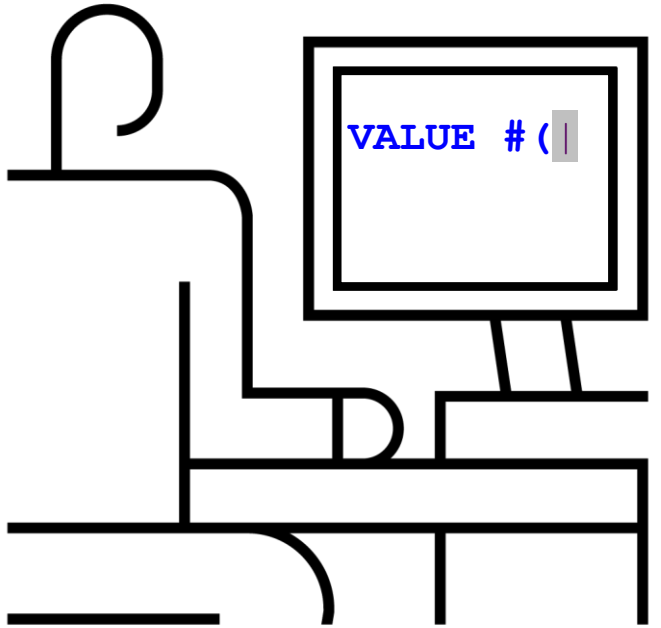


Обучение ABAP.
Операторы конструкторы VALUE и NEW

Василий Ковальский, SAP CIS
12 октября 2021 12:00 – 13:00

Презентация доступна по ссылке
https://cloud.mail.ru/public/WvfK/AgZ9Qcvwc

Василий Ковальский,
инструктор ABAP с 1998 года

vassili.kovalski@sap.com

https://www.facebook.com/Vassili.Kovalski

https://www.facebook.com/abap.education

Серия вебинаров Обучение ABAP. Октябрь 2021

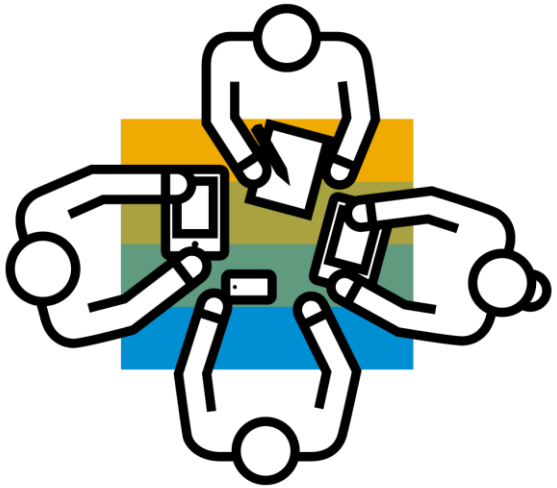
Операторы конструкторы **`VALUE`** и **`NEW`** — 12 октября 2021

Точные вычисления. **`EXACT`** — 13 октября 2021

Передача кластеров данных между программами. **`EXPORT`** и **`IMPORT`** — 14 октября 2021

Интересны другие темы? Пишите, подумаем

Для кого этот вебинар



- в первую очередь для начинающих разработчиков
- для специалистов служб поддержки и проверки качества кода
- для всех интересующихся ABAP

О чем пойдет речь

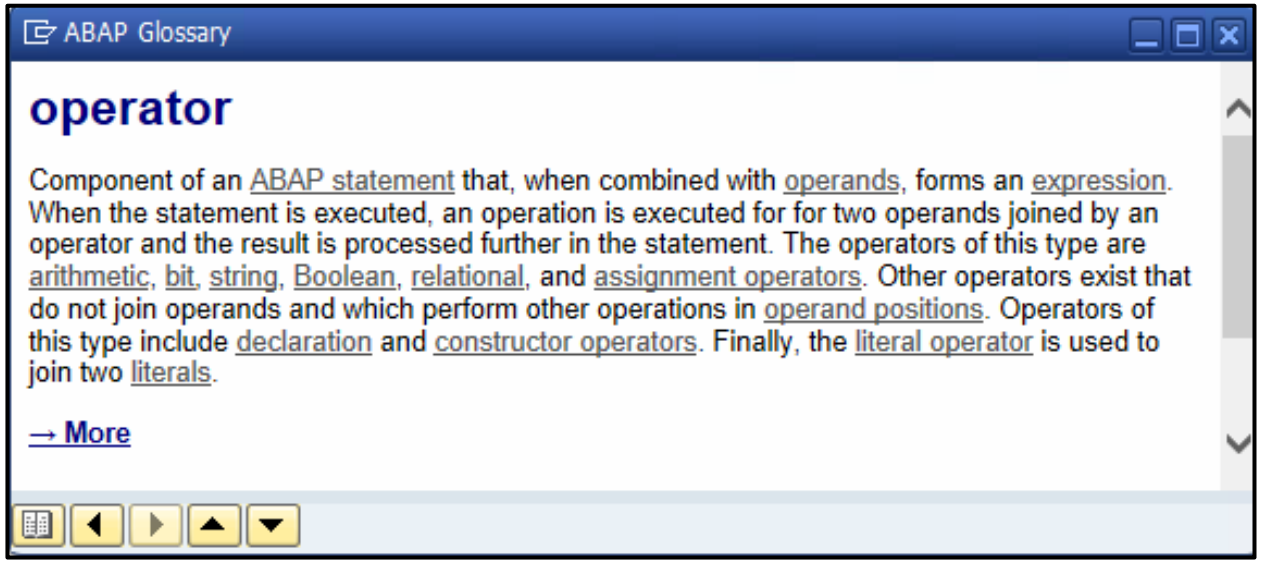
Операторы конструкторы **VALUE** и **NEW** позволяют создавать данные любой сложности «на лету», при работе со структурами и внутренними таблицами позволяют делать очень интересные вещи. Оператор конструктор **NEW** создает объекты

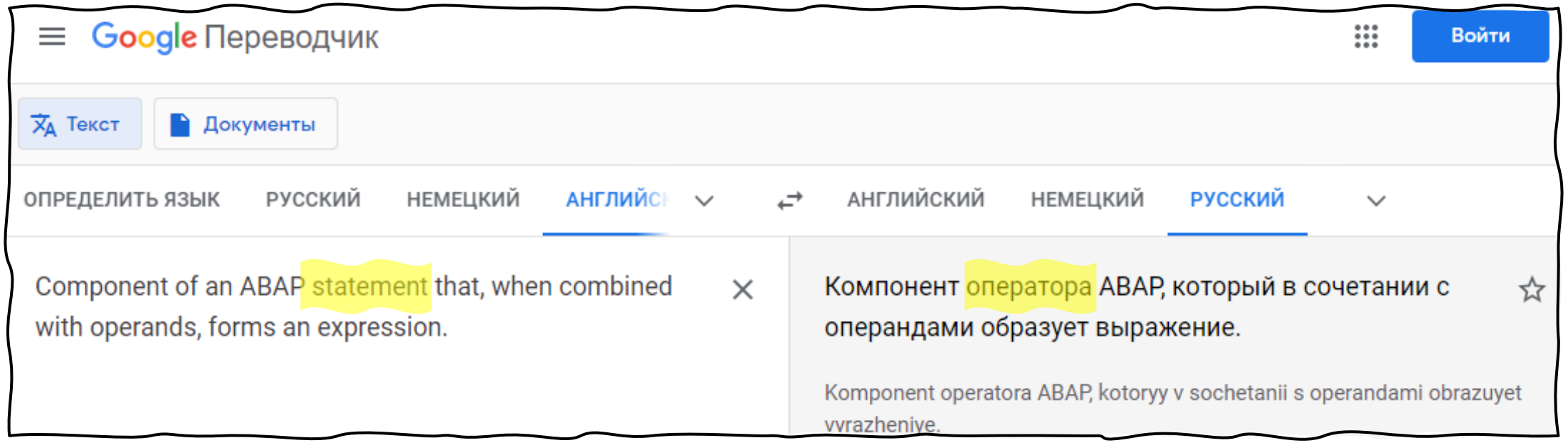
Мы обсудим:

- Основы синтаксиса
- Применение к структурам и внутренним таблицам
- Создание объектов
- Итератор **FOR**
- Где еще об этом узнать
- Что еще новое по ABAP этой осенью.

Предположительная продолжительность **~ 1 час**

“Statement” и “Operator”





Общий синтаксис операторов конструкторов

Constructor Operators for Constructor Expressions

Syntax

```
... NEW | VALUE | CONV | CORRESPONDING | CAST | REF | EXACT | REDUCE | FILTER | COND | SWITCH
      type( ... ) ...
```

Effect

A constructor expression consists of a

- predefined constructor operator,
- a data type or object type **type** that matches the operator and that can be derived implicitly from the operand position using **#**,
- and type-specified parameters specified in parentheses.

Each constructor expression creates a result whose data type is determined using the specified type. The parameters specified in parentheses are used to pass input values. The following constructor operators exist:

- The instance operator **NEW** is used to create objects in operand positions. The result is a reference variable of the static type **type** that points to the new object. The input values determine the content of the new object.
- The value operator **VALUE** is used to fill complex data objects with values in operand positions, create initial values of any data type, or control the results of table expressions. The result is a data object of the specified type **type**. The input values determine the content of the result.

Создание структуры. VALUE <<тип>>|#(...)

```
REPORT zqk_ws081_01."01. VALUE <<type>>|#( ... )
DATA
: gs_msg1 TYPE t100
, gs_msg2 TYPE t100
.

START-OF-SELECTION.
gs_msg1 = VALUE t100( sprsl = 'E' msgnr = 234 text = 'ABAP forever' ).
zcl_qk=>write_flat_structure( gs_msg1 ).
gs_msg2 = VALUE #( sprsl = 'E' arbgb = 'ZQK_WS081' text = 'I like ABAP' ).
zcl_qk=>write_flat_structure( gs_msg2 ).
SKIP.
ULINE.
WRITE: / 'BOT TAk!'.
```

```
01. VALUE <<type>>|#( ... )

Language        EN
Area
Message         234
Message Text    ABAP forever

Language        EN
Area            ZQK_WS081
Message
Message Text    I like ABAP

BOT TAk!
```

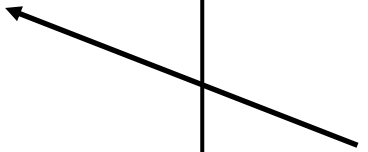
Создание структуры, получение ссылки. NEW <<тип>>|#(...)

```
REPORT zqk_ws081_02."02. NEW <<type>>|#( ... )
DATA
: gs_msg1 TYPE REF TO t100
, gs_msg2 TYPE REF TO t100
.

START-OF-SELECTION.
gs_msg1 = NEW t100( sprsl = 'E' msgnr = 234 text = 'ABAP forever' ).
zcl_qk=>write_flat_structure_by_ref( gs_msg1 ).
gs_msg2 = NEW #( sprsl = 'E' arbgb = 'ZQK_WS081' text = 'I like ABAP' ).
zcl_qk=>write_flat_structure_by_ref( gs_msg2 ).
SKIP.
ULINE.
WRITE: / 'BOT TAk!'.
```

```
01. VALUE <<type>>|#( ... )

Language          EN
Area
Message           234
Message Text      ABAP forever

Language          EN
Area              ZQK_WS081
Message
Message Text      I like ABAP

BOT TAk!
```

Создание объектов. NEW и CREATE OBJECT

```
DATA
: go_cont TYPE REF TO cl_gui_docking_container
, go_grid TYPE REF TO cl_gui_alv_grid
. . .
CREATE OBJECT go_cont
  EXPORTING
    side  = cl_gui_docking_container=>dock_at_bottom
    ratio = 90.
CREATE OBJECT go_grid
  EXPORTING
    i_parent = go_cont.
```



Можно не создавать
ненужных переменных

```
DATA
: go_grid TYPE REF TO              cl_gui_alv_grid
. . .
  go_grid = NEW cl_gui_alv_grid(
      i_parent = NEW cl_gui_docking_container(
          side   = cl_gui_docking_container=>dock_at_bottom
          ratio  = 90
      )
  ).
```

VALUE при передаче параметров

```
DATA gs_layo TYPE lvc_s_layo.
gs_layo-zebra = 'X'.
go_grid->set_table_for_first_display(
  EXPORTING
    i_structure_name = 'T100'
    is_layout        = gs_layo
  CHANGING
    it_outtab        = gt
).
```

Можно не создавать ненужных переменных

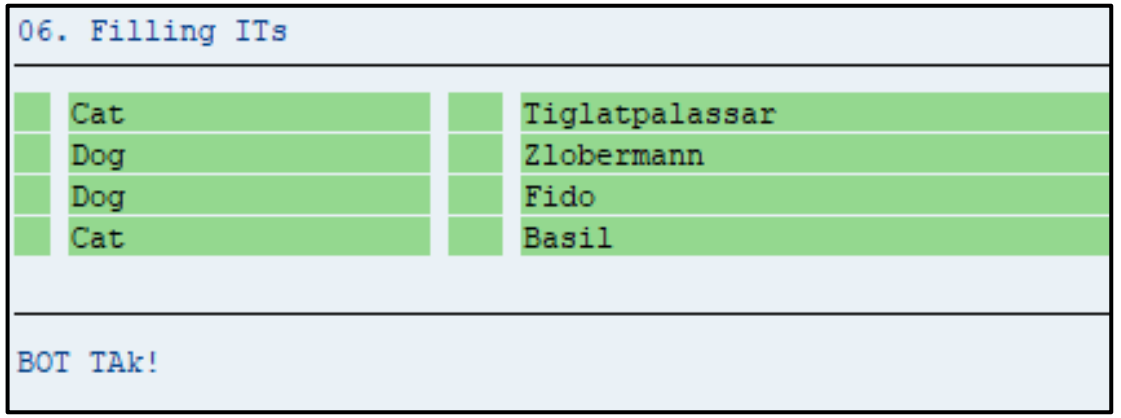
Можно присваивать значения не заранее,

а ровно в нужном месте

```
go_grid->set_table_for_first_display(
   EXPORTING
     i_structure_name = 'T100'
     is_layout        = VALUE lvc_s_layo( zebra = 'X' )
   CHANGING
     it_outtab        = gt
 ).
```

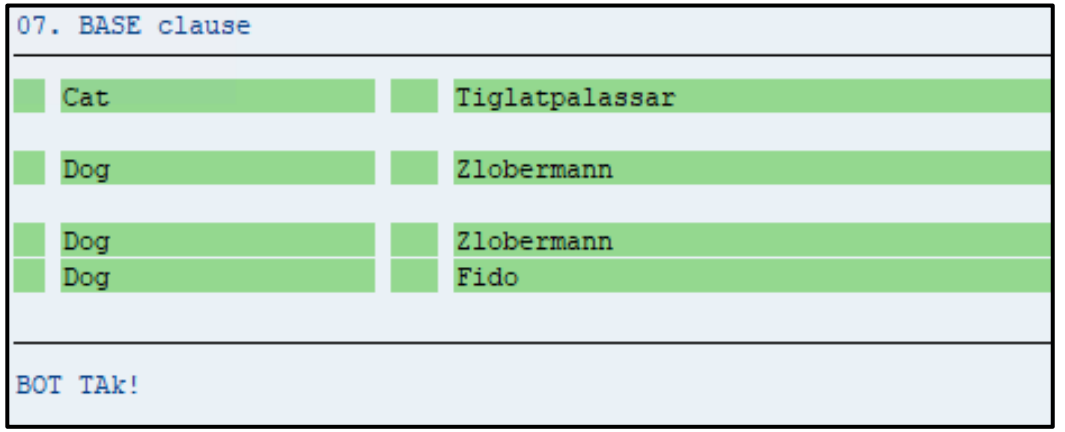
Заполнение внутренних таблиц

```
TYPES tt TYPE STANDARD TABLE OF t100 WITH EMPTY KEY.
DATA  gt TYPE tt.

gt = VALUE tt(
      ( arbgb = 'Cat' text = 'Tiglatpalassar' )
      ( arbgb = 'Dog' text = 'Zlobermann'     )
      ( arbgb = 'Dog' text = 'Fido'           )
      ( arbgb = 'Cat' text = 'Basil'          )
   ).
```

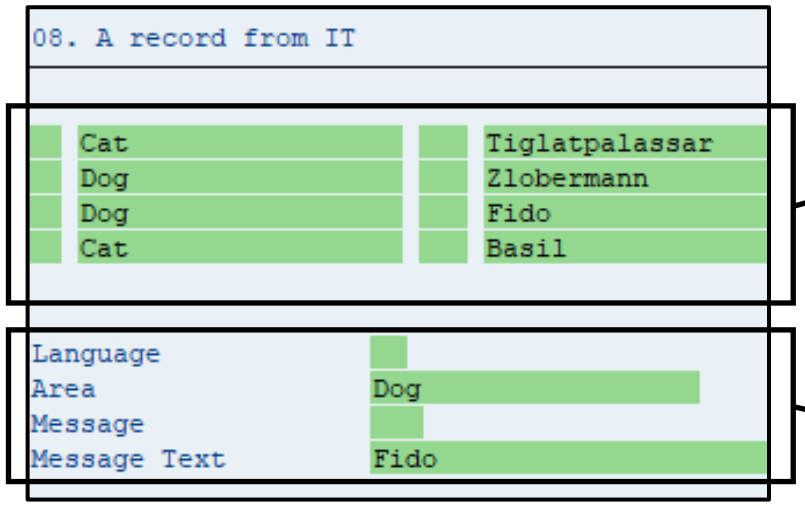


Фраза BASE

```
DATA  gt TYPE tt.
gt = VALUE tt( ( arbgb = 'Cat' text = 'Tiglatpalassar' ) ).
zcl_qk=>write_table( gt ).
gt = VALUE tt( ( arbgb = 'Dog' text = 'Zlobermann'     ) ).
skip.
zcl_qk=>write_table( gt ).
gt = VALUE tt( BASE gt ( arbgb = 'Dog' text = 'Fido'    ) ).
skip.
zcl_qk=>write_table( gt ).
```



Создание данных по записи внутренней таблицы




```
TYPES tt TYPE STANDARD TABLE OF t100 WITH EMPTY KEY.
DATA: gt TYPE tt.

gt = VALUE tt(
      ( arbgb = 'Cat' text = 'Tiglatpalassar' )
      ( arbgb = 'Dog' text = 'Zlobermann'     )
      ( arbgb = 'Dog' text = 'Fido'           )
      ( arbgb = 'Cat' text = 'Basil'          )
   ).

DATA(gs) = VALUE #( gt[ text = 'Fido' ] ).
```

Создание данных по записи внутренней таблицы. Фраза DEFAULT

```
TYPES tt TYPE STANDARD TABLE OF t100 WITH EMPTY KEY.
PARAMETERS p_error AS CHECKBOX.

START-OF-SELECTION.
data(gt) = VALUE tt(
    ( arbgb = 'Cat' text = 'Tiglatpalassar' )
    ( arbgb = 'Dog' text = 'Zlobermann'     )
    ( arbgb = 'Dog' text = 'Fido'           )
    ( arbgb = 'Cat' text = 'Basil'          )
    ).

IF p_error = abap_true.
  data(gs) = VALUE #( gt[ text = 'Fid@' ] )." ! ERROR !
ELSE.
  data(gs_default) = VALUE t100( arbgb = 'Horse' text = 'Sivka' ).
  gs = VALUE #( gt[ text = 'Fid@' ] DEFAULT  gs_default ).
* gs = VALUE #( gt[ text = 'Fid@' ] DEFAULT  VALUE #( arbgb = 'Horse' text = 'Sivka' ) ).
ENDIF.
```

Category	ABAP programming error
Runtime Errors	ITAB_LINE_NOT_FOUND
Except.	CX_SY_ITAB_LINE_NOT_FOUND
ABAP Program	ZQK_WS081_09

Horse
Sivka

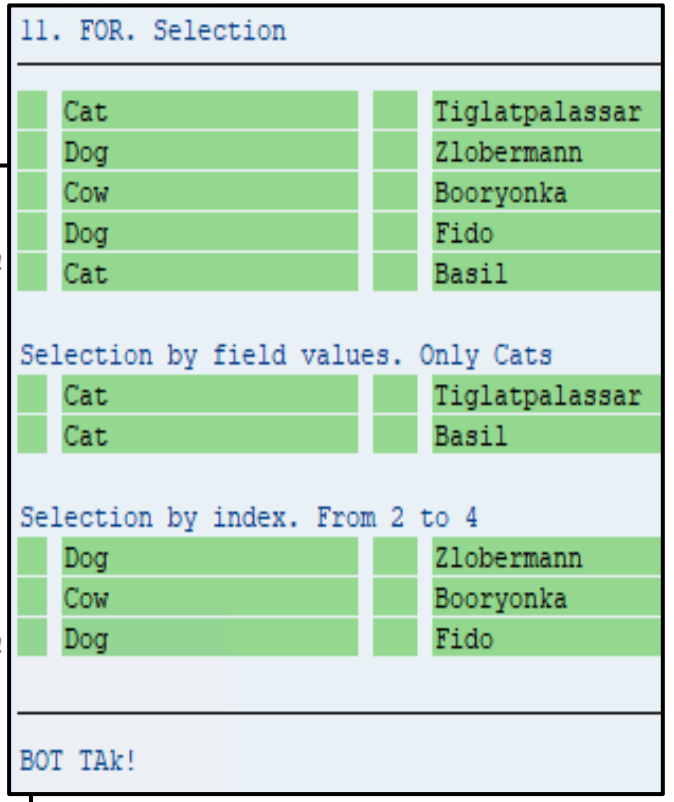
Итератор FOR ... THEN

```
TYPES BEGIN OF ts
,      xx TYPE p DECIMALS 3
,      ff TYPE p DECIMALS 3
,      gg type string
,     END OF ts
,     tt type STANDARD TABLE OF ts WITH EMPTY KEY.

DATA(gt)   = VALUE tt(
     FOR    i = -10
     THEN   i + 1
     WHILE i <= 10
*    UNTIL i > 10
     ( xx = i
       ff = ( exp( i / 10 ) + exp( - i / 10 ) ) / 2
       gg = `|`
          && repeat( val = ` ` occ = ( exp( i / 10 ) + exp( - i / 10 ) ) * 20 - 1 )
          && `*`
     )
  ).
```

```
10. FOR, THEN, WHILE, UNTIL clauses

Hyperbolic cosine: y = ch( x )
10,000-  1,543 *
 9,000-  1,433 *
 8,000-  1,337 *
 7,000-  1,255 *
 6,000-  1,185 *
 5,000-  1,128 *
 4,000-  1,081 *
 3,000-  1,045 *
 2,000-  1,020 *
 1,000-  1,005 *
     0   1,000 *
 1,000   1,005 *
 2,000   1,020 *
 3,000   1,045 *
 4,000   1,081 *
 5,000   1,128 *
 6,000   1,185 *
 7,000   1,255 *
 8,000   1,337 *
 9,000   1,433 *
10,000   1,543 *
```

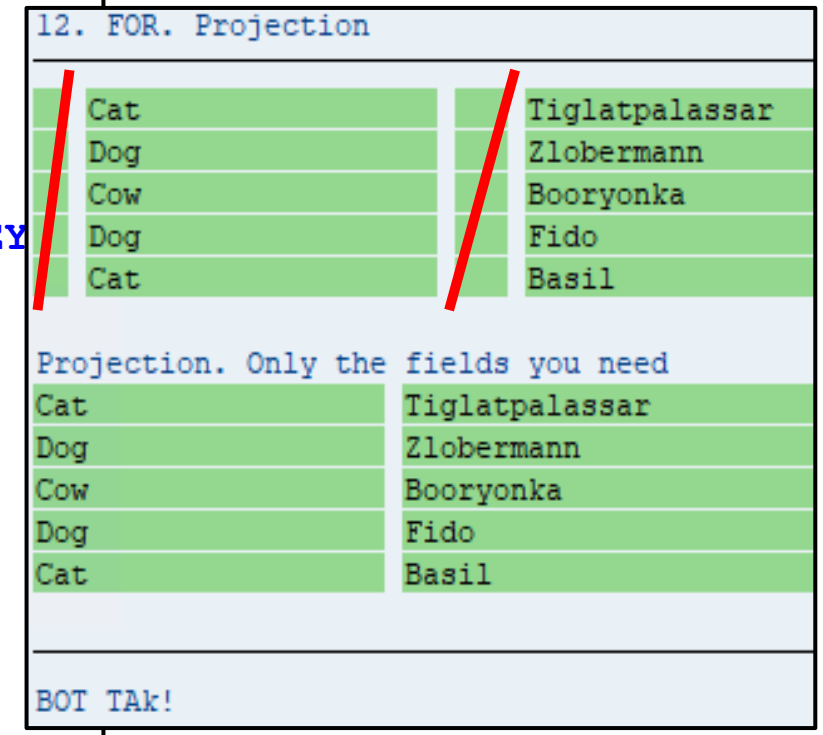
Итератор FOR. Селекция

```
DATA(g1) = VALUE tt(
         FOR    wa                    " Work area definition
         IN     gt                    " Source table
         WHERE ( arbgb = 'Cat' )      " condition
         ( wa )                       " Result
      ).

DATA(g2) = VALUE tt(
         FOR    wa                    " Work area definition
         IN     gt                    " Source table
         FROM   2                     " From
         TO     4                     " To
         ( wa )                       " Result
      ).
```



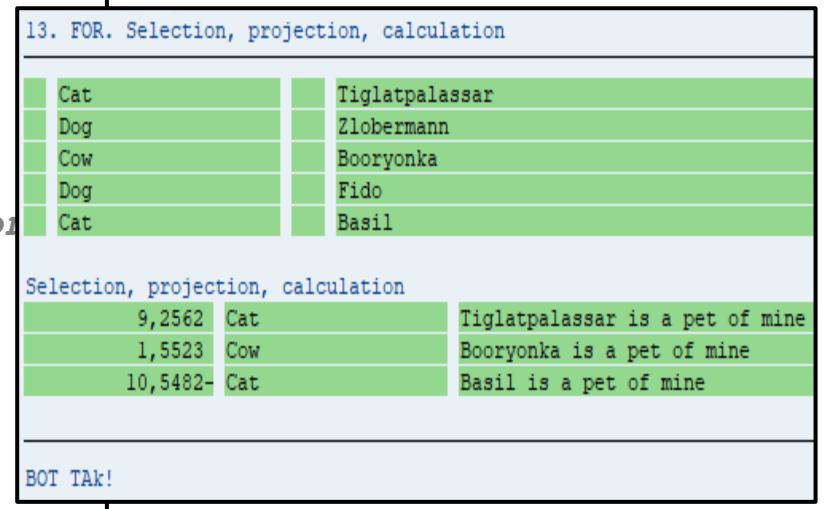
Итератор FOR. Проекция

```
TYPES: tt TYPE STANDARD TABLE OF t100 WITH EMPTY KEY
     , BEGIN OF ts
     ,   species TYPE t100-arbgb
     ,   nick    TYPE t100-text
     , END OF ts
     , td TYPE STANDARD TABLE OF ts WITH EMPTY KEY
     .

  DATA(g1) = VALUE td(
      FOR   wa              " Work area definition
      IN    gt                  " Source table
      ( species = wa-arbgb
        nick    = wa-text
      )                         " Result
    ).
```



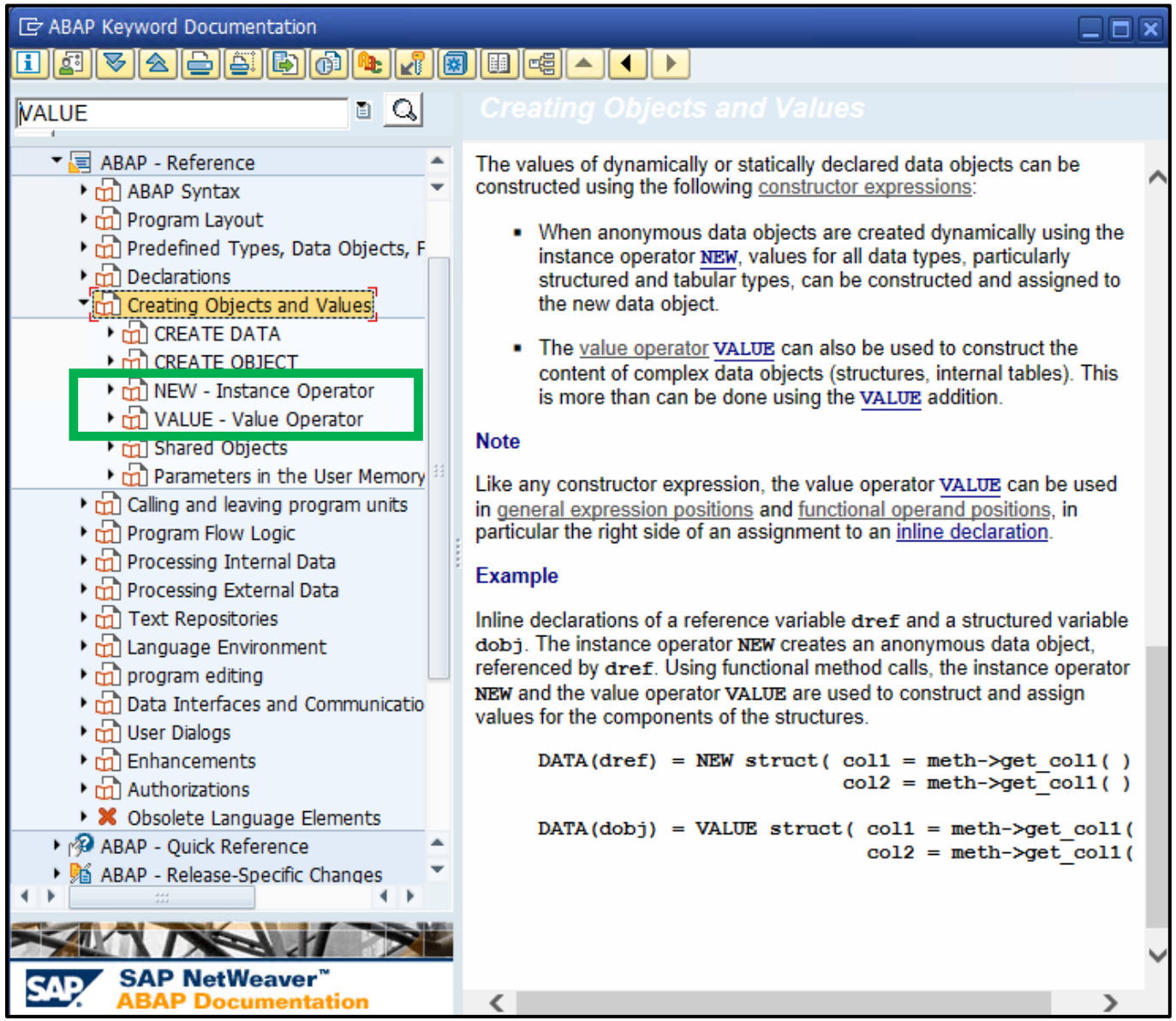
Итератор FOR. Селекция, проекция, вычисления

```
TYPES: tt TYPE STANDARD TABLE OF t100 WITH EMPTY KEY
     , BEGIN OF ts
     ,   no      TYPE p DECIMALS 4
     ,   species TYPE t100-arbgb
     ,   nick    TYPE t100-text
     , END OF ts
     , td TYPE STANDARD TABLE OF ts WITH EMPTY KEY
     .
DATA(g1) = VALUE td(
    FOR   wa                  " Work area definition
    IN    gt                  " Sorce table
    INDEX INTO ind            " Index value definitio
    WHERE ( arbgb CP 'C*' )" Condition
    ( no      = sin( ind ) * 11
      species = wa-arbgb
      nick    = wa-text && ` is a pet of mine`
    )                         " Result
  ).
```



Где еще об этом говорится: Документация

Транзакция **ABAPDOCU**



Семинар D75AW. Дельта-курс: с ABAP Workbench SAP NetWeaver 7.0 на SAP NetWeaver 7.5

Настройка производительности ABAP-программ	
D75AW	3 дн.

- Разработчики
- Консультанты в области разработки

Официальное описание
Дополнительное описание

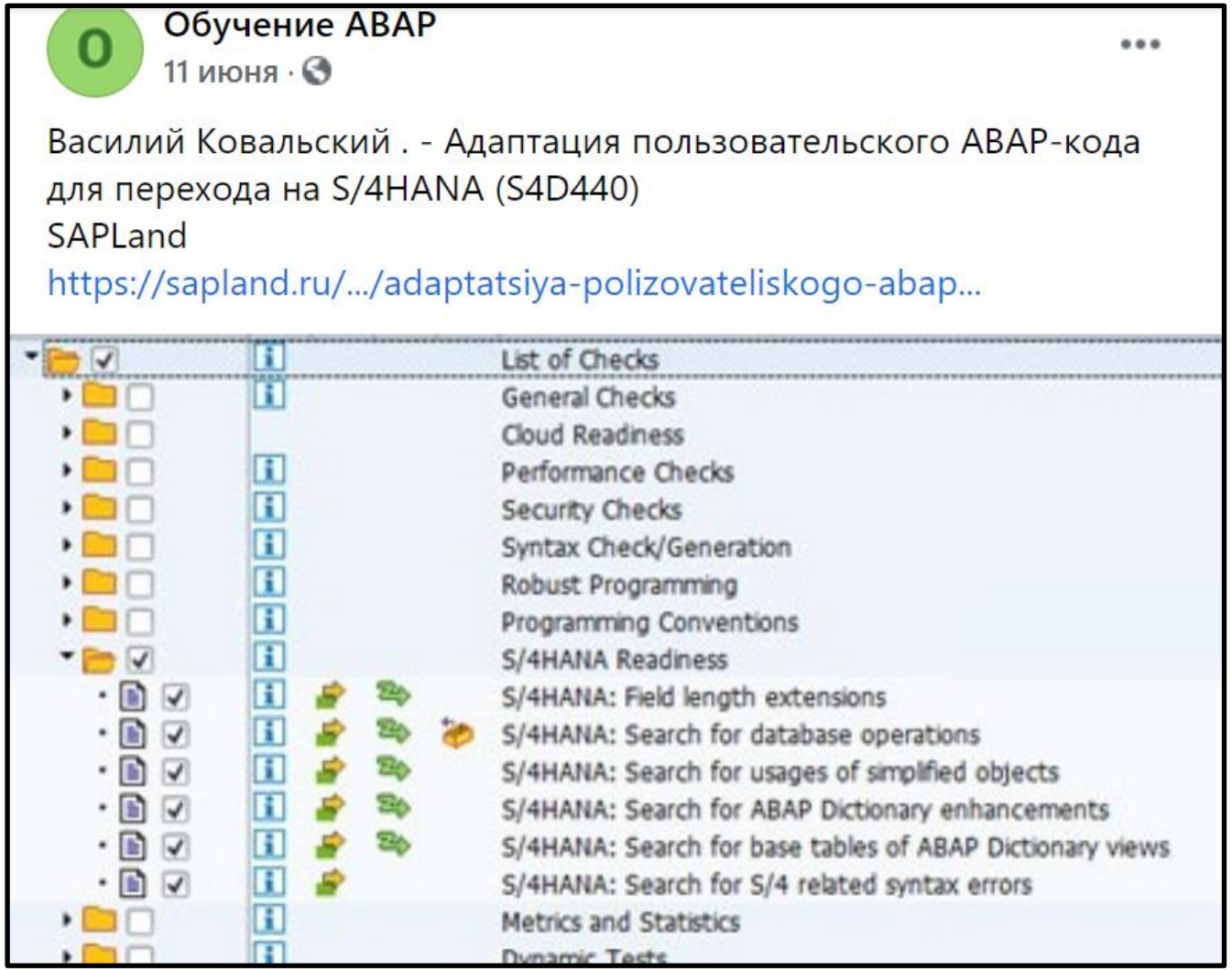
- Введение
- Окружение
- Выражения в ABAP
- Возможности внутренних таблиц
- Изменения в Open SQL
- Core Data Services (CDS)
- Каналы сообщений ABAP
- Новые инструменты анализа

Требования

Обязательно

Опыт процедурно- и объектно-ориентированной разработки на ABAP в SAP NetWeaver 7.0

https://www.facebook.com/abap.education



Семинар WRGCTS. Git-enabled Change and Transport System + Piper Library. 2-3 декабря 2021

Содержание
Обзор gCTS
Базовая конфигурация gCTS
Конфигурация репозитория + упражнение
Разработка с gCTS + упражнение
Обзор инструмента непрерывной интеграции Jenkins
Обзор библиотеки Piper
Конфигурация конвейера + упражнение
gCTS и Change Request Management + упражнение

https://training.sap.com/course/wrgcts--ru-ru

ABAP для консультантов. Основы
Функциональная спецификация на разработку SAP-программ для новичков
Read ABAP: чтение ABAP-кода
ABAP. ООП простыми словами
Использование объектно-ориентированного программирования в ABAP
Экспресс-курс по основам быстрого формирования спецификаций на разработку
Web ABAP: OData-сервисы и их использование в UI5-приложениях
Чистый код. Как писать код, который приятно читать и легко поддерживать
Современная отладка в ABAP: через тернии к звёздам
Unit-тесты на ABAP
Multi Page-разработка и нативный биндинг данных
Шаблоны проектирования в SAP
Разработка HCM в ODATA
Оптимизация ABAP-приложений

https://sapland.ru/click.php?id=260

Приобретайте знания вместе с SAP Training and Adoption

Страница SAP Training and Adoption
https://www.sap.com/cis/training-certification.html

Актуальные новости
https://training.sap.com/content/CIS-RUNews

Электронный магазин семинаров
www.training.sap.com

@SAPtvCIS

@SAPEducation

Москва, **Учебный центр SAP**
+7 (495) 797 27 20
education.russia@sap.com