


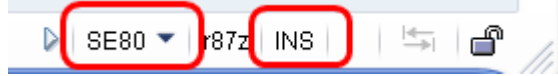


# Übung 1 "Einführung in ABAP"

## Aufgabe 1: SAP-Einführung - Navigation.

### Anmeldung am SAP-System:

Mandant: **101**  
Benutzer: **DEVELOP###** (### zugeteilte Benutzernummer)  
Kennwort: (siehe Vorlesung) (beim Einwählen nach Aufforderung ändern)  
Sprache: **de** (i.d.R. auf **de** voreingestellt)

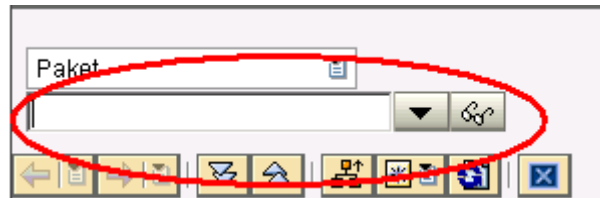
1. Öffnen sie das Eingabefeld für Transaktionen (unter dem ersten Menü), falls es nicht geöffnet ist:
- 
2. Stellen sie das Benutzermenü für den Benutzer **DEVELOP###** ein, falls es nicht angezeigt wird (erster Button in der Funktionsleiste ). Es sollte der Knoten "**Benutzermenü für DEVELOP###**" sichtbar sein.
  3. Öffnen sie den Funktionsbaum für den Benutzer, bis sie den **Object Navigator** sehen (*Benutzermenü > Übersicht > Object Navigator*). Fügen sie das Programm **Object Navigator** per "drag" und "drop" zu den Favoriten hinzu.  
Sie finden die SAP-Funktionen auch im **alternativen SAP Menü** (zweiter Button in der Funktionsleiste ).
  4. Ändern sie die Einstellungen für den Funktionsbaum, so dass die **technischen Namen** im Funktionsbaum angezeigt werden (*Menü Zusätze > Einstellungen*). Öffnen sie den Funktionsbaum nochmals, bis sie den **Object Navigator** sehen. Sie sollten jetzt die Kurzbezeichnung (techn. Namen) der Funktionen sehen.
  5. Stellen sie das System so ein, dass in der Statuszeile (rechts unten) immer der Transaktionscode und der INS-Modus eingestellt ist:
- 
6. Lassen sie sich die Systeminformationen anzeigen (Benutzereigenschaften, SAP-Rechner etc.) (*Menü System > Status*).
  7. Legen sie einen neuen Ordner **WWW** im Favoritenordner an. Fügen sie die Web Adresse (URL) der THM ein (*Kontextmenü – Einfügen weiterer Objekte – Web Adresse*). Rufen sie über den Favoriteneintrag die Webseite auf.

## Aufgabe 2: Arbeiten mit dem Object Navigator (ABAP Workbench)

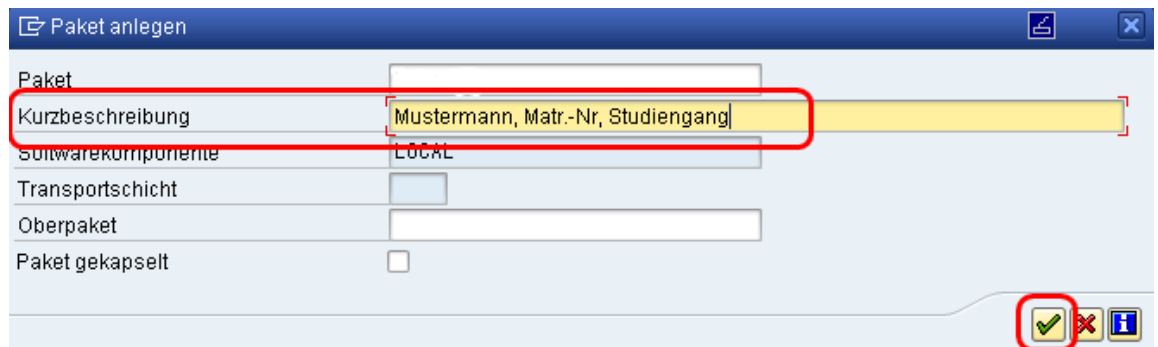
**Vorarbeiten (einmalig):** Festlegen eines **Paketnamens** (Entwickungsklasse).



Das Paket ist ein Oberbegriff (Ordner, Klasse ..), unter dem sie Ihre Programme und alles was sie zu diesem Programm erstellen wieder finden. Den Namen müssen sie wie folgt wählen: **\$SS17\_###**, wobei **###** Ihre Benutzernummer im Benutzernamen ist. **\$** am Anfang des Namens bedeutet, dass die Programme dieses Pakets nicht transportiert werden. Gehen sie wie folgt vor:

1. Öffnen sie den Object Navigator.
2. Wählen sie im ersten Feld **Paket** aus und geben sie im 2. Feld ihren **Paketnamen** **\$SS17\_###** ein (**###** steht für Ihre Benutzernummer) und drücken sie den Funktionsknopf mit der Brille.



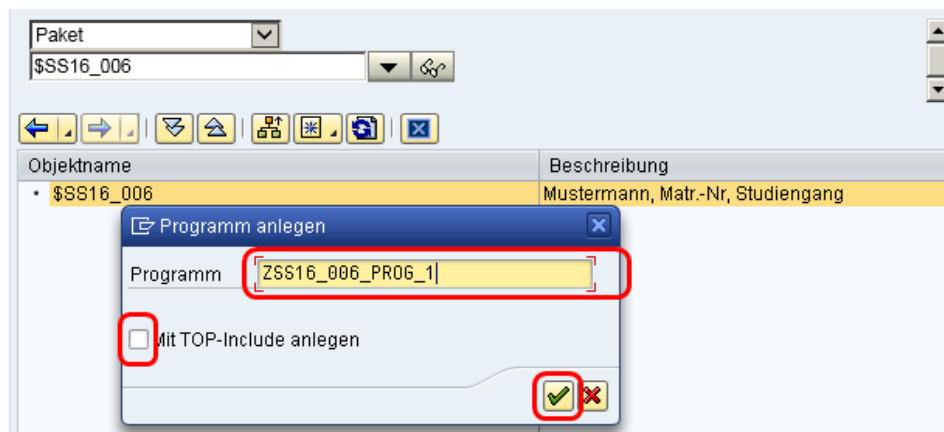
3. Im nachfolgenden Menü tragen sie bitte nur im 2. Feld ihren **Vor-, Nachnamen** und den Studiengang ein, speichern sie die Daten ab.



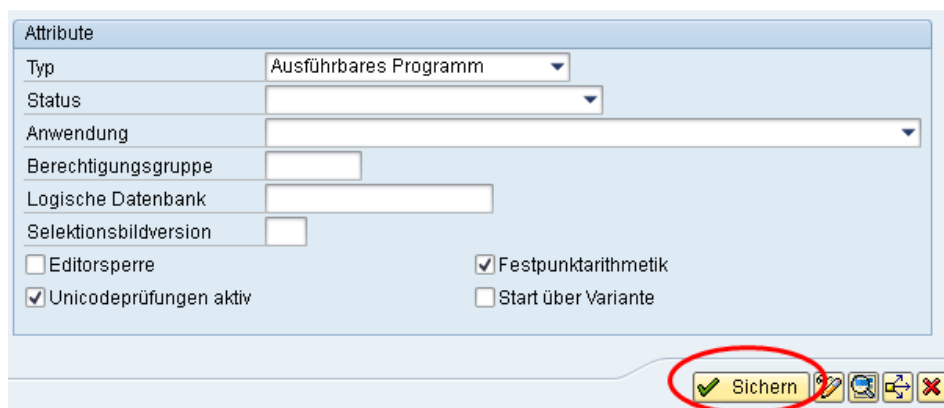
4. Im **Object Navigator** gibt es die Möglichkeit, eine persönliche Favoritenliste mit dem Paket und den entwickelten Programmen anzulegen. Drücken sie die Funktion  (Funktionsleiste des **Object Navigators**) und wählen sie hinzufügen. Das eingestellte Paket wird zu den persönlichen Favoriten hinzugefügt..
5. Mit den blauen Pfeilen  (bzw. über die Listen) in der Funktionsleiste des **Object Navigators** kann man später sehr schnell zu vorherigen oder nachfolgenden Bildern bzw. Verarbeitungsfunktion springen, die man innerhalb einer Sitzung angewählt hat.

**Ein (das erste) Programm eingeben:** Gehen sie wie folgt vor:

1. Machen sie in der Objektliste auf dem Paketnamen mit der rechten Maustaste einen Klick und wählen sie **Anlegen > Programm** in dem Kontextmenü aus.
2. Legen sie den Programmnamen fest. **WICHTIG:** Da der Programmname im ganzen System eindeutig sein muss, müssen sie den Namen wie folgt bilden: **ZSS17\_###\_programmname**. Der erste Buchstabe **Z** ist für ein Benutzerprogramm zwingend vorgeschrieben! Dahinter folgt eine Kennung für das Semester **SS17**. Für **###** tragen sie wieder Ihre Benutzernummer ein, gefolgt von einem Unterstrich **\_**. Den restlichen Teil des Programmnamens können sie frei wählen. **Legen sie das Programm immer ohne TOP-Include an (Selektion entfernen, falls diese gesetzt ist!).**



3. Im nächsten Bild machen sie keinen Eintrag, nur "sichern" drücken.



**Wichtig:** Zum endgültigen Speichern des Katalogeintrags müssen sie auf das **Diskettensymbol drücken**, sonst wird ihr Programm im Standard-Paket **\$TMP** abgelegt! Sollte das Paket nicht voreingestellt sein müssen sie noch ihr Paket eingeben.

4. Nach dem Speichern wird ein neuer Ordner **Programme** mit ihrem Programm angelegt. In diesem Ordner finden sie später alle Programme wieder. Im rechten Fenster (Editor) wird ihr Programm angezeigt.

5. Geben sie das erste Programm (siehe Folie Vorlesung) ein. Speichern sie das Programm, **prüfen**, **aktivieren** und **starten** sie es anschließend mit . Mit den Pfeiltasten (Pfeil nach links oder oben) in der SAP Funktionsleiste kehren sie wieder zurück zum Object Navigator. Beispielprogrammname **ZSS17\_###\_UEB1\_A2**)

```

1  *&-----
2  *& Das erste ABAP-Programm
3  *&-----
4  REPORT z_erstes_programm .
5
6  START-OF-SELECTION .
7      WRITE 'Hello World' .           "Textausgabe
8      SKIP .
9      WRITE / ' ' 'Noch eine Ausgabe' ' .
10     ULINE .

```



**Wichtig:** Beim wiederholten Öffnen eines Programms müssen sie es zum Schreiben öffnen (Button drücken).

### Aufgabe 3: Ein kleines ABAP-Programm

Schreiben sie ein ABAP-Programm, das u.a. den angemeldeten Benutzer, das Systemdatum und die Systemzeit in einem Rahmen wie folgt ausgibt. Programmname sollte ZSS17\_###\_UEB1\_A3 sein.

Benutzername: BCUSER	
Hier steht weiterer Text ....	
Datum 15.03.2003	Zeit 17:39:35

Das Datum steht linksbündig, die Uhrzeit rechtsbündig innerhalb des Rahmens. Benutzen sie ausschließlich die **WRITE**-Anweisung mit der Angabe des Ausgabebereichs und der Spaltenpositionierung.

**Hinweis:** Machen sie pro Ausgabezeile eine WRITE-Anweisung mit Hilfe der Kettensatzoption!

### Aufgabe 4: Ausdruck von festen Werten mit WRITE

Schreiben sie ein ABAP-Programm, das nachfolgende Ausgabe einer Artikelliste d.h. Felder als Konstanten ausgibt. Benutzen sie für die globalen Farbeinstellungen die **FORMAT**-Anweisung, für die partiellen Farbeinstellungen die **COLOR**-Option der WRITE-Anweisung. Programmname sollte ZSS17\_###\_UEB1\_A4 sein.

Artikelliste				
Lfd-Nr	Artikel	Anzahl	Einzelpreis	Preis
1	Jacke	5	20,50	102,50
2	Hose	2	30,55	61,10
3	Schuhe	3	60,25	180,75
4	Hemd	4	20,00	80,00
Summe				424,35
Skonto				10,61
Summe				413,74

## Aufgabe 5: Zuordnung eines Transaktionscodes

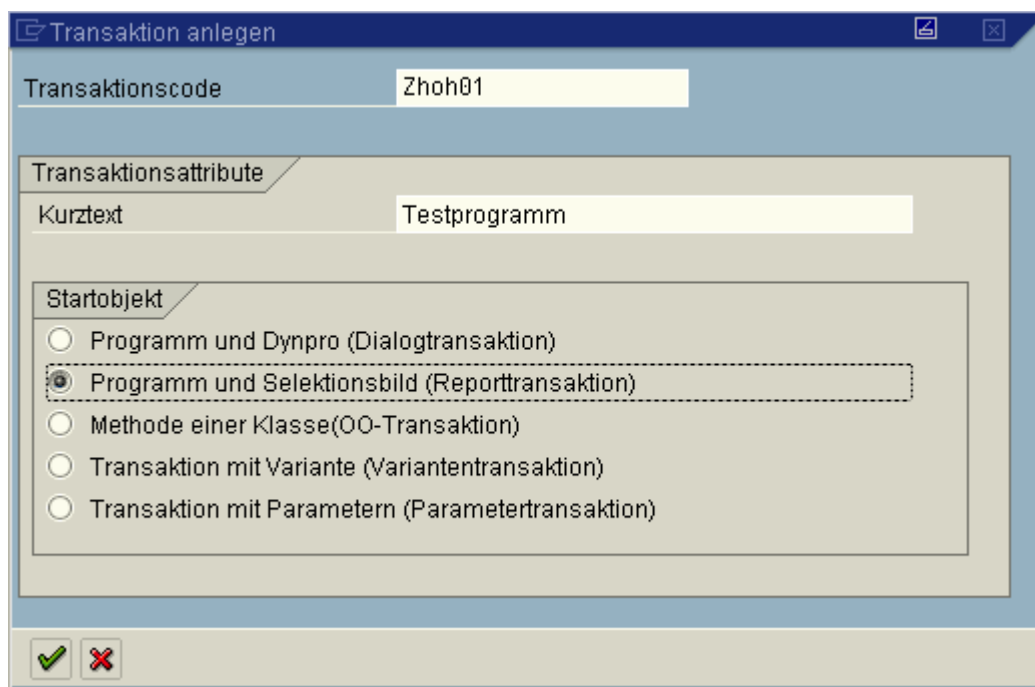
Schreiben Sie folgendes Programm mit dem Namen ZSS17\_###\_UEB2\_A5. Mit dem Programm und dem Befehl PARAMETERS erhalten Sie einen automatischen Eingabebildschirm (Selektionsbildschirm), der in SAP Dynpro genannt wird. Die Eingabe im Feld PARAMETERS wird anschließend mit dem Befehl WRITE ausgegeben.

```
PARAMETERS: eing(20) TYPE c.
```

```
WRITE eing.
```

Übersetzen und aktivieren Sie das Programm. Alternativ zum technischen Namen soll nun das Programm auch über einen Transaktionscode verfügen. Legen Sie die Transaktion unter die rechte Maustaste mit „Anlegen/Transaktion“ mit dem Namen „ZSS17###“ an. Um dies zu können, müssen Sie das Programm im Fenster „Objektname“ markieren. Dann folgen Sie den Dialogen.

### 1.Schritt: Typisierung der Transaktion



Zhoh01 wird durch ihren Programmnamen ZSS17### ersetzt. Der Programmtyp (Startobjekt) ist „Programm und Selektionsbild“, da der Dnpro mit PARAMTERS erzeugt wurde..

## 2.Schritt: Eingabe der Details

Transaktionscode Bearbeiten Springen Hilfsmittel Umfeld System Hilfe

**Reporttransaktion anlegen**

Transaktionscode: ZHOH01  
Paket: ZHOHMANN  
Transaktionstext: Testprogramm  
Programm: z01\_blat1\_ueb4  
Selektionsbild: 1000  
Start mit Variante:   
Berechtigungsobjekt:   
Werte

Objektname  
Z01\_BLATT1\_UEB4  
Felder  
EING

Erfolgreich erstellte Objekte z.B. Transaktionen werden hier angezeigt.

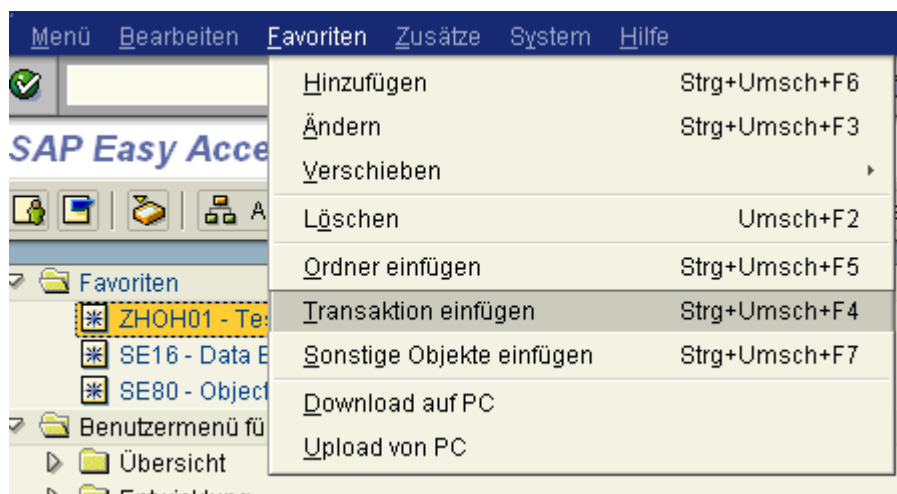
1000=Dynpro/Selektionsbild ist der Name in SAP für eine Dialogmaske mit dem Anwender. Selektionsbilder die mit Parameters erzeugt werden, haben automatisch die Nummer 1000. Eigenprogrammierte Dialogbilder können mit einer eignen Nummer z.B. 100 versehen.

Das Objekt wird mit der Originalsprache Deutsch (DE) angelegt

Der durch

## 3.Schritt: Transaktion im Menü von ABAP

Verlassen Sie nun den Objekt Navigator. Binden Sie ihre Transaktion in das Menü von SAP ein. Rufen Sie Ihr Programm aus dem Menü auf.



Transaktionen sind nun kein Geheimnis mehr für SIE.

## Aufgabe 6: Fügen Sie Kommentare in ihre erstellten Programme ein

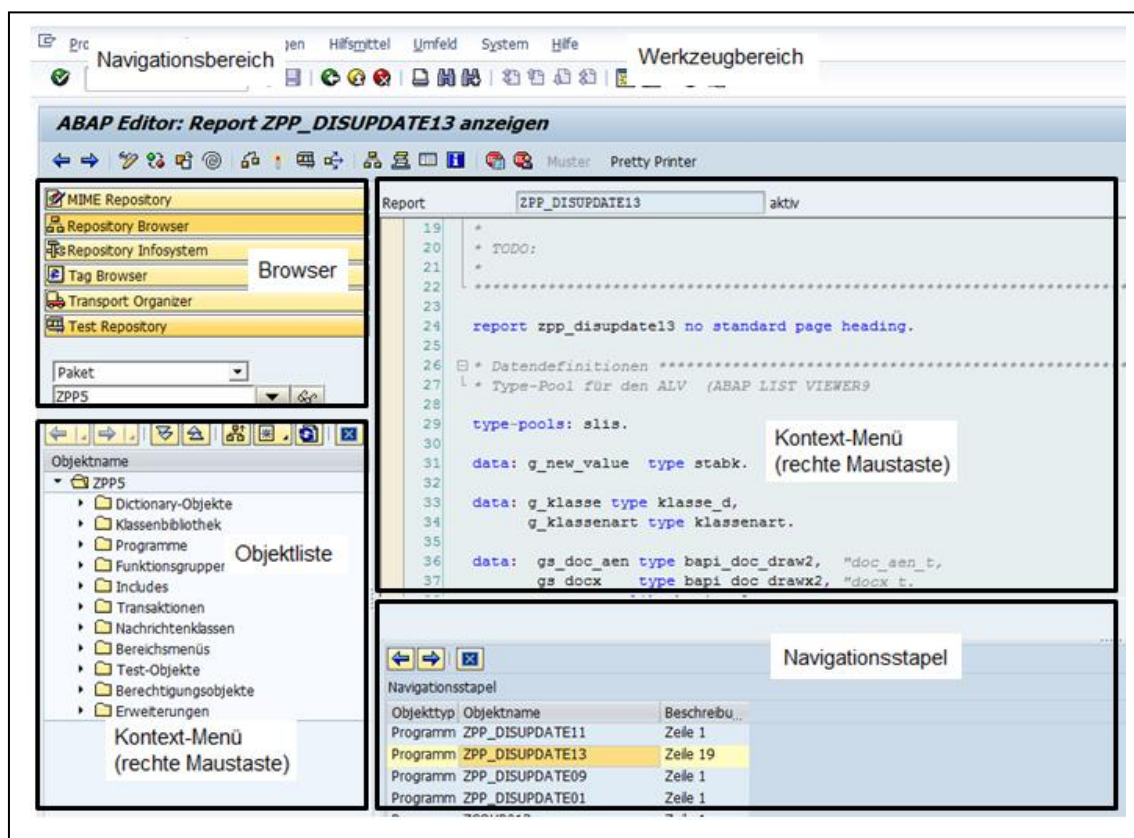
Rufen Sie ihre erstellten Programme auf (ZSS17\_###\_UEB1\_A#) und fügen Sie sinnvolle Kommentar ein:

- \* = Eine Kommentarzeile beginnt mit einem Stern in der 1.Spalte
- „ = Einleiten eines Kommentars an einer beliebigen Stelle im Sourcecode

## Aufgabe 7: Machen Sie sich nochmals mit den Instrumenten von ABAP vertraut

### Repository Browser

Mit dem Repository Browser werden Entwicklungsobjekte angezeigt, aufgerufen und bearbeitet. Komponenten der Oberfläche





Navigationsbereich/links

Browser

Objektauswahlliste

Objektliste

Werkzeugbereich/rechts

Objekt mit Werkzeugen

Navigationsstapel

Die Entwicklungsobjekte werden in **Objektlisten** zusammengefasst, die sämtliche Elemente gleicher Kategorie z.B. eines Pakets, einer Funktionsgruppe oder eines Programms enthalten.

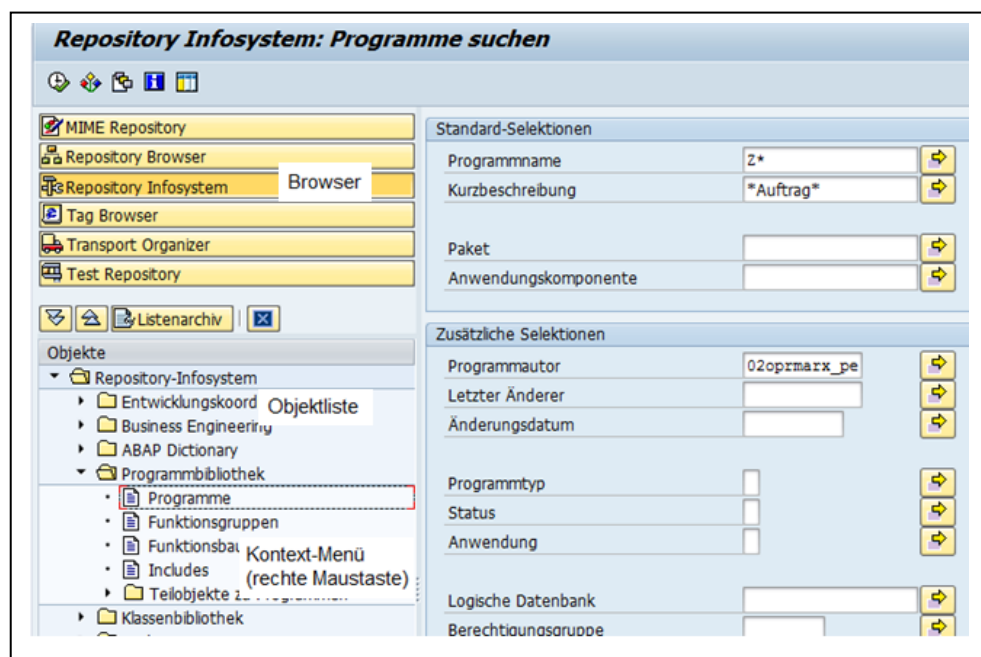
Ausgehend von der Objektliste wird das jeweilige Objekt mit Doppelklick ausgewählt. Damit wird automatisch das entsprechende Werkzeug aufgerufen, mit dem das Objekt erstellt wurde.

Häufig genutzte Objektlisten können in die Favoriten übernommen werden.

**Repository Infosystem**

Das Repository Infosystem wird verwendet um Objekte im System zu suchen.

Komponenten der Oberfläche



Es wird eine Objektliste aller Objektarten in Baumstruktur dargestellt. Bei Doppelklick auf eine bestimmte Objektart bekommen Sie ein Selektionsbild, über das Sie Ihre Suche abgrenzen können.

Es kann die Suche nach Objekten, deren Verwendung und die Umfeldermittlung von Objekten durchgeführt werden (Kontext-Menü).

## Paket anlegen

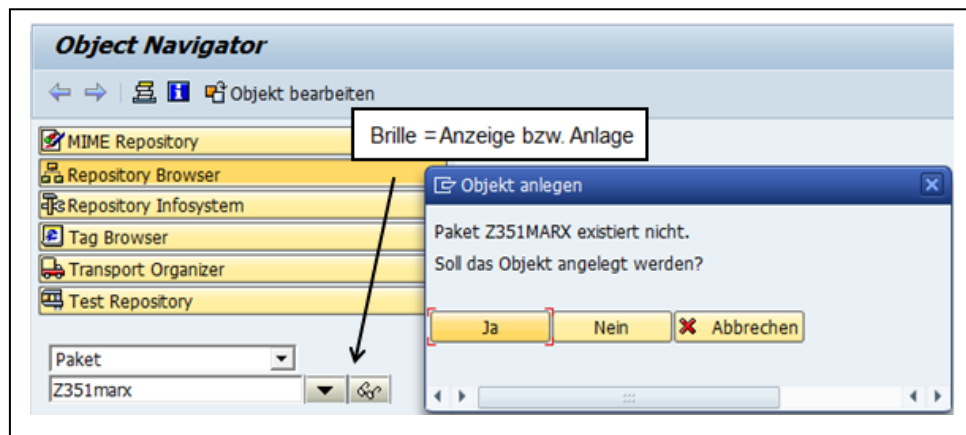
Object Navigator starten (Repository Browser)

### Alternative 1

Objektart Paket und Namen für das neue Paket in der Objektlistenauswahl eintragen.

(Kundenobjekte müssen mit den Buchstaben Z oder Y beginnen)

Drucktaste ‚Anzeigen‘ wählen (oder Enter) und Frage mit ‚Ja‘ beantworten



### Alternative 2

Button „Objekt bearbeiten“ – Reiter Pakete – Paketnamen eingeben – Button „Anlegen“

Attribute im Bildschirm „Paket anlegen“ erfassen. (Detailinfo F1)

**Kurzbeschreibung:** Bezeichnung für das Paket

**Anwendungskomponente:** Siehe Struktur Repository. Das Repository ist in Anwendungskomponenten unterteilt. Diese wiederum in Pakete.

**Softwarekomponente:** Die Softwarekomponente beschreibt eine Menge von Softwareobjekten, die nur gemeinsam auslieferbar sind. Bei Kundenentwicklungen ist die Softwarekomponente immer **Home**.

## Programm anlegen

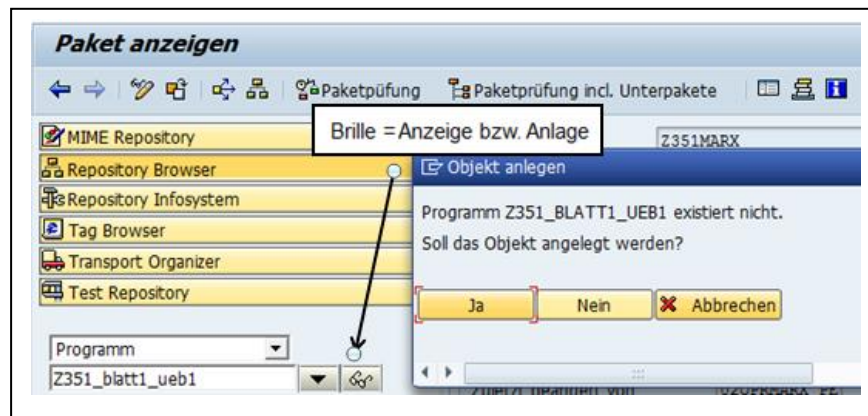
Object Navigator starten (Repository Browser).

### Alternative 1

Objektart Programm und Namen Objektart Paket und Namen für das neue Programm in der Objektlistenauswahl eintragen.

(Kundenobjekte müssen mit den Buchstaben Z oder Y beginnen)

Drucktaste ‚Anzeigen‘ wählen (oder Enter) und Frage mit ‚Ja‘ beantworten



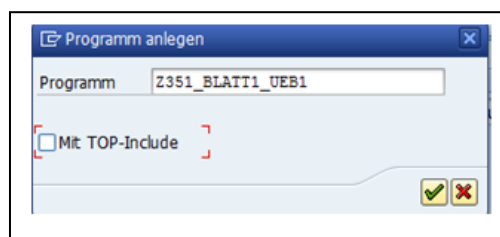
### Alternative 2

Button „Objekt bearbeiten“ – Reiter Programm – Programmnamen eingeben – Button „Anlegen“

### Alternative 3

Paket auswählen zu dem das Programm angelegt werden soll. Über das Kontextmenü zum Paket oder zum Knoten Programme Programm anlegen.

Kennzeichen „Mit TOP-Include“ entfernen (dient der Modularisierung, es würden mehrere Programmteile angelegt).



Attribute im Bildschirm „Programmeigenschaften“ erfassen. (Detailinfo F1)

**Titel:** Überschrift des Programms. Der Titel wird bei der Programmausführung in der Titelzeile angezeigt.

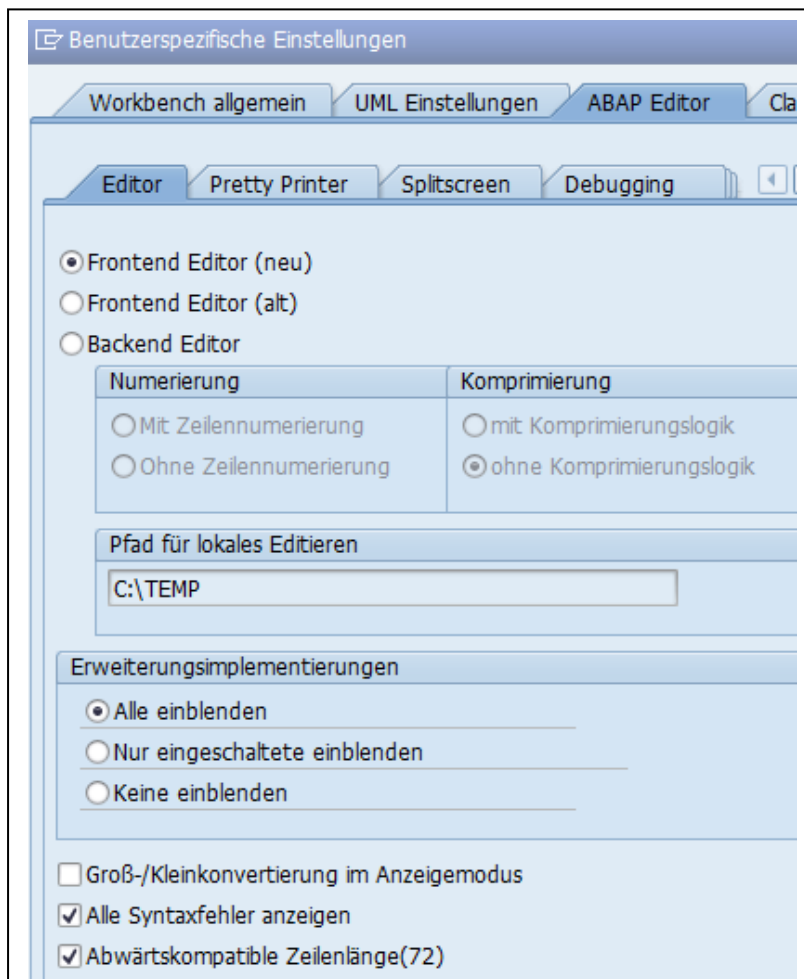
**Typ:** Der Typ gibt an, um welche Art von Programm es sich handelt.

## Die wichtigen Funktionen des ABAP-Editor

Zur Entwicklung von Programmen stehen mehrere Editoren zur Verfügung. Der Backend-Editor ist der traditionelle Editor (Zeileneditor).

Mit den beiden Frontend-Editoren (alt + neu) wurden eine Vielzahl Verbesserungen und Zusatzfunktionen zur Verfügung gestellt.

Der gewünschte Editor und weitere Parameter können über das Menü *Hilfsmittel – Einstellungen-ABAP Editor – Reiter Editor* ausgewählt werden.



## Wichtige Hilfe-Funktionen

Cursor auf eine Anweisung stellen und **F1-Taste** führt zur Dokumentation der Anweisung

**Button ,i‘** führt zur Suche in der Schlüsselwortedokumentation

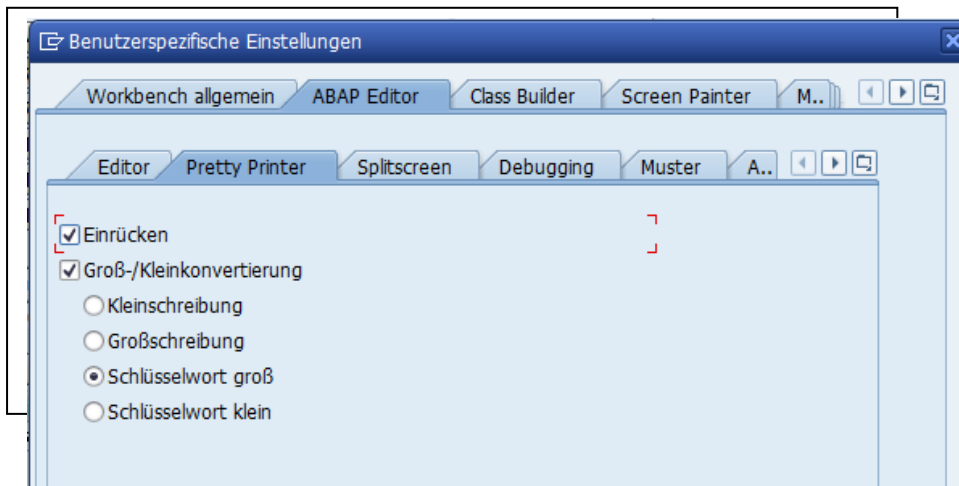
Der **Button ,Muster‘** ist eine wirksame Hilfe für Programmierer. Für einige Anweisungen mit umfangreichen Code oder Schnittstellen kann der Befehl direkt ins Programm übernommen werden.

Nach der Eingabe von Zeichen macht der Editor einen **Vervollständigungs-vorschlag**. Mit der Tab.-Taste kann der Vorschlag übernommen werden.

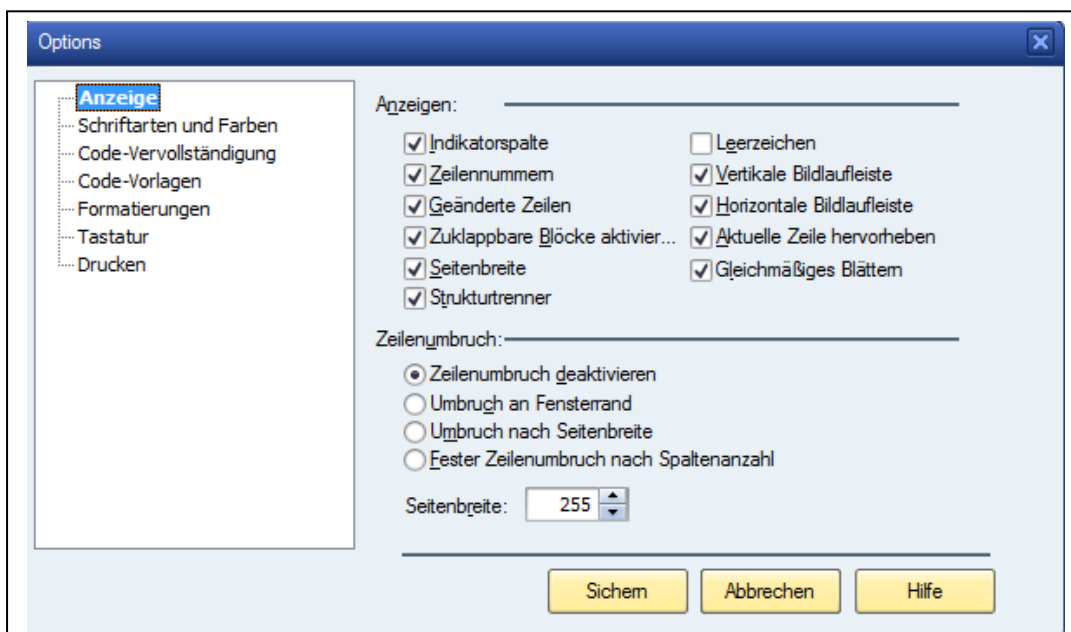
## Pretty Printer

Das ABAP-Laufzeitsystem unterscheidet nicht zwischen Groß- und Kleinschreibung bei Schlüsselwörtern, Zusätzen und Operanten.

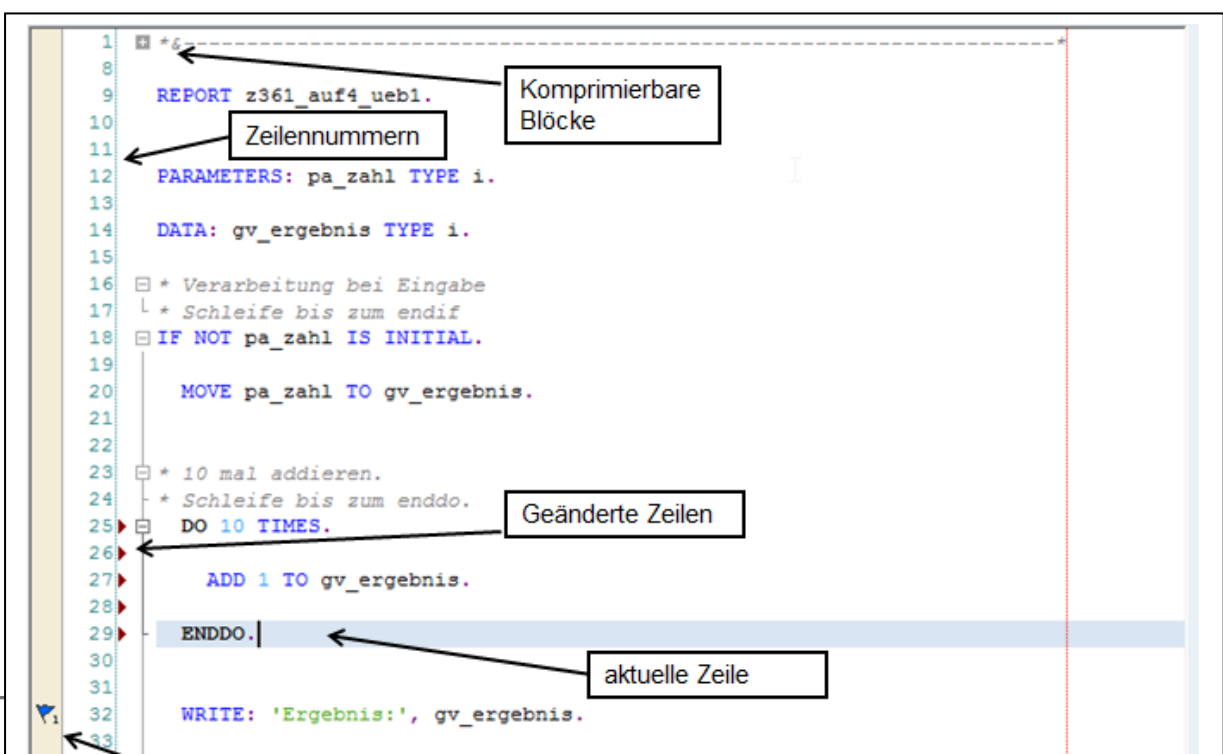
Um die Lesbarkeit eines Programms zu verbessern sollten jedoch die Einstellungen des ‚Pretty Printer‘ genutzt werden.



### Benutzer-Einstellungen



### Beispielprogramm



## Debugging

Logische Fehler können zur Laufzeit mit dem Debugger gesucht werden. Der kann auf verschiedene Arten gestartet werden.

### Alternative 1

Im Object Navigator das Programm markieren und über das Kontextmenü *Ausführen – Debugging*

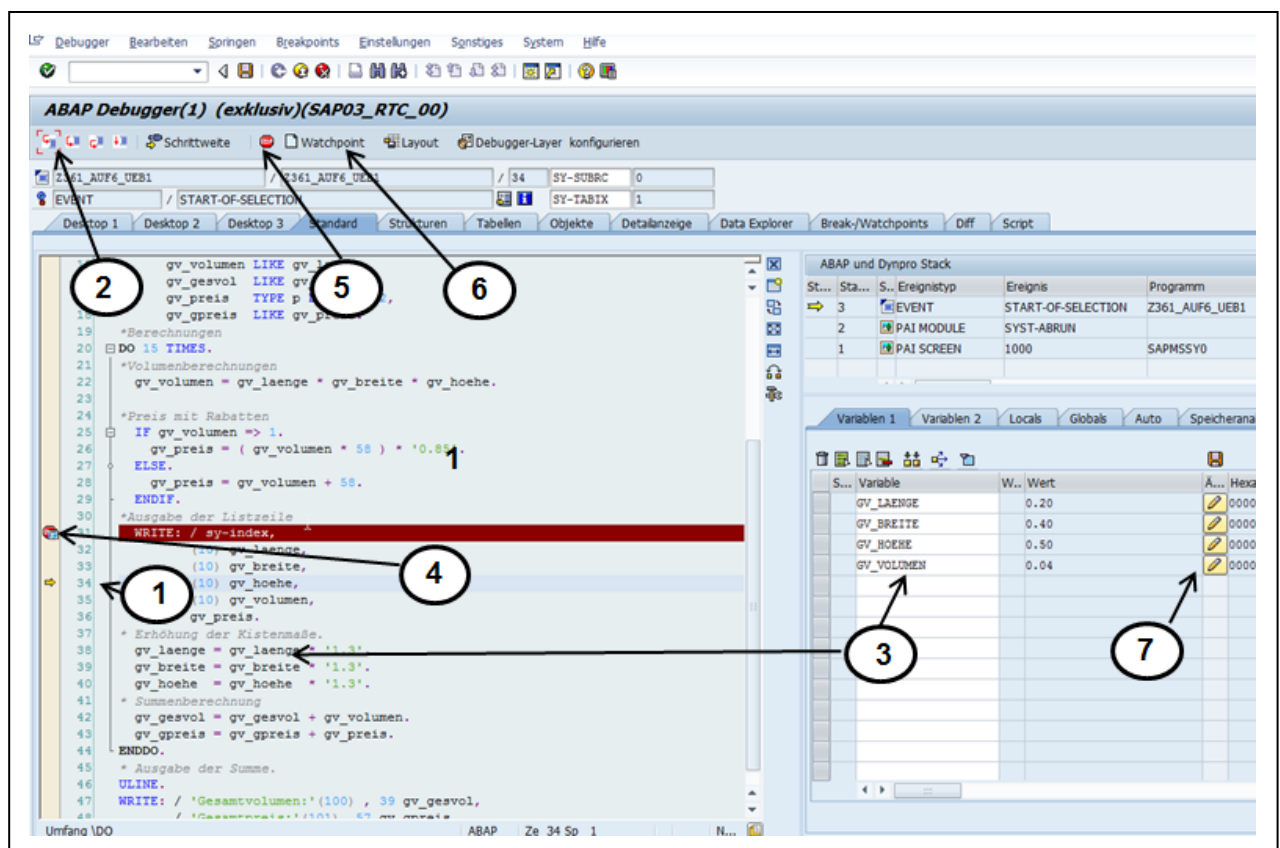
### Alternative 2

Markieren Sie im Werkzeugbereich die gewünschte Programmzeile ab der Sie debuggen wollen. Anschliessend starten Sie das Programm mit F8 oder über das Kontextmenü *Ausführen – Direkt*

### Alternative 3

Bei der Ausführung des Programms an jeder beliebigen Stelle im Kommandofeld */h* eingeben. Oder über das Menü – *System – Hilfsmittel – Debugging*

## Funktionen des Debuggers



1. Aktuelle Anweisung (gelber Pfeil).
2. Einzelschritt ausführen.
3. Doppelklick auf Datenobjekte. Die Objekte werden in die Variablenliste übertragen und ihr Inhalt wird angezeigt.
4. Breakpoint setzen/löschen per Mausklick.

5. Haltepunkt bei einer Anweisung wählen. Das Programm wird danach bis zur gewählten Anweisung ausgeführt (Ausführung mit dem Button „Weiter“).
6. Watchpoint setzen. Dieser Haltepunkt wird abhängig vom Feldinhalt eines Datenobjektes gesetzt.

Möglichkeit 1

Ohne Bedingung = der Debugger hält bei jeder Änderungen des Datenobjektes.

Möglichkeit 2

Mit Bedingung = der Debugger hält wenn die Bedingung erfüllt ist.

7. Während des Debuggings können Feldinhalte geändert werden (Button ‚Stift‘ und Änderung mit ‚Enter‘ bestätigen).

# Notizen