



Modern ABAP

Webinar, 29.3.2019, 10:00 Uhr



Domini Bigl

Senior Consultant at Cadaxo

eMail: dominik.bigl@cadaxo.com

Twitter: [@DomiBiglSAP](https://twitter.com/DomiBiglSAP)

Beratungsschwerpunkte

- Konzeption & Management von Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- ABAP Units
- SAP Fiori, SAP UI5



Johann Föbleitner

Senior Consultant at Cadaxo

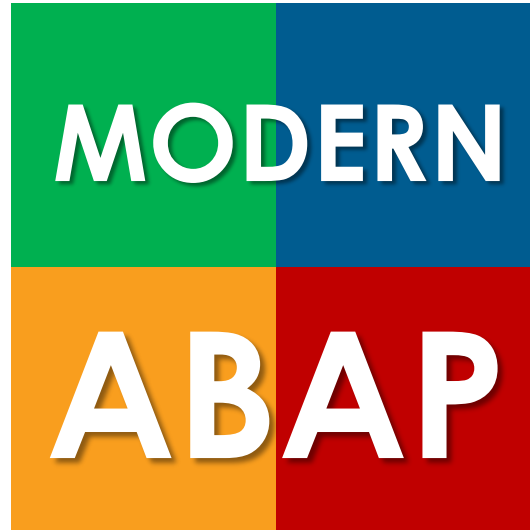
eMail: johann.foessleitner@cadaxo.com

Twitter: [@foessleitnerj](https://twitter.com/foessleitnerj)

Beratungsschwerpunkte

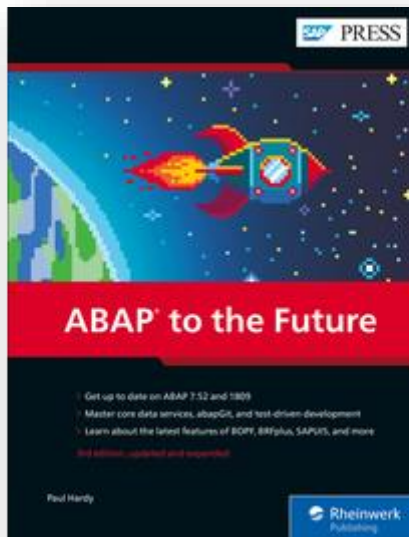
- Konzeption & Management von Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- Clean Code
- S/4HANA Custom Development

- **ABAP Sprachversionen**
- **SAP Cloud Platform ABAP Environment**
- **ABAP Development Tools**
- **ABAP Sprachelemente und SQL Expressions**
- **OOP**
- **Obsolete Sprachelemente**
- **abapGit**
- **CDS Views**
- **ABAP RESTful Programming Model**
- **Code Checks in SAP**
- **Clean Code**
- **Refactoring**
- **ABAP Unit / Testseams**
- **Entwurfsmuster**



- **Warum Modern ABAP?**
 - Performance und Produktivität erhöhen
 - Stabilere und wartbarere Entwicklungen
 - Kürzere Entwicklungszeiten / Fehlerbehebungen
 - Reduzierter & vereinfachter Code





ISBN 978-1-4932-1762-5

ABAP Sprachversionen

- **ABAP Sprachversionen**
 - **X** Standard ABAP
 - **2** ABAP for Key Users
 - **5** ABAP for SAP Cloud Platform (ab 7.53)

General

ABAP Language Version: Standard ABAP (Unicode)

Specific

API State

Fix Point Arithmetic: ☒

Shared Memory Enabled: ☐

Category: General Object Type

Status:

Attribute

Typ: 1 Ausführbares Programm

Status:

Berechtigungsgruppe:

Anwendung:

Name der LDB:

Selektionsbild:

ABAP-Sprachversion: X Standard-ABAP (Unicode)

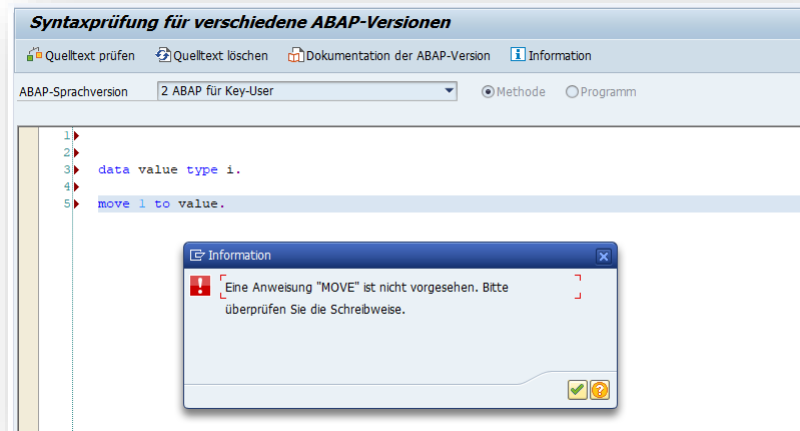
☒ Festpunktarithmetik

- **Freigegebene Objekte**

- CDS Views
- Dictionary Objekte
- Enhancements
- Messages Klassen
- Source Code Library

ABAP Befehle						SY
APPEND	CONCATENATE	ENDCASE	FIND	RAISE	SPLIT	SY-DBCNT
ASSIGN	CONDENSE	ENDDO	GET	READ	TRANSLATE	SY-FDPOS
CASE	CONSTANTS	ENDIF	IF	REPLACE	TRY	SY-INDEX
CATCH	CONTINUE	ENDLOOP	IMPORT	RESUME	TYPES	SY-SUBRC
CHECK	CONVERT	ENDSELECT	INSERT	RETRY	UNASSIGN	SY-TABIX
CLEANUP	DATA	ENDTRY	LOOP	RETURN	WHEN	
CLEAR	DELETE	ENDWHILE	MESSAGE	SELECT	WHILE	
COLLECT	DO	EXIT	MODIFY	SET		
CONCATENATE	ELSE	EXPORT	MOVE-CORRESP.	SHIFT		
CONDENSE	ELSEIF	FIELD-SYMBOLS	OVERLAY	SORT		

- **Syntaxprüfung für ABAP Versionen**
 - Report **DEMO_ABAP_VERSIONS**
 - Paket enthält viele gute **Modern ABAP** Beispiele



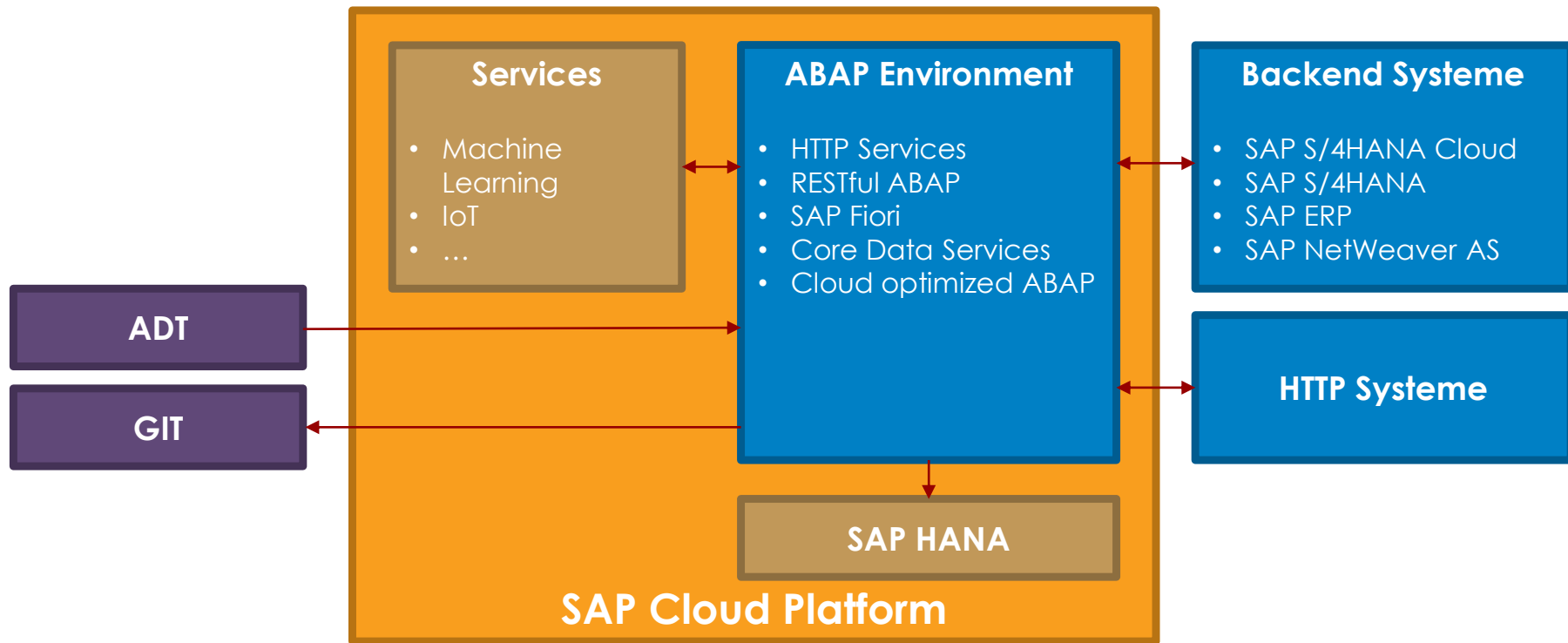
SAP Cloud Platform ABAP Environment

• SAP Cloud Platform ABAP Environment

- Nicht unterstützte ABAP Funktionalität
 - Nicht Cloud Ready ABAP Statements
 - EXEC SQL, OPEN DATASET, CALL SCREEN, ...
 - Obsolete ABAP Statements
 - MOVE, ...
 - Nicht unterstützte ABAP Technologien
 - SAP Gui, Web Dynpro, BSP, ...

● SAP Cloud Platform ABAP Environment

- Whitelisted ABAP Objects und APIs
 - 400 ABAP Entwicklungsobjekte (Klassen, CDS Views, Data Elements, ...)
 - Liste wird mit jedem neuen Release erweitert
 - Später werden auch Funktionalitäten wie Nummernkreise oder Änderungsbelege hinzukommen



**Harald Kuck**

September 4, 2018 14 minute read

SAP Cloud Platform ABAP Environment

87 Likes 41,320 Views 76 Comments

**Karl Kessler**

September 5, 2018 7 minute read

SAP Cloud Platform ABAP Environment is live

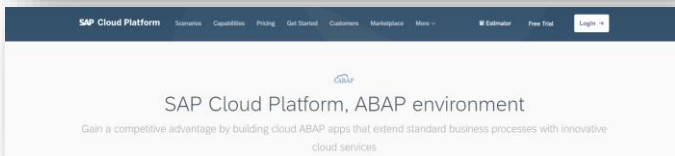
15 Likes 5,165 Views 17 Comments

Article

Take Your ABAP Skills to the Cloud

Develop New Cloud Applications in ABAP with SAP Cloud Platform ABAP Environment

by Karl Kessler, Vice President of Product Management ABAP Platform, SAP SE | SAPinsider, Volume 19, Issue 3



<https://blogs.sap.com/2018/09/04/sap-cloud-platform-abap-environment/>

<https://blogs.sap.com/2018/09/05/sap-cloud-platform-abap-environment-is-live/>

<https://sapinsider.wispubs.com/Assets/Articles/2018/September/SPI-Take-Your-ABAP-Skills-to-the-Cloud>

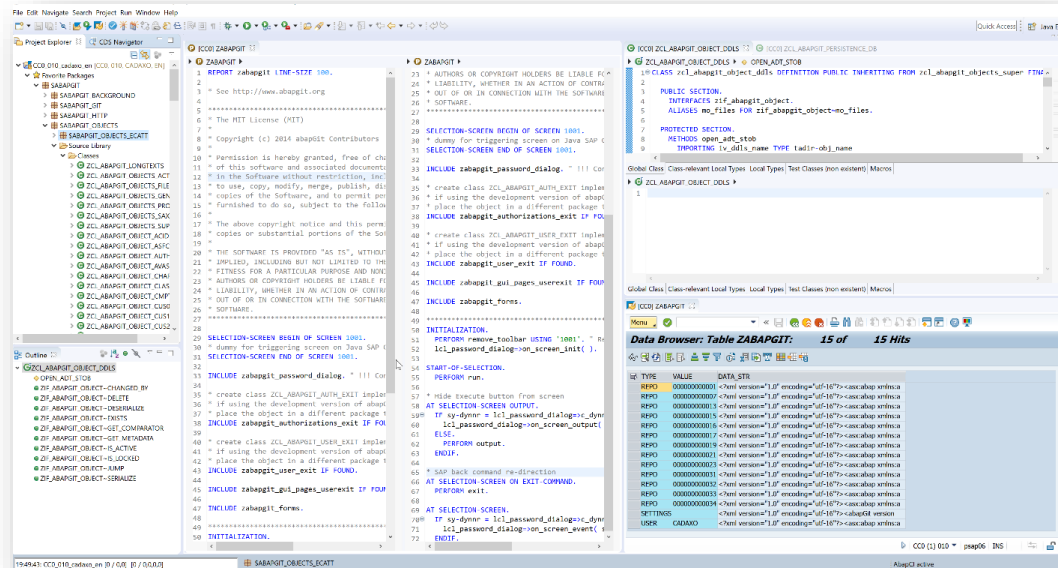
<https://cloudplatform.sap.com/capabilities/product-info.SAP-Cloud-Platform-ABAP-environment.4d0a6f95-42aa-4157-9932-d6014a68d825.html>

ADT – ABAP Development Tools

Eclipse

- Moderne IDE
- Erweiterung zum SAP GUI
- native Editoren für Entwicklungsobjekte
- Inplace GUI – SE11

- Verschiede Views gleichzeitig

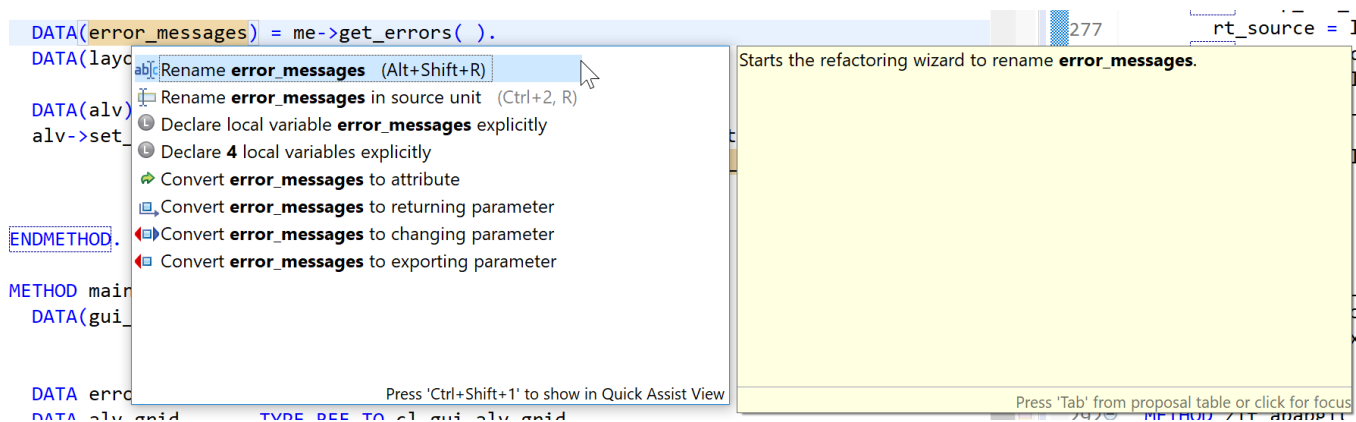


- Detach View – zB Editor auf 2. Monitor

```
1 REPORT zabagit LINE-SIZE 100.
2
3 * See http://www.abagit.org
4
5 *****
6 * The MIT License (MIT)
7 *
8 * Copyright (c) 2014 abagit Contributors
9 *
10 * Permission is hereby granted, free of charge, to any person obtaining a copy
11 * of this software and associated documentation files (the "Software"), to deal
12 * in the Software without restriction, including without limitation the rights
13 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
14 * copies of the Software, and to permit persons to whom the Software is
15 * furnished to do so, subject to the following conditions:
16 *
17 * The above copyright notice and this permission notice shall be included in all
18 * copies or substantial portions of the Software.
19 *
20 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
21 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
22 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
23 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
24 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
25 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
26 * SOFTWARE.
27 *****
28
29 SELECTION-SCREEN BEGIN OF SCREEN 1001.
30 * dummy for triggering screen on Java SAP GUI
31 SELECTION-SCREEN END OF SCREEN 1001.
32
33 INCLUDE zabagit_password_dialog. " !! Contains SELECTION SCREEN
34
35 * create class ZCL_ABAGIT_AUTH_EXIT implementing ZIF_ABAGIT_AUTH in following include,
36 * if using the development version of abagit create a global class instead
37 * place the object in a different package than ZABAGIT
38 INCLUDE zabagit_authorizations_exit IF FOUND.
39
40 * create class ZCL_ABAGIT_USER_EXIT implementing ZIF_ABAGIT_EXIT in following include,
41 * if using the development version of abagit create a global class instead
42 * place the object in a different package than ZABAGIT
43 INCLUDE zabagit_user_exit IF FOUND.
44
45 INCLUDE zabagit_gui_pages_userexit IF FOUND.
46
47 INCLUDE zabagit_forms.
48
49 *****
50 INITIALIZATION.
51 PERFORM remove_toolbar USING '1001'. " Remove toolbar on html screen
52 lcl_password_dialog=>on_screen_init( ).
53
54 START-OF-SELECTION.
55 PERFORM run.
56
```

```
17 * The above copyright notice and this permission notice shall be included in all
18 * copies or substantial portions of the Software.
19 *
20 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
21 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
22 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
23 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
24 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
25 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
26 * SOFTWARE.
27 *****
28
29 SELECTION-SCREEN BEGIN OF SCREEN 1001.
30 * dummy for triggering screen on Java SAP GUI
31 SELECTION-SCREEN END OF SCREEN 1001.
32
33 INCLUDE zabagit_password_dialog. " !! Contains SELECTION SCREEN
34
35 * create class ZCL_ABAGIT_AUTH_EXIT implementing ZIF_ABAGIT_AUTH in following include,
36 * if using the development version of abagit create a global class instead
37 * place the object in a different package than ZABAGIT
38 INCLUDE zabagit_authorizations_exit IF FOUND.
39
40 * create class ZCL_ABAGIT_USER_EXIT implementing ZIF_ABAGIT_EXIT in following include,
41 * if using the development version of abagit create a global class instead
42 * place the object in a different package than ZABAGIT
43 INCLUDE zabagit_user_exit IF FOUND.
44
45 INCLUDE zabagit_gui_pages_userexit IF FOUND.
46
47 INCLUDE zabagit_forms.
48
49 *****
50 INITIALIZATION.
51 PERFORM remove_toolbar USING '1001'. " Remove toolbar on html screen
52 lcl_password_dialog=>on_screen_init( ).
53
54 START-OF-SELECTION.
55 PERFORM run.
56
57 * Hide Execute button from screen
58 AT SELECTION-SCREEN OUTPUT.
59 IF sy-dynnr = lcl_password_dialog=>c_dynnr.
60 lcl_password_dialog=>on_screen_output( ).
61 ELSE.
62 PERFORM output.
63 ENDF.
64
65 * SAP back command re-direction
66 AT SELECTION-SCREEN ON EXIT-COMMAND.
67 PERFORM exit.
68
69 AT SELECTION-SCREEN.
70 IF sy-dynnr = lcl_password_dialog=>c_dynnr.
71 lcl_password_dialog=>on_screen_event( sscrfields-ucomm ).
72 ENDF.
```

- Qickfix – automatisierbare Codeanpassungen



- CDS Views (DDL und DCL Sourcen)
- ABAP in the Cloud
- Flexible Anordnung von Views (zB Editoren, Meldungen,...)
- Inplace GUI
- Quickfixes
- Refactoring / Renaming
- “Dark Theme”
- Plugins – ABAP Favorites, ABAP Continuous Integration,...

- Noch nicht Standardsoftware in Unternehmen
(z.B.: lokaler Admin, Citrix, Softwarecenter,...)
- Debugger
- Batch Input Mappen sichtbare abspielen
(ja, das gibt's noch!)
- Versionen und Updates Eclipse und ADT

- Java Version prüfen (32/64)
- Eclipse IDE for Enterprise Java Developers – Java EE
- Firewalls (Eclipse Proxy Settings)
- Java Version über eclipse.ini setzten

...

-vm

C:\<..>\JDK\1.8\bin\javaw.exe

-vmargs

...

- IDE + JAVA auf USB Stick ;-)



- **ABAP Code Retreat**

- Founded & Supported by [Damir Majer](#) & [Christian Drumm](#)

- <https://wiki.scn.sap.com/wiki/display/events/ABAP+CodeRetreat>

- <https://wiki.scn.sap.com/wiki/display/events/ABAP+CodeRetreat>
- <https://tools.hana.ondemand.com/#abap>
 - ADT Info und Download
- <https://marketplace.eclipse.org/category/free-tagging/abap>
 - Eclipse Plugins für ADT

Neu ABAP Sprachelemente und Open SQL/ABAP SQL Erweiterungen

- **ABAP Code**

- Bessere Lesbarkeit
- Kompakt
- Einfach
- Intuitiver
- Modern
- Schöner

- NEW
- VALUE
- REF
- EXACT
- CONV
- CORRESPONDING
- CAST
- REDUCE
- FILTER
- COND
- SWITCH
- LINES
- LINE_EXISTS
- LINE_INDEX
- IS INSTANCE OF
- CONTAINS
- CONTAINS_ANY_OF
- CONTAINS_ANY_NOT_OF
- MATCHES
- **SQL Expressions**
- **String Expressions**
- **Regular Expressions**
- Table Expressions
 - Chaining
 - Default, Optional
- LOOP AT GROUP
- FOR GROUPS
- FILTER
- REDUCE

```
DATA error_messages TYPE bapiret2_tab.  
DATA alv_grid       TYPE REF TO cl_gui_alv_grid.  
DATA layout         TYPE lvc_s_layo.
```

```
CREATE OBJECT alv_grid EXPORTING i_parent = gui_container.
```

```
layout = me->build_layout( ).
```

Classical

```
error_messages = me->get_errors( ).
```

```
alv_grid->set_table_for_first_display( EXPORTING is_layout = layout  
                                      CHANGING it_outtab = error_messages ).
```

```
DATA(error_messages) = me->get_errors( ).
```

```
NEW cl_gui_alv_grid( gui_container )->set_table_for_first_display( EXPORTING is_layout = me->build_layout( )  
                                                                  CHANGING it_outtab = error_messages ).
```

Modern


```
DATA(error_messages) = me->get_errors( ).  
NEW cl_gui_alv_grid( gui_container )->set_table_for_first_display( EXPORTING is_layout = me->build_layout( )  
                                                                    CHANGING it_outtab = error_messages ).
```

Modern 😞

Debugging 😞

F6 executes: constructor

build_layout

set_table_for_first_display

F5 F7 F5 F7 F5 to enter
set_table_for_first_display

```
DATA(error_messages) = me->get_errors( ).  
DATA(layout) = me->build_layout( ).
```

Modern 😊

```
DATA(alv) = NEW cl_gui_alv_grid( gui_container ).  
alv->set_table_for_first_display( EXPORTING is_layout = layout  
                                CHANGING it_outtab = error_messages ).
```

- **Tabellenzugriff**

```
alv->set_table_for_first_display( EXPORTING is_layout = layouts[ default_layout_index ]  
                                CHANGING  it_outtab = error_messages ).
```

```
layouts[ default_layout_index ]-edit = abap_true.
```

```
layouts[ template = template-samll ]-edit = abap_true.
```

• Moderne SQL Syntax

- Komma getrennte Feldliste
- Hostvariablen mit @ escaped
- INTO Klausel am Ende vom SQL Kommando
- Expression enabled
- Strengere Syntaxprüfung

```
51 SELECT 'Personen' AS type_txt,  
52 COALESCE( name_first && name_last, name_org1 && name_org2 ) AS name  
53 FROM but000  
54 WHERE type = @type_person  
55 INTO TABLE @DATA(persons).  
56
```

Console Problems Debug Shell ABAP Internal Table (Debugger) ✕

Table Data Columns PERSONS [58x2(176)] Standard Table

Filter pattern ✓

Row	TYPE_TXT	NAME
1	Personen	DominikBigl
2	Personen	
3	Personen	JohannFößleitner

- **Code Push Down**
 - Berechnungen
 - Stringbearbeitung
 - Konvertierungen
 - Datum/Zeitberechnung

SQL Funktion	Beschreibung	SQL	CDS	Buffer
ABS(arg)	Absolutbetrag von arg.	7.50	7.40	✓
CEIL(arg)	Kleinste ganzzahlige Zahl, die nicht kleiner als der Wert von arg ist.	7.50	7.40	✓
DIV(arg1, arg2)	Ganzzahliger Anteil der Division von arg1 durch arg2.	7.50	7.40	✓
DIVISION(arg1, arg2, dec)	Division von arg1 durch arg2. Das Ergebnis wird auf dec gerundet.	7.51	7.40	
FLOOR(arg)	Größte ganzzahlige Zahl, die nicht größer als der Wert von arg ist.	7.50	7.40	✓
MOD(arg1, arg2)	Positiver oder negativer ganzzahliger Rest der Division von arg1 durch arg2.	7.50	7.40	✓
ROUND(arg, pos)	Gerundeter Wert von arg. Falls pos größer 0 wird zur Position pos gerundet.	7.50	7.40	

SQL Funktion	Beschreibung	SQL	CDS	Buffer
CONCAT(arg1, arg2)	Verkettung der Zeichenketten in arg1 und arg2. Schließende Leerzeichen werden ignoriert.	7.50	7.40	✓
CONCAT_WITH_SPACE(arg1, arg2, spaces)	Verkettung der Zeichenketten in arg1 und arg2 wie mit CONCAT, jedoch getrennt mit spaces Leerzeichen.	7.51	7.50	✓
INSTR(arg, sub)	Position des ersten Vorkommens der Zeichenkette aus sub in arg.	7.51	7.50	
LEFT(arg, len)	Zeichenkette der Länge len mit den len linken Zeichen von arg ohne Berücksichtigung der schließenden Leerzeichen.	7.51	7.50	
LENGTH(arg)	Anzahl der Zeichen in arg ohne Berücksichtigung möglicher schließender Leerzeichen.	7.50	7.50	
LOWER(arg)	Zeichenkette der Länge arg, in der alle Groß- in Kleinbuchstaben konvertiert sind.	7.51	7.51	
LPAD(arg, len, src)	Zeichenkette der Länge len mit dem rechtsbündigen Inhalt von arg ohne schließende Leerzeichen.	7.50	7.50	
LTRIM(arg, char)	Zeichenkette mit dem Inhalt von arg, in dem alle schließenden Leerzeichen und führenden Zeichen entfernt sind, die dem Zeichen in char entsprechen.	7.50	7.50	
REPLACE(arg1, arg2, arg3)	Zeichenkette arg1, in der unter Beachtung der Groß- und Kleinschreibung alle Vorkommen von arg2 durch den Inhalt von arg3 ersetzt sind.	7.50	7.40	
RIGHT(arg, len)	Zeichenkette der Länge len mit den len rechten Zeichen von arg ohne Berücksichtigung der schließenden Leerzeichen.	7.50	7.50	
RPAD(arg, len, src)	Zeichenkette der Länge len mit dem linksbündigen Inhalt von arg ohne schließende Leerzeichen.	7.51	7.50	
RTRIM(arg, char)	Zeichenkette mit dem Inhalt von arg, in dem alle schließenden Leerzeichen und schließenden Zeichen entfernt sind, die dem Zeichen in char entsprechen.	7.50	7.50	
SUBSTRING(arg, pos, len)	Teilfeld von arg ab der Position pos in der Länge len.	7.50	7.50	✓
UPPER(arg)	Zeichenkette der Länge arg, in der alle Klein- in Großbuchstaben konvertiert sind.	7.51	7.51	

SQL Funktion	Beschreibung	SQL	CDS
FLTP_TO_DEC(arg AS dtype)	Konvertierung von arg (Gleitpunktzahl) in eine gepackte Zahl.	-	7.51
BINTOHEX(arg)	Konvertierung von arg (RAW) in eine Zeichenkette (CHAR).	7.52	7.50
HEXTOBIN(arg)	Konvertierung von arg (CHAR) in eine Bytekette (RAW).	7.52	7.50
UNIT_CONVERSION(p1 => a1, ...)	Konvertierung von Mengeneinheiten.	-	7.40
CURRENCY_CONVERSION(p1 => a1, ...)	Konvertierung von Währungen.	-	7.40
DECIMAL_SHIFT(p1 => a1, ...)	Dezimaltrennzeichen setzen.	-	7.40

SQL Funktion	Beschreibung	SQL	CDS	Buffer
COALESCE(arg1, arg2, ...)	Wert des ersten Arguments, das nicht den Null-Wert hat.	✓	7.40	✓

SQL Funktion	Beschreibung	SQL	CDS
DATS_IS_VALID	Stellt fest, ob ein Datum gültig ist	7.52	7.50
DATS_DAYS_BETWEEN	Differenz zwischen zwei Datumsangaben	7.52	7.50
DATS_ADD_DAYS	Addiert eine Anzahl von Tagen zu einem Datum	7.52	7.50
DATS_ADD_MONTHS	Addiert eine Anzahl von Monaten zu einem Datum	7.52	7.50
TIMS_IS_VALID	Stellt fest, ob eine Zeitangabe gültig ist	7.53	7.50
TSTMP_IS_VALID	Stellt fest, ob ein Zeitstempel gültig ist (YYYYMMDDHHMMSS)	7.53	7.50
TSTMP_CURRENT_UTCTIMESTAMP	Liefert einen aktuellen Zeitstempel	7.53	7.50
TSTMP_SECONDS_BETWEEN	Berechnet die Differenz von zwei Zeitstempel	7.53	7.50
TSTMP_ADD_SECONDS	Addiert eine Anzahl von Sekunden zu einem Zeitstempel	7.53	7.50
ABAP_SYSTEM_TIMEZONE	Gibt die Systemzeitzone des ABAP Servers zurück	7.53	7.51
ABAP_USER_TIMEZONE	Gibt die Benutzerzeitzone des Users vom ABAP Server zurück	7.53	7.51
TSTMP_TO_DATS	Extrahiert aus einem Zeitstempel das Datum	7.53	7.51
TSTMP_TO_TIMS	Extrahiert aus einem Zeitstempel die Uhrzeit	7.53	7.51
TSTMP_TO_DST	Extrahiert aus einem Zeitstempel die Sommerzeitmarkierung	7.53	7.51
DATS_TIMS_TO_TSTMP	Erzeugt aus einem Datum und Uhrzeit einen Zeitstempel	7.53	7.51

- Concat

```
DATA(message) = 'Modern' && | ABAP | && | by { name-cadaxo_short }|.
```

alv-

MESSAGE = Modern ABAP by Cadaxo

- CR/LF und Tab

```
DATA(message) = 'Modern' && | ABAP | && |\r\n\tby { name-cadaxo_short }|.
```

MESSAGE = Modern ABAP \r\n\tby Cadaxo

Modern ABAP
by Cadaxo

• Embedded Expressions

```
DATA(message) = 'Modern' && | ABAP | && | \r\n\tby { name-cadaxo_short } | .
```

Embedded Expressions - format_options

Syntax

```
... [WIDTH      = len]
[ALIGN        = LEFT|RIGHT|CENTER| (dobj) |expr]
[PAD          = c]
[CASE         = RAW|UPPER|LOWER| (dobj) |expr]
[SIGN         = LEFT|LEFTPLUS|LEFTSPACE|RIGHT|RIGHTPLUS|RIGHTSPACE| (dobj) |expr]
[EXPONENT     = exp]
[DECIMALS     = dec]
[ZERO        = YES|NO| (dobj) |expr]
[XSD         = YES|NO| (dobj) |expr]
[STYLE       = SIMPLE|SIGN_AS_POSTFIX|SCALE_PRESERVING
              |SCIENTIFIC|SCIENTIFIC_WITH_LEADING_ZERO
              |SCALE_PRESERVING_SCIENTIFIC|ENGINEERING
              | (dobj) |expr]
[CURRENCY    = cur]
[NUMBER      = RAW|USER|ENVIRONMENT| (dobj) |expr]
[ALPHA       = IN|OUT|RAW| (dobj) |expr]
[DATE        = RAW|ISO|USER|ENVIRONMENT| (dobj) |expr]
[TIME        = RAW|ISO|USER|ENVIRONMENT| (dobj) |expr]
[TIMESTAMP   = SPACE|ISO|USER|ENVIRONMENT| (dobj) |expr]
[TIMEZONE    = tz]
[COUNTRY     = cty] ...
```

- APLHA Konvertierung

```
DATA(partner) = CONV bu_partner( |{ '1' ALPHA = IN }| ).
```

◆ PARTNER = 0000000001

0000000001

```
DATA(partner) = |{ '33' WIDTH = 10 ALPHA = IN }|.
```

◆ PARTNER = 0000000033

0000000033

- https://help.sap.com/doc/abapdocu_753_index.htm/7.53/en-US/index.htm?file=abennews.htm

ABAP - Release-Specific Changes

[Changes in Releases 7.5x](#)

[Changes in Release 7.40 and Its SPs](#)

[Changes in Release 7.0 and its EhPs](#)

[Changes in Releases 6.xx](#)

[Changes in Releases 4.xx](#)

[Changes in Release 3.0](#)

[Changes Before Release 3.0](#)

Object-Oriented Programming

• ABAP Objects – Aber richtig!

```
CLASS zcl_fugr DEFINITION
PUBLIC
FINAL
CREATE PUBLIC .

PUBLIC SECTION.
CLASS-METHODS read_all_partners RETURNING VALUE(e_partners) TYPE bup_but000_t.

CLASS-METHODS save_data_partner IMPORTING i_partner_data TYPE but000.

CLASS-METHODS search_partner IMPORTING i_partner          TYPE bu_partner OPTIONAL
                                i_name                     TYPE bu_name1
                                RETURNING VALUE(e_partners) TYPE bup_but000_t.
```

Class Builder: Display Class ZCL_FUGR

Class/Interface: ZCL_FUGR Implemented / Active

Properties Interfaces Friends Attributes **Methods** Events Types Aliases

Parameters Exceptions **Sourcecode**

Method	Level	Visibility	M...	Description
READ_ALL_PARTNERS	Static Method	Public		
SAVE_DATA_PARTNER	Static Method	Public		
SEARCH_PARTNER	Static Method	Public		

Nur statische Methoden **IST NICHT** ABAP Object/OOP
=> Wie eine FUNKTIONSGRUPPE mit FMs

- **ABAP Objects**
 - **Instanzen - Datenkapselung**
 - **Interfaces**
 - **Abstrakten Klassen**
 - **Vererbung**
 - **Exceptions - Ausnahmeklassen**
 - **Events**
 - **=> ADT Refactoring Support**



- **ABAP OO**

- Videoaufnahme des Vortrags auf der #sitVIE 2018 von Damir Majer

- <https://youtu.be/UAe1EUZZpw>

ABAP Objects - Übersicht

ABAP Objects ist der objektorientierte Bestandteil von ABAP.

https://help.sap.com/doc/abapdocu_753_index.htm/7.53/de-DE/index.htm?file=abenabap_objects_oview.htm



<https://open.sap.com/courses/java1>

MAI
20

Applied ABAP Objects mit Damir Majer

von Majer Consulting

<https://www.eventbrite.de/e/applied-abap-objects-mit-damir-majer-tickets-53966219415>

Obsolete ABAP Sprachelemente

- **Modularisierungseinheiten**

FORM unterprogramm.

ENDFORM.

ON COMMIT / ON ROLLBACK

PERFORM unterprogramm.

FUNCTION <FM>

...

TABLES

Systemfelder

Obsolete Systemfelder

- Frühere Systemfelder
- Spool-Parameter

Frühere Systemfelder

Die folgenden Systemfelder sind obsolet und sollen nicht mehr verwendet werden. Viele stammen noch aus dem R2. Beim Übergang von R2 zu R3 wurden einige Systemfelder übernommen, die nicht mehr versorgt werden.

Name	Typ	Lang	Inhalt
xy-aps11a	2		Im R2 Kernel, welche der SAP-Anwendungen installiert sind. Wird nicht mehr versorgt.
xy-batdefn	1		Im R2 Kennzeichen für tägliche Hintergrundplanung. Wird nicht mehr versorgt.
xy-batmo1	1		Im R2 Kennzeichen für monatliche Hintergrundplanung. Wird nicht mehr versorgt.
xy-batmo2	1		Im R2 Kennzeichen für einwöchige Hintergrundplanung. Wird nicht mehr versorgt.
xy-batmo3	1		Im R2 Kennzeichen für sofortige Hintergrundplanung. Wird nicht mehr versorgt.
xy-batmo4	1		Im R2 Kennzeichen für wöchentliche Hintergrundplanung. Wird nicht mehr versorgt.
xy-brspc1a	4		Im R2 bei Hintergrundverarbeitung Kennzeichen für Ausgabe in Spool. Wird nicht mehr versorgt.
xy-brspc1b	1		Im R2 bei Hintergrundverarbeitung Kennzeichen für Ausgabe in Spool. Wird nicht mehr versorgt.
xy-cconvp	9		Im R2 Kennzahl/Exportkennzeichen aus CURRENCY CONVERSION. Wird nicht mehr versorgt.
xy-cconvp	9		Im R2 Tabellenkurs aus CURRENCY CONVERSION. Wird nicht mehr versorgt.
xy-cctab1a	1		Im R2 Kennzeichen aus CURRENCY CONVERSION. Wird nicht mehr versorgt.
xy-cctab1b	4		Im R2 Kennzeichen aus CURRENCY CONVERSION. Wird nicht mehr versorgt.
xy-cctyp1a	1		Im R2 Kennzahl aus CURRENCY CONVERSION. Wird nicht mehr versorgt.
xy-dconvp	4		Übersystem des R2-Systems. Wird nicht mehr versorgt.
xy-dkey1a	3		Früher alternatives Funktionscode-Memo. Wird nicht mehr versorgt.
xy-lcodb1a	1		Letztes Datenbank, nicht implementiert.
xy-lconvp	1		Letztes Datenbankoperation, nicht implementiert.
xy-latext	16		Programmgesteuerter Bezeichner für Listenstufen. Kann während der Listenanstellung im ABAP-Programm mit Werten versorgt werden. Der bei Fertigstellung einer Liste gültige Wert wird mit der Liste abgespeichert. Bei einem in
xy-machb1	4		Früher Name der Date für Match-Code Zugriff. Wird nicht mehr versorgt.
xy-machb1	1		Abkürzung für Buchstabe für die nicht mehr unterstützte Anweisung <code>sort</code> aus dem R2-System.
xy-neg11a	85		Unbekannt. Wird nicht mehr versorgt.
xy-neg11b	1		Im R2 maximale Anzahl von Seiten pro Liste. Wird nicht mehr versorgt.
xy-prefac1	3		Präfix für Hintergrund-Jobs. Wird nicht mehr versorgt.
xy-rfext1	36		Unbekannt.
xy-spcor1a	16		Im R2 konnten Spool-Daten mit der <code>TRANSFER</code> -Anweisung bearbeitet werden, die auch <code>xy-spcor1a</code> setzte. Wird nicht mehr versorgt.
xy-rfext1a	8		Früher Name der externen Abgabedaten für Extrakte. Wird nicht mehr versorgt.
xy-taext1	1		Früher maximale Anzahl von Einträgen in einer internen Tabelle. Wird nicht mehr versorgt.
xy-taext1a	12		Früher Name einer internen Tabelle nach einem Zugriff. Wird nicht mehr versorgt.
xy-taext1b	1		Bei dem Aufruf <code>xy-taext1a</code> <code>TABLA</code> oder <code>xy-taext1b</code> <code>TABLA</code> wurde <code>xy-taext1b</code> früher mit dem Wert für den initialen Hauptspeicherbedarf der angesprochenen internen Tabelle versorgt. Nicht mehr versorgt.
xy-tspg1a	1		Früher Kennzeichen, ob eine interne Tabelle in das Paging-Datensatz ausgelagert ist. Wird nicht mehr versorgt.
xy-tttab1a	1		Früher Index der letzten gekannten Zeile einer internen Tabelle. Wird nicht mehr versorgt.
xy-tttab1b	1		Früher Offset von internen Tabellen im Stützspeicher. Wird nicht mehr versorgt.
xy-waext1a	5		Früher Buchungsbereich-Währung nach Lesen von Buchungssegment. Wird nicht mehr versorgt.
xy-waext1b	1		Im R2 Nummer der ausgewählten Listenzeile bei Verzeugsungsliste im GUI-Fenster. Statt dessen <code>xy-waext1b</code> verwenden.
xy-waext1c	1		Im R2 Cursorposition bei Verzeugsungsliste im GUI-Fenster. Statt dessen <code>xy-waext1c</code> verwenden.
xy-waext1d	1		Im R2 Index der Liste bei Verzeugsungsliste im GUI-Fenster. Statt dessen <code>xy-latext</code> verwenden.
xy-waext1e	1		Im R2 Cursorposition bei Verzeugsungsliste im GUI-Fenster. Statt dessen <code>xy-waext1e</code> verwenden.
xy-waext1f	79		Im R2 Inhalt der ausgewählten Zeile bei Verzeugsungsliste im GUI-Fenster. Statt dessen <code>xy-latext</code> verwenden.
xy-waext1g	1		Im R2 Fensterkoordinaten bei Verzeugsungsliste im GUI-Fenster. Wird nicht mehr versorgt.
xy-waext1h	1		Im R2 Fensterkoordinaten bei Verzeugsungsliste im GUI-Fenster. Wird nicht mehr versorgt.
xy-waext1i	1		Im R2 Fensterkoordinaten bei Verzeugsungsliste im GUI-Fenster. Wird nicht mehr versorgt.
xy-waext1j	1		Im R2 Fensterkoordinaten bei Verzeugsungsliste im GUI-Fenster. Wird nicht mehr versorgt.

Spool-Parameter

Spool-Parameter werden von der Laufzeitgebung in einer Struktur vom Typ `PHO_FH0000` an das Spool-System übergeben. Bevor diese Struktur verwendet wurde, dienten dazu Systemfelder. Bei der Spool-Aktivierung wird auch heute noch eine Teil

Name	Typ	Lang	Inhalt
xy-paext1a	16		Druckbereich
xy-paext1b	4		Druckbereich
xy-paext1c	1		Verzeichnis
xy-paext1d	12		Name des Spool-Auftrags
xy-paext1e	12		Abschungsname
xy-paext1f	1		Archiviert
xy-paext1g	3		Zeich der Ausgabe
xy-paext1h	6		Name der Spool-Daten
xy-paext1i	1		Druckort
xy-paext1j	1		Neuer Spool-Auftrag
xy-paext1k	12		Druckbereich
xy-paext1l	1		Druckbereich
xy-paext1m	60		Beschreibungstext
xy-paext1n	79		Teil des ausstehenden Programms

```
LOOP AT SCREEN.  
  MODIFY SCREEN.  
ENDLOOP.
```

```
* DATA screen TYPE screen.  
LOOP AT SCREEN INTO DATA(screen).  
  MODIFY screen FROM screen.  
ENDLOOP.
```

```
IF e_parameter IS REQUESTED.  
*   ...  
ENDIF.
```

```
IF e_parameter IS SUPPLIED.  
*   ...  
ENDIF.
```

Obsoleter Operator	Gültiger Operator
><	<>, NE
=<	<=, LE
=>	>=, GE

```
CATCH SYSTEM-EXCEPTIONS exc1 = 1  
                           exc2 = 2  
                           OTHERS = 3.
```

```
    ...  
ENDCATCH.  
IF sy-subrc <> 0.  
    ...  
ENDIF.
```

MOVE
REFRESH
RANGES
WITH HEADER LINE
OCCURS n

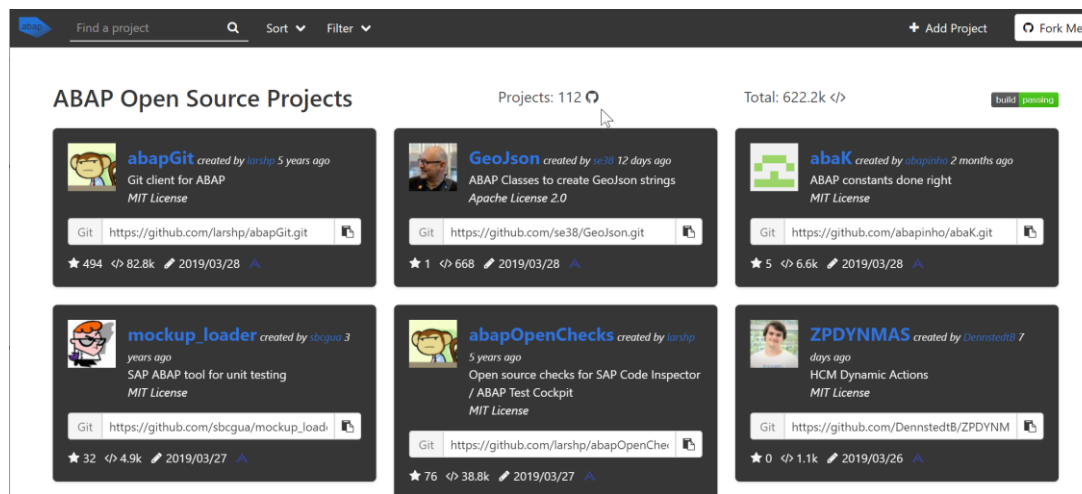
Tabellenoperationen mit impliziter Kopfzeile
 LOOP AT itab.
 READ/INSERT/APPEND/...

https://help.sap.com/doc/abapdocu_753_index.htm/7.53/de-DE/index.htm?file=abensyntax_obsolete.htm

abapGit und Open Source

- Git client for ABAP by **Lars Hvam**
 - <http://www.abapgit.org>
 - Paket <=> (online) Git Repository
- Erweitert SAP Versionierung
- Ermöglicht Austausch von SAP Entwicklungsobjekten
 - Klassen, Tabellen, Strukturen, IDocs, ...
- Aktuell DAS Transporttool für ABAP in the Cloud
 - auch Migration
- Open Source!

- SAPLink
- abap2xlsx
- <https://dotabap.org/>





- **abapGit BunKai**
 - Founded & Supported by 5 SAP Community Mitglieder

- <https://wiki.scn.sap.com/wiki/display/events/abapGit+BunKai>

<https://github.com/larshp/abapGit>

<https://open.sap.com/courses/git1> - **SAP Cloud Platform Version Control with Git**

<https://dotabap.org/>

CDS Views

● **ABAP CDS Views**

- Next Generation zum Definieren von Views
- CDS beinhaltet
 - DDL (Data Definition Language)
 - DCL (Data Control Language)
- 100% in ABAP integriert
 - ABAP Dictionary
 - Transportmanagement
- Pflege ausschließlich über ABAP in Eclipse
- Zugriff via Open SQL

● **ABAP CDS Views - Highlights**

- View Hierarchien
- Eingebaute Funktionen
- Parameter
- Associations
- Annotations
- View Extensions

CDS Views

- **Re-Use Szenarien**
 - Views können verschachtelt werden
- **CDS Features**
 - UNION, UNION ALL, Mengen bzw. Währungs Umrechnungen
 - Associations
 - Berechtigungsprüfungen
 - S/4HANA Programmiermodell

Open SQL

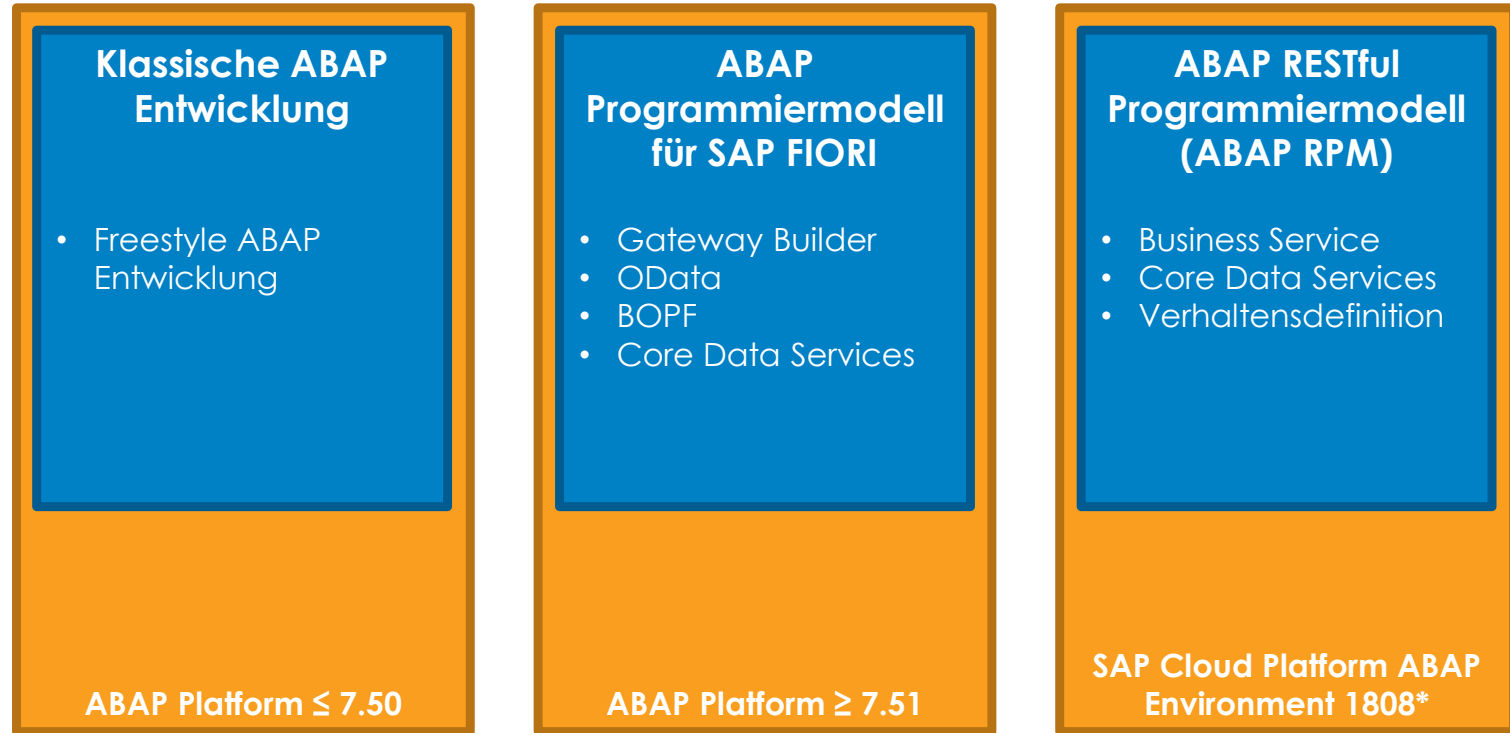
- Einmalverwendungen
- FOR ALL ENTRIES
- Kein Eclipse



<http://www.cadaxo.com/high-class-development/abap-cds-views-6-schritte-zum-verstaendnis/>

<http://www.cadaxo.com/high-class-development/webinar-abap-cds-views/>

ABAP RESTful Programming Model



*On-Premise kommt später

● **ABAP RESTful**

- Für zukünftige Entwicklungen in S/4HANA, SAP Cloud ABAP
- CDS & ABAP
- Business Object
 - Datenmodellierung mit CDS
 - Verhaltensdefinition & Implementierung
 - Managed / Unmanaged



Code Checks in SAP

- **Statische Prüfungen mit dem Code Inspector**
 - Einzelne Objekte oder viele Objekte
 - Namenskonventionen, Performancethemen, ...
 - Seit 7.52 auch Remote Checks möglich

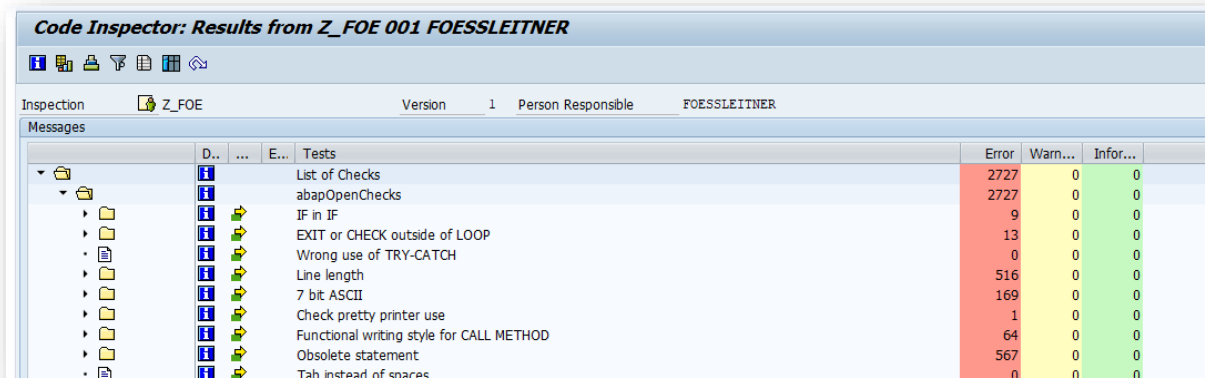
Code Inspector: Ergebnisse von ZFOE 001 CADAXO

Inspektion ZFOE Version 1 Verantwortlicher CADAXO

Meldungen

	D..	...	A..	Tests	Fehler	Warn...	Infor...
▼				Liste der Prüfungen	3	3	2
▼				Performance-Prüfungen	3	0	1
▼				Analyse der WHERE-Bedingung für SELECT	1	0	0
▼				Analyse der WHERE-Bedingung für UPDATE und DELETE	0	0	0
▼				SELECT-Anweisungen, die am Tabellenpuffer vorbei lesen	0	0	0
▼				Problematik SELECT *-Anweisungen suchen	2	0	1
▼				N.zu transformier.SELECT .. FOR ALL ENTRIES-Klauseln such.	0	0	0
▼				Nach SELECT-Anweisung mit DELETE-Anweisung suchen	0	0	0
▼				DB-Operationen in Schleifen über Modularisierungseinheiten	0	0	0
▼				'EXIT' oder keine Anweisung in SELECT...ENDSELECT Schleife	0	0	0
▼				SELECT-Anweisungen mit anschließendem CHECK	0	0	0
▼				Robuste Programmierung	0	1	0
▼				Unsichere Verwendung von FOR ALL ENTRIES	0	1	0
▼				Suchfunktionen	0	2	1
▼				Nach DB-Operationen suchen	0	2	1

- **Lars Hvam's: abapOpenChecks**
 - Umfangreiche ergänzende Checks für den Code Inspector
 - Aktuell bereits über 80 Checks



Code Inspector: Results from Z_FOE 001 FOESSLEITNER

Inspection: Z_FOE Version: 1 Person Responsible: FOESSLEITNER

Messages

	D..	...	E...	Tests	Error	Warn...	Infor...
▼				List of Checks	2727	0	0
▼				abapOpenChecks	2727	0	0
▼				IF in IF	9	0	0
▼				EXIT or CHECK outside of LOOP	13	0	0
▼				Wrong use of TRY-CATCH	0	0	0
▼				Line length	516	0	0
▼				7 bit ASCII	169	0	0
▼				Check pretty printer use	1	0	0
▼				Functional writing style for CALL METHOD	64	0	0
▼				Obsolete statement	567	0	0
▼				Tab instead of spaces	0	0	0

- <https://blogs.sap.com/2015/03/08/abapopenchecks-open-source-checks-sap-code-inspector-atc>
- <https://github.com/larshp/abapOpenChecks>

- **Der SQL Monitor liefert Performance-Daten**
 - Welche SQL Operationen/Tabellenzugriffe treten am häufigsten auf?
 - Welche SQL Operationen/Tagellenzugriffe lesen oder schreiben die meisten Einträge?

SQL-Monitor: Top 200 Datensätze, nicht aggregiert

Hilfe zur Anwendung Zeitreihe anzeigen SQL-Trace aktivieren/deaktivieren

DB-Ausf.	Gesamte DB-Zeit	Gesamtz.Sätze	Mittl.DB-Z	MWve.	Tabellenamen	SQL-Operationstyp	Obj-T.	Objektname	Include-Name	Include-Ze	ABAP-Quelltextfragment	G...	Int. Sess.	Ausf./Sess.	Programmnam
32.002	55.117,325	32.002	1,722	1,000	TRDIR	SELECT (Open S...	CLAS	CL_SCV_SOURCE_REPO...	CL_SCV_SOURCE_REPOSITORY====	3	SELECT SINGLE UCHECK FROM TRD...		1	32.002...	CL_SCV_SOUR
32.002	6.260,987	32.002	0,196	1,000	REPOSRC	SELECT (Open S...	CLAS	CL_SCV_SOURCE_REPO...	CL_SCV_SOURCE_REPOSITORY====	3	SELECT SINGLE SDATE TIME FROM...		1	32.002...	CL_SCV_SOUR
32.002	10.187,131	32.001	0,318	1,000	COVREF	INSERT (Open SQ...	PROG	RSCVR_INIT_START	RSCVR_INIT_START	253	INSERT COVREF_INIT FROM TABLE I...		1	32.002...	RSCVR_INIT_S
30.456	8.604,794	6.481	0,283	0,213	TADIR	Load Buffer (Ope...	CLAS	CL_SCV_SOURCE_REPO...	CL_SCV_SOURCE_REPOSITORY====	5	SELECT SINGLE MASTERLANG FROM ...		1	30.456...	CL_SCV_SOUR
12.682	44.802,026	0	3,533	0,000	<NO_TABLE>	Commit	FUGR	SYD8	LSYDBU27	10	COMMIT WORK ENDEXEC.		22	576,455	SAPLSYD8
11.199	5.411,694	11.199	0,483	1,000	BTCTCL	SELECT (Open S...	PROG	SAPMSSY2	RSBTCTRC	60	SELECT SINGLE * FROM BTCTCL WH...		8.709	1,286	SAPMSSY2
11.199	5.030,679	11.199	0,449	1,000	BTCTCL	UPDATE (Open S...	PROG	SAPMSSY2	RSBTCTRC	131	UPDATE BTCTCL.		8.709	1,286	SAPMSSY2
11.199	42.619,130	0	3,806	0,000	<NO_TABLE>	Commit	PROG	SAPMSSY2	RSBTCTRC	136	COMMIT WORK.		8.709	1,286	SAPMSSY2
9.979	2.934,555	9.975	0,294	1,000	TRDIR	SELECT (Open S...	FUGR	STR7	LSTR7F06	75	SELECT SINGLE * FROM TRDIR INTO...		7	1,425,571	SAPLSTR7
7.971	27.979,236	0	3,510	0,000	<NO_TABLE>	Commit	PROG	SAPMSSY2	<SYSINI>	23	ENDMODULE MODULE %_HDSYSPAI...		4.455	1,789	SAPMSSY2
7.110	2.184,849	7.110	0,307	1,000	TST01	Sonstige (ABAP S...	PROG	SAPMSSY2	<SYSINI>	23	ENDMODULE MODULE %_HDSYSPAI...		3.516	2,022	SAPMSSY2
7.032	3.662,629	7.032	0,521	1,000	TBTCO	Sonstige (ABAP S...	PROG	SAPMSSY2	<SYSINI>	23	ENDMODULE MODULE %_HDSYSPAI...		3.516	2,000	SAPMSSY2
5.413	11.242,193	5.413	2,077	1,000	BTCTCL	SELECT (Open S...	PROG	SAPMSSY2	RSBTCTRC	568	SELECT SINGLE * FROM BTCTCL WH...		5.286	1,024	SAPMSSY2

- Die **SQL Performance Tuning Worklist** kombiniert statische Checks (Code Inspector) mit Laufzeitdaten (SQL Monitor)
 - Liefert eine priorisierte Worklist für potentielle Performanceverbesserungen

Arbeitsvorrat für SQL-Performanceoptimierung

Ergebnisübersicht: 15 Befunde

DB-Ausf.	Gesamte DB-Zeit	Mittl. DB-Zeit	MWw.Sätze	Tabellennamen	Geändert	Reportname	Include-Name	IncZelle	ABAP-Quelltextfragment	Anz.	Befu.	Maximale DB-Zeit	Min.DB-Z.	DB-Alt
4	10,497	2,624	86,000	BUT000		ZPERF01	ZPERF01	16	SELECT * FROM BUT000.	3	3	3,591	2,258	0
4	0,939	0,235	1,000	VARI		ZPERF01	ZPERF01	0	REPORT ZPERF01 TABLES BUT000 PARAMETERS DUMM..			0,316	0,194	0
4	0,813	0,203	0,000	<NO_TABLE>		ZPERF02	[UNRESOLVABLE SOUR	0				0,221	0,188	0
2	0,452	0,226	0,000	<NO_TABLE>		ZPERF01	[UNRESOLVABLE SOUR	0				0,256	0,196	0

SQL-Monitor-Ergebnisse für SSEL START-OF-SELECTION:01 (ZPERF01)

Anfragetyp	EinstPunkt	SQL-Operationstyp	Tabellen	Geändert	ABAP-Quelltextfragment
Hintergrundjob	ZPERF01	SELECT (Open SQL)	BUT000		SELECT * FROM BUT000.

Befunde der statischen Prüfung für SSEL START-OF-SELECTION:01 (ZPERF01)

Prüftitel	Prüfmeldung	Zusatzinformationen	Priorität	Schw.
Problem: SELECT *-Anweisungen suchen	SELECT-Anweisung kann transf.werd. 3.5% d.Felder	ZPERF01 16 : SELECT * FROM	1	
Analyse der WHERE-Bedingung für SELECT	große Tabelle BUT000: Keine WHERE-Bedingung	ZPERF01 16	1	
SELECT-Anweisungen mit anschließendem CHECK	SELECT auf Tabelle BUT000 mit anschließendem CHE	ZPERF01 16	3	

- Das **ABAP Test Cockpit** verwendet u.a. CI Ergebnisse
 - Kombiniert ABAP Code Inspector, ABAP Unit Tests, ...
 - Freigabeszenarien

The screenshot displays the ABAP Test Cockpit (ATC) interface. The main window shows a list of findings under the heading 'CCO: Monday, CW22 2016'. The findings are categorized by priority and type, with a 'Details' view selected for one specific finding.

Findings List:

Pri.	Check Title	Check Message	Object Name	Obj. E.	Contact Person	Package	1st Found
3	Extended Program Chec...	Text element missing in a character string	ZCADAXO_RELEASENOT...	PROG	CADAXO	\$TMP	30.05.2016
3	Extended Program Chec...	Text element missing in a character string	ZCL_CRM_DATA_RUN_B...	CLAS	CADAXO	\$TMP	30.05.2016
3	Extended Program Chec...	Text element missing in a character string	ZCL_DEMO_CHART	CLAS	CADAXO	\$TMP	30.05.2016
3	Extended Program Chec...	Text element missing in a character string	ZCL_DEMO_GOOGLE_C	CLAS	CADAXO	\$TMP	30.05.2016

Details View:

Location / Finding	Description
Package	\$TMP
Function Group	ZCDX_DEMO_MOVIE
Function Module	ZCDX_DEMO_MOVIE_CREATE
Line Number	16
Check Title	Extended Program Check (SLIN)
Check Message	The called function module is obsolete
Priority	Priority 3
Found on	30.05.2016 15:07:06

Request Exemption Dialog:

Approver and Reason

Approver: * Q_MANAGER

Reason: * False Positive - finding does not apply - see justification

Justification: * Text is not relevant for translation (technical text)

Notify me by e-mail

☐ On rejection

☒ On approval or on rejection

☐ Never

Buttons: ? < Back Next > Finish Cancel

- **Der ABAP Call Monitor analysiert die Verwendungen**
 - Zeichnet die Aufrufe von Methoden, Funktionsbausteinen, Unterprogrammen, Zeitpunkten, ...
 - „Nachfolger“ von UPL (Solution Manager)
 - Keine Performanceinformationen

ABAP Call Monitor: Data Browser (28.03.2019)

Application Help

Request Ty...	Request Name	Called Program Name	Obj. Ty.	Called Object Name	Proc. Ty.	Called Processing Block Name	Called Class Name (If Any)	Number of Calls	Summation Coun...
Submit Report	RSUSR000	CL_ABAP_LIST_UTILITIES=====CP	CLAS	CL_ABAP_LIST_UTILITIES	METH	DYNAMIC_OUTPUT_LENGTH	CL_ABAP_LIST_UTILITIES	69	1
Transaction	SCMON	SAPLBTC	FUGR	BTC	FORM	BUILD_WHERE		63	1
Submit Report	SAPLSMPI	CL_ABAP_FILE_UTILITIES=====CP	CLAS	CL_ABAP_FILE_UTILITIES	METH	CHECK_STRING_7BIT_ASCII	CL_ABAP_FILE_UTILITIES	58	1
Submit Report	RSUSR000	CL_SALV_FORM_UIE_LAYOUT_GRID==CP	CLAS	CL_SALV_FORM_UIE_LAYOUT_GR...	METH	GET_COLUMN_COUNT	CL_SALV_FORM_UIE_LAYOUT_GR...	47	1
Submit Report	RSUSR000	CL_ABAP_TYPEDESCR=====CP	CLAS	CL_ABAP_TYPEDESCR	METH	DESCRIBE_BY_DATA	CL_ABAP_TYPEDESCR	40	1
Submit Report	RSUSR000	CL_SALV_FORM_RENDER_TOOL=====CP	CLAS	CL_SALV_FORM_RENDER_TOOL	METH	GET_HEIGHT	CL_SALV_FORM_RENDER_TOOL	37	1
Submit Report	RSUSR000	SAPLSDFRUNTIME	FUGR	SDFRUNTIME	FORM	MOVE_FLAG4_TO_OUTPUTSTYLE			1
Submit Report	RSUSR000	SAPLSDFRUNTIME	FUGR	SDFRUNTIME	FORM	X031L_TO_DFIES			1
Submit Report	RSUSR000	CL_ABAP_LIST_UTILITIES=====CP	CLAS	CL_ABAP_LIST_UTILITIES	METH	DISPLAY_OFFSET	CL_ABAP_LIST_UTILITIES	35	1
Submit Report	RSUSR000	CL_ABAP_SOFT_REFERENCE=====CP	CLAS	CL_ABAP_SOFT_REFERENCE	METH	GET	CL_ABAP_SOFT_REFERENCE	30	1

- <https://blogs.sap.com/2017/04/06/abap-call-monitor-scmmon-analyze-usage-of-your-code/>

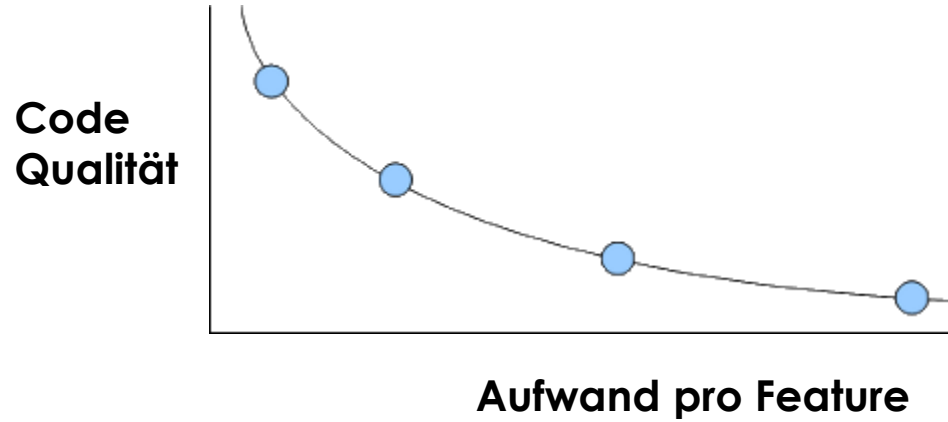
- **Weitere Analysetools**
 - SQL Trace, Performance Trace
 - Laufzeitanalyse
 - Memory Inspector

- <https://blogs.sap.com/2012/03/06/getting-started-with-abap-testing-and-troubleshooting>

Clean Code

*“Any fool can write code that a computer can understand. **Good programmers write code that humans can understand.**”*

Martin Fowler, Pionier der agilen Softwareentwicklung



- **Verständlicher Code ist enorm wichtig**
 - Überwiegende Kosten für Software sind Wartungskosten
 - Code wird daher überwiegend gelesen
 - Je unverständlicher der Code, desto größer die Gefahr, dass er nicht ausreichend verstanden wird

- **Die 3 wichtigsten Clean Code Prinzipien sind**
 - **DRY** - Don't Repeat Yourself
 - **KISS** - Keep it simple, stupid
 - **YAGNI** - You Ain't Gonna Need it

- **Namen sollen den Zweck beschreiben**

- Klingt einfacher, als es ist
- Gute Namen zu finden braucht Zeit, spart später aber Zeit
- Namenslängen ausnutzen

```
DATA D TYPE ...  
DATA LS_DATA TYPE ...  
DELIVERY_DATE->ADD( 5 ).
```

Solche Namen enthüllen nichts

```
DATA PARTNER_INVOICES TYPE TABLE OF ...  
DATA DAYS_SINCE_CREATION TYPE ...  
DELIVERY_DATE->ADDDAYSTO( 5 )  
DELIVERY_DATE->INCREASEBYDAYS( 5 )
```

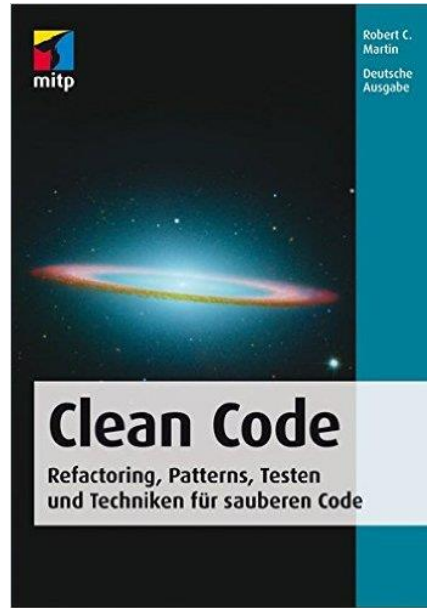
Namen beschreiben den Zweck, Code wird verständlicher

- **Erklärung durch Coding**
 - Ein Beispiel, wie mittels Coding erklärt wird

```
* employee is allowed?  
IF EMPLOYEE->AGE > 65 AND  
  EMPLOYEE->FLAGS = ABAP_TRUE AND  
  EMPLOYEE->ISLOCKED <> ABAP_TRUE.  
...  
ENDIF.
```



```
IF EMPLOYEE->ISALLOWED( ) = ABAP_TRUE.  
...  
ENDIF.
```



ISBN 978-3-8266-5548-7

Refactoring

- **Ziele von Refactoring**

- Code wird verständlicher
- Fehlersuche wird beschleunigt
- Code wird leichter erweiterbar
- Design des Software-Systems wird verbessert

*“We are doing this, to be able to deliver **more functionality more quickly.**”*

Martin Fowler, Pionier der agilen Softwareentwicklung

- **Refactoring Arten**
 - Litter-Pickup
 - Comprehension Refactoring
 - Preparatory Refactoring
 - Planned Refactoring
 - TDD Refactoring
 - Long-Term Refactoring

Refactoring SE80

Refactoring ADT

- Umbenennung von Objekten (Klassen, Methoden, ...)
- Methoden extrahieren
- Konstanten extrahieren
- Variablen extrahieren
- Sichtbarkeit von Attributen ändern
- Attribute von Subklassen in Superklassen verschieben
- ...

Cooler Refactoring Demo

ABAP Units

TEST-SEAM / TEST-INJECTION

- **Ziel von Unit Tests**

- Probleme/Bugs im Developmentprozess so früh wie möglich entdecken
- Probleme/Bugs nach Upgrades etc. rasch entdecken
- Reduktion langer Debugging Sessions
- Reduziert die „Hemmschwelle“ schlechtes Coding zu optimieren -> Clean Code!

- Refactoring ohne Unittests?
- Legacy Code testbar machen ⇔ Refactoring 😞

=> **TEST-SEAM / TEST-INJECTION**

Produktiver Code

```
...  
CLEAR partner_name.  
  
TEST-SEAM but000_select.  
    SELECT SINGLE name_last  
        FROM but000  
        WHERE partner = @i_partner  
        INTO @partner_name.  
END-TEST-SEAM.  
  
IF partner_name...  
...  

```

Test Class

```
...  
CLASS ltc_tests DEFINITION FOR  
TESTING .  
...  
    METHOD setup.  
  
        TEST-INJECTION but000_select.  
            partner_name = 'Bigl'.  
        END-TEST-INJECTION.  
  
    ENDMETHOD.  
...  

```


Normale Ausführung

```
...  
CLEAR partner_name.  
  
SELECT SINGLE name_last  
      FROM but000  
      WHERE partner = @i_partner  
      INTO @partner_name.  
  
IF partner_name...  
...
```

ABAPUnit Test

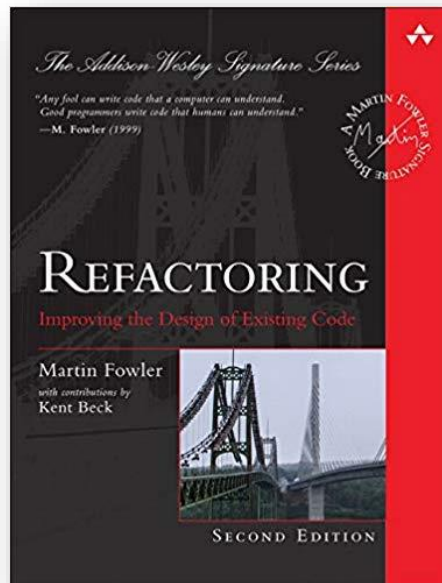
```
...  
CLEAR partner_name.  
  
partner_name = 'Bigl'.  
  
IF partner_name...  
...
```

<http://www.cadaxo.com/high-class-development/webinar-tdd-mit-abap-units/>

https://help.sap.com/doc/abapdocu_750_index.htm/7.50/de-DE/abentest_seams.htm

https://help.sap.com/doc/abapdocu_750_index.htm/7.50/de-DE/abenabap_unit.htm

<https://open.sap.com/courses/wtc1> - Writing Testable Code for ABAP



ISBN 978-0134757599

- <https://refactoring.com/>

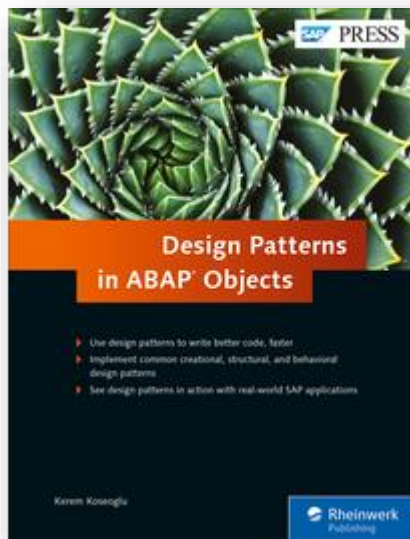
Entwurfsmuster

• Entwurfsmuster

- Sind bewährte Lösungen für wiederkehrende Aufgabenstellungen in der Softwarearchitektur
- Bekannte Arten sind z.B.
 - Erzeugungsmuster (Factory, Singleton, ...)
 - Strukturmuster (Facade)
 - Verhaltensmuster

Erzeugungsmuster (Creational Design Patterns)	Strukturmuster (Structural Design Patterns)	Verhaltensmuster (Behavioral Design Patterns)	Muster für objektrelationale Abbildung	Weitere Muster
abstrakte Fabrik (abstract factory pattern)	Adapter (adapter pattern)	Beobachter (observer pattern)	Data Mapper	Business Delegate
Einzelstück (singleton pattern)	Brücke (bridge pattern)	Besucher (visitor pattern)	Data Access Object	Dependency Injection
Erbauer (builder pattern)	Container (container pattern)	Interceptor (interceptor pattern)	Datentransferobjekt (data transfer object)	Extension Interface
Fabrikmethode (factory method pattern)	Dekorierer (decorator pattern)	Interpreter (interpreter pattern)	Table Data Gateway	Fluent Interface
Multiton (multiton pattern)	Fassade (facade pattern)	Iterator (iterator pattern)	Row Data Gateway	Inversion of Control
Prototyp (prototype pattern)	Fliegengewicht (flyweight pattern)	Kommando (command pattern)	Active Record	Model View Controller (MVC)
	Kompositum (composite pattern oder Whole-Part)	Memento (memento pattern)	Unit of Work	Model View Presenter (MVP)
	Stellvertreter (proxy pattern)	Nullobjekt (Null Object pattern)	Identity Map	Model View ViewModel (MVVM)
	Half Object Plus Protocol	Schablonenmethode (template method pattern)	Lazy Loading	Transaction Script
		Strategie (strategy pattern)	Identity Field	Domain Model
		Vermittler (mediator pattern)	Dependent Mapping	Table Module
		Zustand (state pattern)	Embedded Value	Service Layer
		Zuständigkeitskette (chain of responsibility)	Serialized LOB	Page Controller
		State/Event	Inheritance Mapper	Template View
		Consequences	Metadata Mapping	Transform View
		Accumulator	Query Object	Two-Step View
		MapReduce		Application Controller
		Reduce/Combine		Remote Facade
		Rekursive Erweiterung		Locks
		Software Pipelining		Session States
		Prozess Wrapper		Repository
		Token		Value Object
		Page Object		Registry

- <https://de.wikipedia.org/wiki/Entwurfsmuster>



ISBN 978-1-4932-1465-5

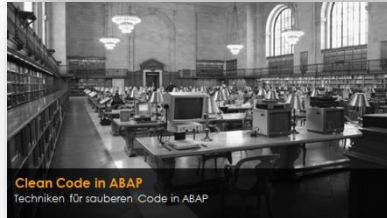
Was noch nicht erwähnt wurde

● Weitere Themen zu Modern ABAP

- Logpoint/Checkpoints
- AMDB / SQL Script / CDS Table Function
- Dynamic Development
- ABAPDoc
- JSON, XML, Transformationen/XSLT
- Debugging
- Pakete
- ABAP Channel, ABAP Daemon
- bgRFC, qRFC
- Regular Expressions
- Gateway / oData
- Agil, Scrum, Kanban, eXtreme Programming
- ...



31. Mai 2019 10:00 – Thema: TBD



<http://www.cadaxo.com/blog/>

See you again!

Thank you for participating!



<https://twitter.com/foessleitnerj>



<https://www.linkedin.com/in/johann-fößleitner-a9851b2a>



https://www.xing.com/profile/johann_foessleitner



johann.foessleitner@cadaxo.com



<https://twitter.com/domibiglsap>



<https://www.linkedin.com/in/dominik-bigl-9b98b68b>



https://www.xing.com/profile/dominik_bigl



dominik.bigl@cadaxo.com