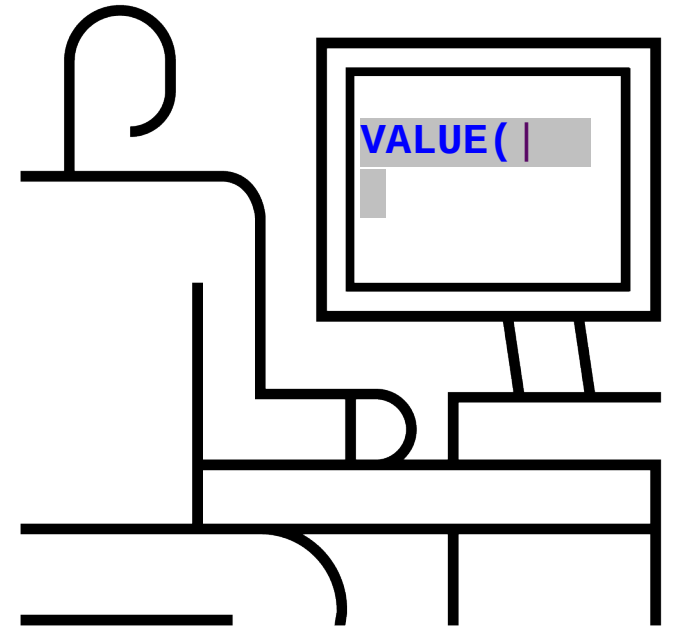


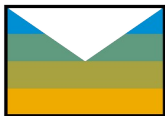
Обучение АВАР. Передача параметров по ссылке и по значению.

Василий Ковальский, SAP CIS
16 декабря 2021 12:00 – 13:00

PUBLIC



Василий Ковальский,
инструктор АВАР с 1998 года



vassili.kovalski@sap.com



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

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



Серия вебинаров Обучение АВАР. Декабрь 2021



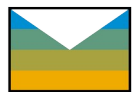
Расширение средств поиска (Search Help)

14 декабря 2021



Передача параметров по ссылке и по значению

16 декабря 2021



Интересны другие темы? Пишите, подумаем

Для кого этот вебинар



- для разработчиков
- для специалистов службы поддержки
- для руководителей подразделений разработки и поддержки
- для консультантов, сталкивающихся с АВАР кодом или общающихся с разработчиками, в особенности для тех, кто пишут спецификации на разработку
- для всех интересующихся АВАР

О чем пойдет речь

Параметры подпрограмм, методов и функциональных модулей могут передаваться по ссылке и по значению. Различные способы передачи параметров по-разному сказываются на возможности работать с ними, на производительность и потребляемую память.

Мы обсудим:

- Способы передачи параметров;
- Влияние способа передачи параметров на возможность работы с ними;
- Влияние способа передачи параметров на производительность и потребляемую память;
- Где еще об этом узнать.



Предположительная продолжительность ~ **1 час**

Способы передачи параметров



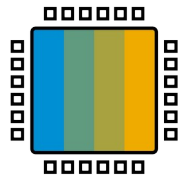
Существует два способа передачи параметров, методов и функциональных модулей:

- по ссылке
- по значению

Умалчиваемым является передача по ссылке. Передачу по значению нужно указать явно фразой **VALUE (имя_параметра)**



Целостность данных



Потребление памяти



Быстродействие

Целостность данных



Подпрограмма

```
FORM form_ref_val DEFINITION
  USING      VALUE(uv) TYPE tv_tst
             ur  TYPE tv_tst
  CHANGING VALUE(cv) TYPE tv_tst
             cr  TYPE tv_tst

  RAISING   zcx_qk.

FORM form_ref_val IMPLEMENTATION.
  uv = to_upper( uv ).
  ur = to_upper( ur ).
  cv = to_upper( cv ).
  cr = to_upper( cr ).
  WRITE: / 'inside form :', gv1, gv2, gv3, gv4.
  IF uv CP '@'.
    RAISE EXCEPTION TYPE zcx_qk
      EXPORTING txt = '@ inside'.
  ENDIF.
ENDFORM.
```



Вызов подпрограммы

```
gv1 = 'aaaa'. gv2 = 'bbbb'. gv3 = 'dddd'.
gv4 = 'eeee'. gv5 = 'ffff'. gv6 = 'gggg'.
WRITE: / 'before form :', gv1, gv2, gv3, gv4.
TRY.
  PERFORM form_ref_val USING gv1 gv2 CHANGING gv3 gv4.
  CATCH zcx_qk INTO gx.
  WRITE: / gx->get_text( ) COLOR COL_NEGATIVE INVERSE.
  ENDTRY.
WRITE: / 'after form :', gv1, gv2, gv3, gv4.
SKIP.
WRITE: / 'before form :', gv1, gv2, gv3, gv4.
TRY.
  gv1 = '@aaa'.
  PERFORM form_ref_val USING gv1 gv2 CHANGING gv3 gv4.
  CATCH zcx_qk INTO gx.
  WRITE: / gx->get_text( ) COLOR COL_NEGATIVE INVERSE.
  ENDTRY.
WRITE: / 'after form :', gv1, gv2, gv3, gv4.
```



Результат подпрограммы без исключений и при исключениях

```
      : uv  ur  cv  cr
before form : aaaa bbbb dddd eeee
inside form : aaaa BBBB dddd EEEE
after form  : aaaa BBBB DDDD EEEE

before form : @aaa bbbb dddd eeee
inside form : @aaa BBBB dddd EEEE
ZCX_QK class exception: @ inside
after form  : @aaa BBBB dddd EEEE
```

По ссылке.

Изменения в источнике
произошли сразу,
независимо от успеха

По значению.

Изменения перенесены в
источник только при
успехе. При исключении
не перенесены



Метод

```
CLASS lcl DEFINITION.  
  PUBLIC SECTION.  
  CLASS-METHODS meth_ref_val  
    IMPORTING VALUE(iv) TYPE char4  
              ir   TYPE char4  
    EXPORTING VALUE(ev) TYPE char4  
              er   TYPE char4  
    CHANGING  VALUE(cv) TYPE char4  
              cr   TYPE char4  
    RAISING   zcx_qk.  
ENDCLASS.
```

```
CLASS lcl IMPLEMENTATION.  
  METHOD meth_ref_val.  
    iv = to_upper( iv ).  
    " ir = to_upper( ir ). " The field  
    "IR" cannot be modified.  
    ev = to_upper( ev ). " Export  
    value has not be imported!  
    er = to_upper( er ).  
    cv = to_upper( cv ).  
    cr = to_upper( cr ).  
    WRITE: / 'inside metod of a local  
class :', gv1, gv2, gv3, gv4, gv5,  
gv6.  
    IF iv CP '*@*'.  
      RAISE EXCEPTION TYPE zcx_qk  
    EXPORTING txt = '@ inside'.  
    ENDIF.  
  ENDMETHOD.  
ENDCLASS.
```



Вызов метода

```
gv1 = 'aaaa'. gv2 = 'bbbb'. gv3 = 'dddd'.
gv4 = 'eeee'. gv5 = 'ffff'. gv6 = 'gggg'.
WRITE: /31 ': iv ir ev er cv cr'.
WRITE: / 'before metod of a local class :', gv1, gv2, gv3, gv4, gv5, gv6.
TRY.
    lcl=>meth_ref_val( EXPORTING iv = gv1 ir = gv2
                      IMPORTING ev = gv3 er = gv4
                      CHANGING  cv = gv5 cr = gv6 ).

CATCH zcx_qk INTO gx.
    WRITE: / gx->get_text( ) COLOR COL_NEGATIVE INVERSE.
ENDTRY.
WRITE: / 'after method of a local class :', gv1, gv2, gv3, gv4, gv5, gv6.
SKIP.
gv1 = 'a@aa'. gv2 = 'bbbb'. gv3 = 'dddd'.
gv4 = 'eeee'. gv5 = 'ffff'. gv6 = 'gggg'.
TRY.
    lcl=>meth_ref_val( EXPORTING iv = gv1 ir = gv2
                      IMPORTING ev = gv3 er = gv4
                      CHANGING  cv = gv5 cr = gv6 ).

CATCH zcx_qk INTO gx.
    WRITE: / gx->get_text( ) COLOR COL_NEGATIVE INVERSE.
ENDTRY.
WRITE: / 'after method of a local class :', gv1, gv2, gv3, gv4, gv5, gv6.
```



Результат подпрограммы без исключений и при исключениях

```
          : iv  ir  ev  er  cv  cr
before method of a globl class : aaaa bbbb dddd eeee ffff gggg
Calling programm variables are unknown inside global class method
after method of a globl class  : aaaa bbbb      EEEE FFFF GGGG

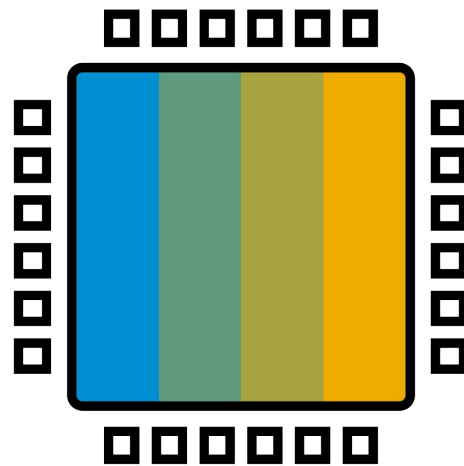
Calling programm variables are unknown inside global class method
ZCX_QK class exception: @ inside
after method of a local class  : aa@a bbbb dddd EEEE ffff GGGG
```

По ссылке.
Изменения в источнике
произошли независимо
от успеха

По значению.
Изменения перенесены в
источник только при
успехе. При исключении
не перенесены



Потребление памяти

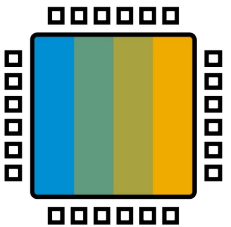


Подпрограммы и класс для тестирования выделения памяти

```
CLASS lcl DEFINITION.  
PUBLIC SECTION.  
CLASS-METHODS meth_ref CHANGING ct LIKE gt.  
CLASS-METHODS meth_val CHANGING VALUE(ct) LIKE gt.  
ENDCLASS.  
CLASS lcl IMPLEMENTATION.  
METHOD meth_ref.  
    DELETE ct INDEX 1.  
ENDMETHOD.  
METHOD meth_val.  
    DELETE ct INDEX 1.  
ENDMETHOD.  
ENDCLASS.
```

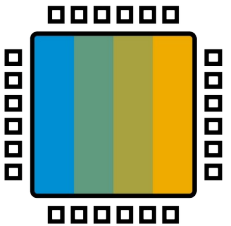
```
FORM form_ref CHANGING ct LIKE gt .  
    DELETE ct INDEX 1.  
ENDFORM.
```

```
FORM form_val CHANGING VALUE(ct) LIKE gt .  
    DELETE ct INDEX 1.  
ENDFORM.
```



Вызовы подпрограмм, методов.

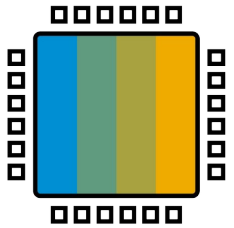
```
START-OF-SELECTION.  
  DO 1000 TIMES.  
    APPEND INITIAL LINE TO gt.  
  ENDDO.  
  
BREAK-POINT. " Examine memory allocation  
PERFORM form_ref CHANGING gt.  
PERFORM form_val CHANGING gt.  
lcl=>meth_ref( CHANGING ct = gt ).  
lcl=>meth_val( CHANGING ct = gt ).  
  
WRITE: / 'BOT TAK.'
```



По ссылке. Память выделяется экономно

```
1 * &-----
2 * & Report ZQK_W0902_03
3 * &-----
4 * &
5 * &-----
6 REPORT zqk_w0902_03. " 03. Parameter Pa
7 DATA: gt TYPE STANDARD TABLE OF char102
8
9 FORM form_ref CHANGING ct LIKE gt .
10   DELETE ct INDEX 1.
11 ENDFORM.
12 FORM form_val CHANGING VALUE(ct) LIKE g
13   DELETE ct INDEX 1.
14 ENDFORM.
15
```

	Rank	Bound (Alloca...	Occupied Bo...
Memory Objects			
{T:14}[1000x1(2048)]Standard Table	1	2056456	100%
\PROGRAM=ZQK_W0902_03\DATA=GT			
{T:2}[1x15(448)]Standard Table	2	2360	23%
{O:3*CL_SRAL_FACTORY_DYNP}	3	0	0%
{O:4*CL_SRAL_SWITCH_CLIENT}	4	0	0%
{O:5*CL_SRAL_DB_PARAMS}	5	0	0%



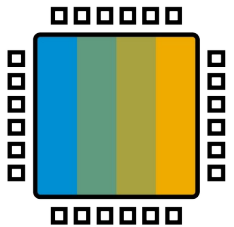
```
1 * &-----
2 * & Report ZQK_W0902_03
3 * &-----
4 * &
5 * &-----
6 REPORT zqk_w0902_03. " 03. Parameter Pa
7 DATA: gt TYPE STANDARD TABLE OF char102
8
9 FORM form_ref CHANGING ct LIKE gt .
10   DELETE ct INDEX 1.
11 ENDFORM.
12 FORM form_val CHANGING VALUE(ct) LIKE g
13   DELETE ct INDEX 1.
14 ENDFORM.
15
```

	Rank	Bound (Alloca...	Occupied Bo...
Memory Objects			
{T:14}[1000x1(2048)]Standard Table	1	2060512	99%
\PROGRAM=ZQK_W0902_03\DATA=GT			
{T:2}[1x15(448)]Standard Table	2	2360	23%
{O:3*CL_SRAL_FACTORY_DYNP}	3	0	0%
{O:4*CL_SRAL_SWITCH_CLIENT}	4	0	0%
{O:5*CL_SRAL_DB_PARAMS}	5	0	0%

По значению. Требуется больше памяти

```
1 *&-----
2 *& Report ZQK_W0902_03
3 *&-----
4 *&
5 *&-----
6 REPORT zqk_w0902_03. " 03. Parameter Pa
7 DATA: gt TYPE STANDARD TABLE OF char102
8
9 FORM form_ref CHANGING ct LIKE gt .
10     DELETE ct INDEX 1.
11 ENDFORM.
12 FORM form_val CHANGING VALUE(ct) LIKE g
13     DELETE ct INDEX 1.
14 ENDFORM.
15
```

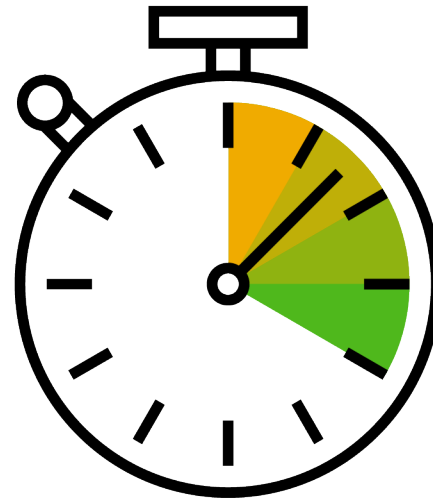
View	Memory Objects	Entries	25	Rank	Bound (Alloca...	Occupied B...
▼	Memory Objects					
▼	{T:14}[1000x1(2048)]StandardTable			1	2060512	99%
•	\PROGRAM=ZQK_W0902_03\FORM=FORM_VAL\DATA=CT					
•	\PROGRAM=ZQK_W0902_03\DATA=GT					
▶	{T:2}[1x15(448)]StandardTable			2	2360	23%
▶	{O:3*CL_SRAL_FACTORY_DYNP}			3	0	0%
▶	{O:4*CL_SRAL_SWITCH_CLIENT}			4	0	0%
▶	{O:5*CL_SRAL_DB_PARAMS}			5	0	0%



```
1 *&-----
2 *& Report ZQK_W0902_03
3 *&-----
4 *&
5 *&-----
6 REPORT zqk_w0902_03. " 03. Parameter Pa
7 DATA: gt TYPE STANDARD TABLE OF char102
8
9 FORM form_ref CHANGING ct LIKE gt .
10     DELETE ct INDEX 1.
11 ENDFORM.
12 FORM form_val CHANGING VALUE(ct) LIKE g
13     DELETE ct INDEX 1.
14 ENDFORM.
15
```

View	Memory Objects	Entries	25	Rank	Bound (Alloca...	Occupied B...
▼	Memory Objects					
▼	{T:14}[1000x1(2048)]StandardTable			1	2060512	99%
•	\PROGRAM=ZQK_W0902_03\DATA=GT					
▼	{T:57}[998x1(2048)]StandardTable			2	2060512	99%
•	\PROGRAM=ZQK_W0902_03\FORM=FORM_VAL\DATA=CT					
▶	{T:2}[1x15(448)]StandardTable			3	2360	23%
▶	{O:3*CL_SRAL_FACTORY_DYNP}			4	0	0%
▶	{O:4*CL_SRAL_SWITCH_CLIENT}			5	0	0%
▶	{O:5*CL_SRAL_DB_PARAMS}			6	0	0%

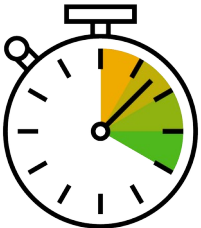
Быстродействие



Будем измерять время только на передачу параметров. Тела пустые

```
FORM form_ref0 CHANGING ct LIKE gt .  
ENDFORM.  
  
FORM form_val0 CHANGING VALUE(ct) LIKE gt .  
ENDFORM.
```

```
CLASS lc1 DEFINITION.  
PUBLIC SECTION.  
  CLASS-METHODS:  
    meth_ref0 CHANGING ct LIKE gt,  
    meth_val0 CHANGING VALUE(ct) LIKE gt.  
ENDCLASS.  
  
CLASS lc1 IMPLEMENTATION.  
  METHOD meth_ref0.  
  ENDMETHOD.  
  
  METHOD meth_val0.  
  ENDMETHOD.  
ENDCLASS.
```



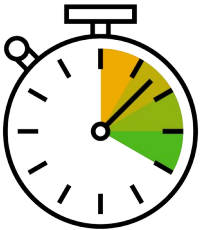
Измерим времена начал и концов

```
DATA
: gt TYPE STANDARD TABLE OF char1024
, a TYPE i, b TYPE i
, t0 TYPE i, t1 TYPE i, t2 TYPE i
, rr TYPE p DECIMALS 3
.
START-OF-SELECTION.
  DO 1000 TIMES.
    APPEND INITIAL LINE TO gt.
  ENDDO.

  GET RUN TIME FIELD a.
  DO 1000 TIMES.
  ENDDO.
  GET RUN TIME FIELD b.
  t0 = b - a.
  WRITE: /31 'ref          val          ratio'.
```

```
DO 1000 TIMES.
  PERFORM form_ref0 CHANGING gt.
ENDDO.
GET RUN TIME FIELD b.
t1 = b - a - t0.
GET RUN TIME FIELD a.
DO 1000 TIMES.
  PERFORM form_val0 CHANGING gt.
ENDDO.
GET RUN TIME FIELD b.
t2 = b - a - t0.
rr = t2 / t1.
WRITE: / 'Forms          :', t1, t2, rr.
```

```
GET RUN TIME FIELD a.
DO 1000 TIMES.
  lcl=>meth_ref0( CHANGING ct = gt ).
ENDDO.
GET RUN TIME FIELD b.
t1 = b - a - t0.
GET RUN TIME FIELD a.
DO 1000 TIMES.
  lcl=>meth_val0( CHANGING ct = gt ).
ENDDO.
GET RUN TIME FIELD b.
t2 = b - a - t0.
rr = t2 / t1.
WRITE: / 'Local class methods :', t1, t2, rr.
```

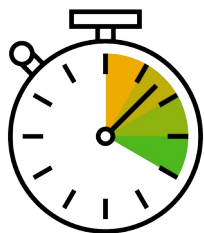


Результаты измерений

	ref	val	ratio
Forms :	188	897	4,771
Local class methods :	153	601	3,928
Global class methods :	437	1.003	2,295
Function modules :	549	1.351	2,461

	ref	val	ratio
Forms :	120	446	3,717
Local class methods :	175	376	2,149
Global class methods :	316	627	1,984
Function modules :	342	1.005	2,939

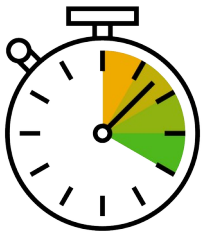
	ref	val	ratio
Forms :	122	482	3,951
Local class methods :	98	369	3,765
Global class methods :	281	568	2,021
Function modules :	348	671	1,928



Маленькие хитрости. Специально для геттеров

```
CLASS lcl DEFINITION.  
PUBLIC SECTION.  
    CLASS-DATA gt TYPE STANDARD TABLE OF char1024 READ-ONLY.  
    CLASS-METHODS get RETURNING VALUE(rt) LIKE gt.  
    CLASS-METHODS class_constructor.  
ENDCLASS.
```

```
CLASS lcl IMPLEMENTATION.  
    METHOD get.  
        rt = gt.  
    ENDMETHOD.  
    METHOD class_constructor.  
        DO 1000 TIMES.  
            APPEND INITIAL LINE TO gt.  
        ENDDO.  
    ENDMETHOD.  
ENDCLASS.
```

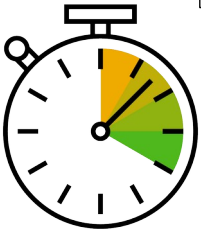


Маленькие хитрости. Специально для геттеров

```
DATA
: gt TYPE STANDARD TABLE OF
char1024
, a TYPE i, b TYPE i
, t0 TYPE i, t1 TYPE i, t2 TYPE i
, rr TYPE p DECIMALS 3
.
START-OF-SELECTION.
  GET RUN TIME FIELD a.
  DO 1000 TIMES.
  ENDDO.
  GET RUN TIME FIELD b.
  t0 = b - a.
  gt = lcl=>gt.
```

```
GET RUN TIME FIELD a.
DO 1000 TIMES.
  gt = lcl=>gt.
ENDDO.
GET RUN TIME FIELD b.
t1 = b - a - t0.
```

```
GET RUN TIME FIELD a.
DO 1000 TIMES.
  gt = lcl=>get( ).
ENDDO.
GET RUN TIME FIELD b.
t2 = b - a - t0.
```

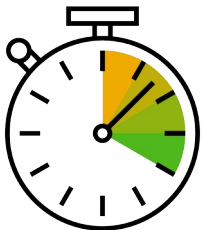


Результаты измерений

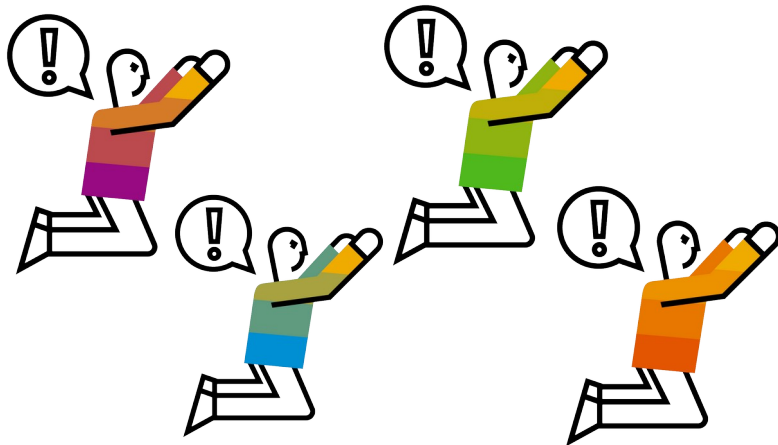
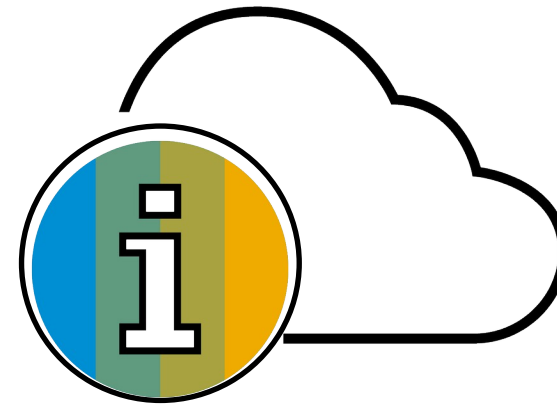
```
Access via read-onky attribute :      50
Access via getter method      :      319
Ratio                          6,380
```

```
Access via read-onky attribute :      82
Access via getter method      :      336
Ratio                          4,098
```

```
Access via read-onky attribute :      96
Access via getter method      :      514
Ratio                          5,354
```



Где еще можно узнать?



Семинар **BC400**. Введение в инструментальные средства АВАР

Введение в
инструментальные
средства АВАР
BC400 5 дн.

 SAP
Live
access



- Разработчики
- Разработчики-консультанты
- Специалисты поддержки
- Менеджеры проектов и программ

[Официальное описание](#)
[Дополнительное описание](#)



- Исполнение АВАР-программ
- Введение в инструментальные средства АВАР
- Основные элементы языка АВАР
- Модуляризация
- Сложные объекты данных
- Моделирование и получение данных
- Классический АВАР-отчет
- Экраны • SAP List Viewer • Web Dynpro АВАР
- Инструменты анализа программ
- Инструментарий разработки «АВАР Development Tools» для SAP NetWeaver
- Адаптация стандартного программного обеспечения SAP



Обязательно

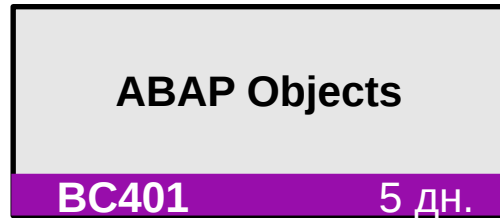
Требования

BC100 или опыт программирования на любом языке

Желательно

SAPTEC

Семинар **BC401**. Объектно-ориентированное программирование на АВАР



- Разработчики
- Разработчики-консультанты

[Официальное описание](#)
[Дополнительное описание](#)



- Введение в объектно-ориентированное программирование
- Основы синтаксиса
- Наследование и приведение типов
- Интерфейсы и приведение типов
- Объектно-ориентированный события
- Объекты репозитария
- Глобальные классы в ABAP Development Tools
- Исключения, основанные на классах
- Модульное тестирование
- Шаблоны объектно-ориентированной разработки
- Службы типов в рантайме
- Создание сложного объектно-ориентированного приложения



Требования
BC400

Семинар **BC402**. Углубленное ABAP программирование

Углубленное
ABAP
программирование
BC402 5 дн.



SAP
Live
access



- Разработчики
- Разработчики-консультанты
- Специалисты поддержки
- Менеджеры проектов и программ

[Официальное описание](#)
[Дополнительное описание](#)



Обязательно

Требования

[BC400](#) и [BC401](#)

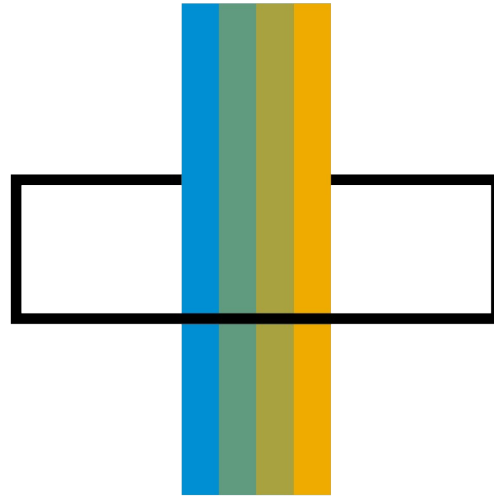
Желательно

Практический опыт объектно-ориентированного программирования на ABAP и [BC430](#)



- Эволюция языка ABAP
- Вызов программ, устройство и управление памятью
- Команды(Statements), функции и выражения для простых данных
- Внутренние таблицы
- Динамическое программирование
- ABAP Open SQL
- Анализ и тестирование

Дополнительно



Семинар **WRGCTS**. Непрерывная интеграция для ABAP с gCTS и Piper.



Семинар
Андрея
Шарапова

Расписание 2022

- 18.04-19.04.2022
- 10.10-11.10.2022

Содержание

- Обзор gCTS
- Базовая конфигурация gCTS
- Конфигурация репозитория
- Разработка с gCTS
- Обзор инструмента непрерывной интеграции Jenkins
- Обзор библиотеки Piper
- Конфигурация конвейера
- gCTS и Change Request Management
- ABAP Git

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

Ранние пташки 2022 -10%



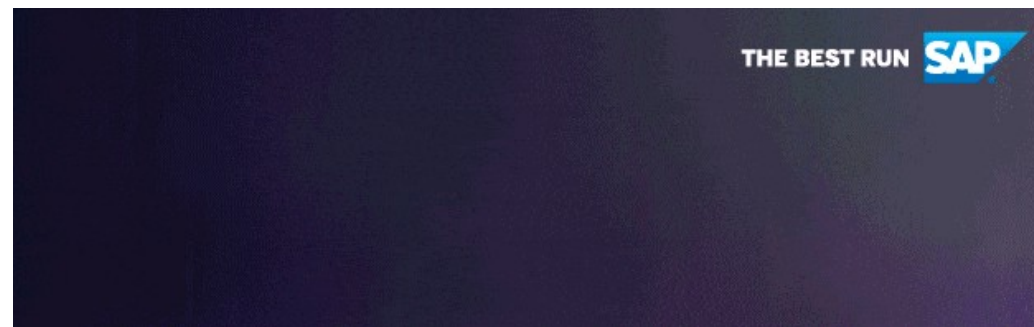
Акция "**Ранние пташки 2022**": с 11 ноября до 30 декабря 2021 регистрируйтесь онлайн на семинары, запланированные с 1 января по 1 апреля 2022, и получите скидку 10%* (расписание 2022).



Используйте промокод **EARLYBIRD2022** в соответствующем поле или в поле "Комментарии". [Подробнее об онлайн-регистрации >](#)



**Скидка не распространяется на опцию «Бежим втроем», референциальную и партнерскую программы, не суммируется с другими скидками по договору, а также другими акциями и специальными предложениями.*



Бежим втроем

Что делать, если :

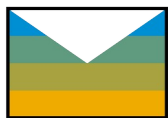
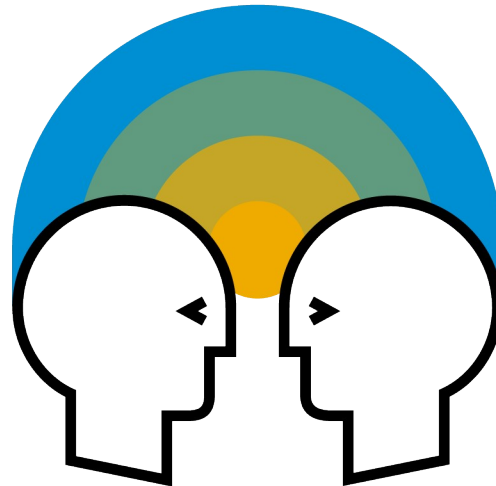
- ни одна из предложенных дат не вписывается в ваш график?
- необходимо развить экспертизу по специфичной и редкой теме,
- незапланированной в расписании?
- предстоит большой проект, важно успеть пройти обучение до его старта, а в расписании нет подходящих дат или нужный семинар доступен лишь по запросу?



- Выберите тему или семинар из [расписания](#) / [глобального портфолио SAP Education](#)
- Соберите команду минимум из трех участников
- Определитесь с удобной датой
- Направьте запрос с желаемым семинаром или темой из семинара, датами и списком участников на education.russia@sap.com

Подробности здесь: <https://training.sap.com/content/ru3torun>

Контакты



vassili.kovalski@sap.com



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

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



Обучение ABAP
11 июня · 🌐

Василий Ковальский . - Адаптация пользовательского ABAP-кода для перехода на S/4HANA (S4D440)
SAPLand
<https://sapland.ru/.../adaptatsiya-polizovateliskogo-abap...>

List of Checks	
General Checks	
Cloud Readiness	
Performance Checks	
Security Checks	
Syntax Check/Generation	
Robust Programming	
Programming Conventions	
S/4HANA Readiness	
• S/4HANA: Field length extensions	👉👉
• S/4HANA: Search for database operations	👉👉👉
• S/4HANA: Search for usages of simplified objects	👉👉👉
• S/4HANA: Search for ABAP Dictionary enhancements	👉👉👉
• S/4HANA: Search for base tables of ABAP Dictionary views	👉👉👉
• S/4HANA: Search for S/4 related syntax errors	👉👉
Metrics and Statistics	
Dynamic Tests	

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](https://www.youtube.com/@SAPtvCIS)



[@SAPEducation](https://www.facebook.com/SAPEducation)

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