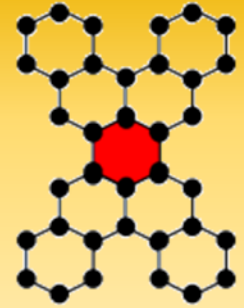


facet.at

by Dr. Roland Schwaiger

- Methoden, Muster und Konzepte
- SAP Development Schulungen
- SAP Development Fachbücher
- SAP Development Projekte
- Akademische Vorlesungen



Enterprise Computing WS 2019/2020

SW-Entwicklung mit SAP-ABAP

sapserver.cosy.sbg.ac.at/00/001
BCUSER
minisap
EN

Dr. Schwaiger Roland

Vorstellung



roland.schwaiger@facet.at

Dr. Roland Schwaiger

Located

Bad Dürrenberg, Hallein, AT

Background

Mathematics (University Salzburg)

Computer Sciences (University Salzburg, Bowling Green State University)

Project & Process Management (SMBS – University of Salzburg Business School)

Profession

SAP Technical Consultant (Cert. SAP Development Consultant)

SAP Trainer

Project Coach (Cert. Scrum Master)

Software Architect

Software Developer (SAP AG, Walldorf, DE and Customer Development Projects)

Author (check out Amazon and/or www.citeseer.com)

Lector (Rheinwerk Verlag)

Lecturer (University Salzburg, FH Salzburg)



Begriffsklärung



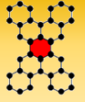
“Enterprise computing involves the development, deployment and maintenance of the information systems required for survival and success in today’s business climate.”

Yen-Ping Shan & Ralph H. Earle,
Enterprise Computing with Objects,
Addison-Wesley, 1998.

Motivation Enterprise Computing

Custom Development in Unternehmen, die SAP einsetzen

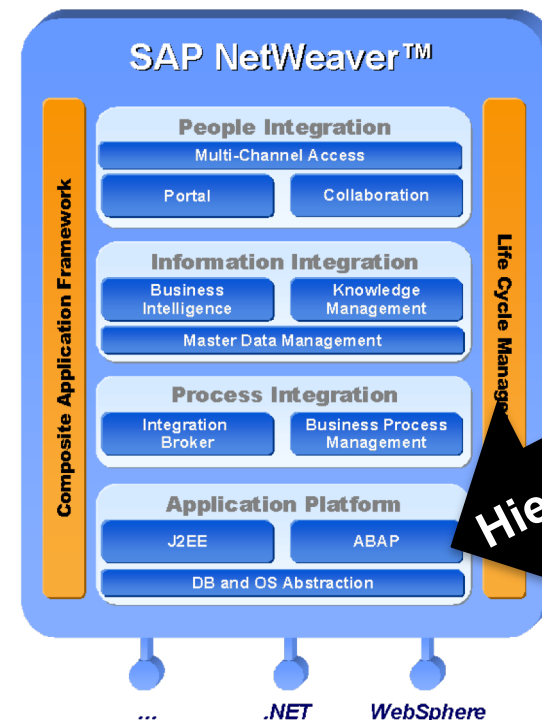
facet.at
Plan-Do-Check-Act



Die zwei Aspekte, die wir betrachten wollen:

1. Das SAP System
2. Das ABAP

und das anhand einer Anwendung zur Segelkursberechnung, die objektorientiert aufgebaut wird.



Hier sind wir

Quelle: [7]

Inhalte/Organisation



Infrastruktur Client-Server, Laufzeitumgebung, Arbeitsprozesse

ABAP Programme Ausführbares Programm, Module Pool

Werkzeuge Object Navigator, Class Builder, ABAP Editor, ABAP Dictionary, [Eclipse, Web IDE, ...]

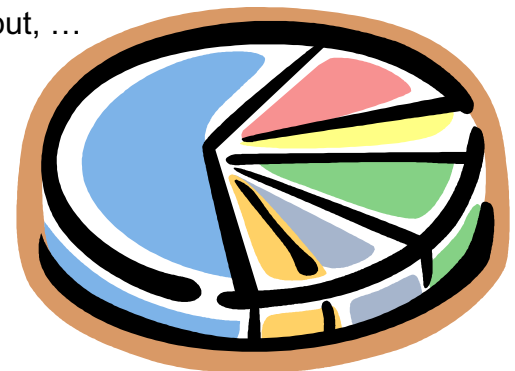
Datentypen und Datenobjekte DATA, TYPES

Modularisierungen Ereignisblock, Unterprogramm, Funktionsbaustein, [REST Services, ...]

ABAP OO Klasse definieren, Objekt instanzieren, Methode, Attribut, ...

GUIs Selektionsbild, Liste, Dynpro, [Web Dynpro, SAP UI5, ...]

DB Tabellen erstellen, auslesen



Abschluss LV



Der Abschluss der LV findet im Rahmen des LV Blocks statt. Dieser besteht aus:

- Implementierung von Repository Objekten im Rahmen der LV
- Abschlussprüfung
 - Theoretische Prüfung mit zehn Fragen aus den Inhalten der LV
 - Entwicklung von Repository Objekten

Einordnung SAP System



- The first version of SAP's **enterprise software** was a **financial Accounting system** named **RF** or better known as **R/1**. (The "R" was for "Real-time data processing")
- This was replaced by **R/2** at the end of the 1970s. **SAP R/2** was a **mainframe based** business application software suite that was very successful in the 1980s and early 1990s
- The client-server based solution **SAP R/3** was officially launched on 6 July 1992. (The 3 stands for the **three layer architecture**)
- In 2008 SAP HANA, an in-memory, column-oriented RDBMS was introduced.

Einordnung

Einordnung SAP



SAP History

- „SAP Systemanalyse und Programmentwicklung“ was founded in 1972 by 5 former IBM engineers
- In 1976 the SAP GmbH was founded, they moved to Walldorf
- In 1988 the SAP Aktiengesellschaft *Systeme, Anwendungen und Produkte in der Datenverarbeitung* was founded

Einordnung SAP History



- SAP is the **largest European** and the third largest software producer world-wide
- SAP has more than **99.700** employees (19.000 software development) (2019)
- Annual Revenue (IFRS) 2018 was **24.7bn** Euros
- SAP products are used by ~ **335.000** customers in 120 subsidiaries.
- SAP has about **12** dev centers worldwide

Motivation

SAP

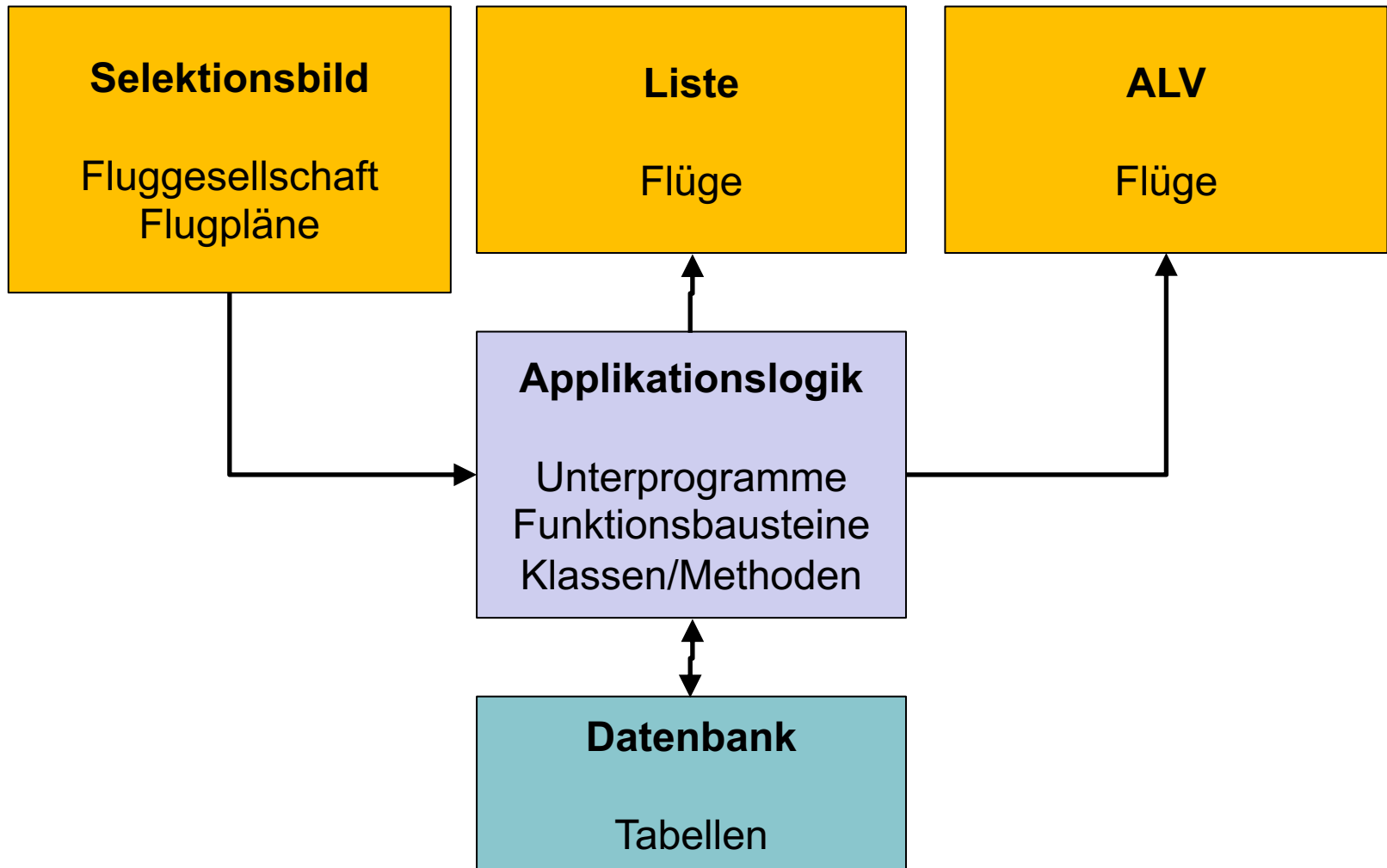
SAP is ...

- the corporation SAP
- the family of products of the SAP, referred to as **THE SAP SYSTEMS**



Unser Beispiel

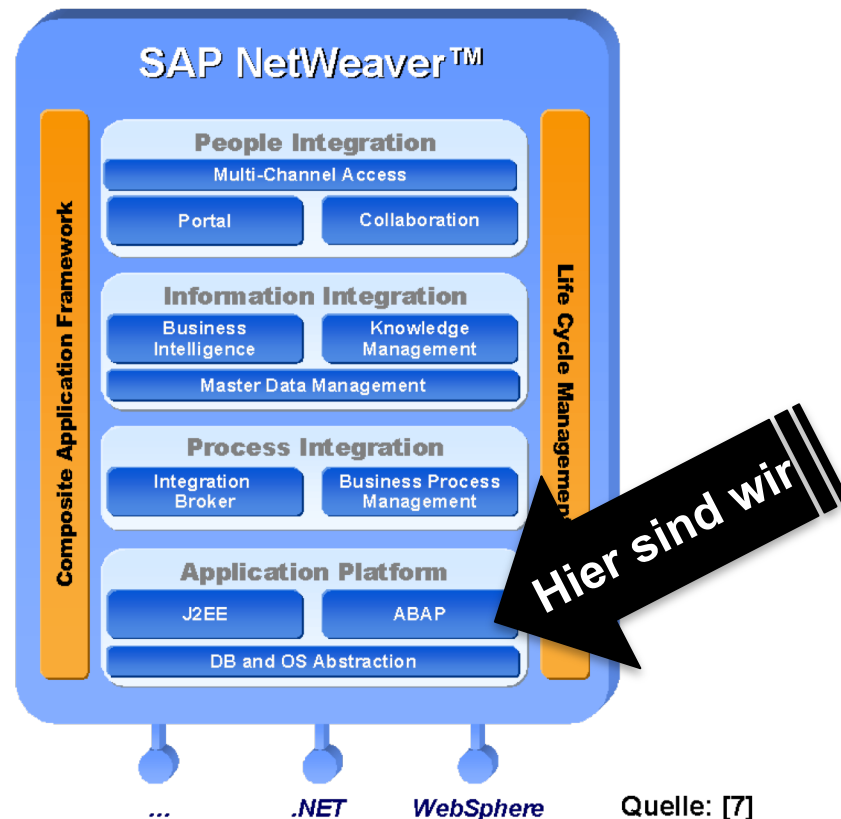
Anwendung ToDev



Begriffe und Handgriffe

Begriffe und Handgriffe

SAP NetWeaver

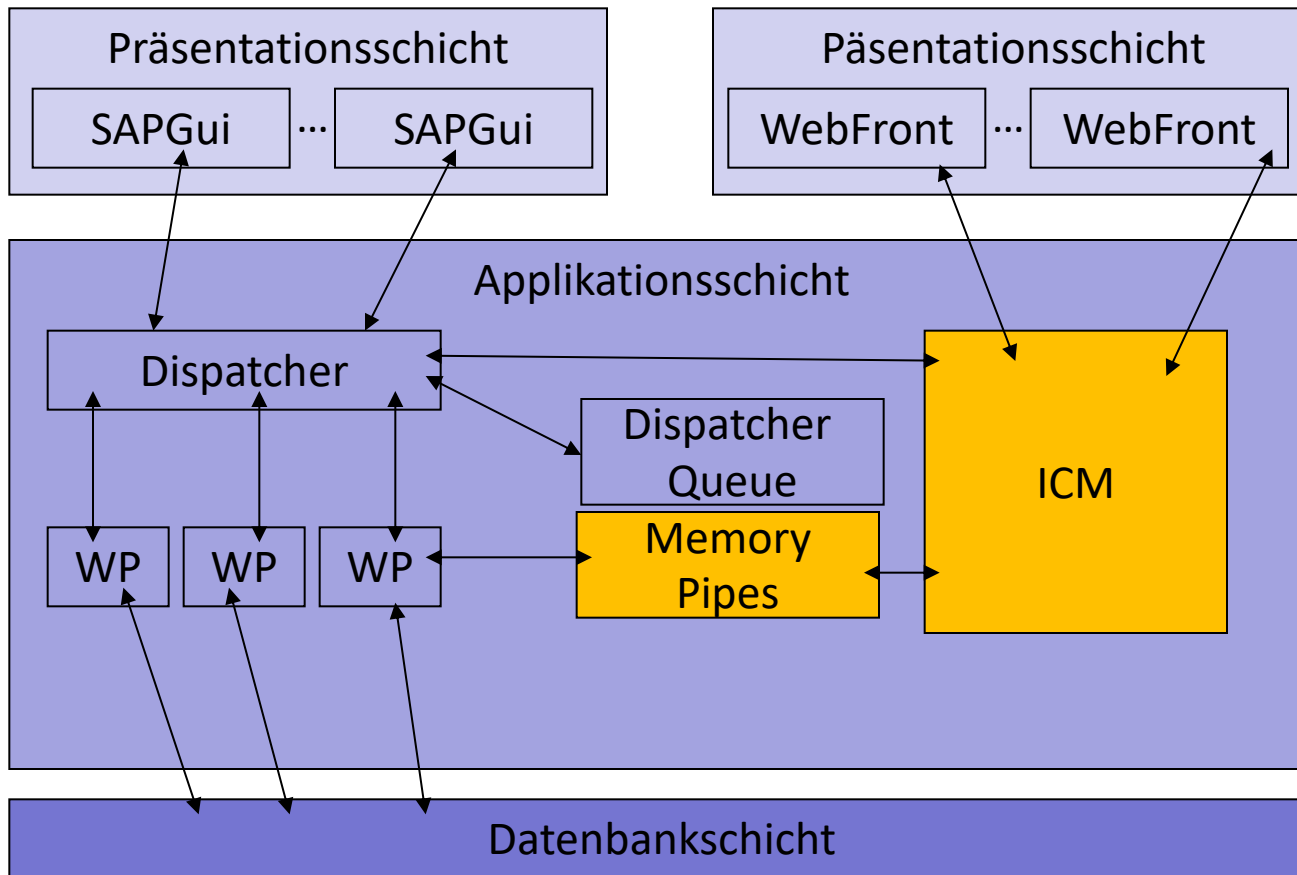


Quelle: [7]

© SAP

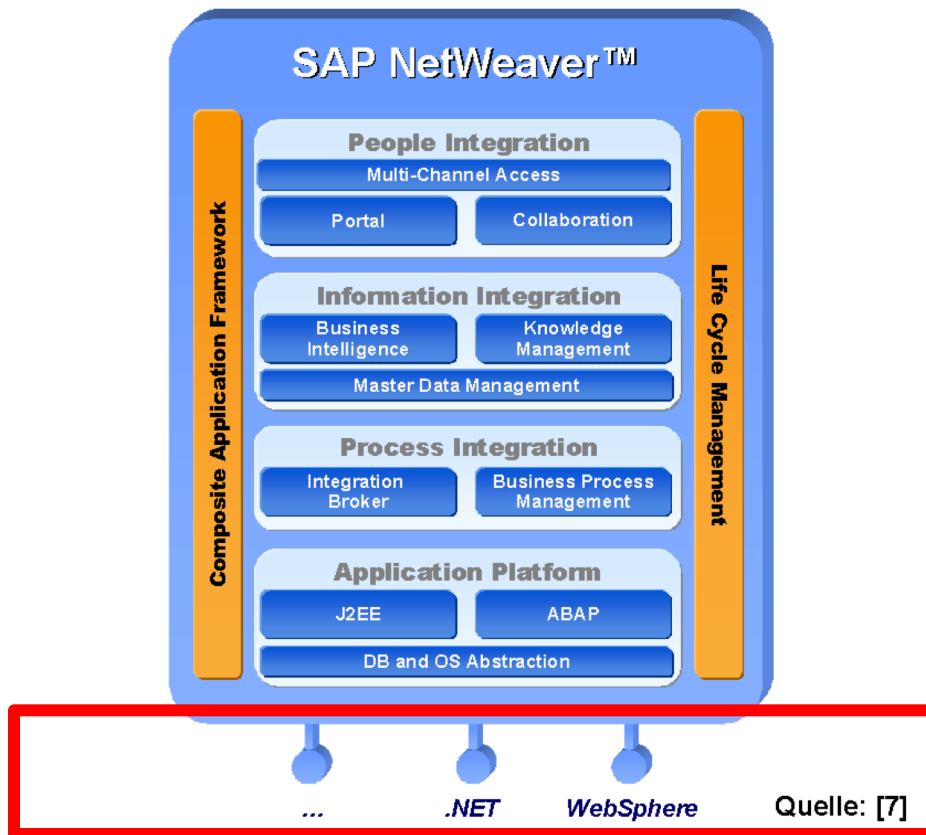
Begriffe und Handgriffe

Application Platform ABAP



Begriffe und Handgriffe

SAP Integrationsmöglichkeiten



UND

- Java Connector (Jco)
- Flash Islands (Flex)
- Silverlight Islands (MS)
 - Web Services
 - Rest Services
 - ...

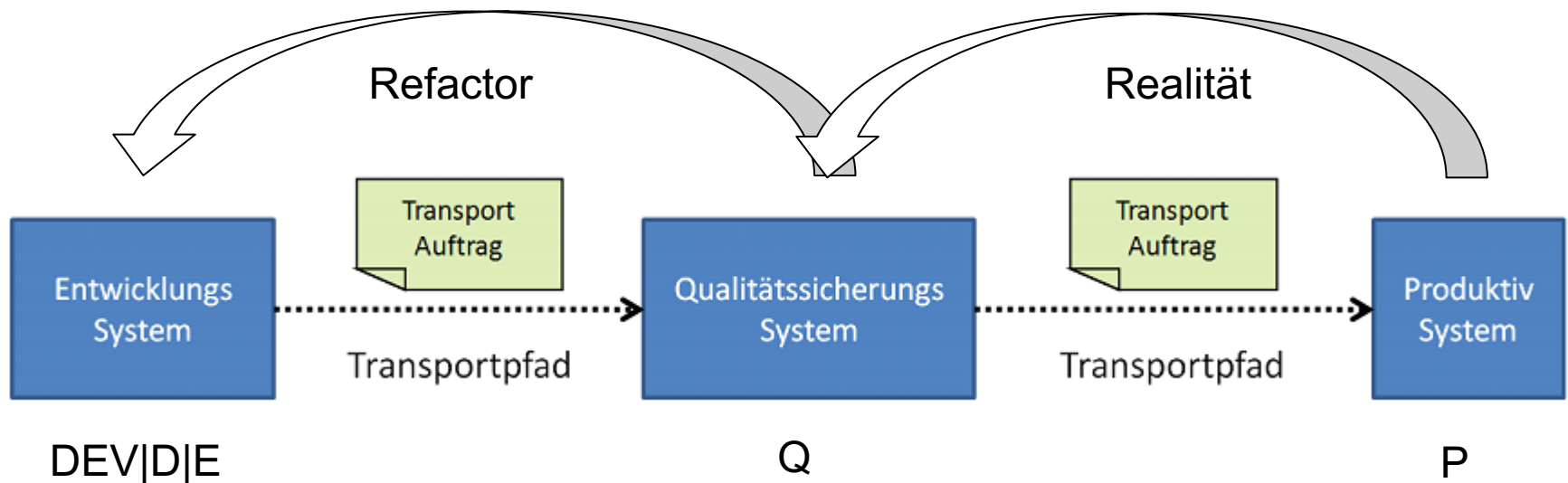
WEITERS

- Operationsmodus Server oder Client

© SAP

Begriffe und Handgriffe

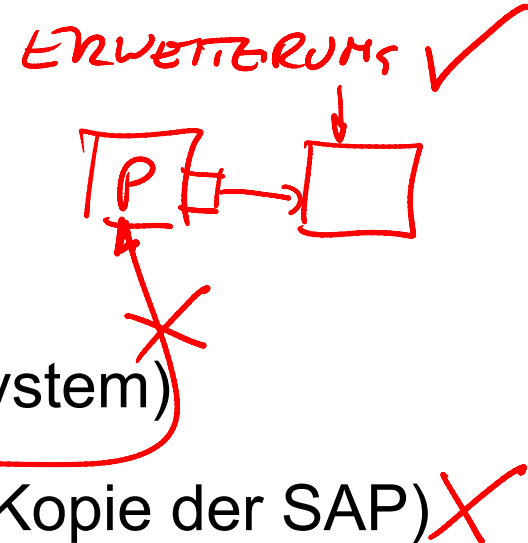
SAP 3-System Landschaft



Begriffe und Handgriffe

SAP Upgrade

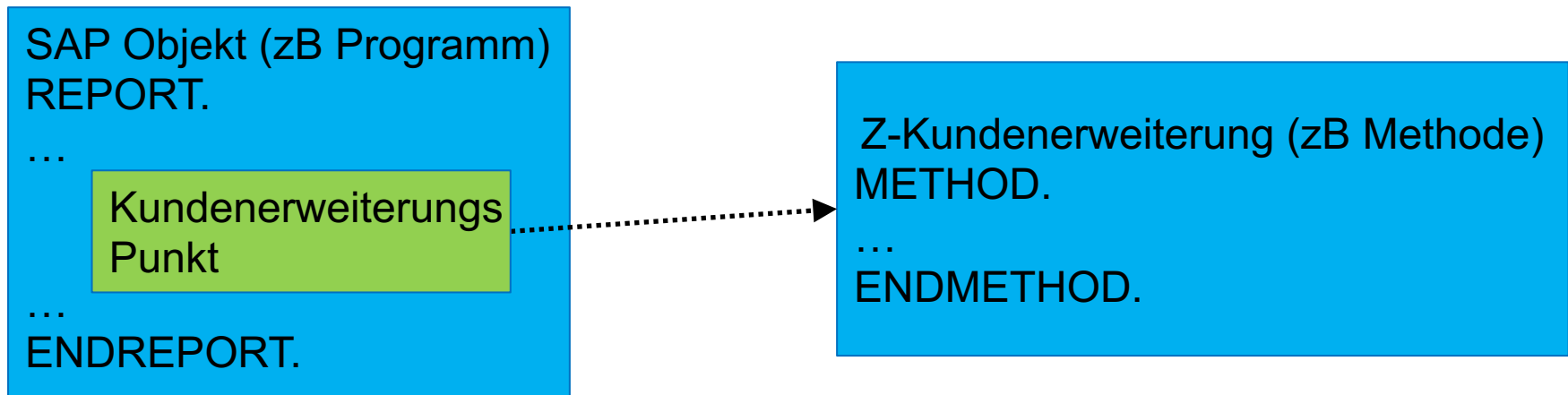
- Original (von SAP oder Kunde)
- Kopie (Ist das Original im Folgesystem)
- Modifikation (Änderung an der Kopie der SAP) ~~X~~
- Modifikationsableich (Abgleich der Modifikationen mit den neuen Auslieferungen der SAP)



Begriffe und Handgriffe

SAP Erweiterbarkeit

- **Prinzip** (SAP sieht einen Absprung für kundeneigene Implementierung vor = Entwicklungsobjekt im Z-Namensraum)
- **Enhancement Framework**



Begriffe und Handgriffe



SAP Standard- vs. Kundenentwicklung

- SAP liefert **Standardfunktionalität**
- Kunden realisieren eigene Prozesse, um einen Wettbewerbsvorteil zu realisieren, d.h. **Kundenentwicklung (Custom Development)**

Begriffe und Handgriffe

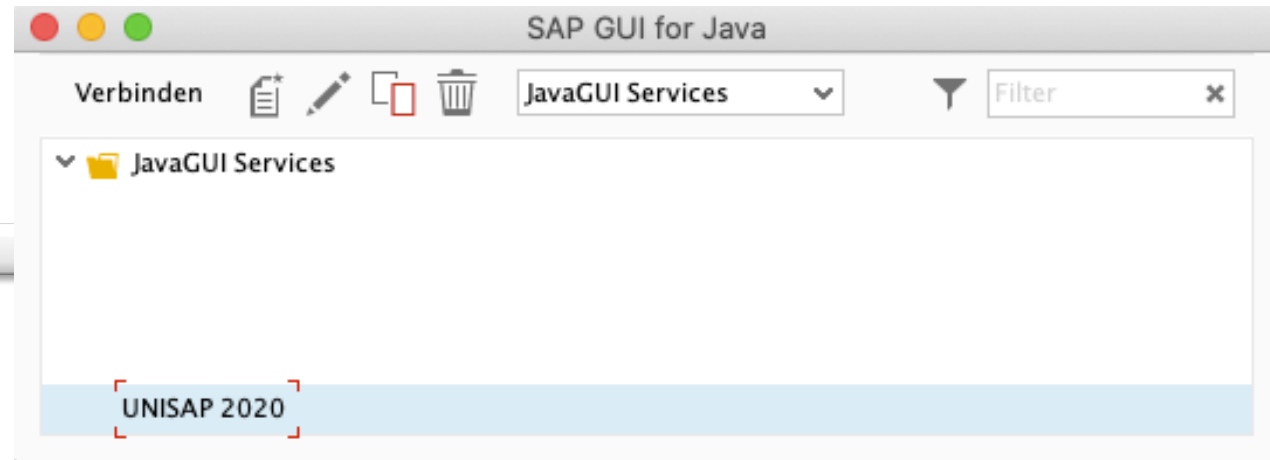
SAP Anmeldung



sdn.sap.com/downloads



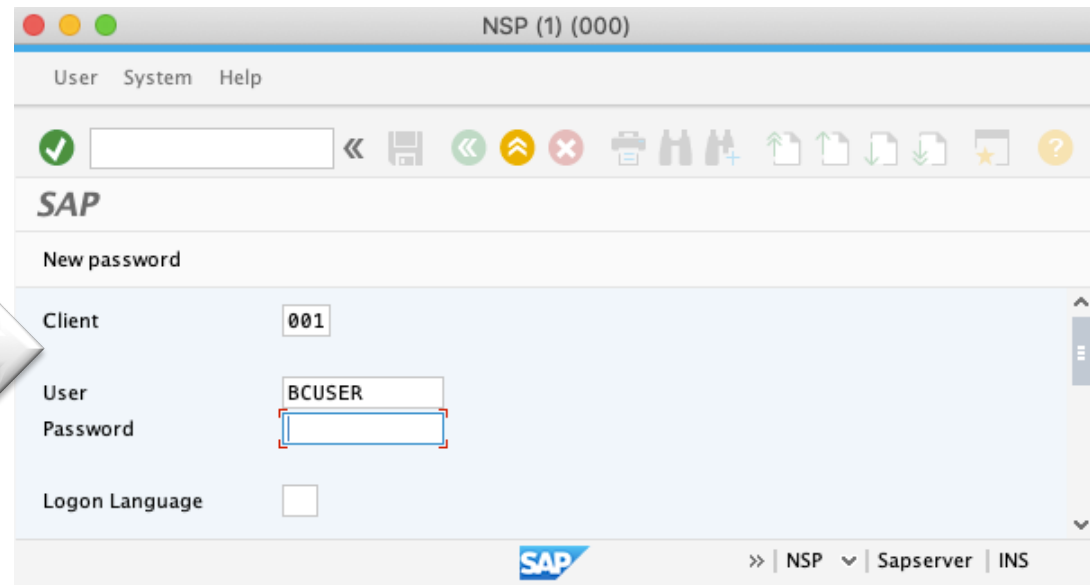
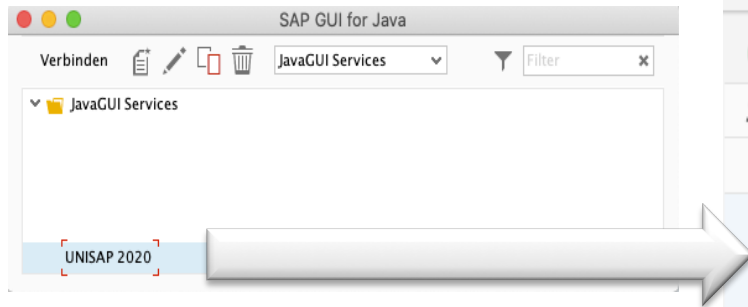
SAPLogon



/nex = Beenden aller Modi

Begriffe und Handgriffe

SAP Anmeldung



Begriffe und Handgriffe

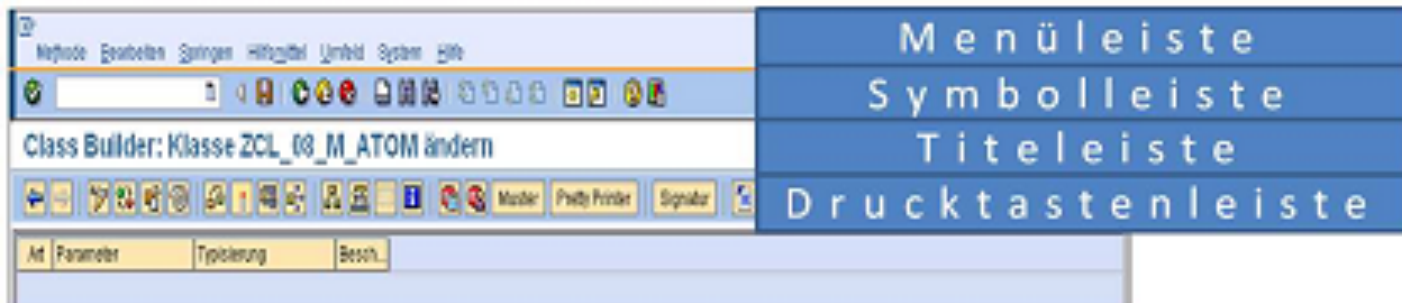


- Anmelden

Begriffe und Handgriffe

SAP GUI Aufbau Die GUI Oberfläche eines SAP

Dynpros besteht aus einer **Menüleiste** (mit variablen und den fixen Menüs System und Hilfe), einer **Symbolleiste** (immer gleich aufgebaut), der **Titelleiste** und der **Drucktastenleiste**. Über **Funktionstastenzuordnungen** können Funktionen direkt ausgewählt werden.



Begriffe und Handgriffe

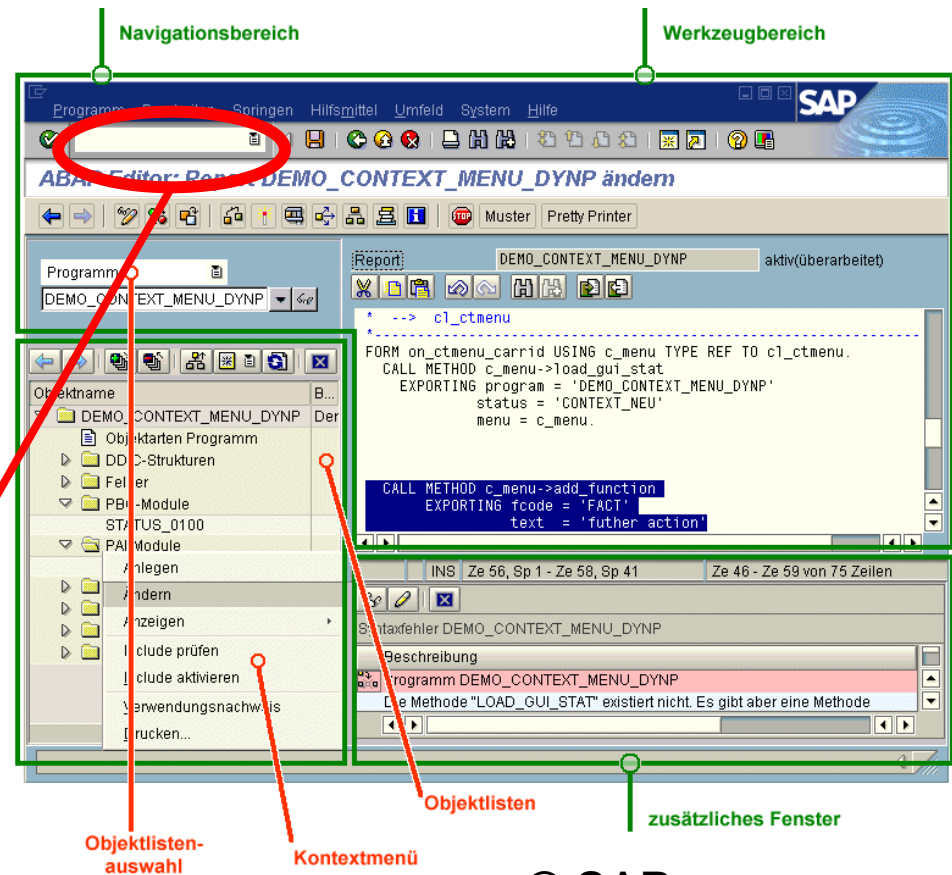
DER Transaktionscode:

SE80

Object Navigator

/n = neuer Start
/o = neues Fenster
/nex = Aussteigen

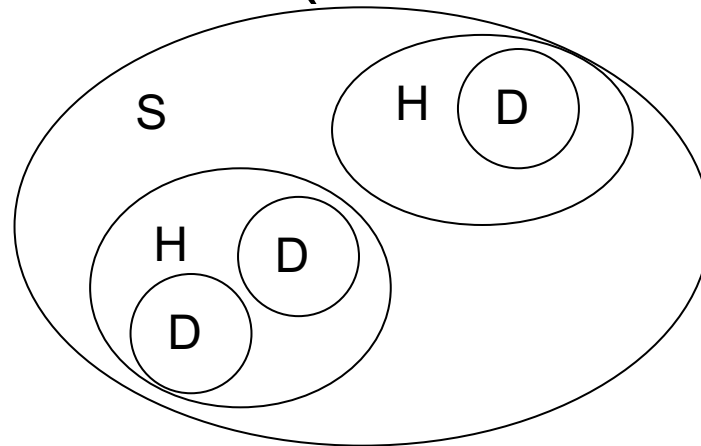
/n + Transaktion,
zB /nse80 = Object Navigator



Begriffe und Handgriffe

Entwicklungsorganisation Pakete (~Java Package)

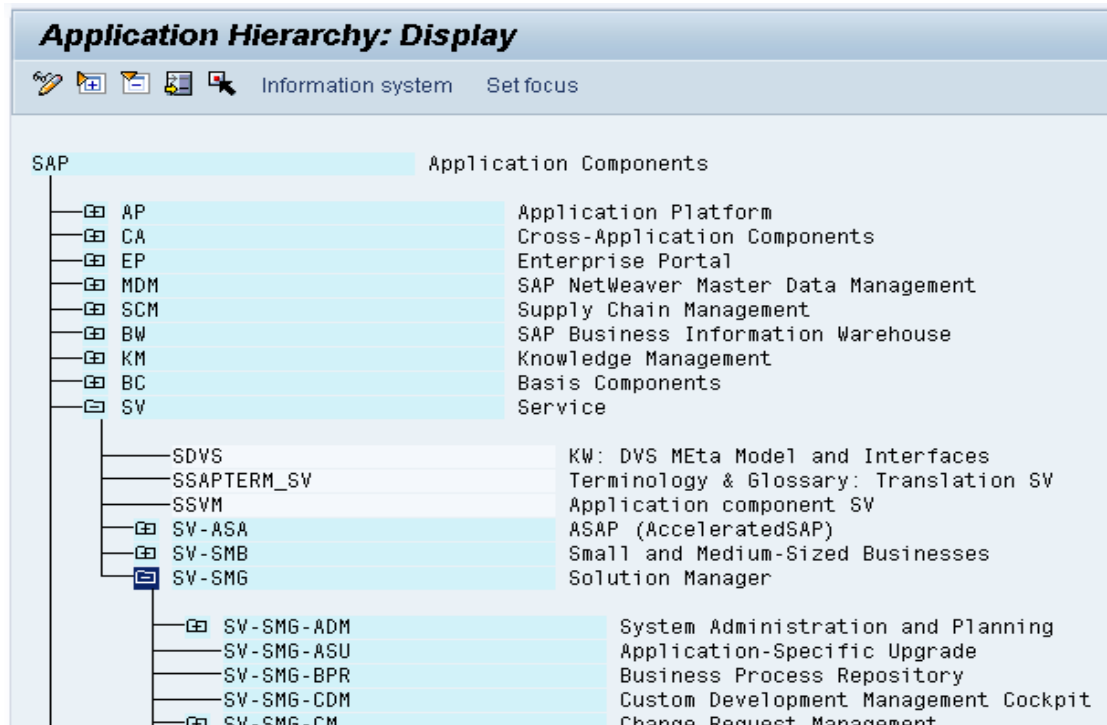
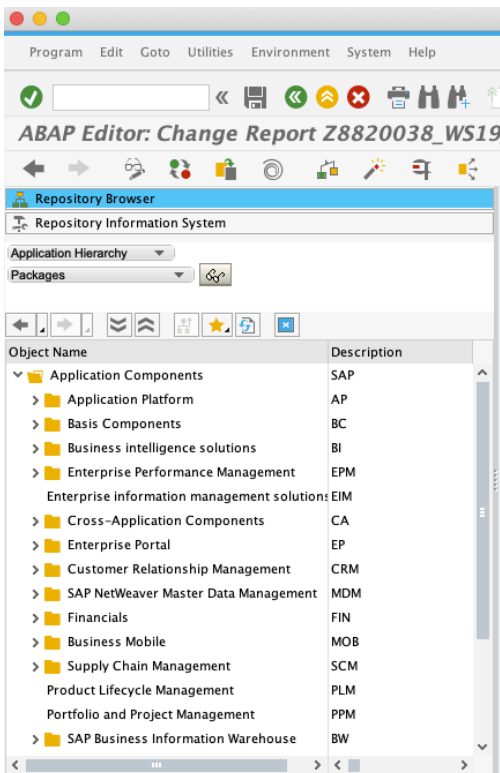
- Strukturpaket (S)
- Hauptpaket (H)
- Standard Paket (“Kein Hauptpaket”) (D)



Begriffe und Handgriffe



Entwicklungsorganisation Anwendungskomponente



Begriffe und Handgriffe



Entwicklungsorganisation

Softwarekomponente

Diese beschreibt eine Menge von **Entwicklungsobjekten**, die nur **gemeinsam auslieferbar** sind.

Ausnahme: Pakete, die nicht an Kunden ausgeliefert werden sollen, müssen abweichend davon der Softwarekomponente **HOME** zugeordnet werden.

Begriffe und Handgriffe

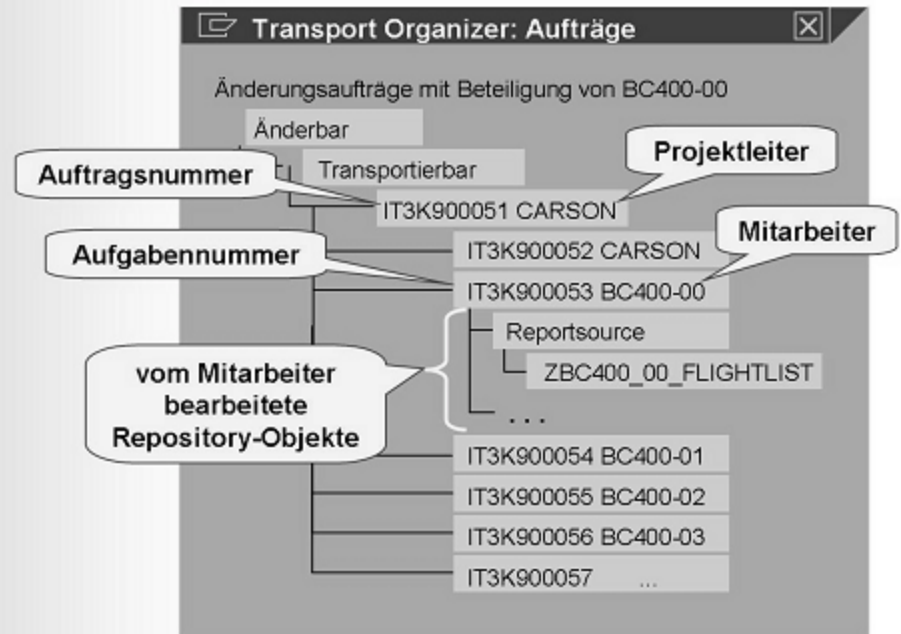
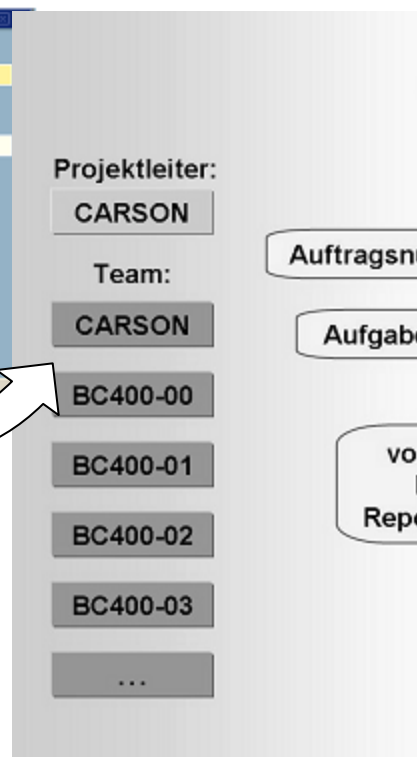
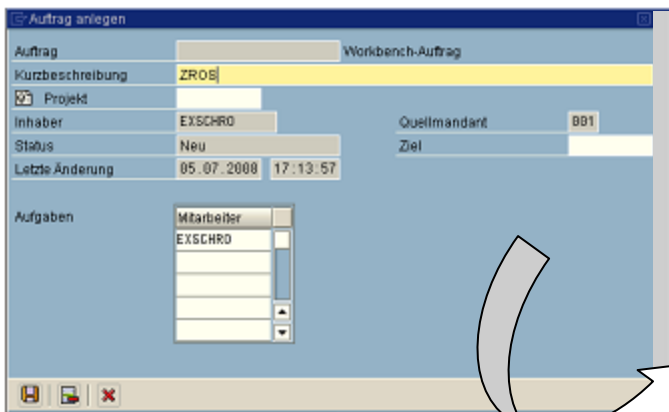


Entwicklungsorganisation Transportschicht

Alle Entwicklungsprojekte, die in einem SAP-System durchgeführt und auf denselben **Transportwegen** transportiert werden, werden zu einer **Transportschicht** zusammengefasst.

Begriffe und Handgriffe

Entwicklungsorganisation **Änderungsauftrag**



Begriffe und Handgriffe

Entwicklungsorganisation: Paket anlegen bzw. Lokales Paket \$TMP

The screenshot shows a dialog box titled "Package Builder: Paket anlegen". It contains several input fields and a checkbox:

Paket	ZROS
Kurzbeschreibung	ROS
Anwendungskomponente	ca
Softwarekomponente	HOME
Transportschicht	<input type="text" value=""/>
Hauptpaket	<input type="checkbox"/>

At the bottom of the dialog, there are three icons: a document icon, a red 'X' icon, and an information icon.

Begriffe und Handgriffe

Namensräume

SAP liefert eigene Softwareentwicklungen an den Kunden aus, die dieser in seine Systeme übernimmt. Der Kunde kann eigene Softwareentwicklungen durchführen. Prinzipiell kann es zu **Namensüberschneidungen** zwischen **SAP Objekten** und **Kunden Objekten** kommen.

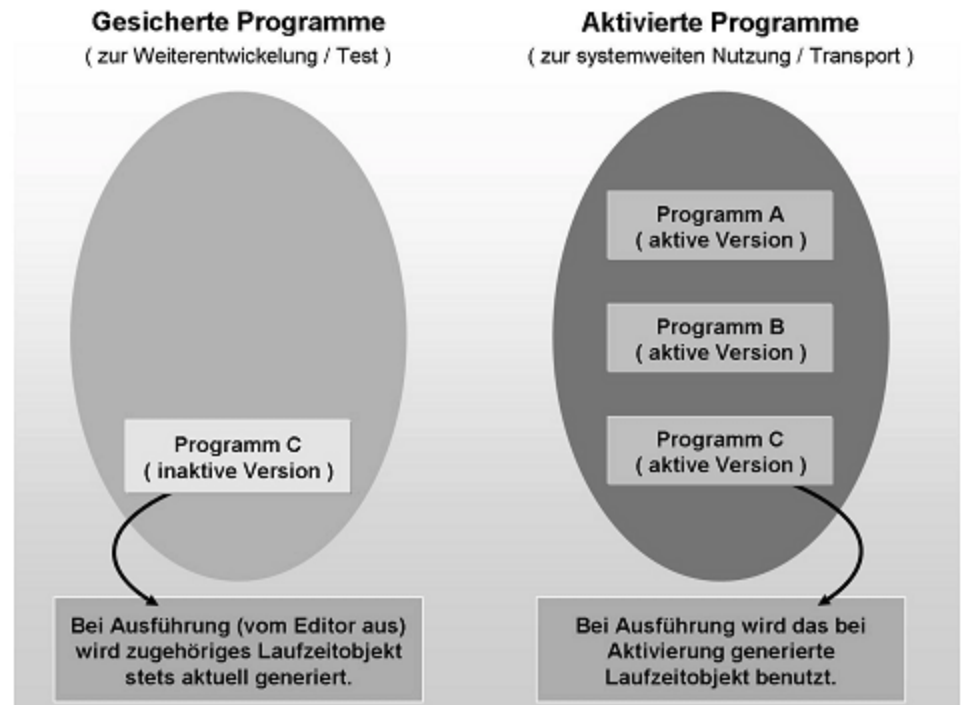
Es gibt **zwei Varianten** (Namensräume), um Kundenobjekte zu schützen.

1. Wenn Entwicklungsobjekte mit **Z oder Y beginnen**, dann sind diese implizit geschützt, da SAP keine Objekte mit Z oder Y beginnend ausliefert (Ausnahme: Kunden-Exits)
2. Der Kunde kann bei SAP einen weltweit eindeutigen **Namensraum** beantragen. Dieser ist 10stellig und beginnt mit / und endet mit /. Dieser Namensraum muss den Entwicklungsobjekten vorangestellt werden. Diese Variante ist speziell für Drittanbieter interessant.



Begriffe und Handgriffe

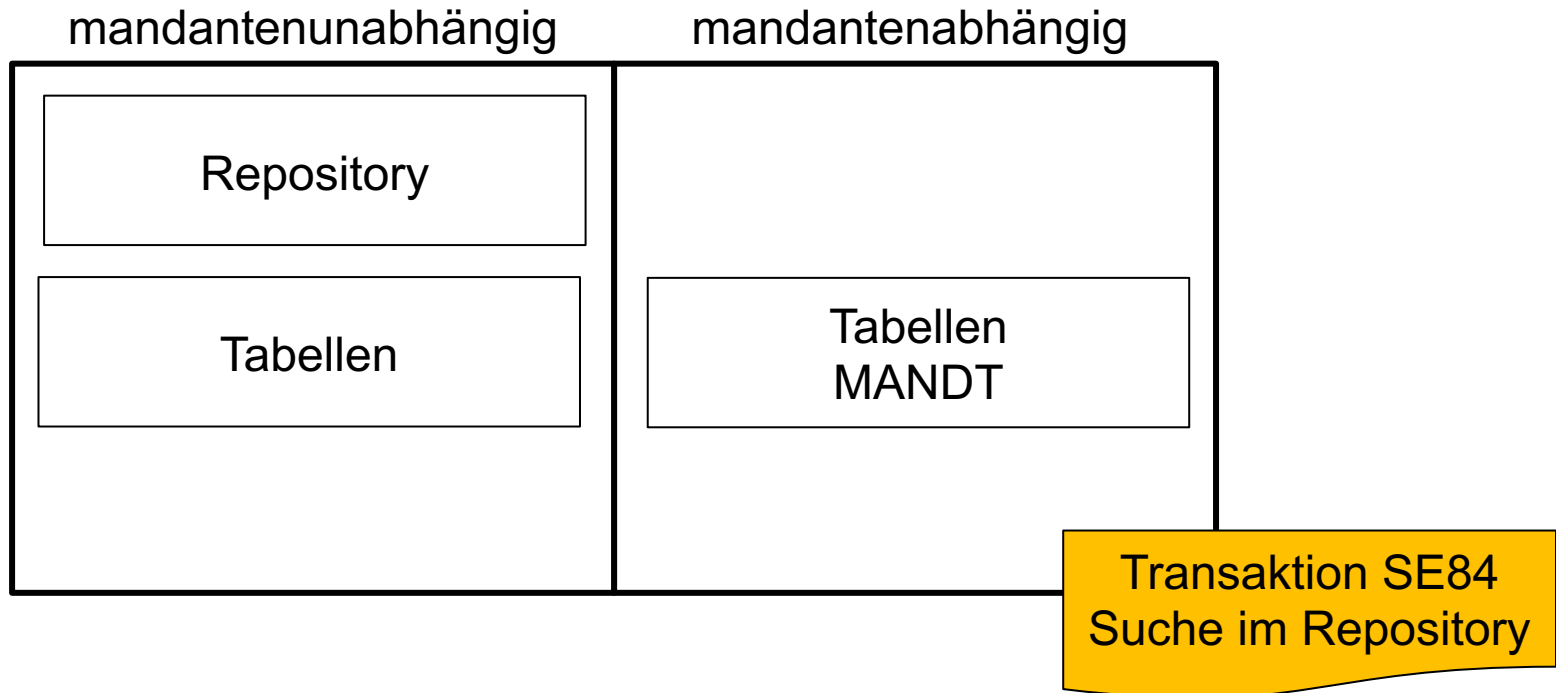
- **Repository Objekte**
 - Aktivitätszustände
 - Programmarten
 - Funktionsbausteine
 - ...
- **Dictionary Objekte**
 - Aktivitätszustände
 - Domäne
 - Datenelement
 - Struktur
 - Transparente Tabelle



Begriffe und Handgriffe

- **Repository Objekte (Aufbau Repository)**

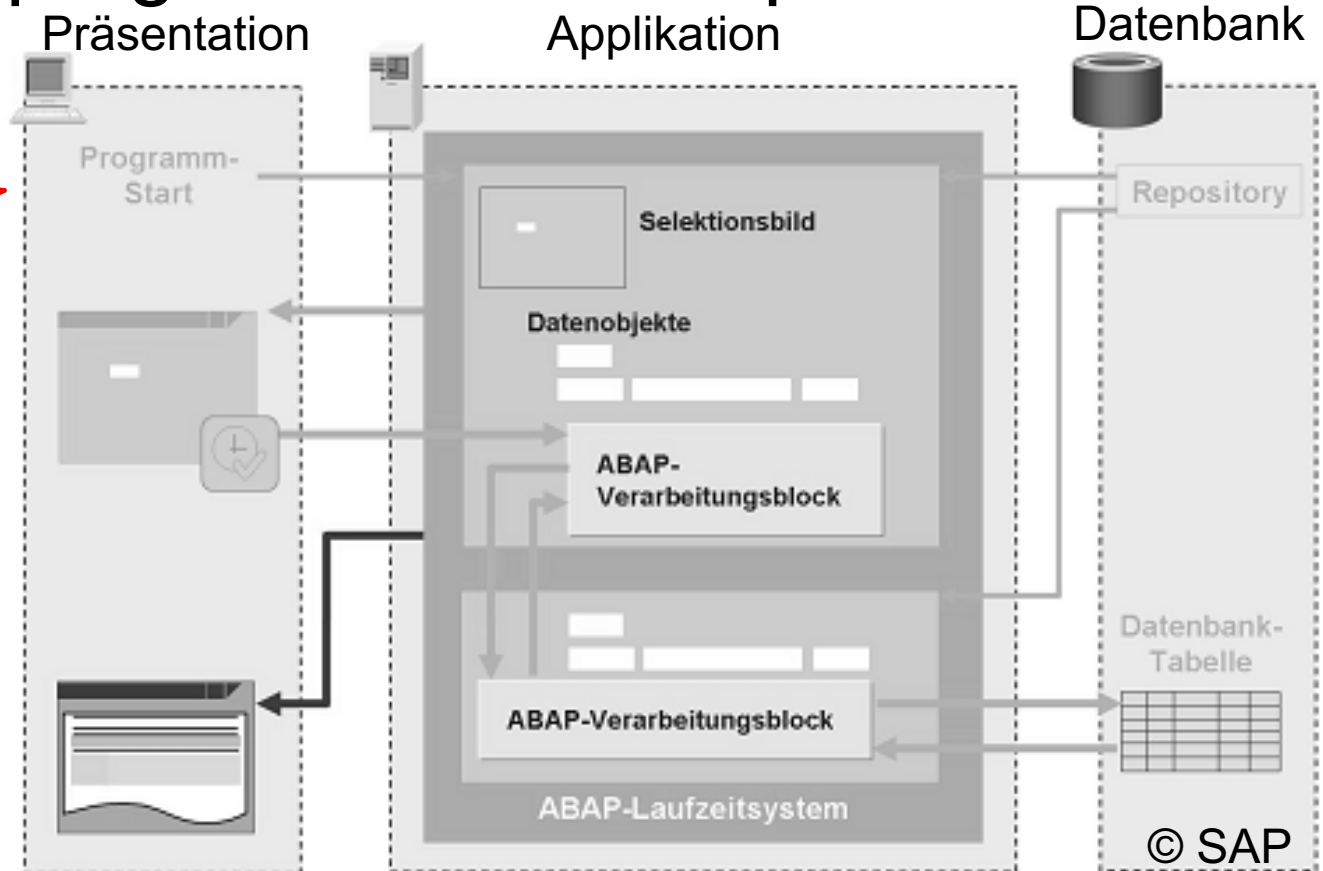
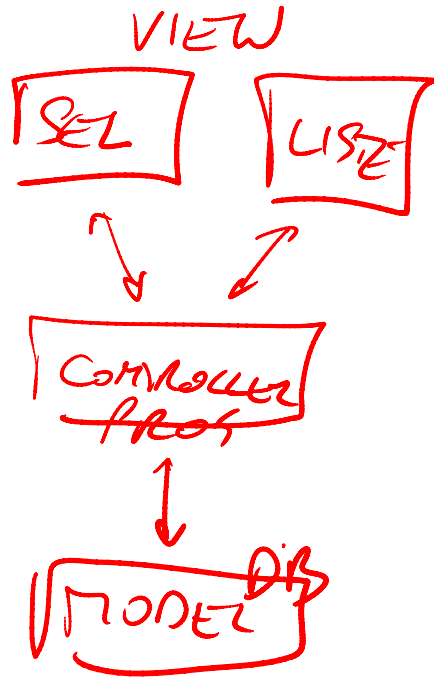
Datenbank



Begriffe und Handgriffe

- Beispielprogramm mit exemplarischen

Ablauf



Beispiel



```
*&-----*  
*& Report Z8820038_WS19_HELLO_WORLD  
*&  
*&-----*  
*&  
*&  
*&-----*
```

REPORT z8820038_ws19_hello_world.

* Eingabefeld für Fluggesellschaft
PARAMETERS: pa_car TYPE scarr-carrid.

* Ziel Datenobjekt für Name der FG
DATA gd_carrname TYPE scarr-carrname.

INITIALIZATION.

* Initialisierung für Eingabefeld
pa_car = 'LH'.

LOAD-OF-PROGRAM.

START-OF-SELECTION.

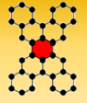
* Lesen der FG von der DB Tabelle SCARR
SELECT SINGLE carrname FROM scarr INTO gd_carrname
WHERE carrid = pa_car.

END-OF-SELECTION.

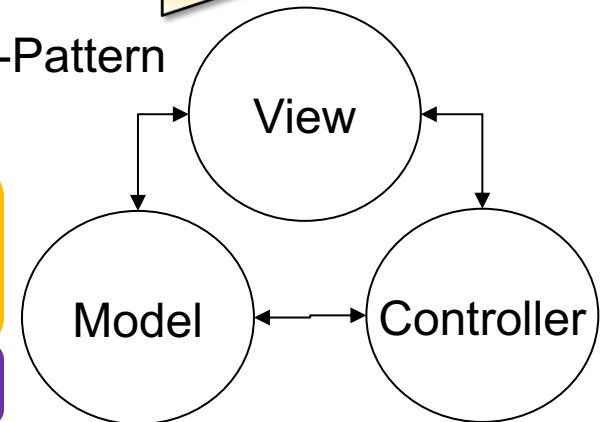
* Ausgabe auf Standardliste
WRITE 'Hello World!'.
* Neue Zeile
new-line.

* Ausgabe des Names der FG
WRITE: / 'Name der FG: ', gd_carrname.

Beispielprogramm



MVC-Pattern



REPORT zwatjab

```
PARAMETERS: pa_carr TYPE bapisfdeta-carrid default 'LH',
pa_cofro TYPE bapisfdeta-countryfr default 'US',
pa_cifro TYPE bapisfdeta-cityfrom default 'NEW YORK',
pa_coto TYPE bapisfdeta-countryto default 'DE',
pa_cito TYPE bapisfdeta-cityto default 'FRANKFURT',
pa_after TYPE bapi_aux-afternoon AS CHECKBOX.
```

Selektionsbild = View

```
DATA: it_bapisflist TYPE TABLE OF bapisflist,
wa_bapisflist LIKE LINE OF it_bapisflist.
```

Geschäftsdaten = Modell

START-OF-SELECTION.

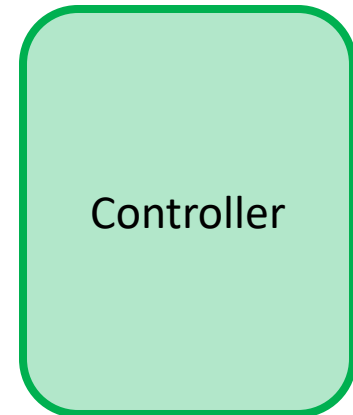
```
CALL FUNCTION 'BAPI_SFLIGHT_GETLIST'
EXPORTING
  fromcountrykey = pa_cofro
  fromcity       = pa_cifro
  tocountrykey   = pa_coto
  tocity         = pa_cito
  airlinecarrier = pa_carr
  afternoon      = pa_after
* MAXREAD       = 0
* IMPORTING
* RETURN        =
TABLES
  flightlist     = it_bapisflist
```

Geschäftslogik = Modell

END-OF-SELECTION.

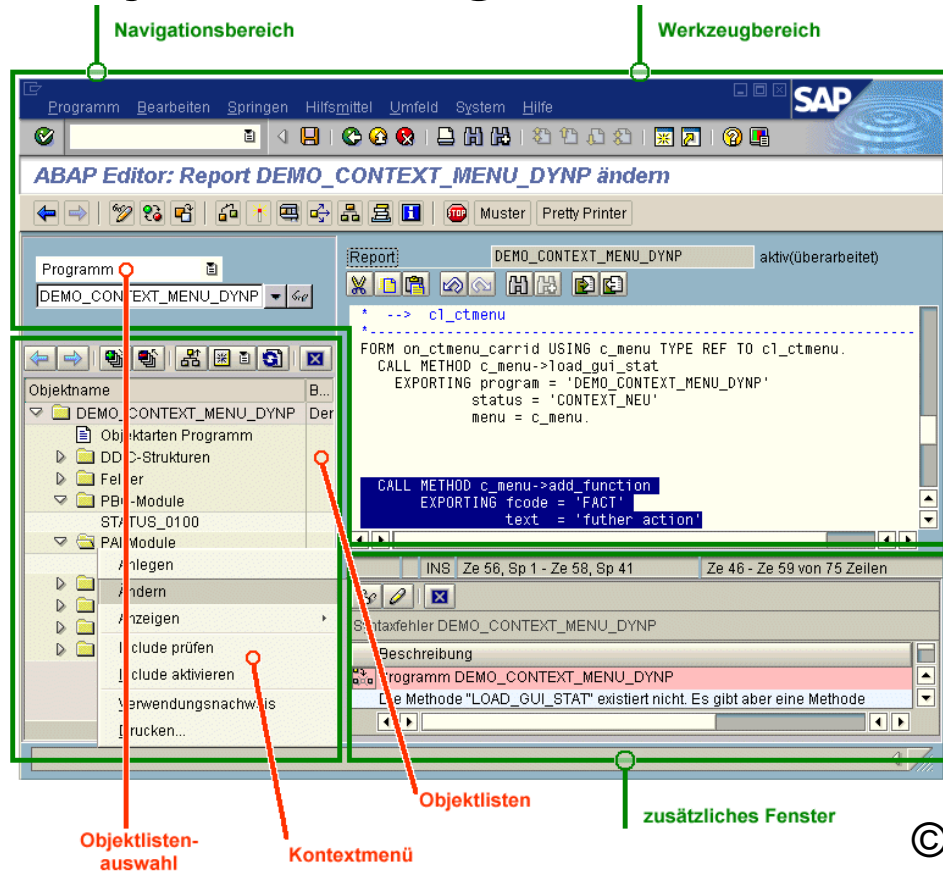
```
LOOP AT it_bapisflist INTO wa_bapisflist.
  WRITE: / wa_bapisflist-carrid,
         wa_bapisflist-connid,
         wa_bapisflist-fldate,
         wa_bapisflist-airpfrom,
         wa_bapisflist-airpto,
         wa_bapisflist-deptime,
         wa_bapisflist-seatsmax,
         wa_bapisflist-seatsocc.
ENDLOOP.
```

Ergebnisliste = View



Motivation SAP

- Object Navigator und Werkzeuge



- ABAP Editor
- Debugger
- ABAP Dictionary
- Screen Painter
- Menu Painter
- Function Builder
- Class Builder

...

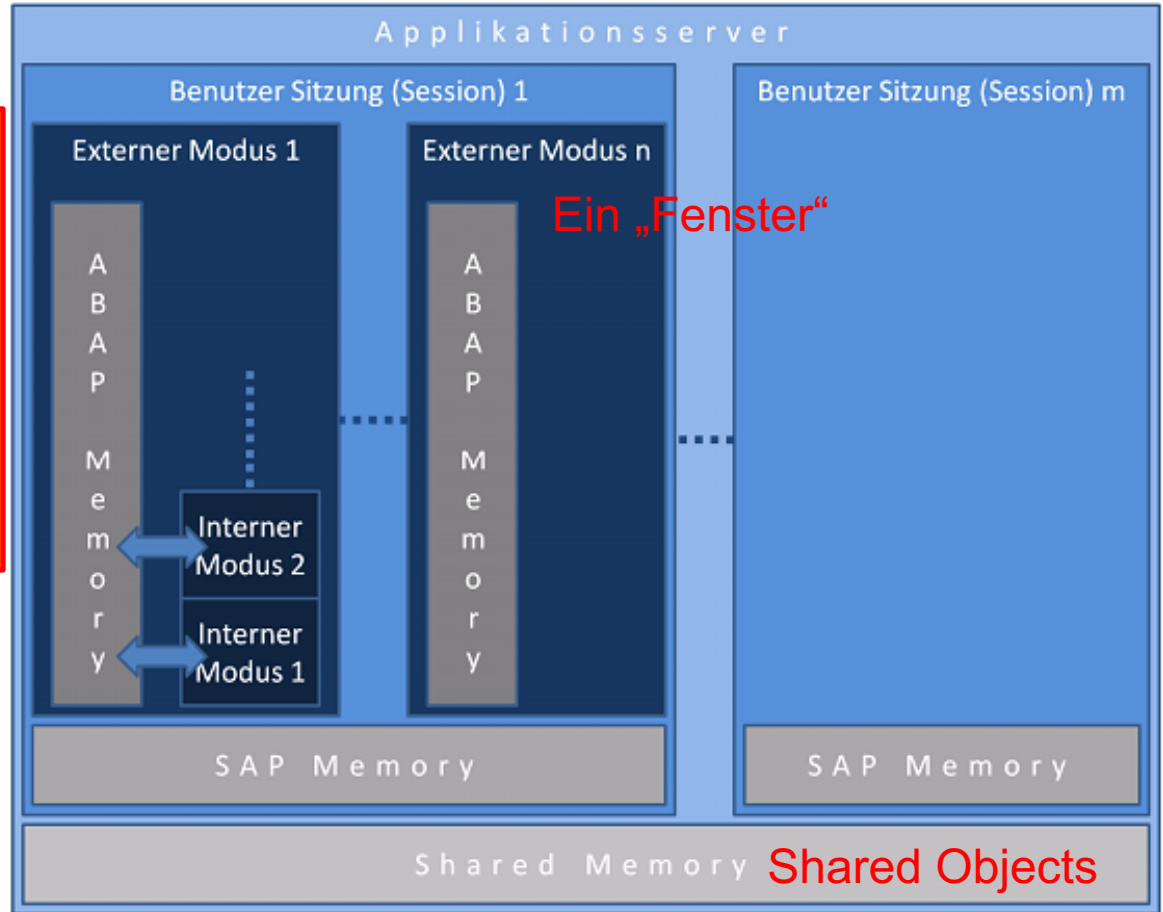
© SAP

Motivation SAP

Speichermodell

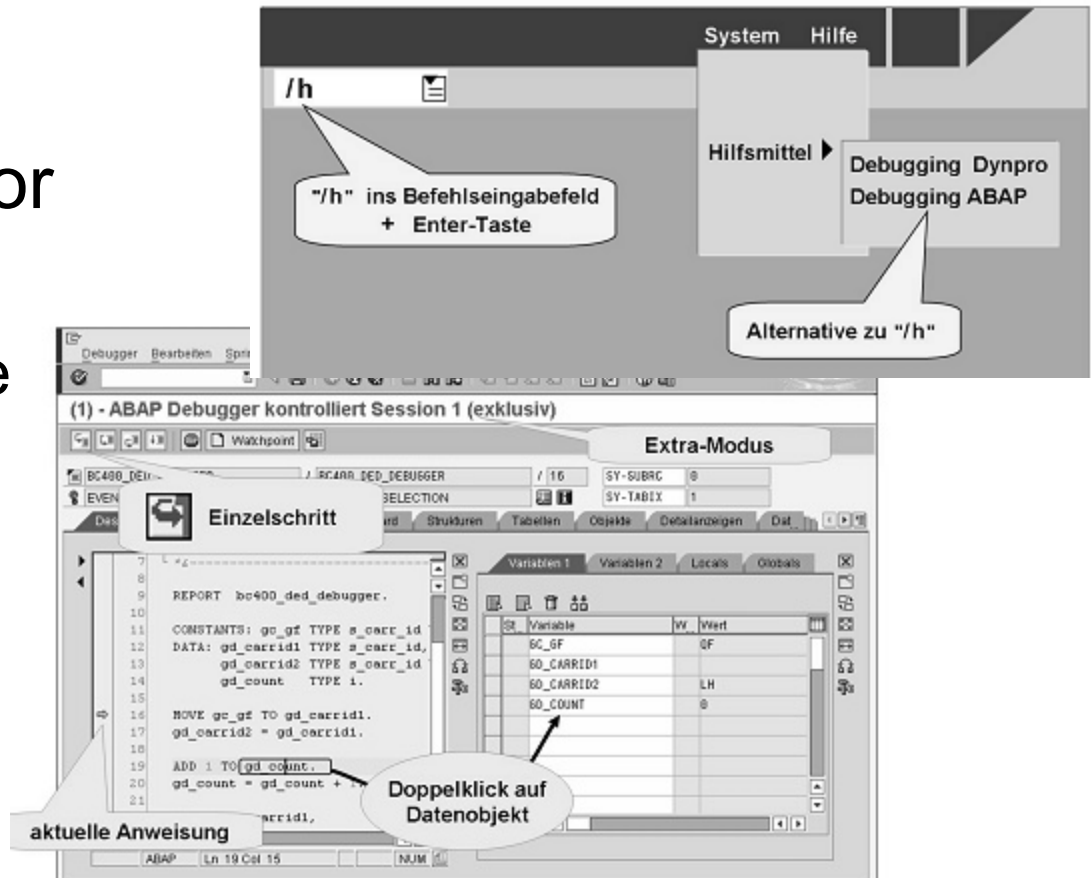
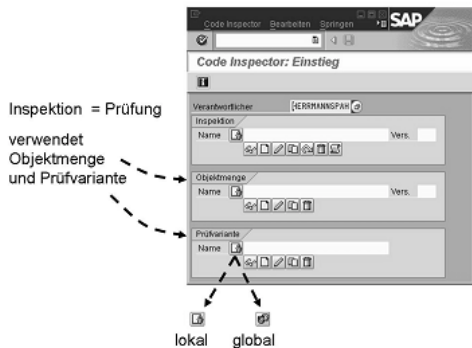
/n + Transaktion = zurücksetzen vom aktuellen externen Modus und Programmstart

/o + Transaktion = neuer externer Modus und Programmstart



Statische vs. Dynamische Analyse

- **Debugger**
- Code Inspector
 - Prüfvariante
 - Objektmenge
 - Inspektion

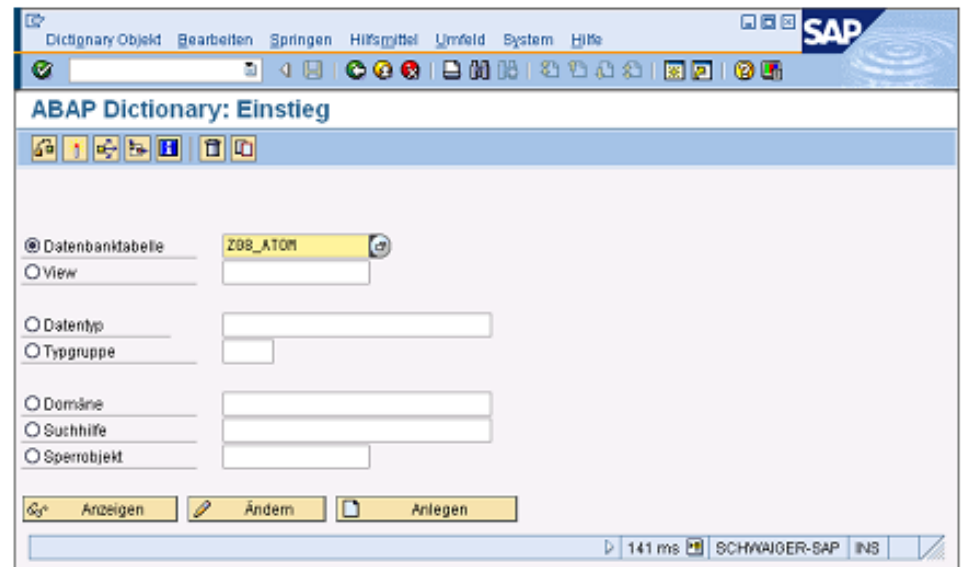


DB Dictionary

Dictionary

ABAP Dictionary: Transparente Tabelle

- Motivation: Dient zur Definition einer DB-Tabelle
- Anlegen: TA SE11
- Indizes
 - Primär
 - Sekundär
- Technische Einstellungen



Dictionary

ABAP Dictionary: Transparente Tabelle

- Spaltendefinition:
 - Name der Spalte
 - Teil des Primärschlüssels?
 - Initialisierung
 - Typisierung

Transp. Table: ZSCARREVAL (Active)
Short Description: Bewertung der FG

Attributes | Delivery and Maintenance | **Fields** | Entry help/check | Currency/Quantity Fields

Field	Key	Initi..	Data element	Data Type	Length	Decim..	Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0	Client
CARRID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	S_CARR_ID	CHAR	3	0	Airline Code
EVAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Z8820038_EVAL	INT1	3	0	Rating

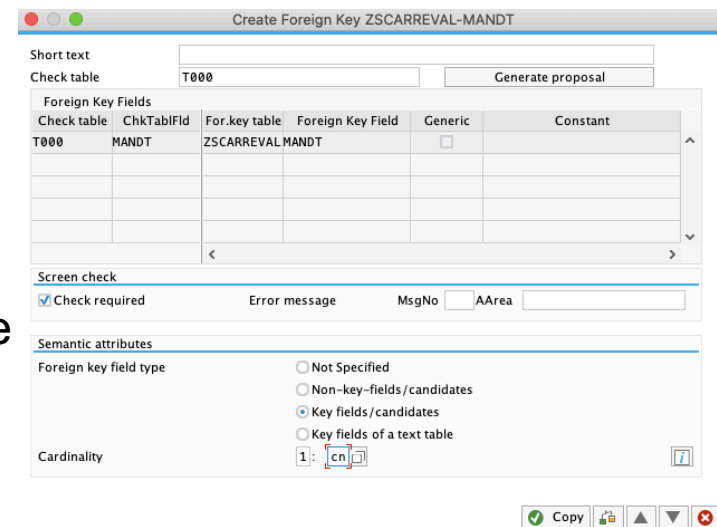
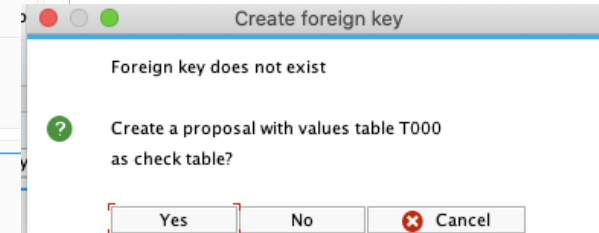
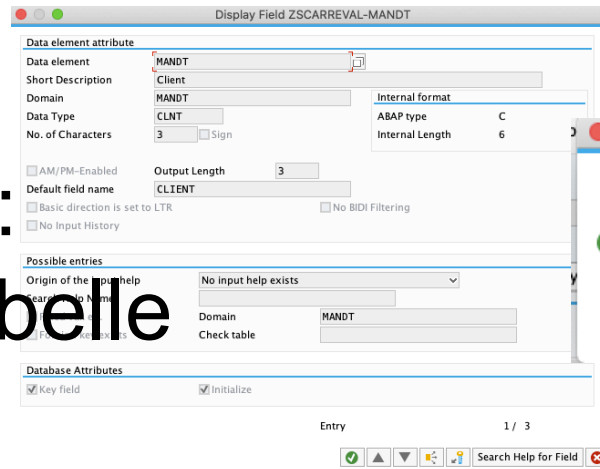
Dictionary

ABAP Dictionary: Transparente Tabelle

Durch die Pflege einer

Fremdschlüsselverprobung wird **nicht** referentielle Integrität auf Datenbankebene definiert.

Die Definition dieser Verprobung hat nur dann eine Auswirkung, sofern ein Dynpro- oder ein Web Dynpro-Feld definiert wird, dass einen Bezug zu dem Fremdschlüsselfeld herstellt.



ABAP Dictionary: Transparente Tabelle

Technische Einstellungen & Erweiterbarkeit

The screenshot shows the 'Maintain Technical Settings' window for a transparent table named 'ZSCARREVAL'. The window title is 'NSP (1) (00)'. The main area contains the following fields and options:

- Name:** ZSCARREVAL (Transparent Table)
- Short text:** Bewertung der FG
- Last Change:** BCUSER, 10.01.2020
- Status:** Active, Saved
- Logical storage parameters:**
 - Data class:** APPL0 (Master Data, Transparent Tables)
 - Size category:** 0 (Data records expected: 0 to 43.000)
- Buffering:**
 - Buffering not allowed
 - Buffering allowed but switched off
 - Buffering switched on
- Buffering type:**
 - Single records buff.
 - Generic Area Buffered (No. of key fields:)
 - Fully Buffered
- Log data changes
- Write access only with JAVA

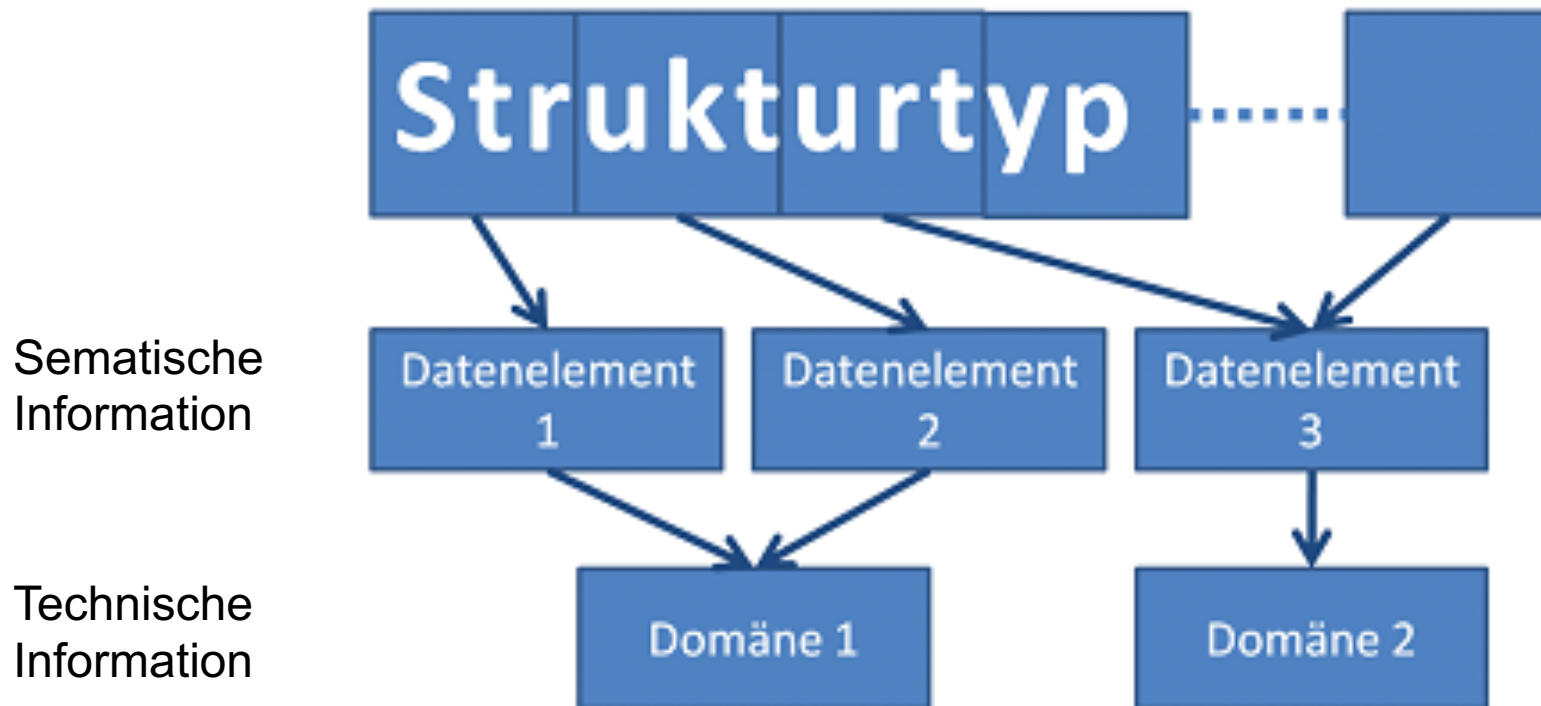
The screenshot shows the 'Erweiterungskategorie für Z08_ATOM pflegen' dialog box. It contains the following options for the 'Erweiterungskategorie der Struktur':

- beliebig erweiterbar
- erweiterbar und zeichenartig oder numerisch
- erweiterbar und zeichenartig
- nicht erweiterbar
- nicht klassifiziert

At the bottom, there are buttons for 'Übernehmen', 'F1', and 'X'.

Dictionary

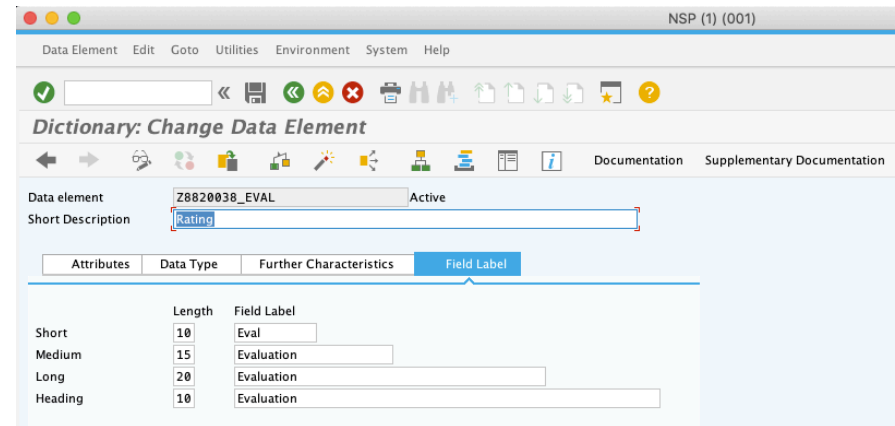
Zweischichtiger Domänenansatz im ABAP Dictionary



Dictionary

ABAP Dictionary: Datenelement

- Motivation: Definition von globalen skalaren Typen im ABAP Dictionary (betriebswirtschaftliche oder auch semantische Ebene)
- Anlegen: TA SE11
- Verwendung:

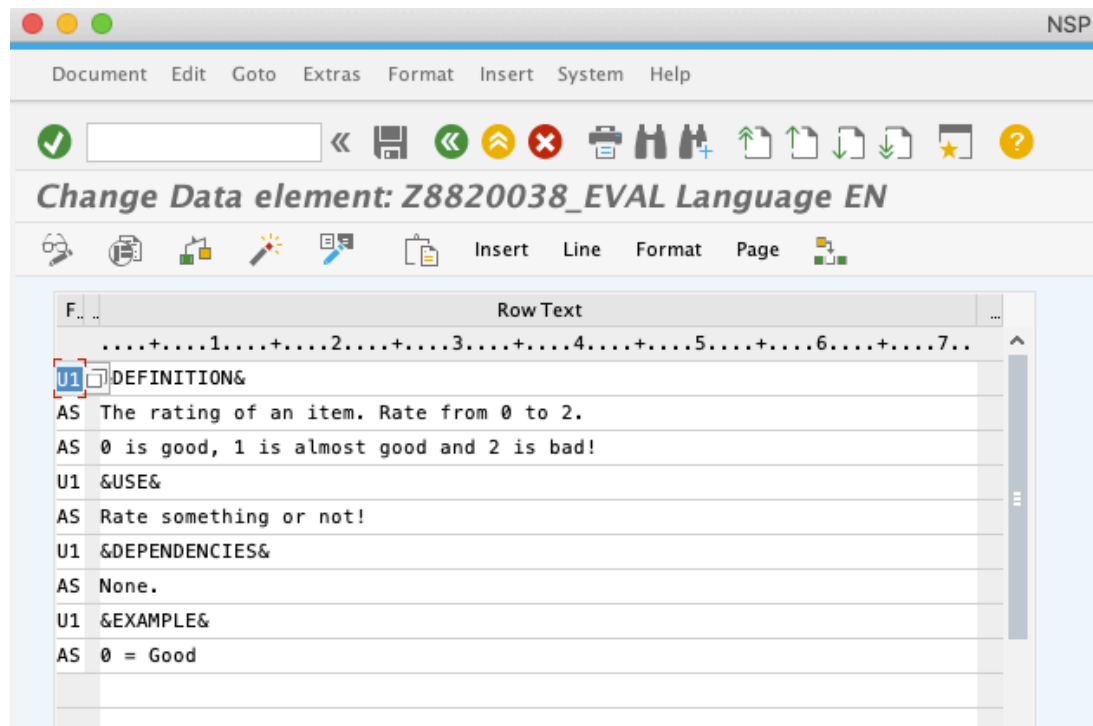


Typisierung von Variablen, Schnittstellen, ...

Dictionary

ABAP Dictionary: Datenelement

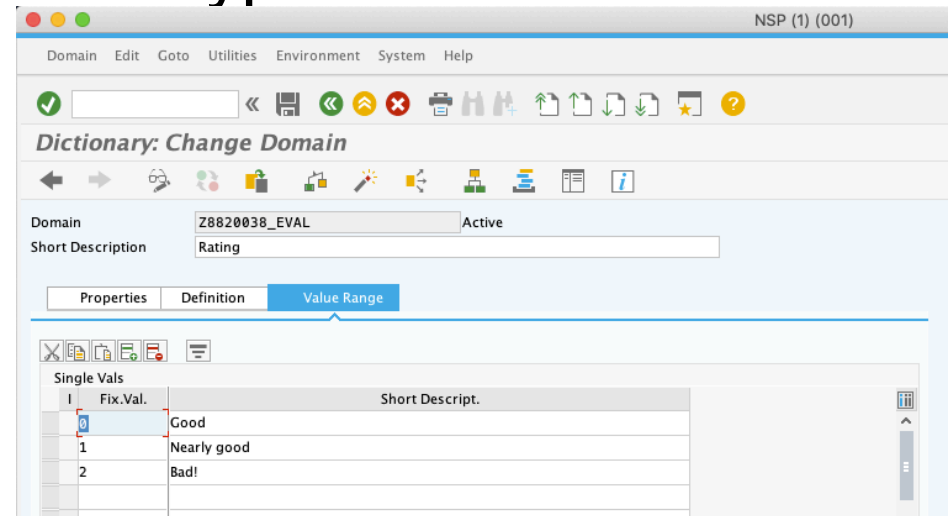
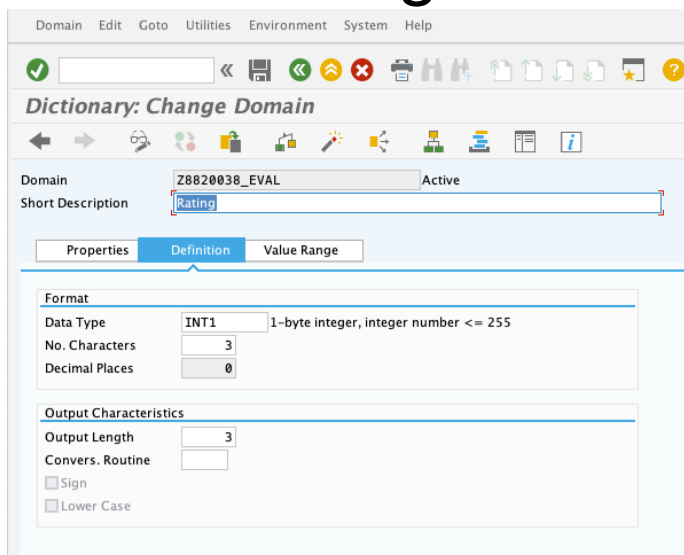
- Dokumentation !



Dictionary

ABAP Dictionary: Domäne

- Motivation: Dient zur Definition von globalen technischen Typen mit Festwerten
- Anlegen: TA SE11
- Verwendung: Technischer Typ für Datenelemente



ABAP Prozedural

“Die Sprache”

- ist proprietär,
- ist typisiert,
- ermöglicht mehrsprachige Anwendungen,
- ermöglicht SQL-Zugriffe,
- ist objektorientiert erweitert worden,
- ist plattformunabhängig,
- ist aufwärtskompatibel.



Syntax

Allgemeiner Aufbau einer ABAP-Anweisung

X X X

ABAP-
Schlüsselwort

Y Y Y

Zusätze und Operanden
(schlüsselwort-spezifisch)

.

Punkt als Abschluss
der Anweisung

Programmbeispiel

```
PARAMETERS pa_num TYPE i.
```

```
DATA gv_result TYPE i.
```

```
MOVE pa_num TO gv_result.
```

```
ADD 1 TO gv_result.
```

```
WRITE 'Your input:' .
```

```
WRITE pa_num .
```

```
NEW-LINE .
```

```
WRITE 'Result: ' .
```

```
WRITE gv_result .
```

message_class =
Nachrichtenklasse = Transaktion SE91

Nachrichten (Messages)

```
MESSAGE tnnn (message_class) [ WITH v1 [ v2 ] [ v3 ] [ v4 ] ] .
```

Typ	Bedeutung	Dialogverhalten	Nachricht erscheint in
i	Info-Nachricht	Programm wird nach Unterbrechung fortgesetzt	modalem Dialogfenster
s	Setznachricht	Programm wird ohne Unterbrechung fortgesetzt	Statuszeile *) des Folgebilds
w	Warnung	kontextabhängig	Statuszeile *)
e	Fehler	kontextabhängig	Statuszeile *)
a	Abbruch	Programm wird abgebrochen	modalem Dialogfenster
x	Kurzdump	Laufzeitfehler MESSAGE_TYPE_X wird ausgelöst	Kurzdump

AT SELECTION-SCREEN



AT SELECTION-SCREEN.

IF pa_eval > 2.

MESSAGE e001(zws19) WITH pa_eval.

* The evaluation & is not valid

ENDIF.

ABAP

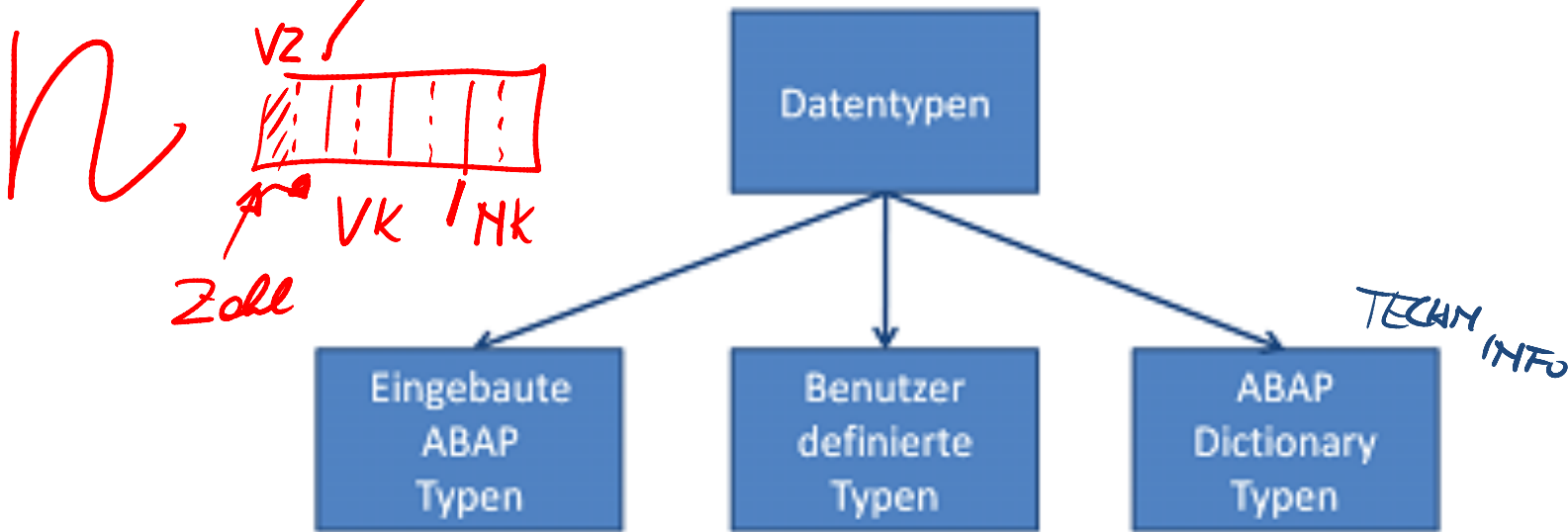


Einige wichtige Anweisungen:

- DATA
- TYPES
- CONSTANTS
- WRITE

Datentypen

Bei der Typisierung von Variablen können **eingebaute** ABAP Typen (d, t, i, c, p, n, f, string, xstring), **benutzerdefinierte** Typen (TYPES) oder **globale Typen** aus dem **ABAP Dictionary** (Datenelemente, Strukturtypen, Tabellentypen) verwendet werden.



Typen Arten:

- Elementare Typen
- Strukturierte Typen

* Zeilentyp für Datenpufferung zu FG und Eval

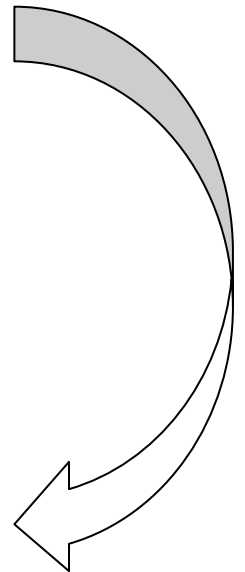
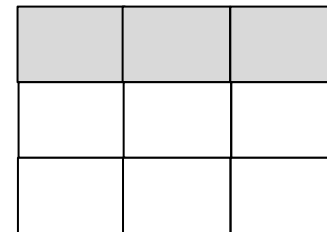
```
TYPES: BEGIN OF gst_fg_eval,  
        carrid TYPE scarr-carrid,  
        carrname TYPE scarr-carrname,  
        eval TYPE zscarreval-eval,  
END OF gst_fg_eval.
```

- Tabellenartige Typen

* Tabellentyp (Standard Tabelle) mit Zeilentyp
gtt_fg_eval TYPE TABLE OF gst_fg_eval.



carrid carrname eval



Debugging

The screenshot shows the ABAP Debugger interface. At the top, there is a menu bar with options: Debugging, Edit, Goto, Breakpoints, Settings, Development, System, and Help. Below the menu is a toolbar with various icons for navigation and execution. The main area is titled "ABAP Debugger" and contains a "Watchpoint" section. Below this, there are tabs for "Fields", "Table", "Breakpoints", "Watchpoints", "Calls", "Overview", and "Settings". The "Fields" tab is active, showing the "Main Program" as "Z8820038_WS19_HELLO_WORLD" and the "Source code of" as "Z8820038_WS19_HELLO_WORLD". The current execution step is "EVENT END-OF-SELECTION". The code being executed is:

```
* Ausgeben der FG mit bestimmten Bewertung
ULINE.
* Zeile für Zeile werden die Daten aus der Tabelle
* ausgegeben
→ LOOP AT gt_fg_eval INTO gs_fg_eval.
   WRITE: / gs_fg_eval-carrid,
          gs_fg_eval-carrname,
          gs_fg_eval-eval.
ENDLOOP.
```

The "Field names" section at the bottom shows a table with the following data:

Field names	Field contents
gt_fg_eval	Table[9x48]
gs_fg_eval-carrid	SQ
gs_fg_eval	Non-charlike structure

At the bottom of the debugger, there are several status fields: SY-SUBRC 0, SY-TABIX 9, SY-DBCNT 9, and SY-DYNNR 1000. A green checkmark icon and the text "Field copied into the display" are visible at the bottom left, and the SAP logo is at the bottom right.

Strukturierte Typen:

- Struktur (Dictionary oder programmlokal)
- Transparente Tabelle (Dictionary)
- View (Dictionary)

Literale

Fixe Datenobjekte ohne Bezeichner

Literale

Zahlenliterale	Textliterale
positive ganze Zahl : 123	Zeichenketten : 'Hallo'
negative ganze Zahl : -123	Bezimalzahlen : '123.45'
	Gleitpunktzahlen : '123.45E01'

Fixe Datenobjekte mit Bezeichner

Konstanten

```
CONSTANTS gc_myconst TYPE type_name VALUE {literal | IS INITIAL}.
```

ABAP

Beispiele

```
* comments ... }
* comments ... }
* comments ... }

PARAMETERS pa_num TYPE i.
DATA gv_result TYPE i.

MOVE pa_num
  TO gv_result.

ADD 1 TO gv_result.

WRITE : 'Your input: '
      pa_num.

NEW-LINE.

WRITE : 'Result: ' , gv_result.
```

Kommentare
(ganze Zeilen)

Kommentare
(Zeilenrest)

Kettensatz

Kettensatz

ReturnCodes (sy-subrc)

- Wichtige Systemfelder werden in der ABAP Dict Struktur **SYST** definiert.
- Das wichtigste ist **sy-subrc**, welches die Aussage über den Ausgang der vorherigen Operation liefert.

IF sy-subrc = 0. “Alles ok

...

ELSE. “Da lief was falsch

...

ENDIF.

```
* Lesen der FG von der DB Tabelle SCARR
SELECT SINGLE carrname FROM scarr INTO gd_carrname
WHERE carrid = pa_car.
IF sy-subrc <> 0.
  message e002(zws19) with pa_car.
ENDIF.
```

Modularisierung

- Ebene lokal – lokale Wiederverwendung
 - Unterprogramme
 - SAP GUI Module
 - Ereignisblöcke
- Ebene global – globale Wiederverwendung
 - Funktionsbausteine (Funktionsgruppen) – TA SE37
 - **ABAP Klassen (ABAP Interfaces) – TA SE24**

ABAP Prozedural

Interne Modularisierung



Unterprogramme

Definition

FORM <name> <schnittstelle>.

<deklarationen,anweisungen>

ENDFORM.

Schnittstelle (Formalparameter)

- USING [VALUE(<parameter>)]
TYPE <typ>
- CHANGING [VALUE(<parameter>)]
TYPE <typ>

Verwendung (Aktualparameter)

PERFORM <name> USING ... CHANGING
+ Schnittstellenverwendung

```
*-----*
*      Form  READ_CARRNAME
*-----*
*      Lesen des Namens der FG
*-----*
*      -->ID_CARRID   ID der FG
*      <--ED_CARRNAME Name der FG
*-----*
FORM read_carrname USING   id_carrid TYPE scarr-carrid
                        CHANGING ed_carrname TYPE scarr-carrname.

DATA: ld_carrname TYPE scarr-carrname.

* Lesen der FG von der DB Tabelle SCARR
SELECT SINGLE carrname FROM scarr INTO ld_carrname
WHERE carrid = id_carrid.
* Nichts gefunden
IF sy-subrc <> 0.
  CLEAR ed_carrname.
* Alles gefunden
ELSE.
  ed_carrname = ld_carrname.
ENDIF.
ENDFORM.                                " READ_CARRNAME

* Aufruf Unterprogramm für die Ermittlung des Carrname
* Verwendungsstelle
PERFORM read_carrname USING pa_car
                        CHANGING gd_carrname.
IF gd_carrname IS INITIAL. "sy-subrc <> 0.
  MESSAGE e002(zws19) WITH pa_car.
ENDIF.
```

Unterprogramm

```
*&-----*
*&  Form READ_EVAL
*&-----*
*   Lesen der Bewertung
*-----*
*   -->ID_CARRID  ID der FG
*   <--ED_EVAL   Bewertung
*-----*
FORM read_eval USING  id_carrid TYPE scarr-carrid
                   CHANGING ed_eval TYPE zscarreval-eval.

SELECT SINGLE eval FROM zscarreval INTO ed_eval
WHERE carrid = id_carrid.

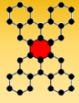
ENDFORM.          " READ_EVAL
```

ABAP Prozedural Struktur



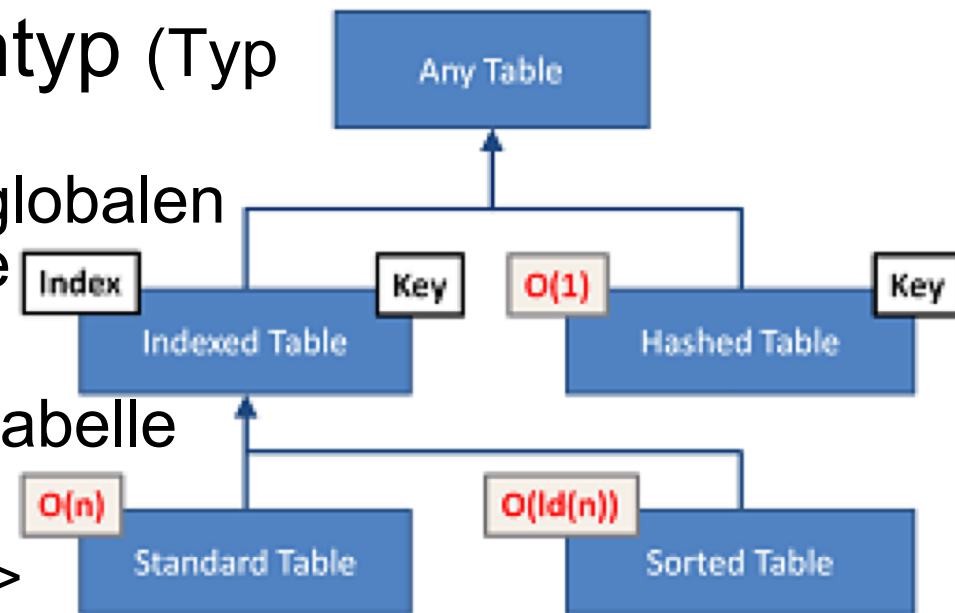
ABAP Dictionary: Struktur

- Motivation: Definition von globalem strukturiertem Typ (vergleichbar: Record)
- Anlegen: TA SE11
- Verwendung: Typisierung von Variablen, Schnittstellen, Tabellentypen, ...
- Zugriff auf Komponenten: Bindestrich Semantik
- Kopieren von Inhalten: MOVE-CORRESPONDING



ABAP Dictionary: Tabellentyp (Typ einer internen Tabelle)

- Motivation: Definiert den globalen Typ einer internen Tabelle
- Anlegen: TA SE11
- Definition einer internen Tabelle
- Wichtige Anweisungen:
 - Sort <tabelle> BY <spalte> [ascending|decending]
 - APPEND <zeile> TO <tabelle>.
 - DELETE <zeile> FROM <tabelle>.

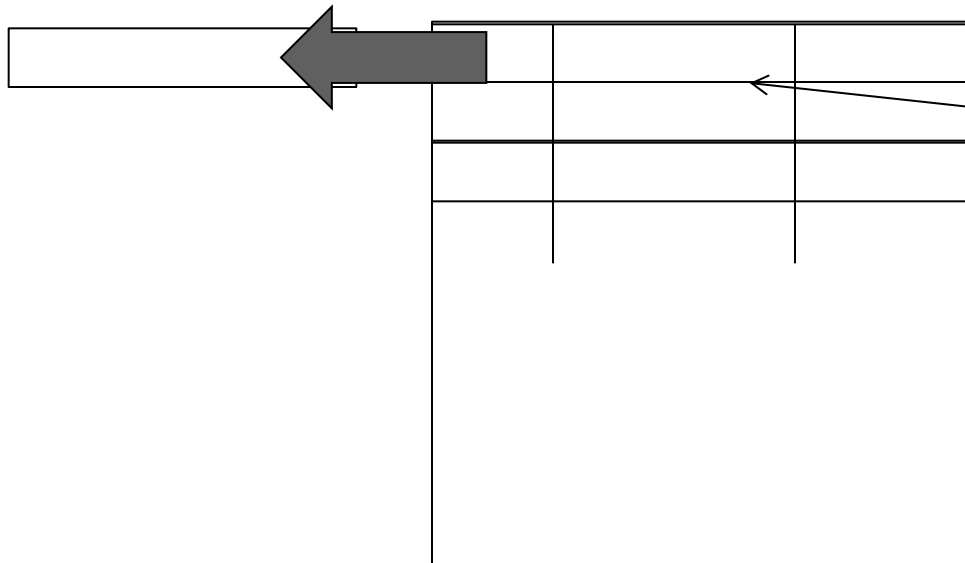


ABAP Prozedural Datenhaltung

Interne Tabelle

Arbeitsstruktur (Struktur)

Ist im Hauptspeicher!



Zeilentyp

Art der Tabelle

- Standard
- Sortierte
- Hashed

Schlüssel

- Feldern
- Reihenfolge
- Eindeutigkeit

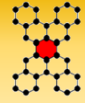
Ausgabe Tabelle

```
*&-----*
*&   Form WRITE_FG_EVAL
*&-----*
*   Ausgabe der Tabelle auf die Liste
*-----*
*   -->IT_FG_EVAL  Tabelle der FV und eval
*-----*
FORM write_fg_eval USING it_fg_eval TYPE gtt_fg_eval .

DATA: ls_fg_eval LIKE LINE OF it_fg_eval.

LOOP AT it_fg_eval INTO ls_fg_eval.
  WRITE: / ls_fg_eval-carrid,
         ls_fg_eval-carnname,
         ls_fg_eval-eval.
ENDLOOP.

ENDFORM.          " WRITE_FG_EVAL
```



Lesen in interne Tabelle

```

*&-----*
*&  Form READ_FG_EVAL
*&-----*
*   Ermitteln der FG zu einer eval
*-----*
*   -->Id_EVAL  Eval
*   <--ET_FG_EVAL FG zu eval
*-----*
FORM read_fg_eval USING  id_eval  TYPE zscarreval-eval
                     CHANGING et_fg_eval TYPE gtt_fg_eval.

DATA: ld_carrid  TYPE scarr-carrid,
      ls_fg_eval LIKE LINE OF et_fg_eval,
      ld_carname TYPE scarr-carname.

* Massenoperation mit SELECT
SELECT carrid FROM zscarreval INTO ld_carrid
WHERE eval = id_eval.
* Hilfsstruktur befüllen
  ls_fg_eval-eval = id_eval.
  ls_fg_eval-carrid = ld_carrid.
* Lesen der fg von der db tabelle scarr
  PERFORM read_carname
    USING
      ld_carrid
    CHANGING
      ld_carname.

  ls_fg_eval-carname = ld_carname.
* Eintrag in der internen Tabelle erzeugen
  APPEND ls_fg_eval TO et_fg_eval.
ENDSELECT.

ENDFORM.          " READ_FG_EVAL

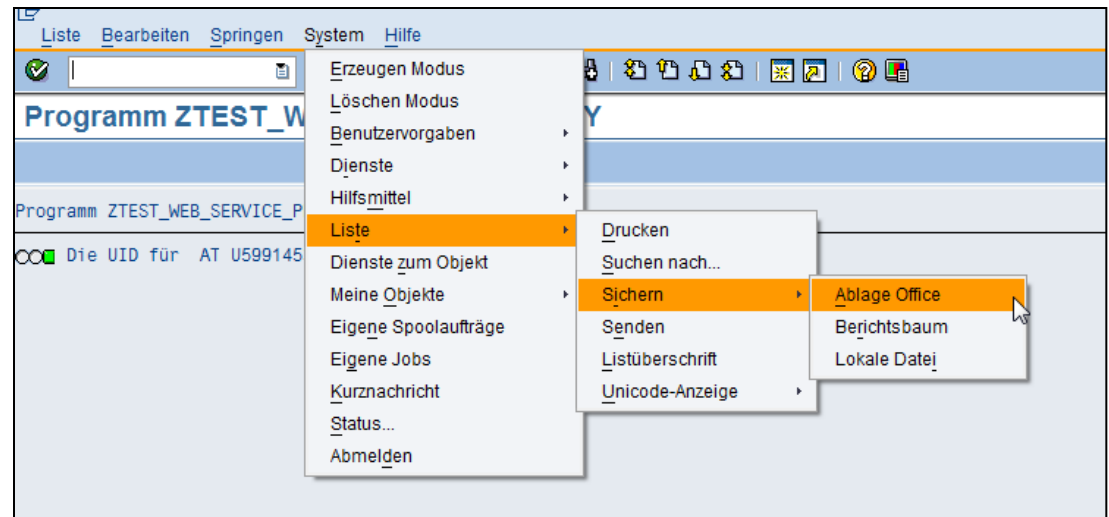
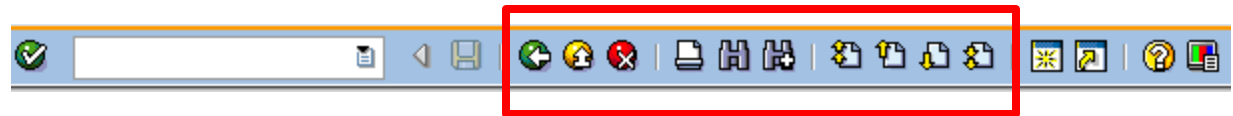
```

ABAP Liste

ABAP Liste

Standard Funktionen

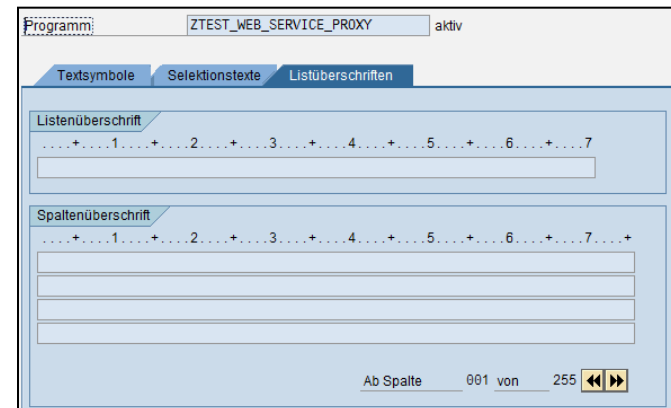
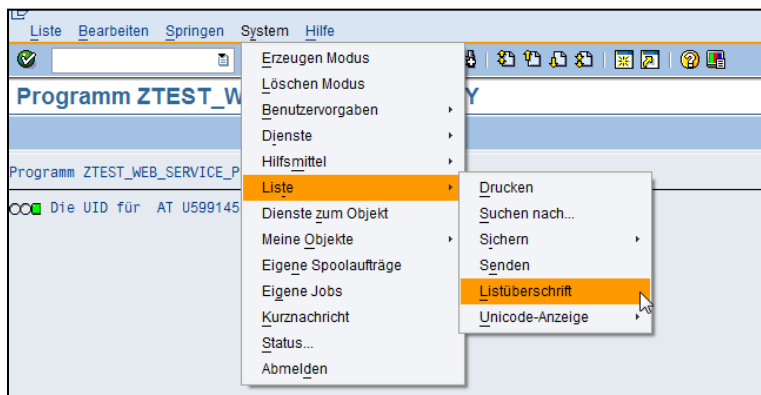
- Suche
- Blättern
- Exportieren
- ...



ABAP Liste

Listen und Spaltenüberschriften

- Direkte Pflege System->Liste->Listenüberschrift
- Über Springen->Textelemente->Listenüberschriften



ABAP Liste

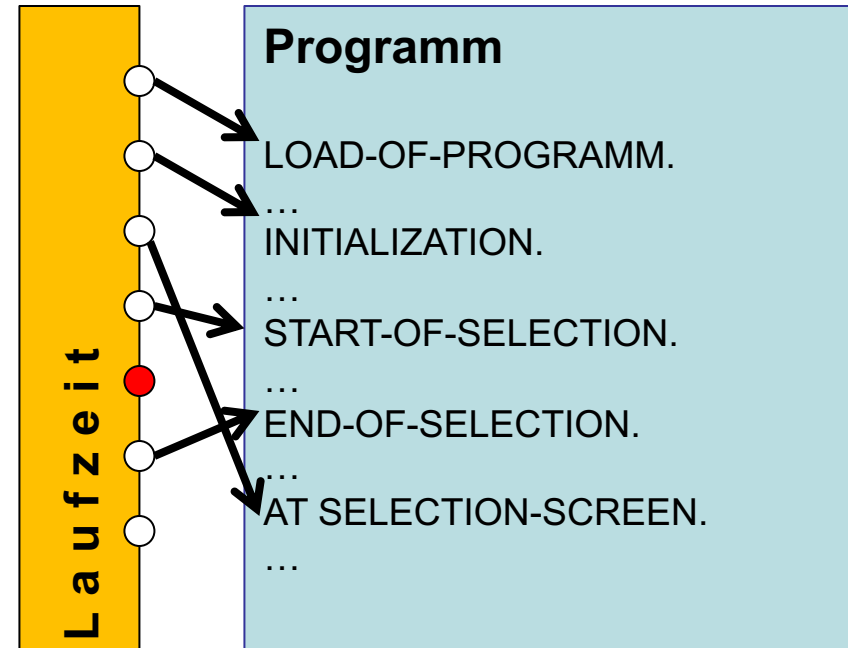
Ereignisse

- START-OF-SELECTION
- END-OF-SELECTION
- TOP-OF-PAGE
- ...

```
TYPE-POOLS: abap,  
            icon.  
START-OF-SELECTION.
```

* Gültigkeit ausgeben

```
IF ls_response-valid = abap_true.  
    WRITE: / icon_green_light AS ICON, 'Die UID für ', pa_ms, pa_vat, ' ist gültig!'.  
ELSE.  
    WRITE: / icon_red_light AS ICON, 'Die UID für ', pa_ms, pa_vat, ' ist ungültig!'.  
ENDIF.
```



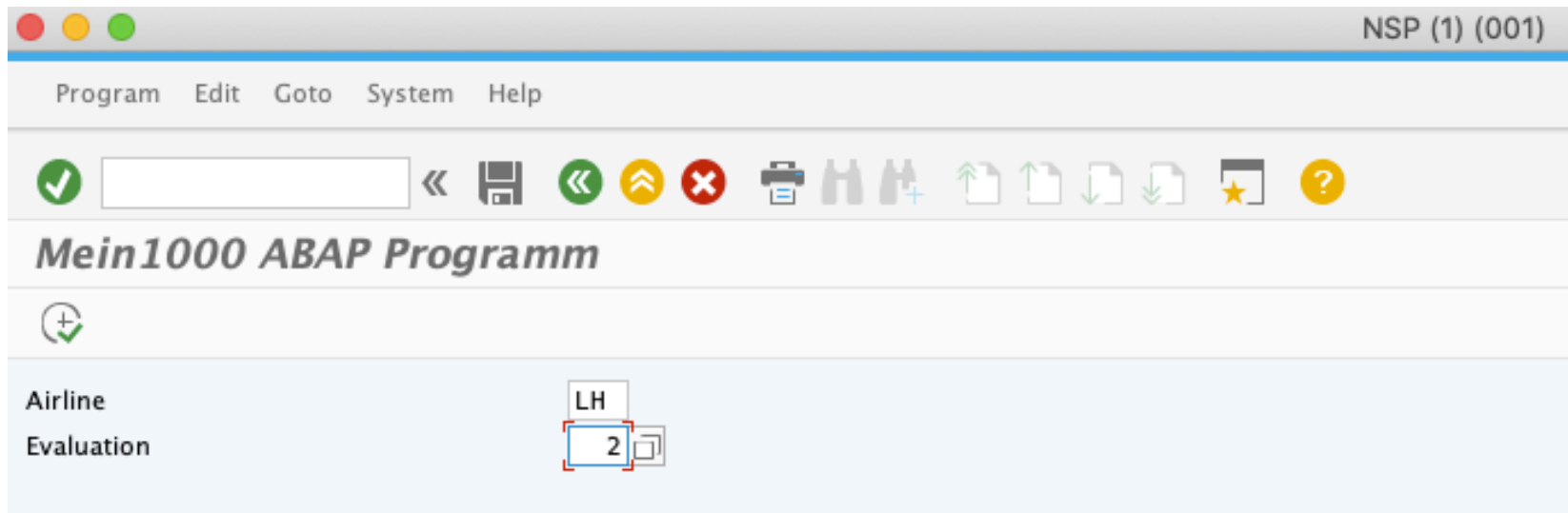
ABAP Selektionsbild

ABAP Selektionsbild



Motivation

- Verwendung

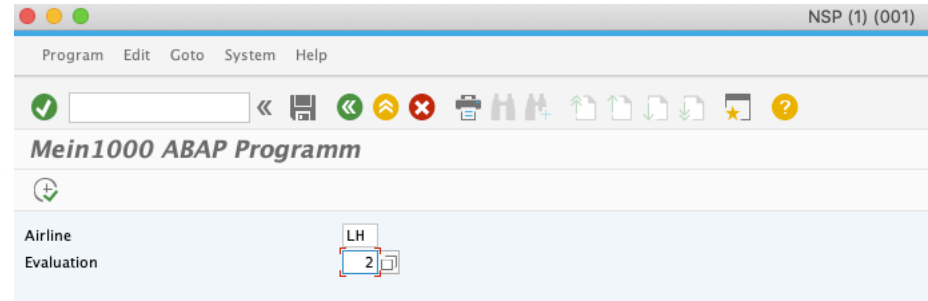


ABAP Selektionsbild



Gestaltung

- Parameters
 - UI Element
 - Datenfeld

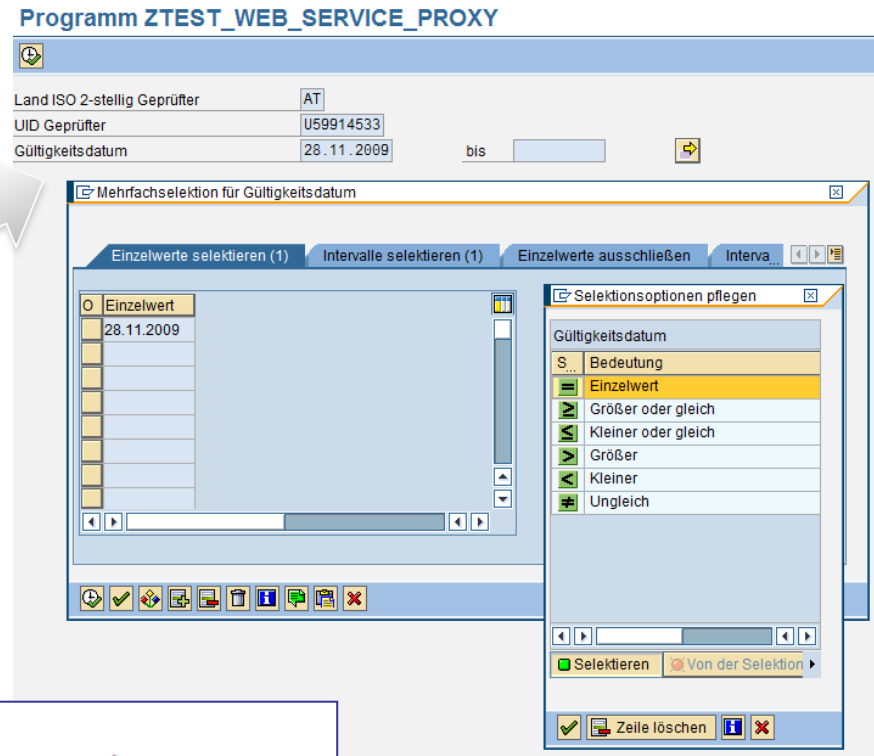


* Eingabefeld für Fluggesellschaft
PARAMETERS: pa_car TYPE scarr-carrid,
pa_eval TYPE zscarreval-eval.

ABAP Selektionsbild

Gestaltung

- Select-Options
 - UI Element
 - Datenfeld



```
* data declarations  
DATA: ld_date TYPE dats. "Datum für Select Option  
  
|SELECT-OPTIONS: so_date FOR ld_date. "Datumsselektion
```

Select Option und Parameter

1. Range deklarieren

* Eingabe für Intervalle

```
SELECT-OPTIONS: so_eval FOR gd_so_eval.
```

* Nur die Daten der SO

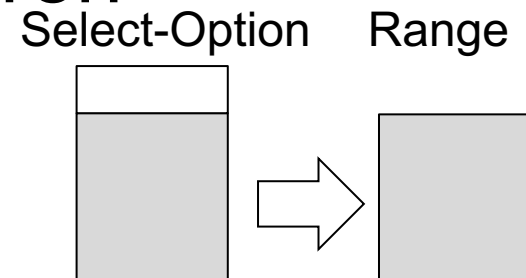
```
data: gt_so_eval like range of gd_so_eval.
```

2. Daten von SO in Range kopieren

* Ermitteln der FG zu einer Eval

```
gt_so_eval = so_eval[[]].
```

```
PERFORM read_fg_eval USING gt_so_eval "pa_eval  
CHANGING gt_fg_eval.
```



3. An Unterprogramm übergeben

```
FORM read_fg_eval USING it_so_eval like gt_so_eval "id_eval TYPE zscarreval-eval  
CHANGING et_fg_eval TYPE gtt_fg_eval.
```

```
DATA: ld_carrid TYPE scarr-carrid,  
ls_fg_eval LIKE LINE OF et_fg_eval,  
ld_carrname TYPE scarr-carrname.
```

* Hilfsstruktur zum Auslesen der Eval Daten

```
DATA: ls_scarreval type zscarreval.
```

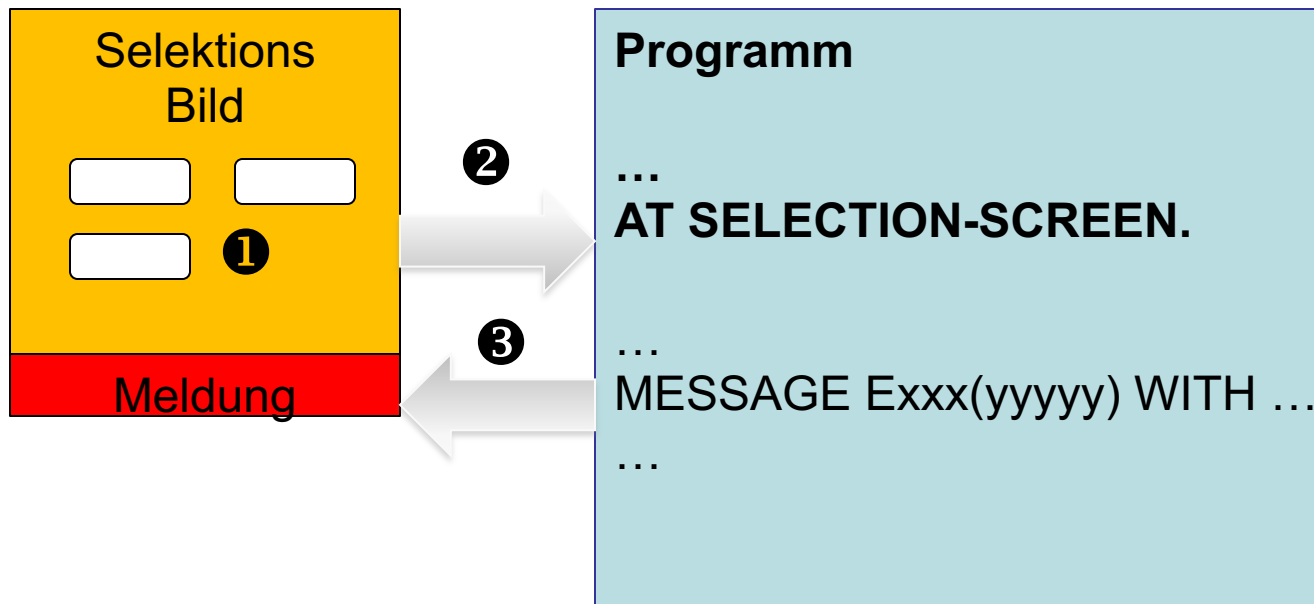
* Massenoperation mit SELECT

```
* Auslesen aller Spalten einer Zeile aus eval  
SELECT * FROM zscarreval INTO ls_scarreval  
WHERE eval in it_so_eval. "=" id_eval.
```


ABAP Selektionsbild

Ereignisse

- AT SELECTION-SCREEN.



Modularisierung

ABAP Prozedural

Interne Modularisierung



Includes

```
13: INCLUDE /co_test_import_writf01.  
14:
```

- Objekte können in **Includes** zusammengefasst werden
- Includes werden beim **Übersetzungsvorgang** eingebunden
- Includes sind wiederverwendbare „Textkonserven“
- Eine Wiederverwendung durch Unterprogramme und Deklarationen in Includes ist zu vermeiden!

ABAP Prozedural

Externe Modularisierung



Funktionsgruppen und Funktionsbausteine

- Definition TA SE37 (Function Builder)
 - Schnittstelle
 - IMPORTING
 - EXPORTING
 - CHANGING
 - EXCEPTIONS (prozedural/OO)
 - Verwenden (Über Drucktaste Muster in Drucktastenleiste)
 - CALL FUNCTION '<name>'

The screenshot shows the 'Function Builder' interface for the function 'BAPI_SFLIGHT_GETLIST'. The title bar reads 'Function Builder: BAPI_SFLIGHT_GETLIST anzeigen'. Below the title bar is a toolbar with various icons and the text 'Muster' and 'Einfügen'. The main area displays the function definition in a code editor. The code is as follows:

```
1 FUNCTION BAPI_SFLIGHT_GETLIST.  
2 ***-----  
3 ***"Lokale Schnittstelle:  
4 *** IMPORTING  
5 ***   VALUE (FROMCOUNTRYKEY) LIKE BAPISFDETA-COUNTRYFR  
6 ***   VALUE (FROMCITY) LIKE BAPISFDETA-CITYFROM  
7 ***   VALUE (TOCOUNTRYKEY) LIKE BAPISFDETA-COUNTRYTO  
8 ***   VALUE (TOCITY) LIKE BAPISFDETA-CITYTO  
9 ***   VALUE (AIRLINECARRIER) LIKE BAPISFDETA-CARRID DEFAULT SPACE  
10 ***   VALUE (AFTERNOON) LIKE BAPI_AUX-AFTERNOON DEFAULT SPACE  
11 ***   VALUE (MAXREAD) LIKE BAPI_AUX-MAXREAD DEFAULT 0  
12 *** EXPORTING  
13 ***   VALUE (RETURN) LIKE BAPIRET2 STRUCTURE BAPIRET2  
14 *** TABLES  
15 ***   FLIGHTLIST STRUCTURE BAPISFLIST  
16 ***-----  
17  
18 CLEAR RETURN
```

FuBa 1



Create Function Module

Function Module: Z8820038_WS19_FG_R_carrname

Function group: Z8820038_WS19_FG

Short text: Read carrname

```
* Lesen der FG von der DB Tabelle SCARR
* SELECT SINGLE carrname FROM scarr INTO gd_carrname
* WHERE carrid = pa_car.
* Aufruf Unterprogramm für die Ermittlung des Carrname
* Verwendungsstelle
* PERFORM read_carrname USING pa_car
*                               CHANGING gd_carrname.

CALL FUNCTION 'Z8820038_WS19_FG_R_CARRNAME'
  EXPORTING
    id_carrid = pa_car
  IMPORTING
    ed_carrname = gd_carrname.
```

Save

Function modules Edit Goto Utilities System Help

Test Function Module: Result Screen

Test for function group: Z8820038_WS19_FG
Function module: Z8820038_WS19_FG_R_CARRNAME
Uppercase/Lowercase:

Runtime: 954 Microseconds

Import parameters	Value
ID_CARRID	LH

Export parameters	Value
ED_CARRNAME	Lufthansa

FuBa 2



```
FUNCTION z8820038_ws19_fg_r_eval.
```

```
*"-----
```

```
***"Local Interface:
```

```
*" IMPORTING
```

```
*" REFERENCE(ID_CARRID) TYPE SCARR-CARRID
```

```
*" EXPORTING
```

```
*" REFERENCE(ED_EVAL) TYPE ZSCARREVAL-EVAL
```

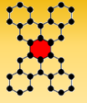
```
*"-----
```

```
SELECT SINGLE eval FROM zscarreval INTO ed_eval  
WHERE carrid = id_carrid.
```

```
ENDFUNCTION.
```

ABAP Objektorientiert

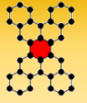
ABAP Objektorientiert



Motivation

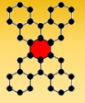
- Wofür?
- Prozedural vs. Objektorientiert
- Eigenschaften der OO
- Abdeckungsgrad in ABAP OO
 - Vererbung
 - Polymorphie
 - Kapselung
- Separate Definition und Implementierung





Modellierung mit der Unified Modeling Language (UML), im Speziellen

- Klassendiagramm
 - Klasse
 - Attribut
 - Methode
 - Assoziation
 - Kardinalität
- Sequenzdiagramm



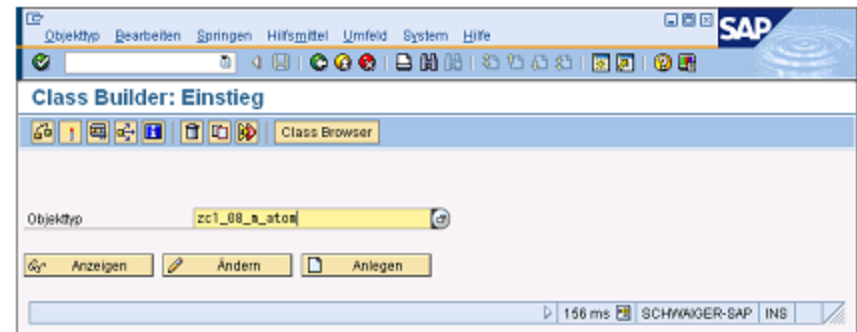
Klasse und Objekt

- Dynamisch vs. Statisch
- Sichtbarkeiten
- Attribut
- Methode
- Ereignis
- Bauplan/Blaupause

ABAP Klasse

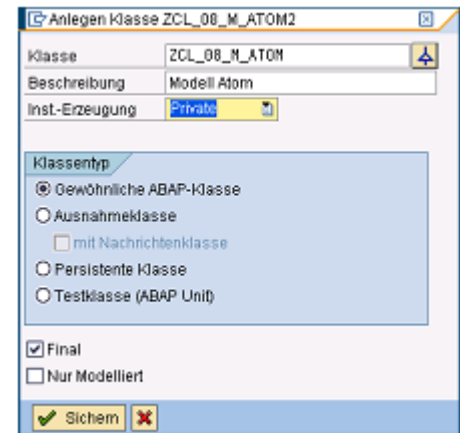
Arten von ABAP Klassen

- **Normale Klasse**
- **Ausnahmeklasse**
- Testklasse
- Persistentes Objekt



Eigenschaften von ABAP Klassen

- Abstrakt
- Final



Klasse SE24

The screenshot shows the 'Create Class' dialog box in SAP. The title bar reads 'Create Class'. There are three colored window control buttons (red, grey, green) on the left. The main area contains the following fields:

- Class:** A text field containing 'ZCL_8820038_WS19_FG' and a small icon to its right.
- Description:** A text field containing 'Carrier'.
- Instantiation:** A dropdown menu showing 'Public'.

Below these fields is a section titled 'Class Type' with a blue underline. It contains five radio button options:

- Usual ABAP Class
- Exception Class
 - With Message Class
- Persistent class
- Test Class (ABAP Unit)

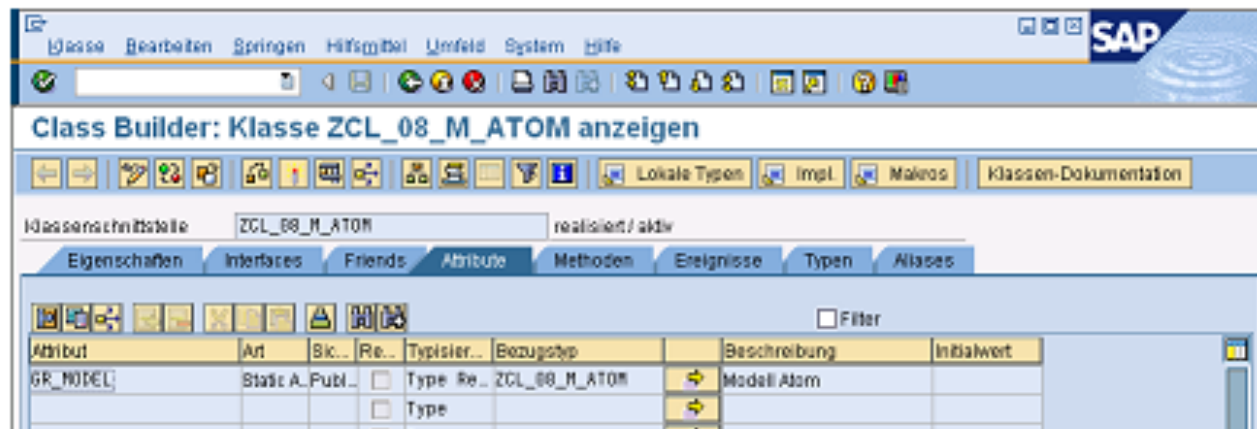
At the bottom left of the dialog is a checkbox labeled 'Final' with red brackets around it. At the bottom right are two buttons: 'Save' with a green checkmark icon and a close button with a red 'X' icon.

ABAP Klasse



Attribut

- Statisch
- Instanz
- Sichtbarkeiten

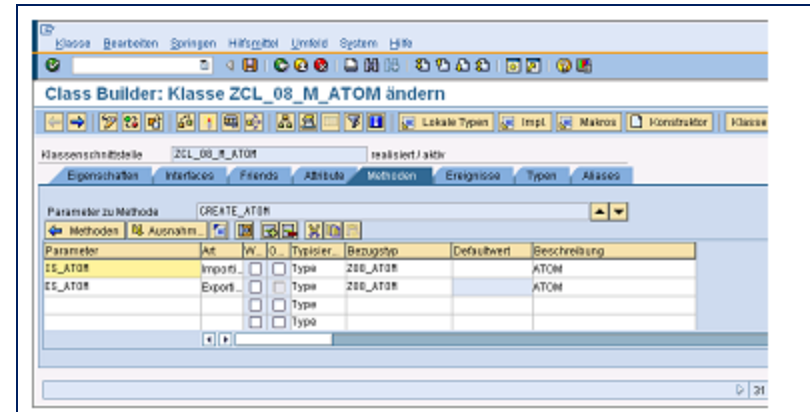
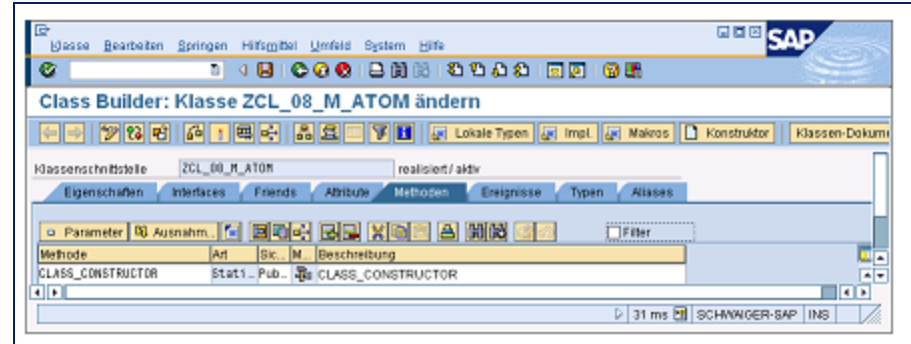


ABAP Klasse



Definition Methode

- Statisch
- Instanz
- Sichtbarkeiten



```

* Daten von der Datenbanktabelle lesen
SELECT * FROM z08_atom INTO TABLE et_atoms
WHERE pk_atom IN it_r_pk_atoms.

IF sy-subrc <> 0.
* Falls keine Daten gefunden wurden, dann Exception
RAISE EXCEPTION TYPE zcx_08_atom
EXPORTING
textid = zcx_08_atom->atom_no_data
.
ENDIF.
    
```

Art	Parameter	Typisierung	Beschreibung
	IT_R_PK_ATOMS	TYPE Z08_TT_R_PK_ATOM	TT zu Range PK_ATOM
	ET_ATOMS	TYPE Z08_TT_ATOM	TT Atom
	ZCX_08_ATOM		Exc Atom

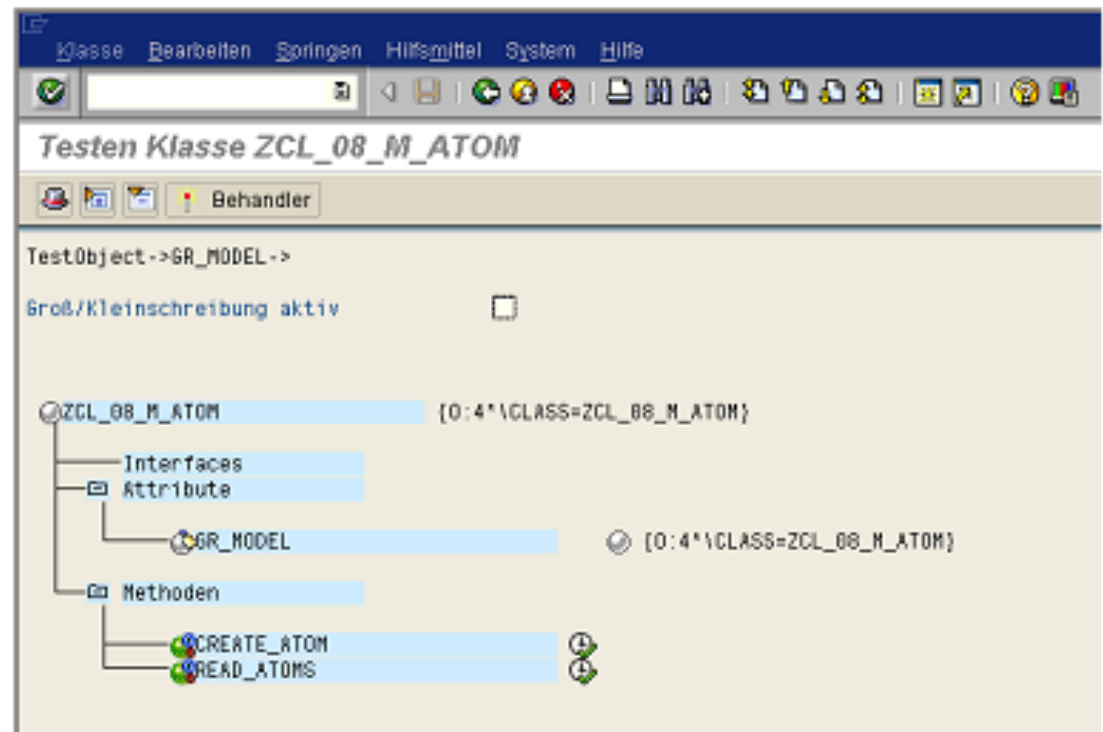
Methode: READ_ATOMS

ABAP Klasse



Testen von Klassen

- Testrahmen
- Testklassen



ABAP Objekt

Zustände von Objekten

- Erzeugen (Constructor)

```
IKT.  
* Objekt instanziiieren  
  CREATE OBJECT go_fg  
  EXPORTING  
    id_carrid = pa_car.  
* Fehlerbehandlung
```

- Terminieren (Grabage Collector)
 - Mittels CLEAR kann die Objektreferenz initialisiert werden und dadurch sammelt der GC das Objekt ein und gibt die Ressourcen frei.
 - Prinzipiell, wenn ein Objekt nicht mehr erreichbar ist, dann gibt der GC die betroffenen Ressourcen frei

ABAP Objekt



Aufruf von Methoden

- Statisch

- Instanz `go_alv->display()`.

- Funktional `gd_carrname_single = go_fg->get_carrname()`.

Verwendung von Attributen

```
cl_salv_table=>factory(  
* EXPORTING  
* list_display = IF_SALV_C_BOOL_SAP=>FALSE  
* r_container =  
* container_name =  
  IMPORTING  
    r_salv_table = go_alv  
  CHANGING  
    t_table = gt_fg_eval  
  ).
```

ABAP OO Ausnahmen

ABAP Ausnahmen (Exceptions)



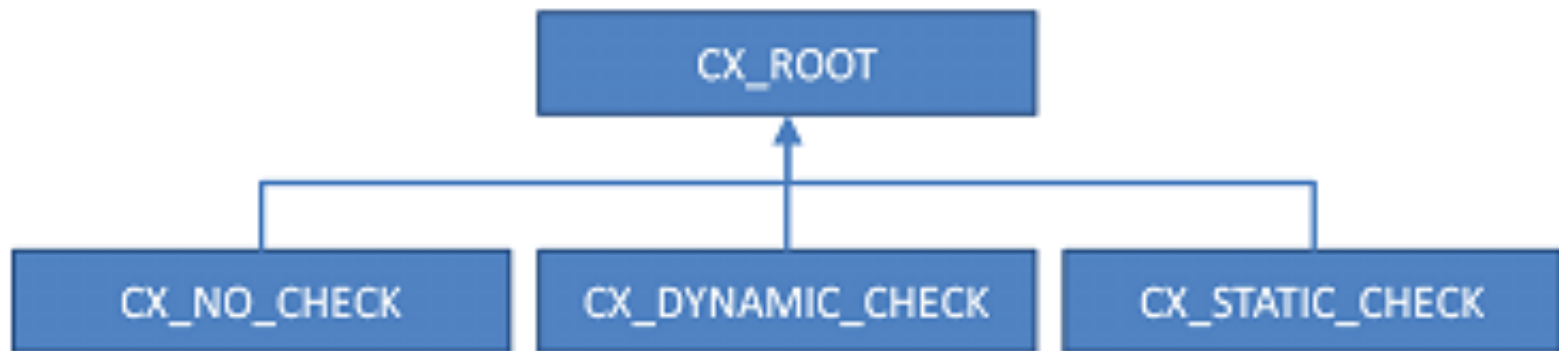
Existierende Ausnahmemechanismen

- Auswertung sy-subrc
- CATCH SYSTEM-EXCEPTION ...
.ENDCATCH. + sy-subrc Auswertung
- TRY. CATCH. ENDTRY.

ABAP Ausnahmen (Exceptions)

Vererbungshierarchie für Exceptions

CX_ROOT



ABAP Ausnahmen (Exceptions)

Exception anlegen

Text zur Ausnahme

The screenshot shows the 'Anlegen Klasse' (Create Class) dialog box in SAP. The title bar reads 'Anlegen Klasse ZCX_08_ATOM'. The dialog has several fields: 'Klasse' (Class) is 'ZCX_08_ATOM', 'Erbt von' (Inherits from) is 'CX_STATIC_CHECK', 'Beschreibung' (Description) is 'Exc Atom', and 'Inst-Erzeugung' (Instance Generation) is 'Public'. Below these fields is a 'Klassentyp' (Class Type) section with radio buttons for: 'Gewöhnliche ABAP-Klasse' (unselected), 'Ausnahmeklasse' (selected), 'mit Nachrichtenklasse' (checkbox), 'Persistente Klasse' (unselected), and 'Testklasse (ABAP Unit)' (unselected). At the bottom, there are checkboxes for 'Final' (checked) and 'Nur Modelliert' (unchecked). The bottom bar contains 'Sichern' (Save) and 'Abbrechen' (Cancel) buttons.

ABAP Ausnahmen (Exceptions)

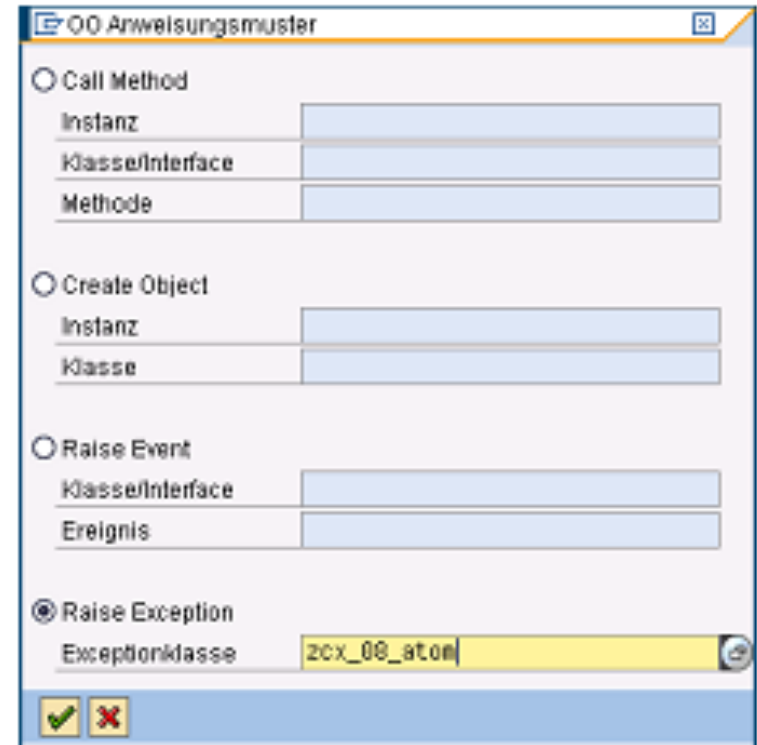


Exception auslösen

RAISE EXCEPTION


Ty.	Parameter	Type spec.	Description
	ID_CARRID	TYPE SCARR-CARRID	Airline Code
	ZCX_8820038_WS19_FG		Exception FG


```
Method CONSTRUCTOR  
METHOD constructor.  
    me->gd_carrid = id_carrid.  
    CALL FUNCTION 'Z8820038_WS19_FG_R_CARRNAME'  
      EXPORTING  
        id_carrid = me->gd_carrid  
      IMPORTING  
        ed_carrname = me->gd_carrname.  
    IF me->gd_carrname IS INITIAL.  
      RAISE EXCEPTION TYPE zcx_8820038_ws19_fg  
* EXPORTING  
*   textid =  
*   previous =  
*  
    ENDIF.
```



ABAP Ausnahmen (Exceptions)

Ausnahmetexte



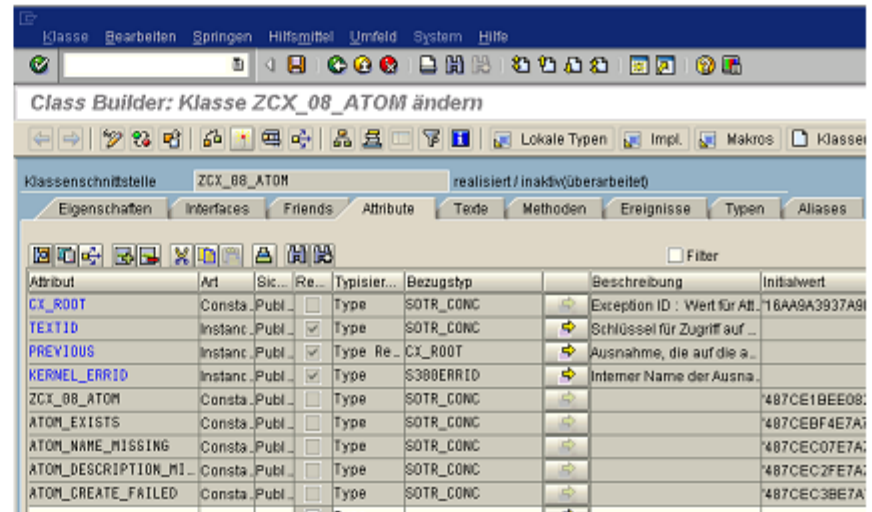
Class Builder: Klasse ZCX_08_ATOM ändern

Klassenschnittstelle ZCX_08_ATOM realisiert / inaktiv(überarbeitet)

Eigenschaften Interfaces Friends Attribute Texte Methoden Ereignisse

Langtext

Exception ID	Text
CX_ROOT	Es ist eine Ausnahme aufgetreten.
ZCX_08_ATOM	Es ist ein Problem im Zusammenhang mit dem Atom aufgetreten
ATOM_EXISTS	Das Atom wurde bereits angelegt
ATOM_NAME_MISSING	Der Name des Atoms ist nicht vorhanden
ATOM_DESCRIPTION_MISSING	Die Beschreibung des Atoms ist nicht vorhanden
ATOM_CREATE_FAILED	Die Anlage des Atoms ist fehlgeschlagen



Class Builder: Klasse ZCX_08_ATOM ändern

Klassenschnittstelle ZCX_08_ATOM realisiert / inaktiv(überarbeitet)

Eigenschaften Interfaces Friends Attribute Texte Methoden Ereignisse Typen Aliases

Filter

Attribut	Art	Sic...	Re...	Typisier...	Bezugstyp	Beschreibung	Initialwert
CX_ROOT	Consta	Publ	<input type="checkbox"/>	Type	SOTR_CONC	Exception ID : Wert für At...	16A9A3937A9
TEXTID	Instanc	Publ	<input checked="" type="checkbox"/>	Type	SOTR_CONC	Schlüssel für Zugriff auf ...	
PREVIOUS	Instanc	Publ	<input checked="" type="checkbox"/>	Type Re...	CX_ROOT	Ausnahme, die auf die a...	
KERNEL_ERRID	Instanc	Publ	<input checked="" type="checkbox"/>	Type	S380ERRID	Interner Name der Ausna...	
ZCX_08_ATOM	Consta	Publ	<input type="checkbox"/>	Type	SOTR_CONC		487CE1BEE08
ATOM_EXISTS	Consta	Publ	<input type="checkbox"/>	Type	SOTR_CONC		487CEBF4E7A
ATOM_NAME_MISSING	Consta	Publ	<input type="checkbox"/>	Type	SOTR_CONC		487CEC07E7A
ATOM_DESCRIPTION_MI...	Consta	Publ	<input type="checkbox"/>	Type	SOTR_CONC		487CEC2FE7A
ATOM_CREATE_FAILED	Consta	Publ	<input type="checkbox"/>	Type	SOTR_CONC		487CEC3BE7A

V1

ABAP OO Anwendung ALV

Vorbereitung Struktur und Tabellentyp



NSP (2) (001)

Structure Edit Goto Utilities Extras Environment System Help

Dictionary: Change Structure

Structure: ZST_8820038_WS19_FG_EVAL Active
Short Description: Eval for FG

Attributes Components Entry help/check Currency/quantity fields

Component	Key	Initi.	Typing Method	Component Type	Data Type	Length	D
CARRID	<input type="checkbox"/>	<input type="checkbox"/>	Types	S_CARR_ID	CHAR	3	
CARRNAME	<input type="checkbox"/>	<input type="checkbox"/>	Types	S_CARRNAME	CHAR	20	
EVAL	<input type="checkbox"/>	<input type="checkbox"/>	Types	Z8820038_EVAL	INT1	3	

NSP (2) (001)

Table Type Edit Goto Utilities Environment System Help

Dictionary: Change Table Type

Table Type: ZTT_8820038_WS19_FG_EVAL Active
Short text: Eval FG

Attributes Line Type Initialization and Access Primary Key Secondary Key

Line Type: ZST_8820038_WS19_FG_EVAL

Predefined Type
Data Type:
No. of Characters: Decimal Places:

Reference type
 Name of Ref. Type:
 Reference to Predefined Type
Data Type:
Length: Decimal Places:

V1: ABAP Klasse CL_SALV_TABLE



GUI Klassen - Lebenszyklus

1. Deklaration Objektreferenz für ALV
2. Instanziierung mittels FACTORY
3. Anzeige mittels DISPLAY

```
** 00 Ausgabe
DATA: go_alv TYPE REF TO cl_salv_table.

TRY.
    cl_salv_table=>factory(
*   EXPORTING
*     list_display   = IF_SALV_C_BOOL_SAP=>FALSE
*     r_container    =
*     container_name =
    IMPORTING
        r_salv_table = go_alv
    CHANGING
        t_table      = gt_fg_eval
    ).

    go_alv->display( ).

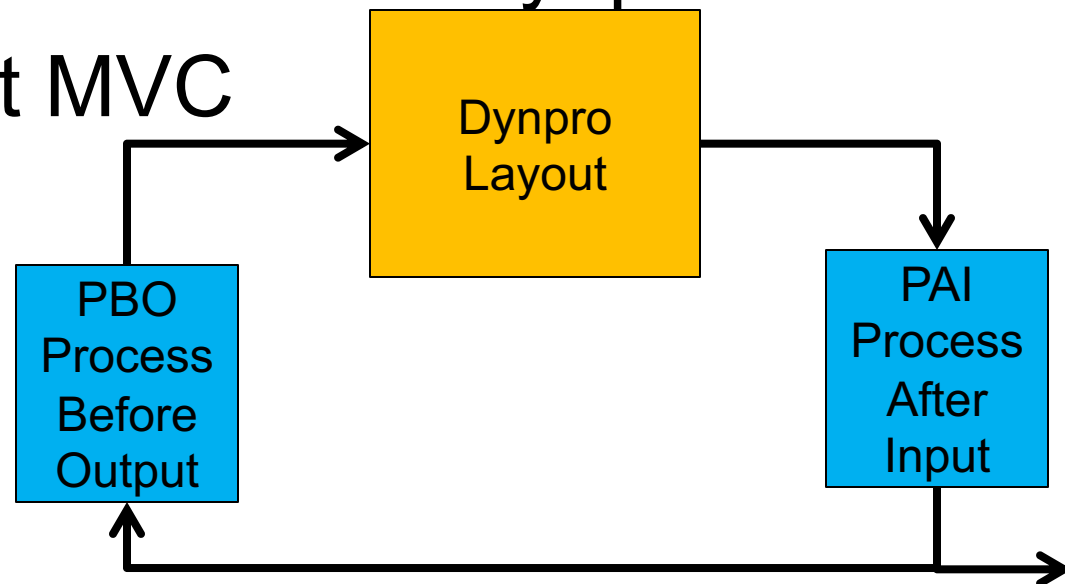
CATCH cx_salv_msg .
ENDTRY.
```

ABAP Dynpro

ABAP Dynpro

Motivation

- Ablauf PBO, PAI, Layout
- Weitere Elemente eines Dynpros
- Vergleich mit MVC



ABAP Dynpro



Dynpro erstellen

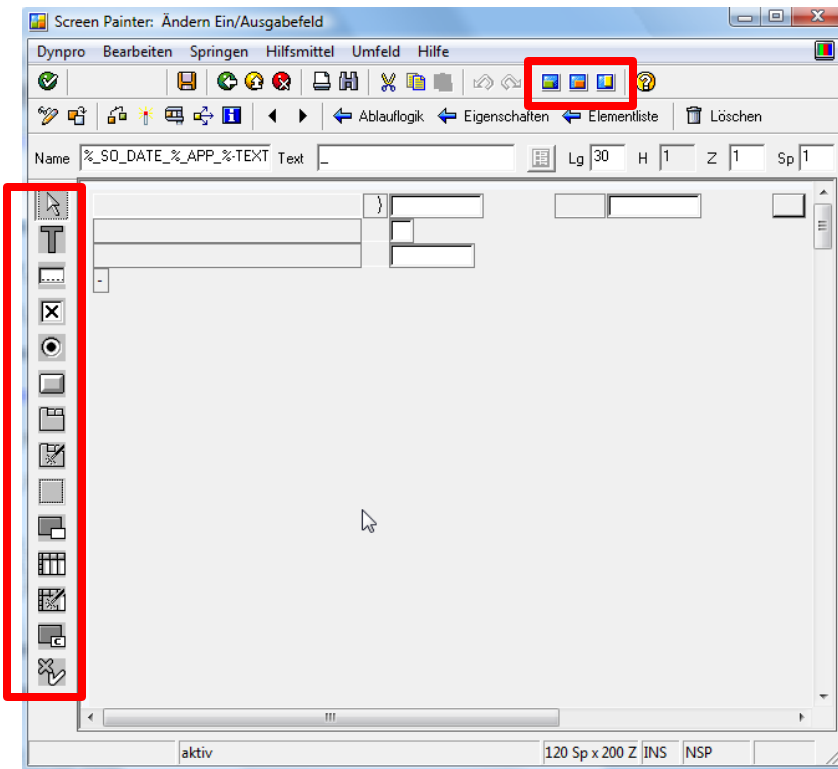
Zum Zeitpunkt END-OF-SELECTION den
Aufruf

```
CALL SCREEN 9000.
```

einfügen und mittels Vorwärtsnavigation
anlegen.

ABAP Dynpro

Grafischen Layout Editor



ABAP Dynpro



- Eine Drucktaste definieren, um wieder zur Aufrufstelle navigieren zu können
 - Funktionscode `BACK` zuordnen
- OK Feld definieren;
 - In den Eigenschaften des Dynpros
 - Im rufenden Programm

```
DATA: ok_code TYPE sy-ucomm.
```


ABAP Dynpro



In der Dynpro Ablauflogik

- Ein Process After Input (PAI) Modul zur Behandlung des Drucktastenfunktionscodes BACK
- Ein Process Before Output (PBO) Modul zum Setzen des Titels einfügen

- Kommunikationsstruktur im ABAP Dictionary definieren bzw. einen vorhandenen `<ABAP Dictionary Strukturtyp>` verwenden
- Layout Elemente basierend auf dem `<ABAP Dictionary Strukturtyp>` im Screen Painter definieren
- Datenstruktur
TABLES: `<ABAP Dict Strukturtyp>`.
Im Programm definieren

ABAP Dynpro



Dynpro erstellen

- Weitere diverse PBO/PAI Module
 - Logik
 - Initialisierung
 - Prüfung Benutzereingaben

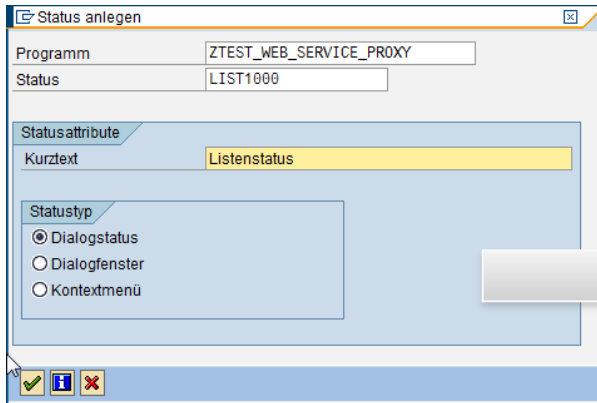
ABAP Dynpro



- Zum Setzen der Oberfläche, d.h. Menü, Drucktastenleiste, Symbolleiste, Funktionstastenzuordnung, wird der Menu Painter verwendet.
- Für unseren Anwendungsfall reicht es, die Drucktaste BACK zu aktivieren

ABAP Dynpro

Menu Painter



Oberfläche ZTEST_WEB_SERVICE_PROXY aktiv(überarbeitet)

Menüleiste Zeilenauswahlliste

Normen einmischen

Liste Bearbeiten Springen

Code Text

PRI	Drucken
%EX	Sichern/Senden
SAVE	Beenden
	Sichern

MYFUNCTION Meine Funktion

Drucktastenleiste

Positionen 1 - 7	PICK	%PRI	%SC	%SC+	RW		
Positionen 8 - 14	MYFUNCTION						

Funktionstasten Zeilenauswahlliste

Symbolleiste

SAVE	BACK	%EX	RW	%PRI	%SC	%SC+	P--	P-	P+
------	------	-----	----	------	-----	------	-----	----	----

Empfohlene Funktionstastenbelegungen

F2	PICK	Auswählen
F9	<.>	Markieren
Umsch-F2	<.>	Löschen
Umsch-F4	<.>	Sich.o.Prüfen
Umsch-F5	<.>	Anderes <Objekt>

Frei belegbare Funktionstasten

F5	MYFUNCTION	Meine Funktion
----	------------	----------------

V2

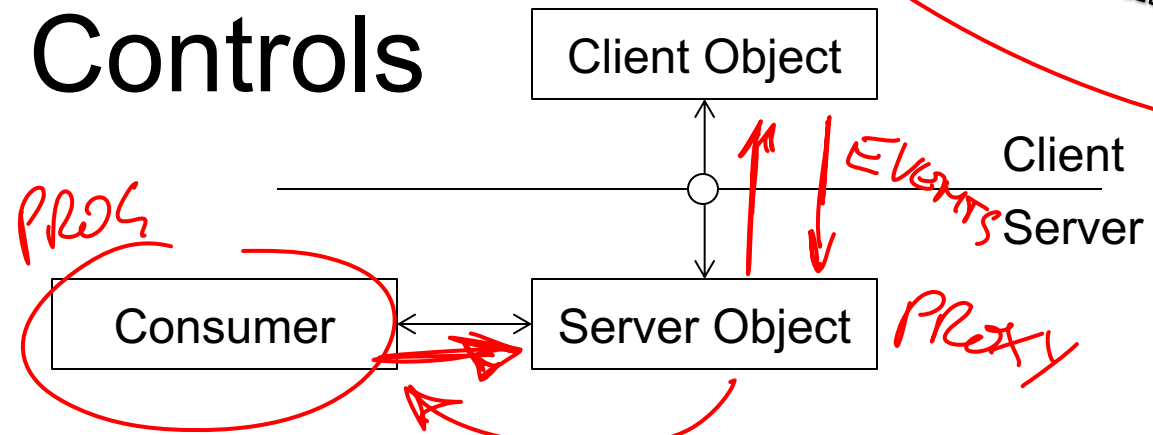
ABAP OO Anwendung ALV

V2: ABAP Klasse CL_GUI_ALV_GRID

GUI Klassen

Enjoy Controls

Control Framework



Container

Enjoy Controls

CL_GUI_CUSTOM_CONTAINER

CL_GUI_ALV_GRID

ABAP Klasse

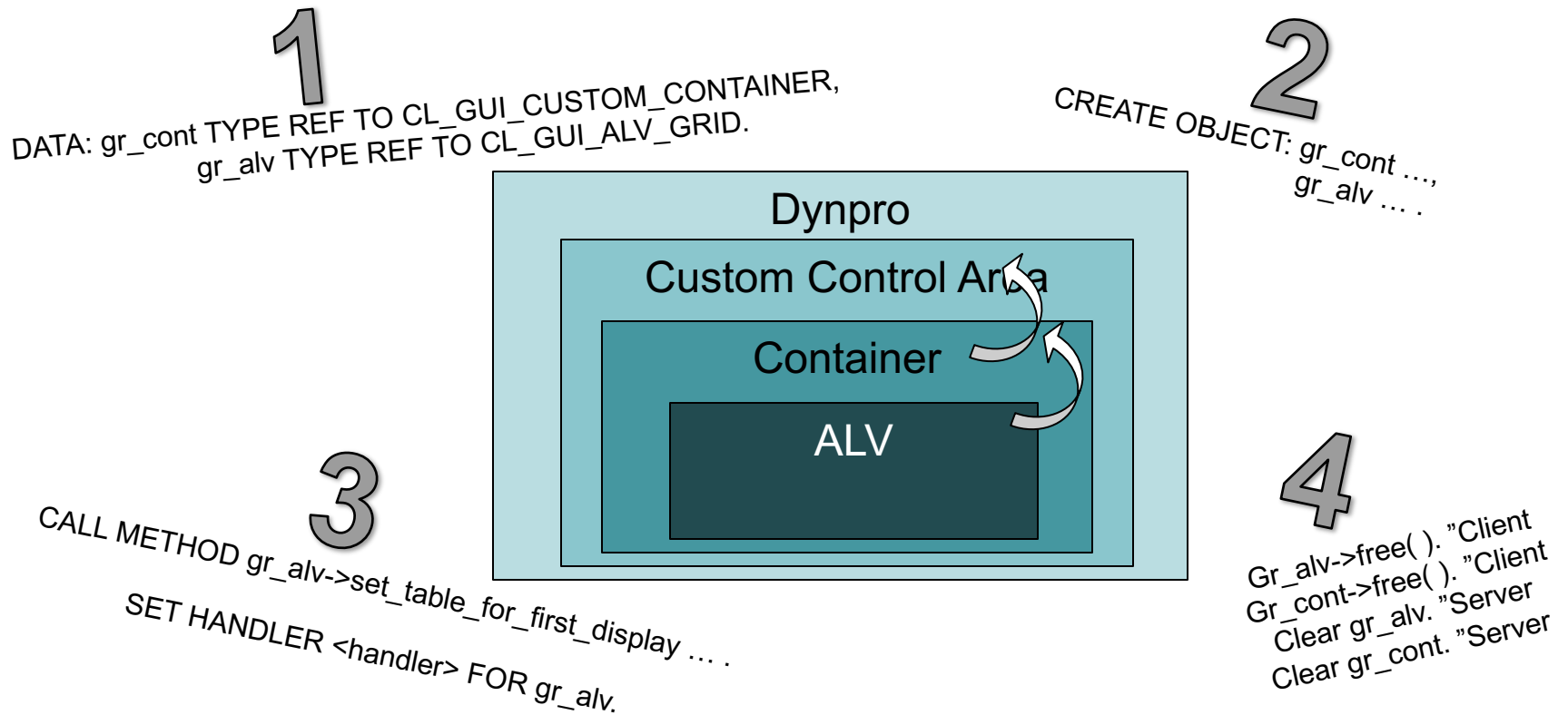


GUI Klassen - Lebenszyklus

1. Deklaration Objektreferenzen für Container und Enjoy Control
2. Instanziierung Container und Enjoy Control
3. Verwendung
4. Abbau der Client und Server Objekte

ABAP Klasse

GUI Klassen - Lebenszyklus



ABAP OO Events

ABAP Klasse



Ereignisse dienen zur losen Kopplung zwischen Objekten.

- Ein Objekte, der **Sender**, schickt Nachrichten.
- Ein/mehrere Objekte, der/die **Empfänger**, empfängt/empfangen die Nachricht

ABAP Objekt

Sender

1. Event Definition: EVENTS
2. Event auslösen: RAISE EVENT
EXPORTING <parameter> = <wert>

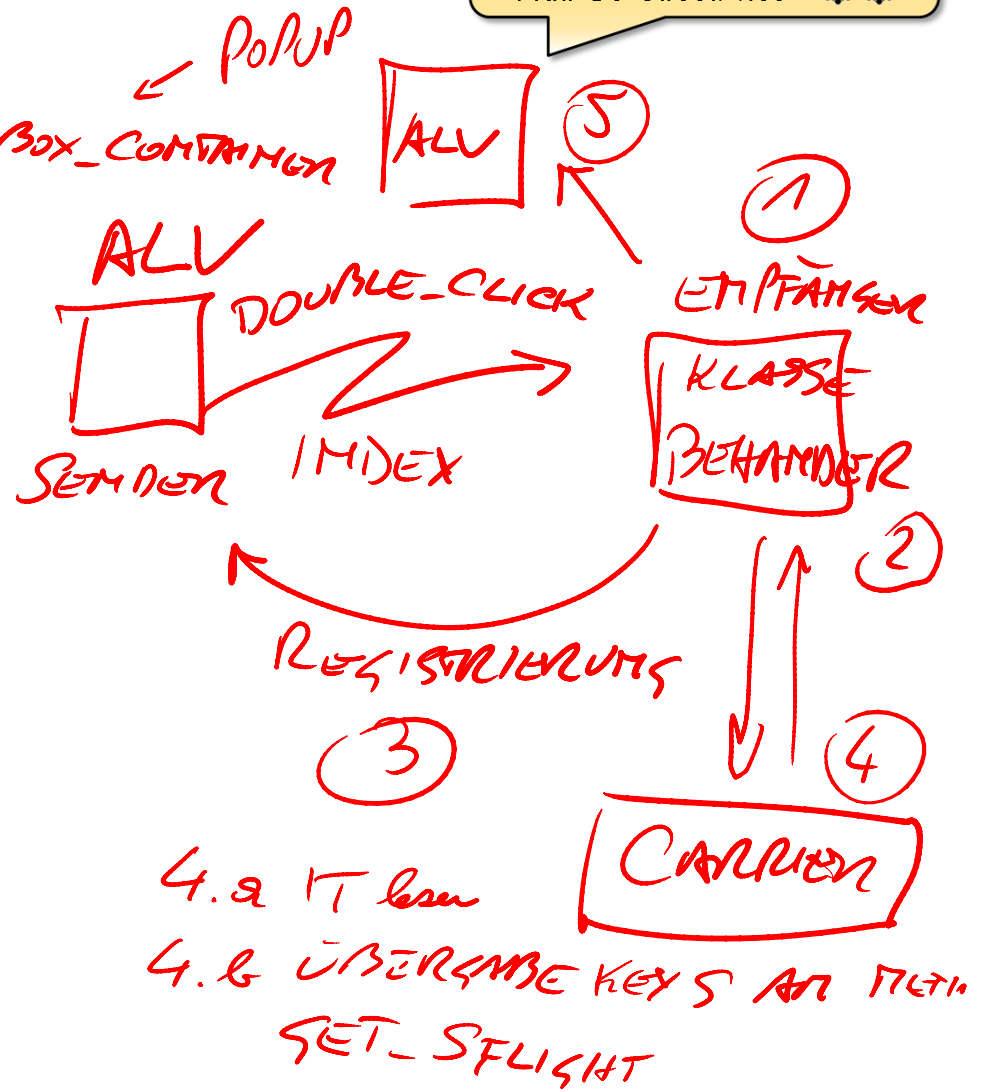
Empfänger

3. Event Handler: ON_ <event> FOR EVENT
<event> OF <sender> IMPORTING sender
<parameter>
4. Behandler registrieren: SET HANDLER
<empfänger>->ON_ <event> FOR <sender>.

ABAP Objekt

Beispiel Doppelklick

- Sender
- Empfänger
- De-/Registrierung
- Behandlung



Done!



Done!

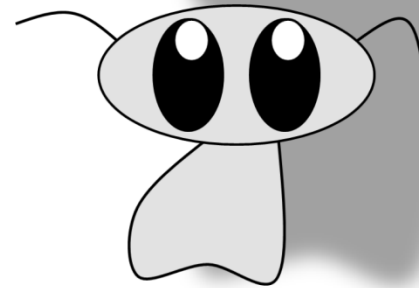
Transaktionscodes

<i>Abkürzung</i>	<i>Erläuterung</i>
SE80	Object Navigator
SE11	ABAP Dictionary
SE24	Class Builder
SE63	Übersetzungswerkzeug für Texte
SU01	Benutzerverwaltung
SICF	Servicebaum
SE84	Repository Infosystem
STMS	Transport und Management System
SE09	Transport Organizer
SE16	Tabellenpflege

Konventionen

*l** Lokal
*g** Global
*i** Input
*r** Returning
*c** Changing
*+d_** elementares Datenobjekt (DO)
*+s_** strukturiertes DO
*+t_** tabellenartiges DO
*pa_** Parameter
*READ_** Lese Op auf DB
*OUTPUT_** Ausgabe Op auf Liste

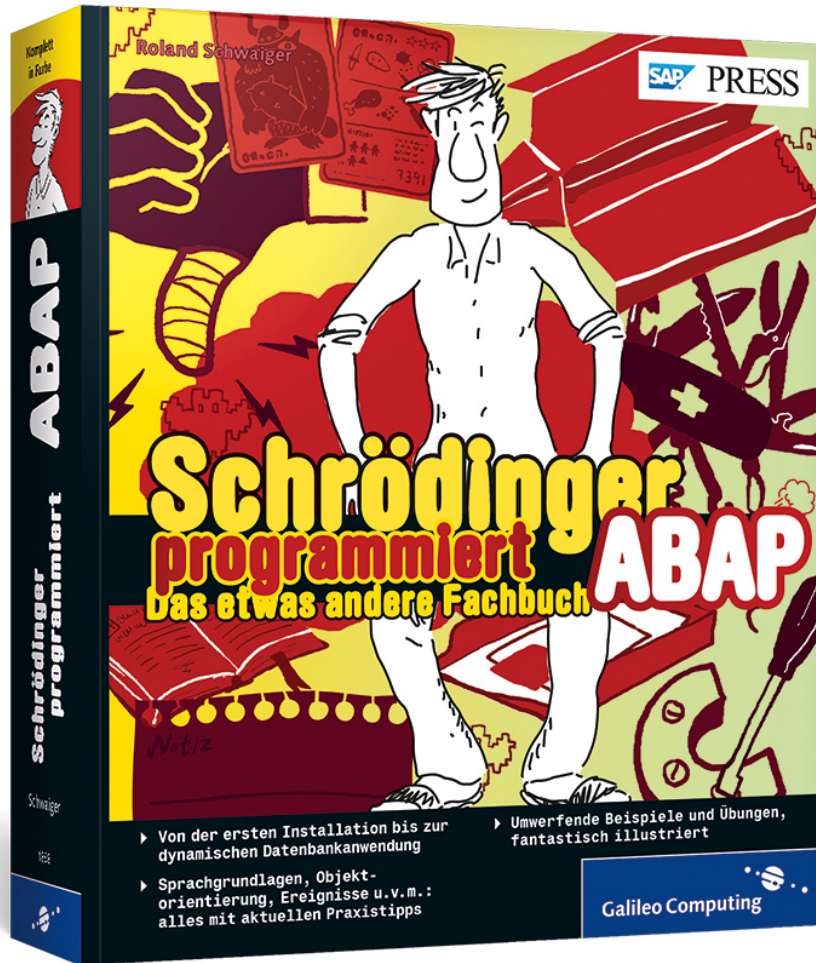
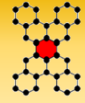
Literaturliste



- www.wikipedia.org
 - sdn.sap.com
 - help.sap.com
 - www.rolandschwaiger.at
-
- (2016) Dominik Ofenloch und Roland Schwaiger (2016). "**Getting Started with Web Dynpro ABAP**" SAP PRESS, End of 11/2016, ISBN(13) 978-1-5922-9311-7
 - (2016) Dominik Ofenloch und Roland Schwaiger (2016). "**Einstieg in Web Dynpro ABAP**" SAP PRESS, Ende 04/2016, ISBN(13) 978-3-8362-1315-8
 - (2008) Roland Schwaiger (2008). "**Sprachen und Standards für IST- und SOLL-Prozessbeschreibungen im betrieblichen Umfeld**", Books on Demand, 2008, ISBN(13) 978-3-8370-6322-6
 - (2016) Roland Schwaiger und Martin Schwaiger (2016). "**Agile Prozessfassung**", Books on Demand, 09.2016, ISBN(13) 978-3-8391-6919-3

Möge der ABAP mit dir sein!

facet.at
Plan-Do-Check-Act

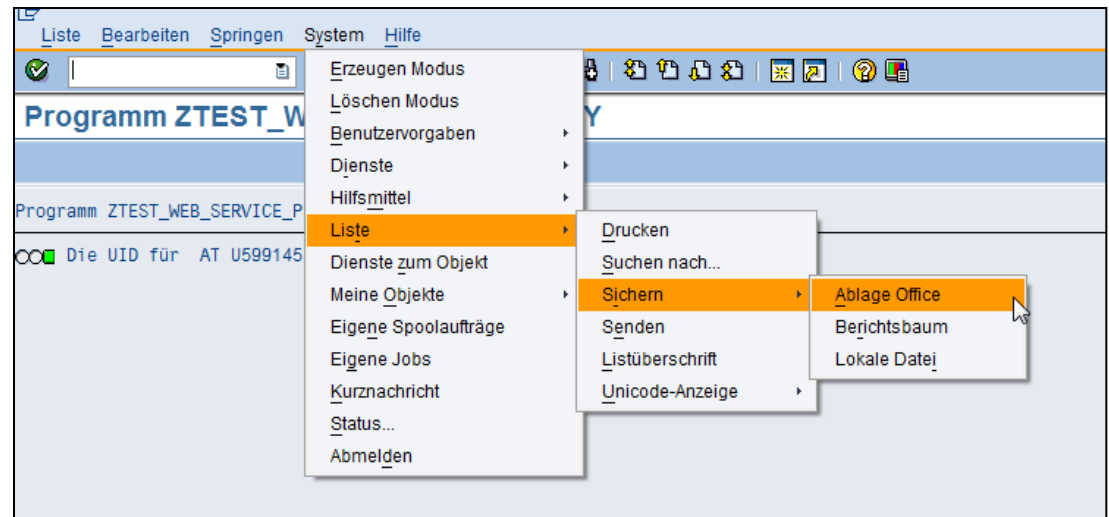
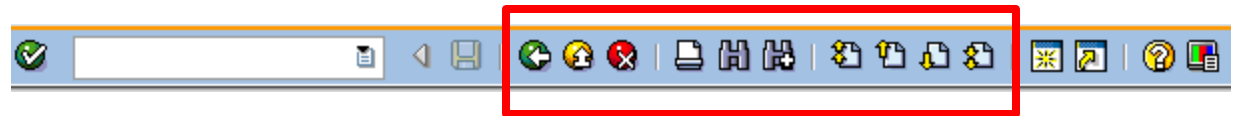


ABAP Liste

ABAP Liste

Standard Funktionen

- Suche
- Blättern
- Exportieren
- ...



ABAP Liste

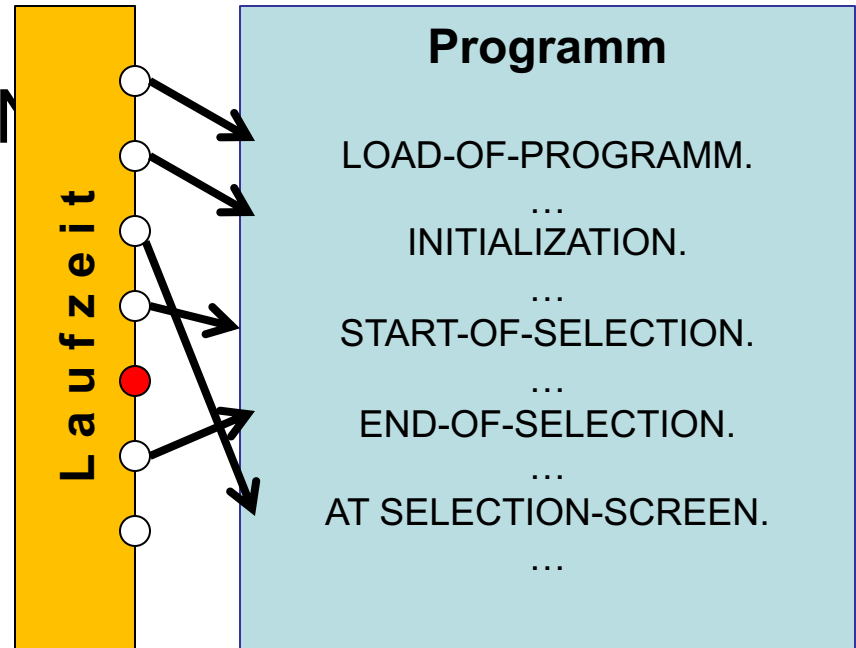
Ereignisse

- START-OF-SELECTION
- END-OF-SELECTION
- TOP-OF-PAGE
- ...

```
TYPE-POOLS: abap,  
            icon.  
START-OF-SELECTION.
```

```
* Gültigkeit ausgeben
```

```
IF ls_response-valid = abap_true.  
  WRITE: / icon_green_light AS ICON, 'Die UID für ', pa_ms, pa_vat, ' ist gültig!'.  
ELSE.  
  WRITE: / icon_red_light AS ICON, 'Die UID für ', pa_ms, pa_vat, ' ist ungültig!'.  
ENDIF.
```



ABAP OO

ABAP Klasse

Erbung/Vererbung

- Redefinition
- Casting (Up/Down)

