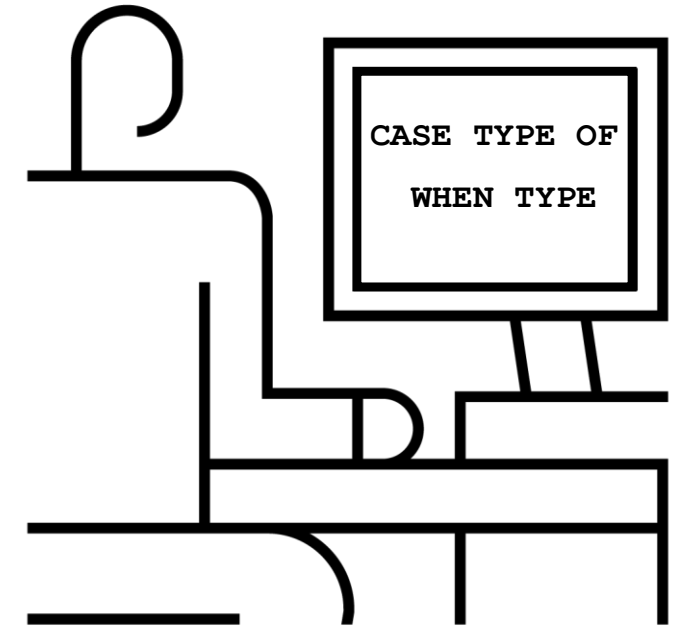


Обучение АВАР. Определение класса по ссылке

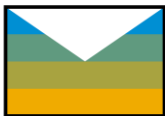
Василий Ковальский, SAP CIS
26 февраля 2020 11:00 – 12:00

PUBLIC



Презентация доступна по ссылке
<https://cloud.mail.ru/public/iH86/5NkkNVGP7>

Василий Ковальский,
инструктор АВАР с 1998 года



vassili.kovalski@sap.com



Серия вебинаров Обучение АВАР. Февраль 2020



Определение класса по ссылке

26 февраля 2020



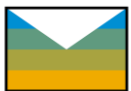
Коротко о блокировках

27 февраля 2019



Диалоговые экраны. Навигация

28 февраля 2019



Интересны другие темы? Пишите, подумаем

Для кого этот вебинар



- для АВАР разработчиков
- для руководителей подразделений разработки
- для специалистов службы поддержки,
- для консультантов, сталкивающихся с АВАР кодом или общающихся с разработчиками

О чем пойдет речь

В переменную типа ссылки на предка можно положить ссылку на любого потомка данного класса. Это называется приведением типа к предку.

Часто по ссылке на объект бывает нужно установить, к какому конкретно классу он относится. Мы рассмотрим:



- приведение типа к потомку, обработка исключений
- новый логический оператор IS INSTANCE OF
- новые ABAP-команды CASE TYPE OF и WHEN TYPE
- Run Time Type Identification



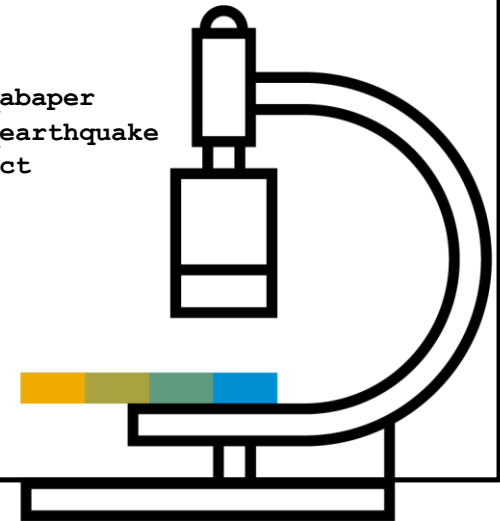
Предположительная продолжительность ~ **1 час**

Постановка задачи.

```
CLASS lcl_earthquake DEFINITION.  
  PUBLIC SECTION.  
    DATA: m_when TYPE d  
          , m_where TYPE c LENGTH 20 VALUE 'Atlantida'  
          , m_magn TYPE p DECIMALS 2  
          .  
    METHODS  
      : display  
      , happen IMPORTING i_where LIKE m_where OPTIONAL  
                i_magn LIKE m_magn  
          .  
ENDCLASS.
```

```
CLASS lcl_abaper DEFINITION.  
  PUBLIC SECTION.  
    DATA: m_name TYPE c LENGTH 20 VALUE 'Someone'  
          , m_cert TYPE abap_bool  
          .  
    METHODS: show  
              , certify.  
ENDCLASS.
```

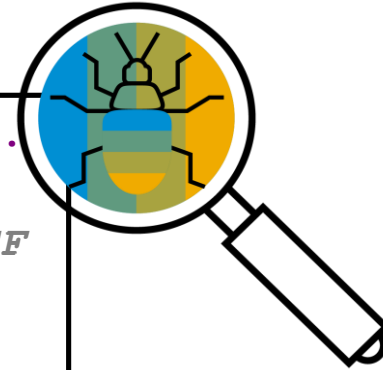
```
CLASS lcl_test DEFINITION.  
  PUBLIC SECTION.  
    CLASS-METHODS test IMPORTING i_ref TYPE REF TO object.  
ENDCLASS.  
CLASS lcl_test IMPLEMENTATION.  
  METHOD test.  
    BREAK-POINT. " Check I_REF  
  ENDMETHOD.  
ENDCLASS.  
  
DATA: go_abaper TYPE REF TO lcl_abaper  
      , go_quake TYPE REF TO lcl_earthquake  
      , go_obj TYPE REF TO object  
      .  
  
START-OF-SELECTION.  
  go_abaper = NEW #( ).  
  go_quake = NEW #( ).  
  lcl_test=>test( go_quake ).  
  lcl_test=>test( go_abaper ).  
  WRITE: / 'BOT Tak!'.
```



В переменной могут оказаться ссылки на объекты разных классов. Нужно определить, к какому относится объект

Отладка

```
CLASS lcl_test IMPLEMENTATION.  
  METHOD test.  
    BREAK-POINT. " Check I_REF  
  ENDMETHOD.  
ENDCLASS.
```



The screenshot shows the SAP IDE interface. On the left, the class implementation code is displayed with line numbers 46 to 54. Line 51, containing the `BREAK-POINT. " Check I_REF` statement, is highlighted. On the right, the 'Variables 1' window is open, showing a table with the following data:

S...	Variable	V...	Val.
	I_REF		{0:11*\PROGRAM=ZQK_W0304_0\CLASS=LCL_EARTHQUAKE}

Приведение к потомку

```
METHOD test.  
  DATA : lo_quake TYPE REF TO lcl_earthquake  
         , lo_abaper TYPE REF TO lcl_abaper.  
  lo_abaper ?= i_ref.  
ENDMETHOD.
```




Category	ABAP programming error
Runtime Errors	MOVE_CAST_ERROR
Except.	CX_SY_MOVE_CAST_ERROR
ABAP Program	ZQK_W0304_2

Приведение к потомку. Перехват исключения по-старому.

```
METHOD test.  
  DATA: lo_quake TYPE REF TO lcl_earthquake  
        , lo_abaper TYPE REF TO lcl_abaper.  
  CATCH SYSTEM-EXCEPTIONS move_cast_error = 1.  
    lo_abaper ?= i_ref.  
    lo_abaper->show( ).  
    lo_abaper->certify( ).  
    lo_abaper->show( ).  
  ENDCATCH.  
  CATCH SYSTEM-EXCEPTIONS move_cast_error = 1.  
    lo_quake ?= i_ref.  
    lo_quake->display( ).  
    lo_quake->happen( '6.7' ).  
    lo_quake->display( ).  
  ENDCATCH.  
ENDMETHOD.
```



```
CATCH SYSTEM-EXCEPTIONS   
Obsolete Syntax  
CATCH SYSTEM-EXCEPTIONS [exc1 = n1 exc2 = n2...]  
                        [OTHERS = n_others].  
  [statement_block]  
ENDCATCH.
```

Работает.

Устарело.



Приведение к потомку. Перехват исключения по-новому

```
METHOD test.  
  DATA: lo_quake TYPE REF TO lcl_earthquake  
        , lo_abaper TYPE REF TO lcl_abaper  
        , lx_csterr TYPE REF TO cx_sy_move_cast_error  
        .  
  TRY.  
    lo_abaper ?= i_ref.  
    lo_abaper->show( ).  
    lo_abaper->certify( ).  
    lo_abaper->show( ).  
    CATCH cx_sy_move_cast_error INTO lx_csterr.  
  ENDTRY.  
  TRY.  
    lo_quake ?= i_ref.  
    lo_quake->display( ).  
    lo_quake->happen( '6.7' ).  
    lo_quake->display( ).  
    CATCH cx_sy_move_cast_error INTO lx_csterr.  
  ENDTRY.  
ENDMETHOD.
```

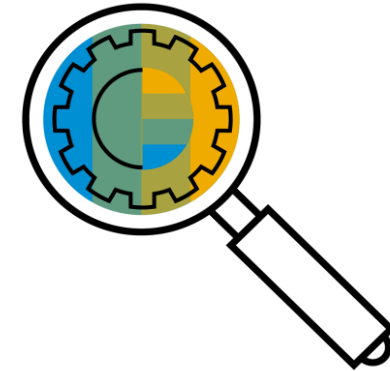


Можно получить ссылку на объект класса исключительной ситуации. Эту ссылку можно использовать при возбуждении других исключений, можно передавать сквозь операторные скобки.

Ссылка на исключение нужна не всегда

Логический оператор IS INSTANCE OF

```
METHOD test.  
  DATA: lo_quake TYPE REF TO lcl_earthquake  
        , lo_abaper TYPE REF TO lcl_abaper  
        .  
  IF i_ref IS INSTANCE OF lcl_abaper.  
    lo_abaper ?= i_ref.  
    lo_abaper->show( ).  
    lo_abaper->certify( ).  
    lo_abaper->show( ).  
  ELSEIF i_ref IS INSTANCE OF lcl_earthquake.  
    lo_quake ?= i_ref.  
    lo_quake->display( ).  
    lo_quake->happen( '6.7' ).  
    lo_quake->display( ).  
  
  ELSE . . .  
  ENDIF.  
ENDMETHOD.
```



Не предполагается
возникновение исключений

Команды CASE TYPE OF И WHEN TYPE

```
METHOD test.  
  DATA: lo_quake TYPE REF TO lcl_earthquake  
        , lo_abaper TYPE REF TO lcl_abaper  
        .  
  CASE TYPE OF i_ref.  
    WHEN TYPE lcl_abaper.  
      lo_abaper ?= i_ref.  
      lo_abaper->show( ).  
      lo_abaper->certify( ).  
      lo_abaper->show( ).  
    WHEN TYPE lcl_earthquake INTO lo_quake.  
      lo_quake->display( ).  
      lo_quake->happen( '6.7' ).  
      lo_quake->display( ).  
  ENDCASE.  
ENDMETHOD.
```

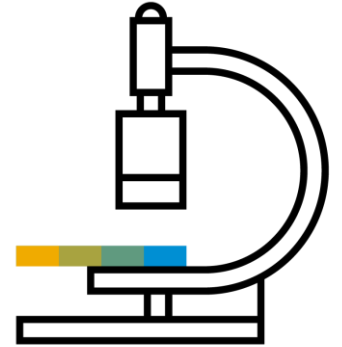


Не предполагается
возникновение исключений

Возможно присвоение с
приведением типа
(вместо **?=**)

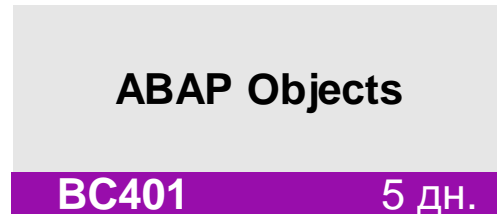
RTTI

```
METHOD test.  
  DATA: lo_quake  TYPE REF TO lcl_earthquake  
          , lo_abaper TYPE REF TO lcl_abaper  
          , lo_clsdesc TYPE REF TO cl_abap_classdescr  
          .  
  lo_clsdesc ?= cl_abap_classdescr=>describe_by_object_ref( i_ref ).  
  CASE lo_clsdesc->get_relative_name( ).  
    WHEN 'LCL_ABAPER'.  
      lo_abaper ?= i_ref.  
      lo_abaper->show( ).  
      lo_abaper->certify( ).  
      lo_abaper->show( ).  
    WHEN 'LCL_EARTHQUAKE'.  
      lo_quake ?= i_ref.  
      lo_quake->display( ).  
      lo_quake->happen( '6.7' ).  
      lo_quake->display( ).  
    WHEN OTHERS.  
  ENDCASE.  
ENDMETHOD
```



Можно получить полное описание класса, включая имя.

Где об этом говорится: **BC401**. Объектно-ориентированное программирование на ABAP



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



Обязательно **BC400**
Требования



- Разработчики
- Разработчики-консультанты



- 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



Live access

Ссылки кликабельны!

Где об этом говорится: **BC402**. Углубленное ABAP программирование

Углубленное
ABAP
программирование

BC402

5 дн.



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

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



- 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

Требования



Обязательно

Желательно

BC400 и **BC401**

Практический опыт объектно-ориентированного программирования на ABAP и **BC430**

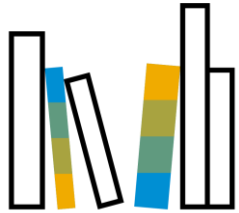


Live access

Ссылки кликабельны

Learning Hub. Учебная комната «АВАР в СНГ»

Бесплатный сервис в рамках Learning Hub.



Коллекция ссылок на учебные материалы по теме в нужном порядке.
В настоящее время в комнате две программы обучения на сертификации:

C_TAW12_740

C_TAW12_750



Кроме стандартных учебных материалов есть и будут появляться дополнительные:

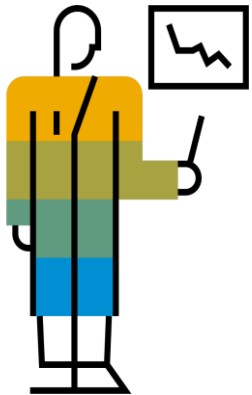
- описание функциональных возможностей
- описание синтаксических конструкций
- сообщения о вебинарах, мастер-классах
- материалы вебинаров



Интерактивный ресурс. Можно задавать вопросы и получать ответы на вопросы.
Общение в этой комнате на русском языке

Ссылки кликабельны

Мастер-классы [SAPLand](#). Май 2020



- 18 мая 2020 [BRF+Business Rule Framework. Примеры и способы использования](#)
- 18 мая 2020 [Read ABAP: Чтение ABAP кода \(отладка и мониторинг кода\)](#)
- 19 мая 2020 [Call ABAP: дизайн SAP приложений](#)
- 20 мая 2020 [Web ABAP: работа с OData-сервисами и прототипирование UI5-приложений](#)
- 21 мая 2020 [Использование объектно-ориентированного программирования в ABAP](#)
- 22 мая 2020 [Знакомство с SAP UI5: применение и средства разработки](#)
- 22 мая 2020 [Особенности подготовки спецификаций на разработку пользовательских ABAP-программ](#)



Ссылки кликабельны

Приобретайте знания вместе с SAP Training & Enablement



<https://www.sap.com/cis/training-certification.html>



<https://training.sap.com/content/CIS-RUNews>



[JAM](#)



www.training.sap.com



[@SAPtvCIS](#)



[@SAPEducation](#)

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