-SCREEN COMMENT 1(10) TEXT-001.
```

```
PARAMETERS : P1(3), P2(5), P3(1).
```

```
SELECTION-SCREEN END OF LINE.
```

Результат:



Вывод элемента на экране выбора в рамках линии можно позиционировать с помощью дополнения POSITION:

```
SELECTION-SCREEN POSITION <pos>.
```

Для <pos> можно воспользоваться константами POS_LOW и POS_HIGH. Пример:

```
REPORT SAPMZTST.
```

```
TABLES SPFLI.
```

```
SELECT-OPTIONS AIRLINE FOR SPFLI-CARRID.
```

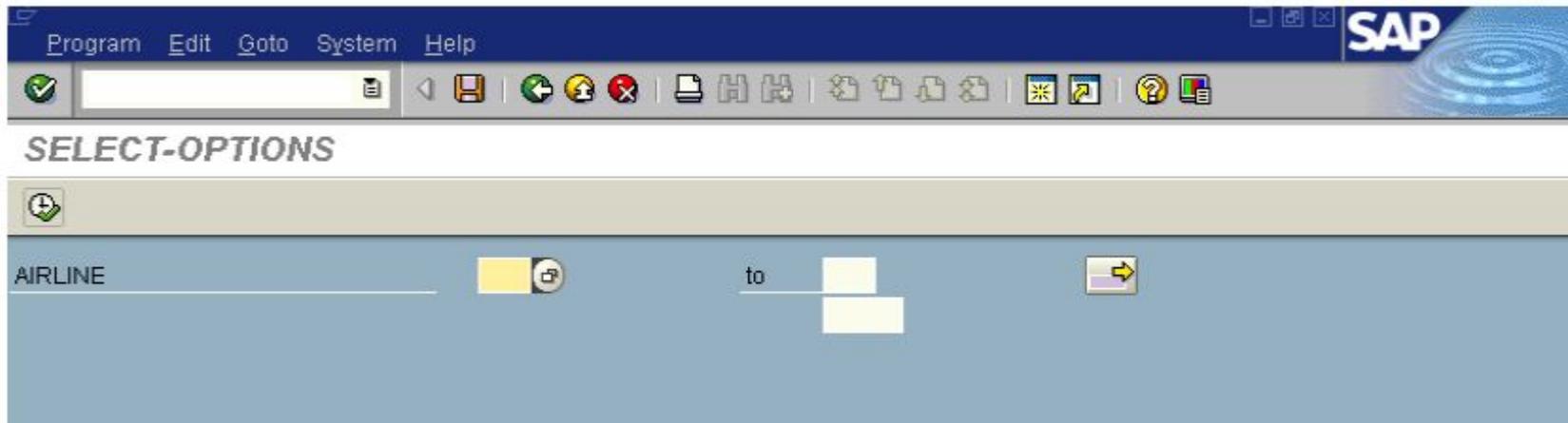
```
SELECTION-SCREEN BEGIN OF LINE.
```

```
SELECTION-SCREEN POSITION POS_HIGH.
```

```
PARAMETERS FIELD(5).
```

```
SELECTION-SCREEN END OF LINE.
```

Результат:



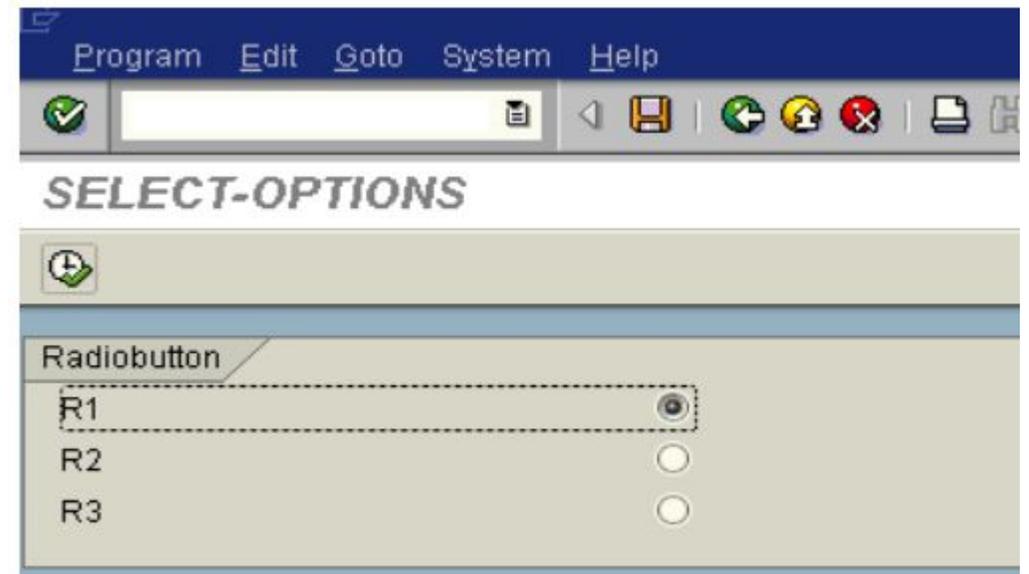
Объединение элементов на экране выбора в блоки.

```
SELECTION-SCREEN BEGIN OF BLOCK <block>
[WITH FRAME [TITLE <title>]] [NO INTERVALS].
...
SELECTION-SCREEN END OF BLOCK <block>.
```

Для каждого нового блока необходимо обозначить новый <block>. Если будет использовано дополнение WITH FRAME блок будет выведен вместе с рамкой, если использовать TITLE вместе с текстом в рамке. Если будет использовано дополнение NO INTERVALS, для всех критериев выбора расположенных внутри блока будет отключена возможность выбора интервальных значений.

Пример:

```
SELECTION-SCREEN BEGIN OF BLOCK RAD1 WITH FRAME TITLE
TEXT-002.
PARAMETERS R1 RADIOBUTTON GROUP GR1.
PARAMETERS R2 RADIOBUTTON GROUP GR1.
PARAMETERS R3 RADIOBUTTON GROUP GR1.
SELECTION-SCREEN END OF BLOCK RAD1.
```



Создание кнопок на панели инструментов отчёта (Application toolbar)

На экране выбора в панели инструментов можно задать до 5 кнопок, используя следующий синтаксис:

```
SELECTION-SCREEN FUNCTION KEY <i>.
```

<i> должен быть от 1 до 5. Тексты кнопок задаются во время инициализации отчёта в блоке события INITIALIZATION, для этого необходимо присвоить значение в заголовке системной таблицы:

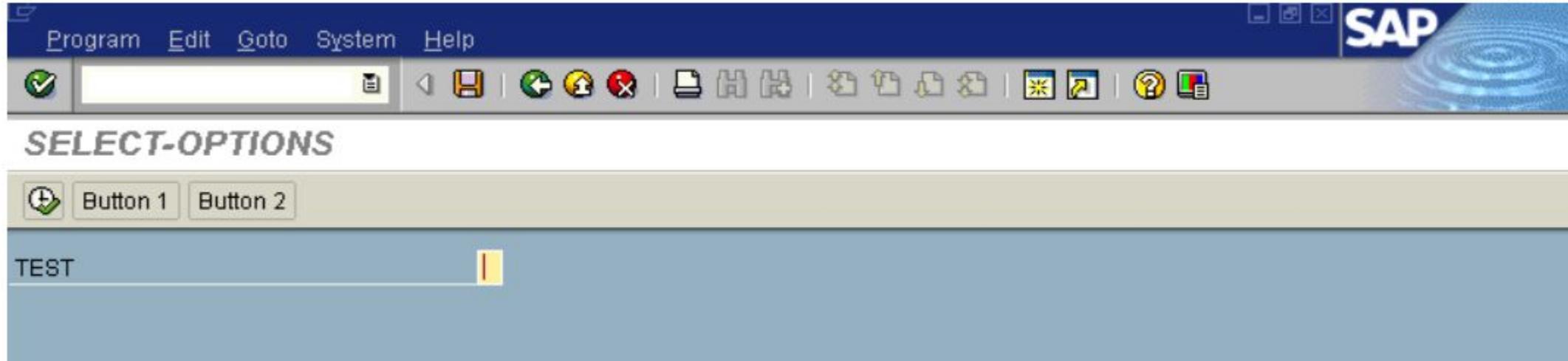
```
SSCRFIELDS-FUNCTXT_0<i>
```

Чтобы иметь возможность работать с этой таблицей, её необходимо объявить в начале программы через оператор TABLES: SSCRFIELDS .

Пример:

```
TABLES SSCRFIELDS.
DATA FLAG.
PARAMETERS TEST:
SELECTION-SCREEN FUNCITON KEY 1.
SELECTION-SCREEN FUNCITON KEY 2.
INITIALIZATION.
    SSCRFIELDS-FUNCTXT_01 = 'Button 1'. "Задаем имя для кнопки
    SSCRFIELDS-FUNCTXT_02 = 'Button 2'.
AT SELECTION-SCREEN.
    IF SSCRFIELDS-UCOMM = 'FC01'. "Если нажали на кнопку 1 ...
        FLAG = '1'.
    ELSEIF SSCRFIELDS-UCOMM = 'FC02'.
        FLAG = '2'.
    ENDIF.
START-OF-SELECTION.
    IF FLAG = '1'.
        WRITE : / 'Button 1 was clicked'.
    ELSEIF FLAG = '2'.
        Write : / 'Button 2 was clicked'.
```

Результат:



В данном примере, программа запоминает на какую кнопку было нажатие в переменной FLAG и в зависимости от неё после запуска отчёта выводит либо один текст, либо другой. Используемые в программе события AT SELECTION SCREEN, INITIALIZATION, START-OF-SELECTION будут рассмотрены далее.

Кроме кнопок на панели инструментов, так же есть возможность создания кнопок на самом экране выбора:

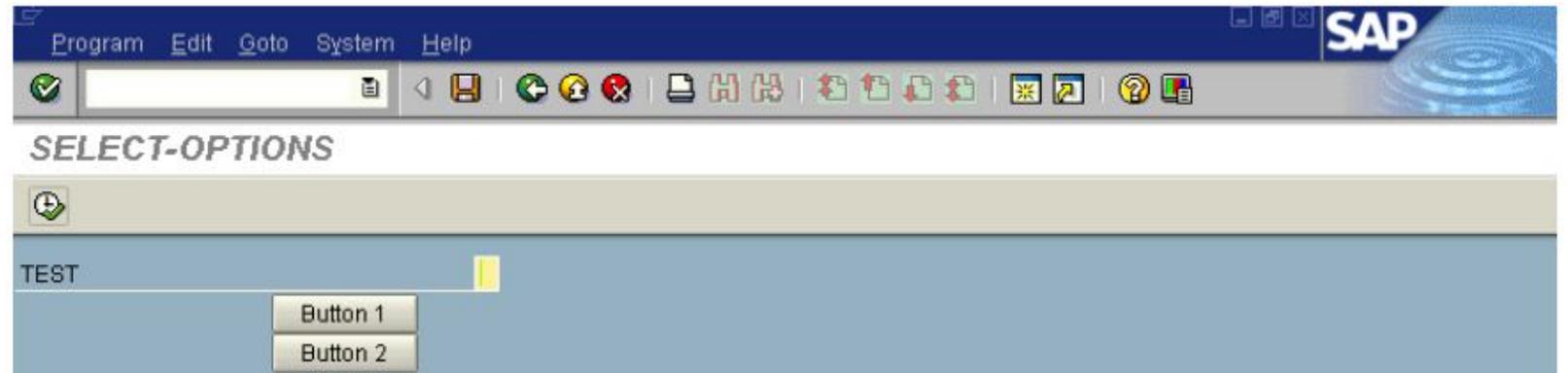
```
SELECTION SCREEN PUSHBUTTON [/]<pos(len)> <name> USER-COMMAND <ucom> [MODIF ID <key>].
```

Параметры <pos(len)> аналогичны ранее рассмотренным для команды вывода комментария COMMENT. <name> - задает имя кнопки. <ucom> - код из 4-х символов, необходимый для идентификации команды передаваемый в обработчик экрана.

Далее будет рассмотрен пример аналогичный предыдущему, за исключением того, что кнопки расположены на самом экране.

Пример:

```
TABLES SSCRFIELDS.  
DATA FLAG.  
PARAMETERS TEST:  
SELECTION-SCREEN PUSHBUTTON /20(10) BUT1  
USER-COMMAND CLI1.  
SELECTION-SCREEN PUSHBUTTON /20(10) TEXT-020  
USER-COMMAND CLI2.  
INITIALIZATION.  
BUT1 = 'Button 1'.  
AT SELECTION-SCREEN.  
IF SSCRFIELDS-UCOMM = 'FCI1'.  
FLAG = '1'.  
ELSEIF SSCRFIELDS-UCOMM = 'CLI2'.  
FLAG = '2'.  
ENDIF.  
START-OF-SELECTION.  
IF FLAG = '1'.  
WRITE : / 'Button 1 was clicked'.  
ELSEIF FLAG = '2'.  
Write : / 'Button 2 was clicked'.  
ENDIF.
```



Варианты программы

При отображении любого экрана выбора, его значения могут быть сохранены в так называемом варианте, чтобы при повторном запуске программы не указывать все значения заново.

Варианты используются как в онлайн режиме, так и при планировании запуска программы в фоновом режиме.

Варианты могут быть созданы как для конкретного пользователя (user variant), так и для всех пользователей.

Кроме непосредственно хранения значений в виде констант, варианты позволяют динамически заполнять значения переменных. Пример: если требуется при выборе варианта указывать всегда текущую дату. При сохранении варианта необходимо указать заполнения параметра динамически:

Атрибуты варианта

Скопировать присвоение экрана

Имя варианта: TEST

Значение: TEST

Только для фоновой обработки

Защитить вариант

Просмотр т. в каталоге

Системный вариант (автоперенос)

Техническое имя

Объекты экрана выбора

Экраны выбо...	Имя поля	Тип	Защитить поле	Скр
1.000	P_TEST	P	<input type="checkbox"/>	

Атрибуты варианта

Выбор переменных выбора

Вид переменной	Значение
T	T: табличная переменная из TVARV
D	D: динамический расчет даты (местная дата)
X	X: динамический расчет даты (системная дата)

е обязательного ввода

Перем
X

В качестве значений для параметров (критериев выборов) внутри варианта могут быть использованы так называемые переменные варианта. Это значения критериев выбора или параметров сохраненные через транзакцию STVARV (в таблице TVARV):

Просмотр таблицы TVARVC: переменные в выборках

Ведение по отдельности

Содержимое таблицы TVARVC Включить измененные записи в запрос на перенос

Параметры | Оции выбора

Имя	Зн.	ПРО...
SAPSCLAS_RADIO_B_INIT_DISP	X	<input type="checkbox"/>
SAP_SCMA_DETAIL_LIST		<input type="checkbox"/>
SAP_SCMA_FISC_YEAR		<input type="checkbox"/>
SAP_SCMA_PERIOD		<input type="checkbox"/>
SAP_SCMA_POST_DATE		<input type="checkbox"/>
SAP_SCMA_POST_PERIOD		<input type="checkbox"/>
SAP_SCMA_SELV_PROJECT		<input type="checkbox"/>
SAP_SCMA_TEST_RUN		<input type="checkbox"/>
SAP_SCMA_VALUE_DATE		<input type="checkbox"/>

Сохранение вариантов производится либо через АВАР редактор на его начальном экране, либо нажав кнопку сохранить на любом из экранов выбора, после чего будет показан следующий экран:

Атрибуты варианта

Скопировать присвоение экрана

Имя варианта: ZTEST

Значение: Описание варианта

Только для фоновой обработки
 Защитить вариант
 Просмотр т. в каталоге
 Системный вариант (автоперенос)

Техническое имя

Объекты экрана выбора

Экраны выбора	Имя поля	Тип	Защитить поле	Скрыть поле	Скрыть поле "ПО"	Сохранить поле без значений
1.000	Финансовый год	P	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ПрисвЭкрана

Создан.	Экраны выбора
<input checked="" type="checkbox"/>	1000
<input type="checkbox"/>	1001

На экране указываются все критерии выбора/параметры, значения которых можно сохранить в варианте.

Программа Обработать **Перейти к** Система Справка

Монитор контроля

Финансовый год: 2015

Варианты

- Пользовательские переменные...
- Справка/Экран выбора
- Назад F3
- Вызвать... Shift+F5
- Просмотреть...
- Удалить...**
- Сохранить как вариант... Ctrl+S

Контроль за потоком выполнения АВАР программы через обработку событий.

Язык АВАР является событийно-ориентированным. Это означает что основной поток выполнения АВАР программы контролируется внешними событиями. Части программной логики создаются специально под отдельные события. Эти части начинают свое выполнение в тот момент, когда система запускает соответствующие события.

События:

- INITIALIZATION – событие инициализации, срабатывает до того момента как пользователь увидит селекционный экран (экран выбора)
- AT SELECTION-SCREEN – обработка событий возникающих на экране выбора (нажатие кнопок, выбор радиокнопки и т.п.)
- START-OF-SELECTION – событие возникающее сразу после ввода всех данных на экране выбора и нажатия выполнения
- GET <table> - событие возникающее в момент когда система готова предоставить данные строки таблицы <table> из логической базы данных связанной с АВАР программой. Логические базы позволяют получать определенный набор данных, при этом сами генерируют экраны выбора на основе которых эти данные выбираются.
- END-OF-SELECTION – событие возникающее после обработки всех данных в логической базе данных

События релевантные для отчётов представленных в виде АВАР списка:

- TOP-OF-PAGE – вызывается каждый раз, когда создается новая страница АВАР списка. Используется обычно для вывода текста заголовка для каждой страницы.
- END-OF-PAGE – событие аналогичное предыдущему, только вызываемое в момент завершения страницы АВАР списка.

События отвечающие за интерактивность отчётов:

- AT LINE-SELECTION – в отчётах в виде списка, данное событие возникает при выборе пользователем линии из списка.
- AT USER-COMMAND – событие возникающие когда пользователь нажимает функциональную клавишу (или пункт в меню) или когда вводит команду в окне ввода команд.
- AT PF<nn> - событие возникающие при обработке события нажатия на функциональную клавишу, которой присвоен код <nn>.

Вывод сообщений

Сообщение в ABAP вызывается через оператор MESSAGE, при этом используется разный синтаксис, но смысл остается одним, необходимо указать тип сообщения. Рассмотрим один из возможных вариантов вызова:

```
MESSAGE 'Текст сообщения' TYPE 'E'.
```

Типы сообщений могут быть следующими:

- A – abort, при вызове сообщения с таким типом текущая транзакция моментально завершается, используется при необходимости срочного «ошибочного» завершения работы транзакции.
- E – error, вызывается для того чтобы предупредить пользователя о возможной ошибке (пример: пользователь вводит в качестве номера паспорта не цифровые символы).
- I – information, вызывается для того чтобы проинформировать пользователя о чём либо, после нажатия на ENTER происходит дальнейшая обработка события.
- S – confirmation, подтверждение – отображается уже на следующем экране. Например: после считывания данных из БД, можно перед непосредственно выводом их на экран, уведомить пользователя о количестве считанных записей. В стандартных транзакциях SAP системы данное сообщение обычно показывается после успешного сохранения какого-либо объекта в системе.
- W – warning, уведомляет пользователя о возможных проблемах, после нажатия на ENTER программа продолжает свое выполнение.

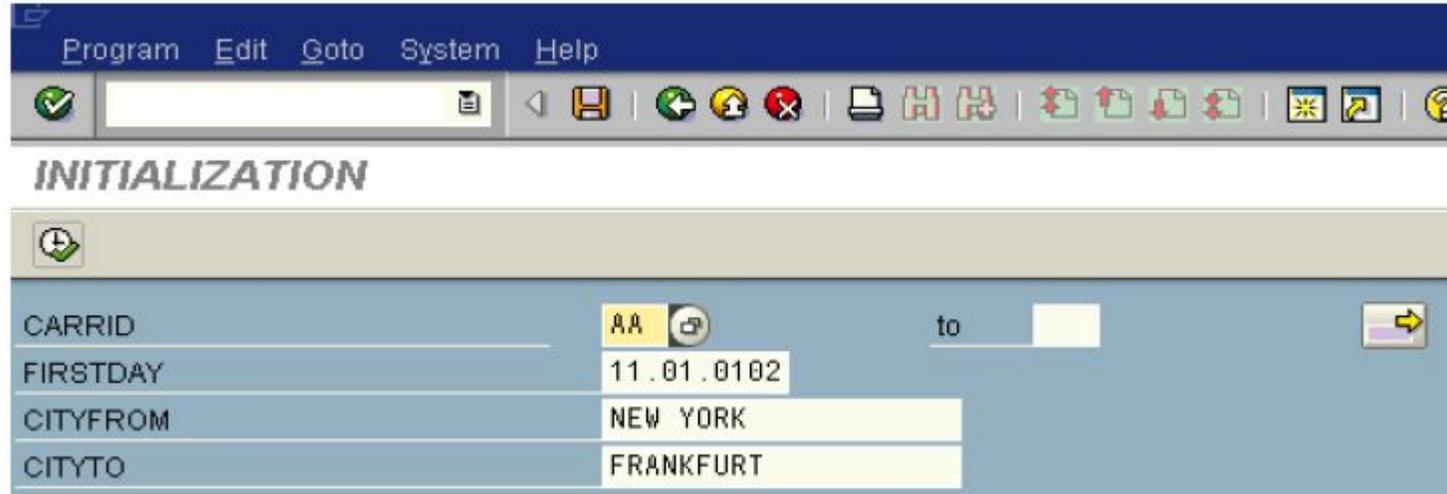
Событие инициализации

Событие вызывается сразу после старта программы, но перед выводом экрана выбора (или если программа связана с логической базой данных, с экраном выбора ЛБД). Используется как правило для инициализации текстов (или значений) на экране выбора или каких-либо внутренних переменных необходимых для работы отчёта.

Пример:

```
REPORT SAPMZTST.  
TABLES SPFLI.  
SELECT-OPTIONS : CARRID FOR SPFLI-CARRID.  
PARAMETERS : FIRSTDAY LIKE SY-DATUM DEFAULT SY-DATUM,  
CITYFROM LIKE SPFLI-CITYFROM,  
CITYTO LIKE SPFLI-CITYTO.  
INITIALIZATION.  
CITYFROM = 'NEW YORK'.  
CITYTO = 'FRANKFURT'.  
CARRID-SIGN = 'I'.  
CARRID-OPTION = 'EQ'.  
CARRID-LOW = 'AA'.  
APPEND CARRID.  
FIRSTDAY+(2) = '01'.
```

В данном примере событие инициализации используется для заполнения параметров и критериев выбора на экране выбора начальными значениями.



The screenshot shows a software window with a menu bar containing 'Program', 'Edit', 'Goto', 'System', and 'Help'. Below the menu bar is a toolbar with various icons. The main area is titled 'INITIALIZATION' and contains a table with the following data:

Field	Value
CARRID	AA
FIRSTDAY	11.01.0102
CITYFROM	NEW YORK
CITYTO	FRANKFURT

Обработка событий экрана выбора

Для обработки событий с экрана выбора используется событие AT SELECTION-SCREEN с указанием дополнений:

- AT SELECTION-SCREEN ON <field>. Позволяет обработать правильность заполнения критерия выбора или параметра <field>
- AT SELECTION-SCREEN ON END OF <seltab>. Позволяет обработать таблицу критерия выбора <seltab>, после заполнения её значениями в диалоге.
- AT SELECTION-SCREEN ON VALUE-REQUEST FOR <field>. Позволяет назначить свой обработчик в момент выбора значения для поля (по нажатию F4) <field>
- AT SELECTION-SCREEN ON HELP-REQUEST FOR <field>. Позволяет переопределить реакцию на нажатие F1 по полю.
- AT SELECTION-SCREEN ON RADIOBUTTON GROUP <radi>. Позволяет среагировать на изменение значения в радио группе.
- AT SELECTION-SCREEN ON BLOCK <block>. Срабатывает когда все поля в блоке были заполнены и обработаны.
- AT SELECTION-SCREEN OUTPUT. Событие срабатывающее до непосредственного вывода экрана выбора, позволяющее динамически менять его содержимое (скрывать поля ввода, менять их свойства).

Без использования дополнений, событие будет вызвано, когда обработка экрана выбора завершена.

START-OF-SELECTION

Данное событие вызывается сразу после завершения обработки экрана выбора. В контексте данного события обычно вызывается основная логика работы программы: считывание данных из БД, генерация пользовательского интерфейса вывода и т.д.

Если в своей программе Вы не определите никакого обработчика события, любой код после определения имени программы PROGRAM Z... будет выполнен в контексте этого события не явным образом. Т.е. система подставит весь ваш код в обработку этого события.

Следующие две программы равны:

```
PROGRAM ZTEST01.  
PERFORM main_logic.  
FORM main_logic .. ENDFORM.
```

```
PROGRAM ZTEST01.  
START-OF-SELECTION.  
    PERFORM main_logic.  
FORM main_logic .. ENDFORM.
```

END-OF-SELECTION

Событие вызывается системой после того как все таблицы из логической базы данных были обработаны.

Дополнительная документация по всем ключевым словам может быть найдена на официальном сайте компании SAP:

http://help.sap.com/abapdocu_740/en/index.htm