



## Leseprobe

Dieses Buch ist Ihnen ein unverzichtbarer Begleiter im ABAP-Kosmos. In dieser Leseprobe zeigt Ihnen Autor Felix Roth, wie Sie das ABAP Dictionary einsetzen. Lernen Sie, wie Sie Datendefinitionen anlegen und verwalten sowie datenbankunabhängige Entwicklungsobjekte wie Tabellen, Strukturen, Datenelemente und Domänen erstellen.

-  **»Das ABAP Dictionary«**  
**»Einleitung«**
-  **Inhaltsverzeichnis**
-  **Index**
-  **Der Autor**
-  **Leseprobe weiterempfehlen**

Felix Roth

### **ABAP Objects – Das umfassende Handbuch**

1.056 Seiten, gebunden, Dezember 2016  
69,90 Euro, ISBN 978-3-8362-4270-7

 [www.sap-press.de/4201](http://www.sap-press.de/4201)

*Das ABAP Dictionary ist die zentrale Stelle für alle im System vorhandenen Datendefinitionen und damit unerlässlich für jede ABAP-Programmierung. Von Tabellen über Strukturen bis hin zu Suchhilfen – hier werden Sie glücklich!*

### 3 Das ABAP Dictionary

Das ABAP Dictionary erreichen Sie über den Transaktionscode SE11. Es ist die zentrale Stelle im System, um Datendefinitionen anzulegen und zu verwalten. Es enthält die Beschreibung aller im System vorhandenen Datenstrukturen und stellt diese allen anderen Systemkomponenten auf Bedarf zur Verfügung. Diese Informationen können auch direkt in ABAP-Anweisungen konsumiert werden.

Das ABAP Dictionary ist darüber hinaus die Schnittstelle zur an das SAP-System angebundenen, unter dem System liegenden Datenbank und damit das Tool, um Tabellen bzw. Views auf dieser Datenbank zu erzeugen und zu verwalten. Dazu kann mit ABAP und Open SQL über das ABAP Dictionary auf die Datenbanktabellen zugegriffen werden, ohne diesen Zugriff (z. B. den Aufbau und Abbau der Verbindung) explizit orchestrieren zu müssen. Diese Art des Zugriffs ist einer der Hauptgründe für die Stärke von ABAP, wenn es darum geht, mit großen Datenmengen umzugehen.

Fast der ganze Funktionsumfang des ABAP Dictionarys ist bereits auf dem Einstiegsbildschirm der Transaktion SE11 zu erkennen (siehe Abbildung 3.1), auch wenn sich hinter den Eingabefeldern VIEW und DATENTYP mehr Auswahlmöglichkeiten verstecken.

Sie können mit dem ABAP Dictionary folgende Objekte anzeigen, bearbeiten und anlegen:

- ▶ Domänen (siehe Abschnitt 3.1)
- ▶ Datenelemente (siehe Abschnitt 3.2)
- ▶ Strukturen (siehe Abschnitt 3.3)
- ▶ Tabellentypen (siehe Abschnitt 3.4)
- ▶ Datenbanktabellen (siehe Abschnitt 3.5)

- ▶ Typgruppen (siehe Abschnitt 3.7)
- ▶ Views (siehe Abschnitt 3.8)
- ▶ Suchhilfen (siehe Abschnitt 3.10)
- ▶ das Sperrkonzept (siehe Abschnitt 3.12)

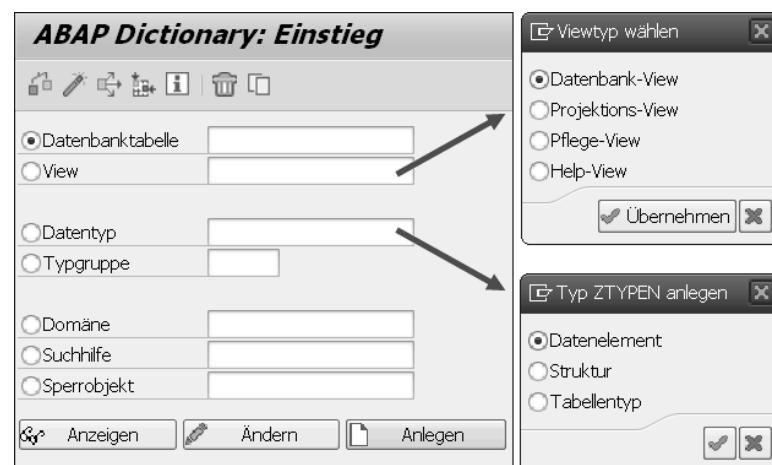


Abbildung 3.1 Einstiegsbildschirm des ABAP Dictionarys

Darüber hinaus bietet das ABAP Dictionary noch etwas verstecktere Funktionen an:

- ▶ Pflegedialoge (siehe Abschnitt 3.9)
- ▶ Datenbank-Utility-Tool (siehe Abschnitt 3.11)
- ▶ Indizes (siehe Abschnitt 3.6)

Im Rahmen der vielen Änderungen in der ABAP-Programmierung, die im Zuge der Einführung von SAP HANA vorgenommen wurden, wurde Transaktion SE11 an einer oder anderen Stelle leicht angepasst. So wurde z. B. die Möglichkeit eingeführt, die Speicherart von Datenbanktabellen zu beeinflussen (siehe Abschnitt 3.5.9) und Volltext-Indizes zu erstellen (siehe Abschnitt 28.5.1, »Volltextindex anlegen«).

#### Beispiel

Wie gut ABAP mithilfe des ABAP Dictionarys mit großen Datenmengen umgehen kann, zeigt das folgende Beispiel. Das folgende ABAP-Programm selektiert 100 Einträge aus der SAP-Standardtabelle für Materialien MARA:

```
DATA: lt_mara TYPE TABLE OF mara.
SELECT * FROM mara INTO TABLE lt_mara UP TO 100 ROWS.
```

Das Programm deklariert lediglich eine interne Tabelle auf Basis der im ABAP Dictionary enthaltenen Tabelle MARA mit all seinen Feldnamen, Datentypen und Feldlängen. Alle Programme greifen, wenn es um Daten geht, also auf das ABAP Dictionary als zentrale Stelle zu. Das heißt, wenn Sie im ABAP Dictionary Änderungen an einer Tabelle vornehmen, müssen Sie keine Änderungen am Quelltext von Programmen vornehmen. Beim nächsten Aufruf des Programms wird automatisch die Änderung festgestellt, und das Programm wird mit den neuen Informationen neu generiert.

## 3.1 Domänen

In diesem Abschnitt erläutere ich das dem ABAP Dictionary zugrunde liegende zweistufige Domänenprinzip sowie die Anlage und den Wertebereich einer Domäne.

### 3.1.1 Das zweistufige Domänenprinzip

Im ABAP Dictionary gibt es ein zweistufiges Domänenprinzip, das technische und semantische Domänen vorsieht:

- ▶ Die *technische Domäne* beschreibt den Wertebereich eines Feldes, der durch die Angabe eines eingebauten Datentyps, der Ausgabelänge und eventueller Festwerte festgelegt wird. Im ABAP-Umfeld werden diese technischen Domänen nur *Domänen* genannt.
- ▶ Die *semantischen Domänen* auf der anderen Seite weisen den technischen Domänen durch die Vergabe von Texten einen bestimmten Zusammenhang zu. Im ABAP-Umfeld werden diese semantischen Domänen *Datenelement* genannt.

Wie in Abbildung 3.2 dargestellt, kann ein Feld einer Struktur oder Tabelle auf ein Datenelement verweisen, das wiederum auf eine Domäne verweist.

Eine Domäne kann demzufolge in mehreren Datenelementen verwendet werden, und ein Datenelement kann in vielen Feldern von Tabellen und Strukturen verwendet werden. Die Domäne fasst also technische Informationen über mehrere Tabellen hinweg zusammen und kann in Form von Datenelementen verschiedene Ausprägungen haben. Darüber hinaus können Sie auf die Datenelemente auch aus ABAP heraus zugreifen und z. B. einen Parameter für eine Eingabemaske definieren.

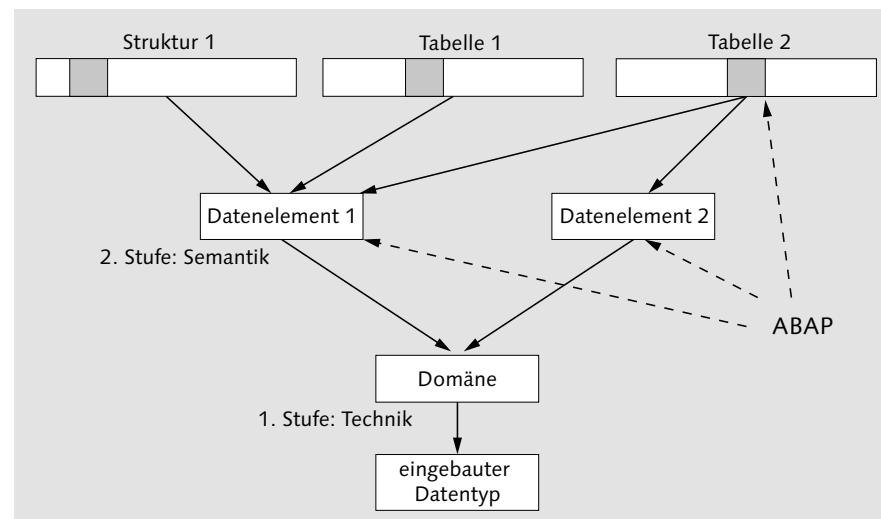


Abbildung 3.2 Das zweistufige Domänenprinzip

### Beispiel

Eine Identifikationsnummer (ID) wird als Feld nicht nur für eine, sondern in der Regel für viele Tabellen verwendet. Beispielsweise arbeitet eine Universitätsverwaltung mit einer Tabelle für Professoren und einer für Studenten, vielleicht auch einer für Kantinenangestellte. Die Gemeinsamkeit dieser Tabellen: Es wird ein Feld benötigt, das z. B. eine bis zu zehnstellige Identifikations- oder Personalnummer abspeichern kann.

Um diese Aufgabenstellung zu lösen, würden Sie als ABAP-Entwickler nun eine Domäne (CHAR der Länge 10) und für die unterschiedlichen Ausprägungen (Professor, Student) jeweils ein Datenelement anlegen. In ABAP können Sie sich jetzt in mehreren Reports und Selektionsbildschirmen jeweils auf eines der angelegten Datenelemente beziehen und so den dahinterliegenden Text aufrufen und sich damit viel Arbeit sparen. Es dreht sich bei den Domänen und Datenelementen also primär um das Thema der Wiederverwendung.

### 3.1.2 Domänen anlegen

Um eine Domäne anzulegen, gehen Sie wie folgt vor:

1. Geben Sie den gewünschten Namen für die Domäne auf dem Übersichtsbildschirm der Transaktion SE11 im Feld DOMÄNE ein, und klicken Sie anschließend auf ANLEGEN.

2. Es öffnet sich der Domänen-Editor, in dem Sie nun Ihre Domäne ausprägen können (siehe Abbildung 3.3). Sie müssen mindestens im Eingabefeld DATENTYP aus den ca. 30 verschiedenen Datentypen einen Datentyp auswählen. Zusätzlich können Sie die ZAHL DER STELLEN (im Beispiel mit der Personalnummer waren dies 10) und für numerische Typen die gewünschten DEZIMALSTELLEN einstellen (bei der Zahl 3,41 sind das z. B. zwei).

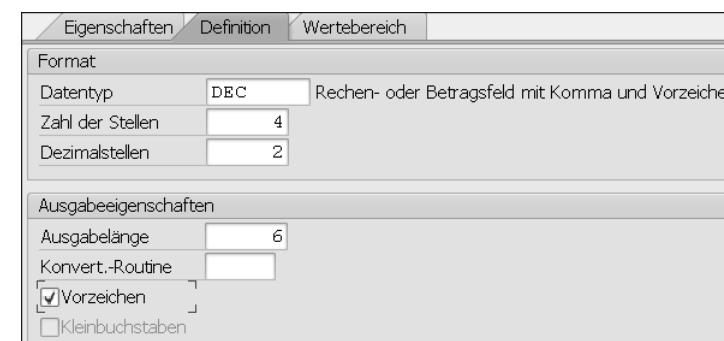


Abbildung 3.3 Definition einer Domäne

3. Unter den AUSGABEEIGENSCHAFTEN können Sie zusätzlich die AUSGABELÄNGE festlegen. Wichtig ist, dass Sie bei numerischen Datentypen und negativen Zahlen das Häkchen in der Checkbox VORZEICHEN setzen, da ansonsten das Minuszeichen nicht berücksichtigt wird.
4. Aktivieren Sie die Domäne anschließend über den Button AKTIVIEREN (EDITOR) in der Menüleiste.

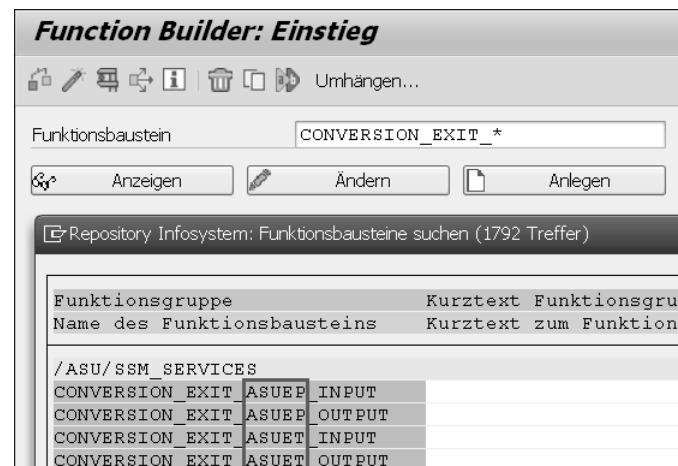
Darüber hinaus können Sie für Domänen eine Konvertierungsroutine hinterlegen und einen Wertebereich definieren, was in den folgenden Abschnitten beschrieben ist.

### 3.1.3 Konvertierungsroutinen

Konvertierungsroutinen sind spezielle Routinen, die die Umwandlung von Werten von und zur Datenbank ermöglichen. Im SAP-System gibt es knapp 2.000 Konvertierungsroutinen. Die beiden bekanntesten sind die Konvertierungsroutinen ALPHA, um einem Wert führende Nullen hinzuzufügen und zu entfernen, und die Konvertierungsroutine ISOLA, um zweistellige ISO-Sprachschlüssel in einstellige SAP-Sprachschlüssel umzuwandeln und umgekehrt.

Jede Konvertierungsroutine hat einen fünfstelligen Namen und wird als Konvertierungs-Exit in Form von zwei Funktionsbausteinen im System abge-

legt. Diese Konvertierungs-Exits finden Sie, indem Sie in Transaktion SE37 (siehe Kapitel 5, »Der Function Builder«) nach Funktionsbausteinen suchen, die wie in Abbildung 3.4 mit dem Namen CONVERSION\_EXIT\_ anfangen.



**Abbildung 3.4** Konvertierungs-Exits finden

Der für Sie interessante Namensbestandteil der Funktionsbausteine in dieser Liste steht zwischen CONVERSION\_EXIT\_ und \_INPUT bzw. \_OUTPUT: Dies ist der Name, den Sie im ABAP Dictionary bei der Anlage von Domänen als Konvertierungsroutine im Feld KONVERT.-ROUTINE angeben müssen (siehe Abbildung 3.3).

Um eigene Konvertierungsroutinen anzulegen, müssen Sie lediglich eine Funktionsbausteingroup mit zwei Funktionsbausteinen anlegen, einen Funktionsbaustein für die Eingabe und einen für die Ausgabe. Wie Sie eine Funktionsgruppe anlegen können, ist in Abschnitt 5.4 beschrieben.

Die Namen müssen dabei folgendem Schema entsprechen, wobei NAME durch Ihren maximal fünfstelligen Namen ersetzt werden kann:

- ▶ CONVERSION\_EXIT\_NAME\_INPUT
- ▶ CONVERSION\_EXIT\_NAME\_OUTPUT

Die beiden Funktionsbausteine müssen folgende Schnittstelle aufweisen:

- ▶ einen Eingabeparameter unter IMPORTING mit einem Werteparameter INPUT
- ▶ einen Ausgabeparameter unter EXPORTING mit einem Werteparameter OUTPUT

Der einfachste Weg, dies zu erreichen, ist es, einen bestehenden Funktionsbaustein (z. B. CONVERSION\_EXIT\_ALPHA\_INPUT und CONVERSION\_EXIT\_ALPHA\_OUTPUT) zu kopieren und diesen anschließend entsprechend abzuändern.

### 3.1.4 Wertebereich einer Domäne

Auf der Registerkarte WERTEBEREICH zu einer Domäne (siehe Abbildung 3.3) haben Sie die Möglichkeit, die Werte einer Domäne über Festwerte (Einzelwerte), Festwertintervalle oder über eine Wertetabelle einzuschränken. Falls Festwerte für eine Domäne definiert wurden, können diese zusätzlich für eine Eingabeverprüfung in Dynpros herangezogen werden. Dies gilt allerdings nur für die Datentypen CHAR und NUMC.

Sie können die Werteüberprüfung für folgende Elemente aktivieren:

- ▶ für Parameter mit dem Zusatz VALUE CHECK (siehe Abschnitt 12.2.1, »Parameter«)
- ▶ für Selektionsoptionen durch das Event AT SELECTION-SCREEN ON (siehe Abschnitt 12.4, »Ereignisse eines Selektionsbildschirms«)

Sollte keine andere Hilfemöglichkeit, wie z. B. eine Suchhilfe (siehe Abschnitt 3.10), definiert sein, werden die Festwerte auch als Eingabehilfe (**[F4]**-Hilfe) angezeigt.

#### Festwerte und Festwertintervalle

Die Angabe von Festwerten und Intervallen ist nur für Domänen der Datentypen CHAR, NUMC, DEC, INT1, INT2 und INT4 möglich. Beide können beliebig miteinander kombiniert werden.

#### Wertetabelle

Es können auch alle Werte einer Domäne gegen eine Wertetabelle geprüft werden. Diese Tabelle muss dazu lediglich bei der Angabe des Wertebereichs in der Domäne eingetragen werden. Diese Angabe können Sie auf der Registerkarte WERTEBEREICH im Feld WERTETABELLE machen.

Durch das Eintragen einer Wertetabelle wird aber noch keine Prüfung implementiert. Die Prüfung in Form des Abgleichs mit der Wertetabelle wird erst nach Definition eines Fremdschlüssels wirksam. Eine Fremdschlüsselbeziehung definiert eine Abhängigkeit zwischen zwei Tabellen. Dazu werden die Schlüsselfelder der ersten Tabelle mit dazu passenden Feldern der zweiten Tabelle verknüpft. Dies wäre beispielsweise möglich, wenn zwei Tabellen jeweils

ein Feld `MATNR` mit einer Materialnummer hätten. Solange eine solche Verknüpfung nicht definiert wurde, weiß das System nicht, anhand welchen Feldes der angegebenen Wertetabelle die Werte der Domäne geprüft werden soll.

#### Beispiel

In Abbildung 3.5 ist beispielweise ein Feld `CARRID` mit dem Datenelement und der gleichnamigen Domäne `S_CARR_ID` definiert. In der Domäne ist die Wertetabelle `SCARR` als Prüftabelle eingetragen, die alle möglichen Fluggesellschaften enthält. Durch die für das Feld `CARRID` über den Button `FREMDSCHLÜSSEL` (🔗) eingetragene Fremdschlüsselbeziehung ist die Prüfung aktiviert worden. In dem Pop-up-Fenster in Abbildung 3.5 ist erkennlich, dass die beiden Tabellen über die Felder `MANDT` und `CARRID` miteinander verknüpft wurden.



Abbildung 3.5 Wertetabelle aktivieren

Wenn ein Anwender nun einen Wert für das Feld `CARRID` auf einer Selektionsmaske eingibt, wird dieser anhand der Wertetabelle `SCARR` geprüft. Ist der eingegebene Wert nicht in der Prüftabelle enthalten, wird eine Fehlermeldung ausgegeben.

## 3.2 Datenelemente

Um ein Datenelement anzulegen, gehen Sie wie folgt vor:

1. Geben Sie den gewünschten Namen für das Datenelement auf dem Hauptbildschirm der Transaktion SE11 im Feld `DATENTYP` ein, und klicken Sie auf `ANLEGEN`.

2. Wählen Sie in dem Pop-up-Fenster `DATENELEMENT` aus.
3. Es öffnet sich der Datenelement-Editor, in dem Sie nun, wie in Abbildung 3.6 dargestellt, auf der Registerkarte `DATENTYP` den Typ spezifizieren können. Sie können hier aus den folgenden Datentyparten auswählen:
  - **DOMÄNE** (siehe Abschnitt 3.1)
  - **EINGEBAUTER TYP**
  - **REFERENZTYP** (kann eine Referenz auf eine Klasse, ein Interface oder auf einen eingebauten Typ sein)

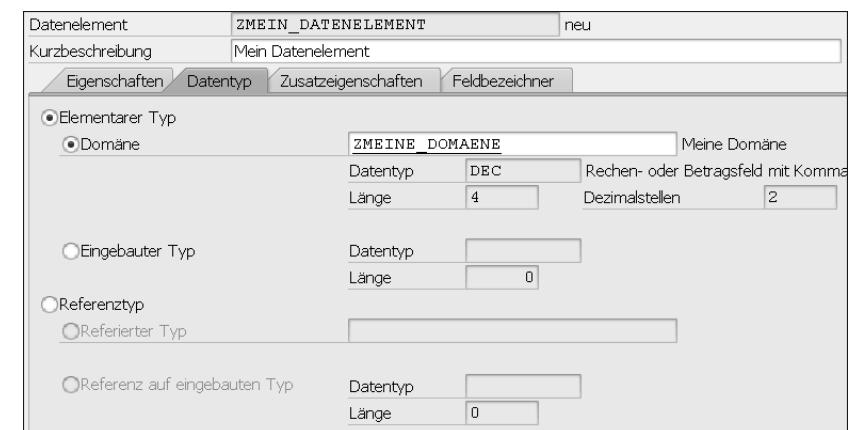


Abbildung 3.6 Datentyp eines Datenelements

4. Danach können Sie Ihr Datenelement bereits aktivieren (📝).

Sie haben darüber hinaus aber noch die Möglichkeit, auf der Registerkarte `FELDBEZEICHNER` die Texte zum Datenelement und auf der Registerkarte `ZUSATZEIGENSCHAFTEN` beispielsweise eine Suchhilfe anzugeben. Beide Registerkarten erläutere ich in den folgenden Abschnitten.

### 3.2.1 Feldbezeichner

Da das Datenelement ja die semantische Bedeutung zur Typdefinition liefert (siehe Abschnitt 3.1.1, »Das zweistufige Domänenprinzip«), können Sie dem Datenelement auf der Registerkarte `FELDBEZEICHNER` verschiedene Texte zuordnen (siehe Abbildung 3.7).

Um die Texte zu pflegen, geben Sie sie in die Eingabefelder `FELDBEZEICHNER` ein, und bestätigen Sie Ihre Eingabe anschließend mit ↵. Die Länge der Eingabefelder gibt dabei die maximale Länge der Texte vor.

Eigenschaften		Datentyp	Zusatzeigenschaften	Feldbezeichner
Länge	Feldbezeichner			
kurz	10	MDatenel		
mittel	17	Mein Datenelement		
lang	20	Mein Datenelement		
Überschrift	17	Mein Datenelement		

Abbildung 3.7 Feldbezeichner eines Datenelements

### 3.2.2 Übersetzung

Fast alle in Transaktion SE11 zu pflegenden Objekte haben Texte (z. B. eine Kurzbeschreibung), die sich übersetzen lassen. Doch nirgends hat dies eine wichtigere Bedeutung als bei den Feldbezeichnern eines Datenelements. Diese Texte werden später beispielsweise beim Aufbau einer ALV-Tabelle als Spaltentexte verwendet (siehe Kapitel 19, »Tabellenanzeige mit dem SAP List Viewer (ALV)«) oder auf einem Selektionsbildschirm in Form eines Parameters oder eines Select-Option-Namens dargestellt (siehe Kapitel 12, »Reports und Selektionsbildschirme«). Ist der Text übersetzt, wird abhängig von der Anmeldesprache des Nutzers jeweils die passende Übersetzung angezeigt.

Um einen Text zu einem Datenelement zu übersetzen, gehen Sie wie folgt vor:

1. Rufen Sie das Datenelement im ABAP Dictionary auf, und wählen Sie SPRINGEN • ÜBERSETZUNG im Hauptmenü.
2. Wählen Sie, wie in Abbildung 3.8 dargestellt, im sich öffnenden Pop-up-Fenster die gewünschte Zielsprache aus.



Abbildung 3.8 Angabe der Zielsprache

3. In der nächsten Maske werden Ihnen nun alle zu übersetzenden Texte angezeigt. Diese können Sie jeweils unter dem originalen Textblock übersetzen, indem Sie den übersetzten Text eintragen, wie in Abbildung 3.9 dargestellt.

DTEL(..) von deDE nach enUS: ZMEIN_DATENELEMENT	
[DDTEXT 00001] Mein Datenelement	My data element
[REPTEXT 00001] Mein Datenelement	My data element

Abbildung 3.9 Übersetzung von Textelementen

4. Sie haben darüber hinaus die Möglichkeit, sich über den Button QUELL-TEXTE IN SAPTERM SUCHEN (🔍) auf der Funktionsleiste Vorschlagswerte aus der SAPterm-Datenbank anzeigen zu lassen.
5. Wenn Sie anschließend auf SPEICHERN (💾) klicken, werden Ihnen Ihre eingegebenen Texte gelb angezeigt.

Prinzipiell können Texte folgende Status haben:

- ▶ Rot: neuer Text
- ▶ Gelb: beanstandeter Text
- ▶ Grün: korrekt übersetzter Text

Wenn Sie lediglich eine Übersetzung benötigen, reicht der Status Gelb aus. Damit verzichten Sie allerdings auf die Wiederverwendung von bereits angefertigten Übersetzungen, da damit Ihre Übersetzung nicht in den Vorschlagspool übernommen wird. Möchten Sie Ihre Übersetzung dagegen in den Vorschlagspool übernehmen, um z. B. die Wiederverwendung zu ermöglichen, können Sie auf den Button VORSCHLAG DIREKT ANLEGEN rechts neben dem gelben Statusfeld klicken. Damit wechselt der Status von Gelb auf Grün. Speichern Sie gegebenenfalls erneut.

### 3.2.3 Zusatzeigenschaften

Auf der Registerkarte ZUSATZEIGENSCHAFTEN haben Sie die Möglichkeit, dem Datenelement eine Suchhilfe zuzuordnen (siehe Abschnitt 3.10). Über eine Parameter-ID kann Ihr Feld mit einem Vorschlagswert aus dem SAP-Memory gefüllt werden. Für jeden Benutzer können Sie einen solchen Wert in dessen Benutzerstammdaten auf der Registerkarte PARAMETER pflegen. Die Benutzerstammdaten erreichen Sie über den Pfad SYSTEM • BENUTZERVORGABEN im Hauptmenü.

### 3.3 Strukturen

Eine Struktur ist ein Verbund von einzelnen Feldern. Wie Sie eine Struktur programmatisch definieren, ist in Abschnitt 7.4.3 erläutert. Um eine Struktur im ABAP Dictionary anzulegen, gehen Sie wie folgt vor:

1. Geben Sie den gewünschten Namen auf dem Hauptübersichtsbildschirm der Transaktion SE11 im Feld DATENTYP ein. Klicken Sie anschließend auf ANLEGEN.
2. Wählen Sie im sich öffnenden Pop-up-Fenster den Datentyp STRUKTUR aus.
3. Es öffnet sich der Struktur-Editor, in dem Sie nun, wie in Abbildung 3.10 dargestellt, auf der Registerkarte KOMPONENTEN die gewünschten Felder Ihrer Struktur eintragen können. Für jedes Feld müssen Sie die TYPISIERUNGSART und einen Typ in Form eines Datenelements oder anderer Strukturen (*tiefe Strukturen*) vergeben.

Komponente	Typisierungsart	Komponententyp	Datentyp	Länge	DezSt...	Kurzbeschreibung
MATNR	Type	MATNR	CHAR	18	0	Materialnummer
MEINS	Type	MEINS	UNIT	3	0	Basismengeneinheit
MTART	Type	MTART	CHAR	4	0	Materialart
BRGEW	Type	BRGEW	QUAN	13	3	Bruttogewicht

Abbildung 3.10 Komponenten einer Struktur

4. Danach können Sie Ihre Struktur bereits AKTIVIEREN ().

Die in Abbildung 3.10 angegebene Struktur wird allerdings eine Warnung und eine Fehlermeldung produzieren:

- Warnung: Die Erweiterungskategorie fehlt.
- Fehler: Für das Feld BRGEW fehlen die Referenztabelle und das Referenzfeld.

Dies schauen wir uns in den folgenden beiden Abschnitten an.

#### 3.3.1 Erweiterungskategorie

Die Erweiterungskategorie können Sie über den Pfad ZUSÄTZE • ERWEITERUNGSKATEGORIE DEFINIEREN im Hauptmenü definieren (siehe Abbildung 3.11). Damit bestimmen Sie, ob und wie die Struktur erweitert werden darf. Die folgenden Optionen stehen Ihnen dazu zur Verfügung:

- BELIEBIG ERWEITERBAR: Die Struktur darf mit beliebigen Komponenten erweitert werden.
- ERWEITERBAR UND ZEICHENARTIG ODER NUMERISCH: Die Struktur darf erweitert werden, die Erweiterung darf aber keine tiefen Datentypen enthalten.
- ERWEITERBAR UND ZEICHENARTIG: Die Struktur darf mit zeichenartigen Komponenten (c, n, d oder t) erweitert werden.
- NICHT ERWEITERBAR: Die Struktur darf nicht erweitert werden.
- NICHT KLASIFIZIERT: Undefiniert, kann für einen Übergangszustand verwendet werden.

Darf eine Struktur erweitert werden, kann dies mithilfe einer Append-Struktur geschehen. Mehr Informationen zu Append-Strukturen finden Sie in Abschnitt 21.4.7, »Struktur-Erweiterungen«.

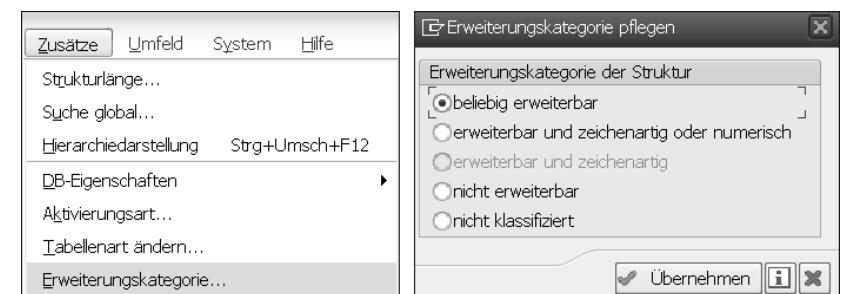


Abbildung 3.11 Erweiterungskategorie einer Struktur

#### 3.3.2 Referenztabellen und das Referenzfeld

Mengen- und Währungsfelder benötigen immer eine Zuweisung zu einer Mengen- bzw. Währungseinheit, damit das System weiß, in welcher Form die Menge bzw. die Währung darzustellen ist. Diese Zuweisung können Sie auf der Registerkarte WÄHRUNGS-/MENGENFELDER vornehmen (siehe Abbildung 3.12). Der Verweis kann dabei auf jede beliebig andere Tabelle bzw. Struktur zeigen.

Komponente	Typisierungsart	Datentyp	Referenztabelle	Referenzfeld	Kurzbeschreibung
MATNR	Type	CHAR			Materialnummer
MEINS	Type	UNIT			Basismengeneinheit
MTART	Type	CHAR			Materialart
BRGEW	Type	QUAN	ZMEINE_STRUKTUR	MEINS	Bruttogewicht

Abbildung 3.12 Verweis auf Einheiten

## 3.4 Tabellentypen

Tabellentypen sind spezielle Typen, die den Aufbau einer internen Tabelle beschreiben. Das Besondere ist, dass eine interne Tabelle auf Basis des Tabellentyps einfach mit dem Zusatz **TYPE** definiert werden kann und dazu nicht der Zusatz **TYPE TABLE OF** benötigt wird. Dies ist insbesondere in Schnittstellen von z. B. Funktionsbausteinen und Methoden von Klassen wichtig, da dort nicht **TYPE TABLE OF**, sondern nur **TYPE** angegeben werden kann.

Um einen Tabellentyp anzulegen, gehen Sie wie folgt vor:

1. Geben Sie den gewünschten Namen des Tabellentyps auf dem Hauptübersichtsbildschirm der Transaktion SE11 im Feld **DATENTYP** ein. Klicken Sie anschließend auf **ANLEGEN**.
2. Wählen Sie im Pop-up-Fenster den Datentyp **TABELLENTYP** aus.
3. Es öffnet sich der Tabellentyp-Editor, in dem Sie nun, wie in Abbildung 3.13 dargestellt, auf der Registerkarte **ZEILENTYP** den gewünschten Aufbau Ihres Tabellentyps anhand einer Struktur definieren können.



Abbildung 3.13 Zeilentyp eines Tabellentyps

4. Auf der Registerkarte **INITIALISIERUNG UND ZUGRIFF** können Sie die Art der internen Tabelle für den Tabellentyp auswählen (siehe Abschnitt 8.1, »Tabellenarten«).
5. Auf den Registerkarten **PRIMÄRSCHLÜSSEL** und **SEKUNDÄRSCHLÜSSEL** können Sie nun noch die Schlüsselfelder definieren.
6. Anschließend können Sie Ihren Tabellentyp **AKTIVIEREN** () und z. B. in Methodensignaturen oder Funktionsbausteinschnittstellen verwenden.

Neben den hier beschriebenen globalen Tabellentypen im ABAP Dictionary können Sie Tabellentypen auch programmatisch mit ABAP für einen lokalen Einsatz definieren. Mehr Informationen dazu finden Sie in Abschnitt 8.2, »Interne Tabellen definieren«.

## Ranges-Tabellentypen anlegen

Neben den Tabellentypen für Standardtabellen können Sie im ABAP Dictionary auch globale Ranges-Tabellen als Typ definieren (siehe Abschnitt 8.2.3):

1. Legen Sie dazu, wie in Abschnitt 3.4, »Tabellentypen«, beschrieben, einen neuen Tabellentyp an, und wählen Sie **BEARBEITEN • ALS RANGES-TABELLENTYP DEFINIEREN** im Hauptmenü (siehe Abbildung 3.14).



Abbildung 3.14 Einen Ranges-Tabellentyp anlegen

2. Tragen Sie im folgenden Bildschirm, wie in Abbildung 3.15 dargestellt, eine **KURZBESCHREIBUNG** und ein **DATENELEMENT** bzw. einen eingebauten Typ ein, für das oder den Sie den Ranges-Tabellentyp definieren wollen.
3. Vergeben Sie anschließend noch einen Namen für die dahinterliegende Struktur im Feld **STRUKTURIERTER ZEILENTYP**.
4. Speichern () Sie nun Ihren Ranges-Tabellentyp, da sonst die Anlage des strukturierten Zeilentyps nicht funktioniert.

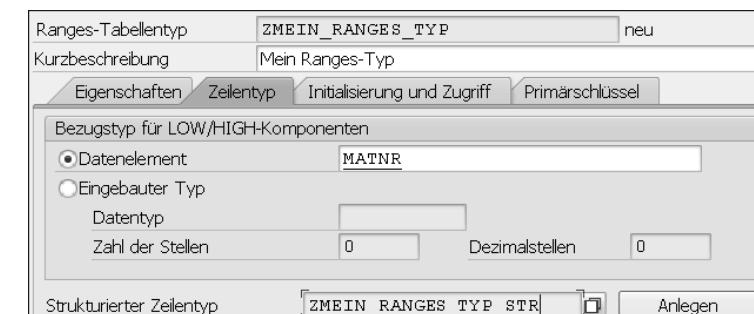


Abbildung 3.15 Anlage des strukturierten Zeilentyps

5. Nun können Sie auf den Button **ANLEGEN** klicken, wodurch eine neue Struktur auf Basis Ihrer Eingabe angelegt wird. Wie Sie in Abbildung 3.16 sehen können, ist dies die Standardstruktur einer Ranges-Tabelle.
6. Vergeben Sie hier eine **KURZBESCHREIBUNG**, aktivieren () Sie die Struktur, und wechseln Sie über die Navigationsleiste () zurück zum Bildschirm **TABELLENTYP ÄNDERN**.

7. Aktivieren ( ) Sie auch dort Ihren Ranges-Tabellentyp

Struktur	ZMEIN_RANGES_TYP_STR	neu(überarbeitet)			
Kurzbeschreibung	Meine Ranges-Struktur				
Eigenschaften	Komponenten	Eingabehilfe/-prüfung			
		Währungs-/Mengenfelder			
        	Eingebauter Typ				
Komponente	Typisierungsart	Komponententyp	Datentyp	Länge	Def.
SIGN	Type	▼ DDSIGN	CHAR	1	
OPTION	Type	▼ DDOPTION	CHAR	2	
LOW	Type	▼ MATNR	CHAR	18	
HIGH	Type	▼ MATNR	CHAR	18	

**Abbildung 3.16** Komponenten der Ranges-Struktur

### 3.5 Datenbanktabellen

Eine der Hauptfunktionen des ABAP Dictionarys ist die Verwaltung der zentralen Datenbanktabellen des SAP-Systems. Eine Datenbanktabelle beinhaltet eine Menge von Daten, die wie in einem Excel-Sheet in Zeilen und Spalten strukturiert sind. Die Spaltennamen definieren, was für Daten in jeder Spalte stehen sollen, während die einzelnen Zeilen die einzelnen Datensätze repräsentieren. Über das ABAP Dictionary als Schnittstelle zur an das SAP-System angebundenen Datenbank können Sie sowohl die Strukturen als auch die Daten aller Datenbanktabellen anzeigen lassen. Zusätzlich können Sie auch eigene Datenbanktabellen für Ihre Entwicklungen anlegen und so ermöglichen, dass Daten langfristig gespeichert werden.

### 3.5.1 Datenbanktabellen anzeigen

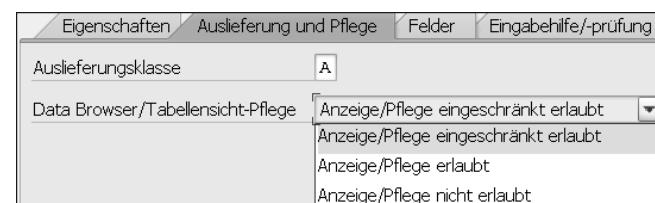
Um eine Datenbanktabelle anzuzeigen, gehen Sie wie folgt vor:

1. Tragen Sie den Namen der gewünschten Tabelle auf dem Hauptübersichtsbildschirm der Transaktion SE11 im Feld DATENBANKTABELLE ein.
  2. Anschließend gelangen Sie durch einen Klick auf den Button ANZEIGEN zur Definition der Tabelle. Wenn Sie nun den Inhalt der Datenbanktabelle sehen möchten, klicken Sie auf der Funktionsleiste auf den Button DATENANZEIGE ( ).
  3. In der folgenden Selektionsmaske können Sie die anzuzeigende Datensetzmengen einschränken. Nehmen Sie hier keine Einstellung vor, werden standardmäßig 200 Datensätze angezeigt.

4. Nun können Sie innerhalb der Datenbanktabelle durch Eingabe von Suchkriterien nach Datensätzen suchen. Klicken Sie auf den Button ANZAHL EINTRÄGE, wird Ihnen angezeigt, wie viele Einträge zu Ihren Suchkriterien passen. Klicken Sie auf AUSFÜHREN () , um Ihre Suche innerhalb der Datenbanktabelle zu starten.

### 3.5.2 Datenbanktabellen anlegen

Zum Anlegen einer neuen Datenbanktabelle tragen Sie den gewünschten Namen der Tabelle ebenfalls auf dem Hauptübersichtsbildschirm der Transaktion SE11 im Feld DATENBANKTABELLE ein, und klicken Sie anschließend auf ANLEGEN. Für den Namen gibt es keine speziellen Namenskonventionen, er muss lediglich im Kundennamensraum (z. B. Z oder Y) liegen. Es öffnet sich nun der Tabellen-Editor (siehe Abbildung 3.17).



**Abbildung 3.17** Registerkarte »Auslieferung und Pflege« im Tabellen-Editor

Zur Anlage einer Tabelle müssen Sie die folgenden Einstellungen vornehmen, auf die ich in den folgenden Abschnitten näher eingehe:

- ▶ Auf der Registerkarte AUSLIEFERUNG UND PFLEGE:
    - AUSLIEFERUNGSKLASSE: Soll die Datenbanktabelle einen Transportanschluss haben (siehe Abschnitt 3.5.3)?
    - DATA BROWSER/TABELLENSICHT-PFLEGE: Wie soll auf die Datenbanktabelle zugegriffen werden können (siehe Abschnitt 3.5.4)?
  - ▶ In den technischen Einstellungen:
    - DATENART: Was soll in der Tabelle gespeichert werden (siehe Abschnitt 3.5.5)?
    - GRÖSSENKATEGORIE: Wie viele Daten werden erwartet (siehe Abschnitt 3.5.6)?
    - PUFFERUNG (optional): Darf die Datenbanktabelle gepuffert werden (siehe Abschnitt 3.5.7)?
  - ▶ Ausprägen der Tabellenfelder auf der Registerkarte FELDER:  
Welche Spalten soll die Datenbanktabelle haben (siehe Abschnitt 3.5.8)?

Darüber hinaus gibt es seit der Einführung von SAP HANA eine neue Registerkarte SPEICHERART, auf die ich in Abschnitt 3.5.9, »Speicherart von Datenbanken mit SAP HANA«, eingehe.

### 3.5.3 Auslieferungsklasse

Eine Auslieferungsklasse wählen Sie auf der Registerkarte AUSLIEFERUNG UND PFLEGE (siehe Abbildung 3.17). Die Auslieferungsklasse bestimmt, wie sich die Datenbanktabelle bei der Installation, beim Upgrade, bei einer Mandantenkopie und beim Transport zwischen Kundensystemen verhält. Die Frage, die Sie sich hier also stellen müssen, lautet: »Wie soll sich die Datenbanktabelle beim Transport verhalten?« Möchten Sie die Datenbanktabelle in jedem System (z.B. Entwicklungs-, Qualitätssicherungs- und Produktivsystem) einzeln pflegen, oder soll die Datenbanktabelle mithilfe der Transporttechnologie über alle Systeme hinweg den gleichen Inhalt haben?

Wenn Sie eine Tabelle mit der Auslieferungsklasse »A« anlegen (siehe Abbildung 3.18), kann die Datenbanktabelle in jedem System über einen Pflegedialog geändert werden. Legen Sie Ihre Datenbanktabelle dagegen mit der Auslieferungsklasse »C« an, kann diese Datenbanktabelle nur in Entwicklungssystemen geändert werden (wenn das System auf ÄNDERBAR gesetzt ist), und die Inhalte müssen über das *Transportwesen* im Rahmen eines Customizing-Auftrags transportiert werden.

Wie Sie einen Pflegedialog anlegen, ist in Abschnitt 3.9, »Pflegedialoge«, beschrieben, während Sie in Kapitel 17, »Das Transportwesen«, eine Erläuterung des Transportmechanismus finden.



Abbildung 3.18 Auslieferungsklasse einer Tabelle

### 3.5.4 Tabellensicht-Pflege

Auf der Registerkarte AUSLIEFERUNG UND PFLEGE (siehe Abbildung 3.17) stellen Sie die Berechtigungen für die Anzeige bzw. Pflege Ihrer Datenbanktabelle ein. Die grundlegende Frage, die Sie sich hier stellen müssen, lautet: »Wie soll auf die Datenbanktabelle zugegriffen werden können?«.

Hier haben Sie die folgenden Optionen zur Auswahl:

- ▶ ANZEIGE/PFLEGE EINGESCHRÄNKKT ERLAUBT

Die Anzeige der Datenbanktabelle ist nur im ABAP Dictionary (Transaktion SE11) und in der allgemeinen Tabellenanzeige (Transaktion SE16 bzw. SE16N) möglich.

- ▶ ANZEIGE/PFLEGE ERLAUBT

Die Tabelle kann in der allgemeinen Tabellenanzeige angezeigt und in der Tabellensicht-Pflege (Transaktion SM30) über einen Pflegedialog (siehe Abschnitt 3.9, »Pflegedialoge«) gepflegt werden.

- ▶ ANZEIGE/PFLEGE NICHT ERLAUBT

Anzeige bzw. Pflege ist nur über ABAP-Anweisungen (d. h. via Open SQL) möglich.

### 3.5.5 Datenart

Die Art der in Ihrer Datenbanktabelle gespeicherten Daten können Sie in den technischen Einstellungen auswählen (siehe Abbildung 3.19). Klicken Sie dazu auf den Button TECHNISCHE EINSTELLUNGEN in der Funktionsleiste.

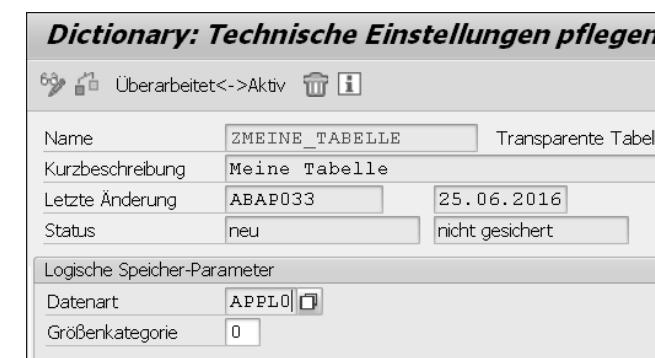


Abbildung 3.19 Technische Einstellungen einer Tabelle

Die wichtigsten Datenarten sind:

- ▶ APPL0 für Stammdaten

- ▶ APPL1 für Bewegungsdaten

- ▶ APPL3 Organisations- und Customizing-Daten

Die Frage, die Sie sich hierbei stellen müssen, lautet: »Welche Daten wollen Sie speichern?« Sind es Daten, die sich häufig ändern (*Bewegungsdaten*), sind es Daten, auf die zwar häufig lesend zugegriffen wird, die sich jedoch sehr

selten ändern (*Stammdaten*), oder sind es Daten, die für das Customizing des Systems benötigt werden? Die Auswahl hat für Sie als Programmierer keine großen Auswirkungen, Sie beeinflusst lediglich, wo die Daten auf der Datenbank abgelegt werden, und dies auch nur für die Datenbanksysteme Oracle und Informix.

### 3.5.6 Größenkategorie

Die Größenkategorie Ihrer Datenbanktabelle können Sie ebenfalls in den technischen Einstellungen auswählen (siehe Abbildung 3.19). Hier geht es darum anzugeben, wie viel Speicherplatz Ihre Tabelle in Zukunft vermutlich einnehmen wird. Beim Anlegen der Datenbanktabelle wird aufgrund der hier angegebenen Größe ein initialer Speicherplatz auf der Datenbank reserviert. Hintergrund ist, dass aufwendige Reorganisationen des Speicherplatzes vermieden werden sollen, die entstehen, wenn der reservierte Speicherplatz überschritten wurde. Die Frage, die Sie sich in diesem Kontext stellen müssen, lautet also: »Wie viele Daten werden vermutlich in der Datenbanktabelle gespeichert werden?«.

### 3.5.7 Pufferung

Die Pufferung können Sie ebenfalls in den technischen Einstellungen konfigurieren. Der *Puffer* ist ein spezieller Bereich auf dem Applikationsserver, in dem Datensätze Ihrer Tabelle vorgehalten werden. Greift eine SQL-Anweisung auf eine gepufferte Tabelle zu, wird geprüft, ob die angefragten Daten sich in diesem Pufferbereich befinden. Ist dies der Fall, werden die Daten direkt aus dem Puffer gelesen. Ist dies nicht der Fall, werden die Daten von der Datenbank gelesen und dabei in den Puffer geladen. Die Pufferung einer Tabelle erhöht die Performance bei jedem Zugriff auf die in der Tabelle enthaltenen Datensätze, da nicht jedes Mal auf die Datenbank zugegriffen werden muss.

Ob Sie die Pufferung für eine Datenbanktabelle zulassen sollten oder nicht, hängt davon ab, wie später mit der Tabelle gearbeitet wird. Sind viele lesende Zugriffe abzusehen, lohnt sich eine Pufferung, bei vielen schreibenden Zugriffen lohnt sich dagegen keine Pufferung. Wenn Sie die Pufferung einschalten, müssen Sie auch eine Pufferungsart auswählen. Die Fragen, die Sie sich an dieser Stelle stellen müssen, lauten also: »Erfolgen auf meine Datenbanktabelle überwiegend lesende Zugriffe, und möchte ich die Geschwindigkeit des Lesezugriffs erhöhen?«.

Zwei grundlegende Pufferungsarten stehen Ihnen zur Verfügung (siehe Abbildung 3.20):

- ▶ **EINZELSÄTZE GEPUFFERT:** Lohnt sich bei großen Tabellen mit vielen einzelnen Zugriffen, da auch für nicht vorhandene Einträge im Puffer ein Vermerk angelegt wird.
- ▶ **VOLLSTÄNDIG GEPUFFERT:** Je kleiner eine Tabelle ist, je häufiger sie gelesen und je seltener in sie geschrieben wird, desto günstiger ist es, sie vollständig zu puffern

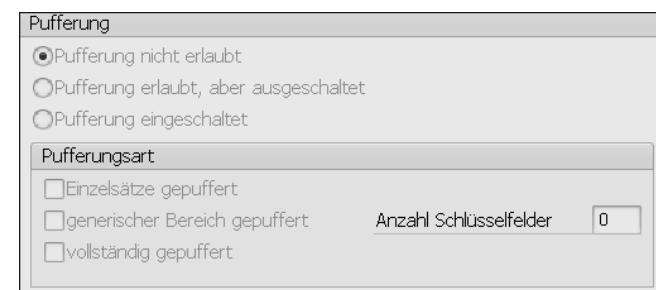


Abbildung 3.20 Pufferung einer Tabelle

### 3.5.8 Felder ausprägen

Nachdem Sie die Eigenschaften und technischen Einstellungen gepflegt haben, können Sie sich endlich dem wichtigsten Punkt bei der Anlage einer Datenbanktabelle widmen: den Tabellenfeldern auf der Registerkarte FELDER. Wie in Abbildung 3.21 zu sehen, können Sie hier wie bei der Anlage von Strukturen (siehe Abschnitt 3.3) Ihre gewünschten Felder eintragen und über Datenelemente typisieren.

Transp.Tabelle		ZMEINE_TABELLE		aktiv
Kurzbeschreibung		Meine Tabelle		
		Eigenschaften	Auslieferung und Pflege	Felder
				Eingabehilfe/-prüfung
				Suchhilfe
Feld	Key	Initi...	Datenelement	Datentyp
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT
MATNR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MATNR	CHAR
MTART	<input type="checkbox"/>	<input type="checkbox"/>	MTART	CHAR
MEINS	<input type="checkbox"/>	<input type="checkbox"/>	MEINS	UNIT
ERSDA	<input type="checkbox"/>	<input type="checkbox"/>	ERSDA	DATS
BRGEW	<input type="checkbox"/>	<input type="checkbox"/>	BRGEW	QUAN

Abbildung 3.21 Felder einer Tabelle ausprägen

Wenn Ihre Tabelle mandantenabhängig sein soll (das ist in der Regel der Fall), müssen Sie als erstes Feld das Feld `MANDT` vom Typ `MANDT` hinzufügen. Dieses Feld wird dann automatisch bei jeder Open-SQL-Anweisung mit dem aktuellen Mandanten gefüllt bzw. beim lesenden Zugriff entsprechend verarbeitet.

### 3.5.9 Speicherart von Datenbanken mit SAP HANA

Mit SAP NetWeaver Application Server ABAP 7.40 wurde im ABAP Dictionary eine neue Registerkarte `DB SPEZIFISCHE EIGENSCHAFTEN` eingeführt (siehe Abbildung 3.22). Hier können Sie steuern, mit welcher Speicherart eine Datenbanktabelle auf dem Datenbanksystem angelegt werden soll.

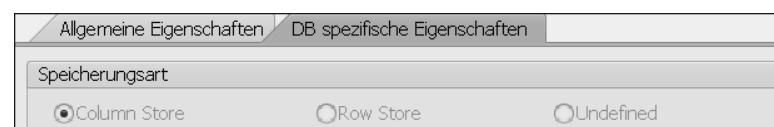


Abbildung 3.22 Speicherart einer Tabelle

Sie können zwischen drei verschiedenen Speicherarten wählen:

- ▶ **COLUMN STORE:** spaltenbasierte Speicherung als die neue Speicherart der SAP-HANA-Datenbank (siehe Kapitel 28, »SAP HANA«)
- ▶ **ROW STORE:** zeilenbasierte Speicherung als herkömmlicher Standard
- ▶ **UNDEFINED:** andere

Diese Einstellung kann jederzeit nach der Erstellung der Tabelle geändert werden. Die Tabelle wird anschließend entsprechend umgesetzt.

## 3.6 Indizes

Mit Indizes können Sie das Durchsuchen einer Tabelle nach Datensätzen beschleunigen. Ein Index kann als sortierte Teilmenge einer Datenbanktabelle verstanden werden und ermöglicht so einen schnelleren Zugriff auf die Datensätze. Im SAP-System wird zwischen folgenden Arten von Indizes unterschieden:

### ▶ Primärindizes

Der Primärindex besteht aus dem Primärschlüssel einer Tabelle sowie einem Zeiger auf ihre Nicht-Schlüsselfelder, damit diese bei Bedarf schnell nachgelesen werden können. Der Primärindex wird beim Anlegen der Tabelle auf der Datenbank automatisch erstellt.

### ▶ Sekundärindizes

Darüber hinaus können Sie im ABAP Dictionary sogenannte Sekundärindizes anlegen. Diese Indizes sind notwendig, wenn häufig über Nicht-Schlüsselfelder auf die Tabelle zugegriffen wird, da hier der Primärindex nicht genutzt werden kann, der ja nur Schlüsselfelder enthält.

Es können mehrere dieser Sekundärindizes für eine Tabelle existieren. Erst zur Laufzeit wird vom Datenbanksystem entschieden, welcher Index verwendet werden muss.

### Sekundärindizes und SAP HANA

Durch die Einführung von SAP HANA und die spaltenorientierte Speicherung haben Sekundärindizes an Bedeutung verloren. Weitere Informationen dazu finden Sie unter der Überschrift »Die goldenen Regeln für SAP HANA« in Abschnitt 9.1 unter Regel 4.

[«]

### Indizes anlegen

Um einen Index anzulegen, gehen Sie wie folgt vor:

1. Lassen Sie sich die Definition der Tabelle, für die Sie einen Index anlegen möchten, in Transaktion SE11 anzeigen (siehe Abschnitt 3.5.1, »Datenbanktabellen anzeigen«).
2. Klicken Sie im Tabellen-Editor in der Funktionsleiste auf INDIZES.
3. In der Übersicht der vorhandenen Indizes für die Tabelle klicken Sie nun, wie in Abbildung 3.23 dargestellt, in der Menüleiste auf den Button ANLEGEN () und wählen aus dem Auswahlmenü die Aktion INDEX ANLEGEN aus.



Abbildung 3.23 Index zu einer Tabelle anlegen

4. Hier können Sie nun angeben, für welches Datenbanksystem der Index angelegt werden soll und welche Felder der Index beinhalten soll (siehe Abbildung 3.24). Wenn Ihr Index den Primärschlüssel enthält, also eine Zeile eindeutig identifiziert, können Sie auch die Option UNIQUE-INDEX auswählen.

5. Aktivieren Sie Ihren Index anschließend über den Button AKTIVIEREN ( ).

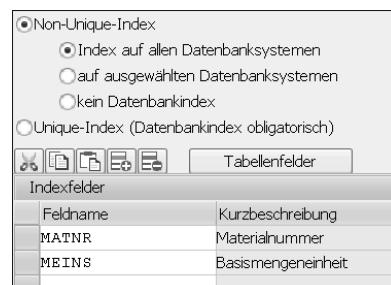


Abbildung 3.24 Definition der Indexfelder

## 3.7 Typgruppen

Eine Typgruppe ist ein historisch bedingtes Konstrukt und dient der Sammlung mehrerer Typen. Es war nötig, da es vor Release 4.5A im ABAP Dictionary keine eigenständigen Datentypen gab, auf die in ABAP-Programmen mit dem TYPE-Zusatz verwiesen werden konnte.

Um eine Typgruppe anzulegen, geben Sie den gewünschten Namen auf dem Hauptübersichtsbildschirm der Transaktion SE11 im Feld TYPGRUPPE ein, und klicken Sie anschließend auf ANLEGEN. Im sich öffnenden Pop-up-Fenster tragen Sie eine KURZBESCHREIBUNG ein und speichern die Typgruppe mit einem Klick auf SICHERN. Innerhalb der angelegten Typgruppe können Sie nun über den Editor Ihre benötigten Typen jeglicher Art definieren (siehe Abbildung 3.25).

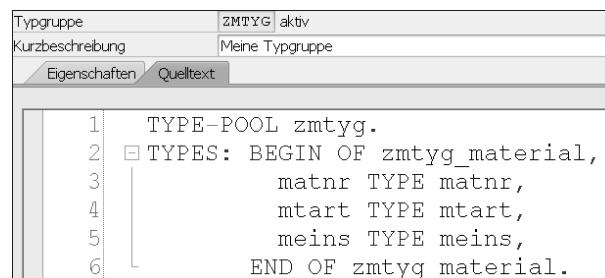


Abbildung 3.25 Eine Typgruppe und seine Typen

Im ABAP-Programm können Sie Ihre Typgruppe im Beispiel aus Abbildung 3.25 mit der folgenden Anweisung einbinden:

TYPE-POOLS zmtyg.

## 3.8 Views

Ein View ist eine Zusammenfassung von ausgewählten Feldern aus mehreren Tabellen, ähnlich einem Join zur Verknüpfung von Datenbanktabellen (siehe Abschnitt 9.2.13, »JOIN: Verknüpfung«). Der Vorteil von Views gegenüber selbst mit ABAP programmierten Joins ist, dass sie rein über Definitionsmasken und nicht durch Programmierung definiert werden und dass Views durch die zentrale Anlage im ABAP Dictionary systemweit wiederverwendet werden können. Die so ausgewählte Schnittmenge wird als eigene Struktur definiert und kann in ABAP-Programmen mit Open SQL konsumiert werden.

Die Daten eines Views werden wie bei einem Join nicht physisch in einer separaten Tabelle gespeichert, sondern als Teilmenge zur Laufzeit aus einer oder mehreren Tabellen durch Selektion (Weglassen von Spalten) und Projektion (Weglassen von Zeilen) abgeleitet.

Um einen View anzulegen, geben Sie den gewünschten Namen auf dem Hauptübersichtsbildschirm der Transaktion SE11 im Feld VIEW ein, und klicken Sie anschließend auf ANLEGEN. Im sich öffnenden Pop-up-Fenster haben Sie nun folgende View-Typen zur Auswahl:

- ▶ Datenbank-View (siehe Abschnitt 3.8.1): normale Umsetzung einer Tabellenverknüpfung
- ▶ Projektions-View (siehe Abschnitt 3.8.2): Hier ist keine Verknüpfung von Tabellen möglich; dieser View-Typ dient dem Ausblenden von Spalten einer Tabelle.
- ▶ Pflege-View (siehe Abschnitt 3.8.3): Pflege-Views ermöglichen die Pflege von über mehrere Tabellen verteilten Daten.
- ▶ Help-View (siehe Abschnitt 3.8.4): Help-Views dienen als Selektionsmethode in Suchhilfen.

### 3.8.1 Datenbank-View

Wenn Sie in dem Pop-up-Fenster zur Anlage eines Views den Typ DATENBANK-VIEW ausgewählt haben, gelangen Sie in den Datenbank-View-Editor:

1. Hier geben Sie auf der Registerkarte TABELLEN/JOINBEDINGUNGEN die zu verknüpfenden Tabellen an. In Open SQL wäre dies die JOIN-Klausel. Die Tabellen müssen dabei über ihre Primär- und Fremdschlüssel miteinander verknüpfbar sein. Tragen Sie die Tabellen unter TABELLEN auf der linken Seite ein (siehe Abbildung 3.26), und klicken Sie anschließend auf den

Button BEZIEHUNGEN unterhalb der Tabelle. Dadurch werden Ihnen bereits mögliche Verknüpfungen vorgeschlagen, von denen Sie eine auswählen können.

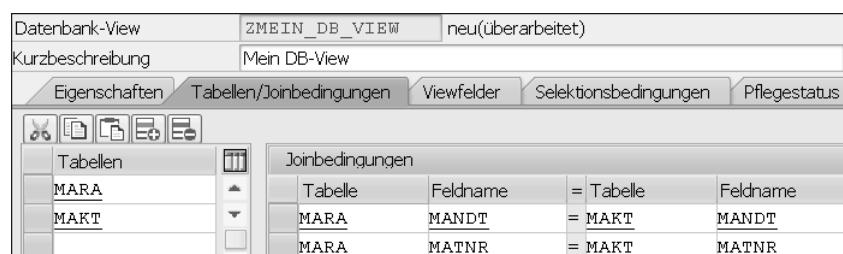


Abbildung 3.26 Join-Bedingung eines Views

- Nun wählen Sie auf der Registerkarte VIEWFELDER (siehe Abbildung 3.27) die gewünschten Felder der ausgewählten Tabellen aus. Bei einer Definition in Open SQL entspräche das der Feldleiste nach der SELECT-Anweisung. Die View-Felder können Sie entweder manuell in die Tabelle eintragen oder sie über den Button TABELLENFELDER automatisch hinzufügen lassen.

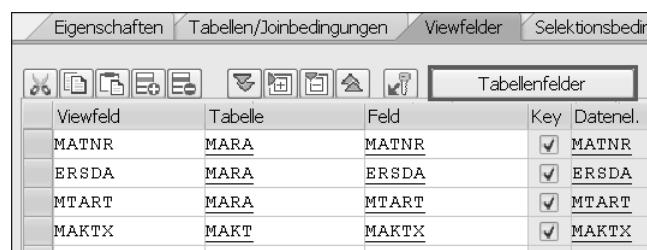


Abbildung 3.27 Felder eines Views

- Jetzt können Sie Ihren View AKTIVIEREN ( aktivieren) und testen.

Sie haben darüber hinaus aber noch die Möglichkeit, auf der Registerkarte SELEKTIONSBEDINGUNGEN (siehe Abbildung 3.28) eine Selektionsbedingung analog zur einer WHERE-Klausel in einer Open-SQL-Anweisung anzugeben, um so nur eine Teilmenge der Daten zu selektieren. Auch dabei können Sie die Felder wieder über den Button TABELLENFELDER automatisch hinzufügen lassen. Die möglichen Operatoren und Vergleichswerte entsprechen dabei genau der Schreibweise einer Open-SQL-Anweisung. Für die Operatoren steht als Unterstützung eine F4-Hilfe bereit. Über die Spalte AND/OR können Sie Ihre Bedingungen verknüpfen.

Tabellenfelder			
Tabelle	Feldname	Operator	Vergleichswert
MAKT	SPRAS	EQ	'D'

Abbildung 3.28 Selektionsbedingung eines Views

AKTIVIEREN ( aktivieren) Sie gegebenenfalls erneut. Nun können Sie Ihren View testen. Dies ist über den Button DATENANZEIGE ( Datenanzeige) in der Funktionsleiste oder – wie eingangs beschrieben – über eine Open-SQL-Anweisung möglich. Wie Sie in Listing 3.1 sehen können, können auch hier nochmals eine Selektion sowie eine Projektion vorgenommen werden. Die Selektion von Datenbanktabellen mit ABAP ist in Kapitel 9, »Zugriff auf Datenbanken«, beschrieben.

```
DATA: lt_mat_texte TYPE TABLE OF zmein_db_view.
SELECT matnr mtktx FROM zmein_db_view
  INTO CORRESPONDING FIELDS OF TABLE lt_mat_texte
  WHERE ersda = sy-datum.
```

Listing 3.1 Verwendung eines Views in einer SELECT-Anweisung

### 3.8.2 Projektions-View

Wenn Sie im Pop-up-Fenster zum Anlegen eines Views den Typ PROJEKTIONSVIEW ausgewählt haben, gelangen Sie zum Projektions-View-Editor. Hier haben Sie im Vergleich zum Datenbank-View-Editor nicht die Möglichkeit, Tabellen miteinander zu verknüpfen, sondern können lediglich auf Basis einer Tabelle eine *Projektion*, also eine Auswahl von Feldern, vornehmen:

- Dazu tragen Sie auf der Registerkarte VIEWFELDER, wie in Abbildung 3.29 dargestellt, die Basistabelle ein und wählen die gewünschten Felder aus. Der Button TABELLENFELDER hilft Ihnen wieder bei der Auswahl der Felder.
- Darüber hinaus können Sie auf der Registerkarte PFLEGESTATUS steuern, ob auf die Tabelle nur lesend oder auch schreibend zugegriffen werden darf. Für die Auswahl der Felder für die DATA BROWSER/TABELLENSICHT-PFLEGE gelten dieselben Erläuterungen wie bei der Anlage einer Tabelle (siehe Abschnitt 3.5.4, »Tabellensicht-Pflege«).

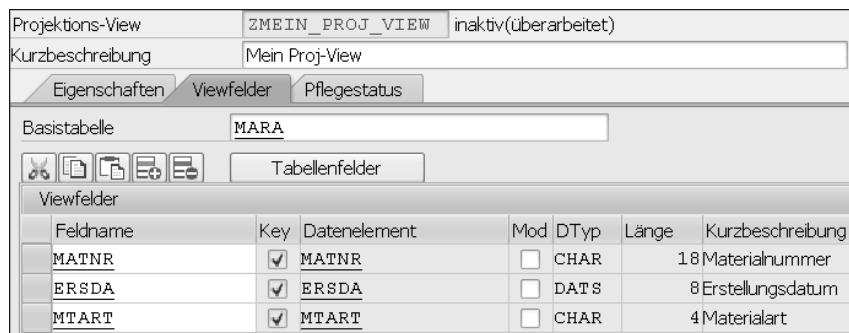


Abbildung 3.29 View-Felder eines Projekts-Views

### 3.8.3 Pflege-View

Wenn Sie in dem Pop-up-Fenster zum Anlegen eines Views den Typ PFLEGE-VIEW ausgewählt haben, gelangen Sie in den Pflege-View-Editor. Ihnen wird vermutlich auffallen, dass der Pflege-View-Editor genau denselben Aufbau hat wie der für Datenbank-Views (siehe Abschnitt 3.8.1). Der Unterschied ist, dass Sie hier auf der Registerkarte TABELLEN/JOINBEDINGUNGEN gezwungen sind, bei der Definition von Tabellenverknüpfungen über den Button BEZIEHUNGEN unterhalb der linken Tabelle zu arbeiten, die Tabellen also nicht frei eintragen können. Hiermit soll sichergestellt werden, dass nur mit Fremdschlüsselbeziehungen gearbeitet wird. Hintergrund ist, dass ein Pflege-View die Pflege mehrerer verknüpfter Tabellen ermöglicht. Sie können damit im Unterschied zu normalen Datenbank-Views auch ändernde Open-SQL-Befehle anwenden bzw. einen Pflegedialog generieren (siehe Abschnitt 3.9).

Um den Pflege-View anzulegen, gehen Sie wie folgt vor:

1. Tragen Sie, wie in Abschnitt 3.8.1, »Datenbank-View«, erläutert, die Basistabelle unter TABELLEN ein, und verknüpfen Sie diese über den Button BEZIEHUNGEN.
2. Anschließend können Sie auf der Registerkarte PFLEGESTATUS auswählen, inwiefern Ihr Pflege-View bearbeitet werden darf (siehe Abbildung 3.30). Für die Auswahlmöglichkeiten AUSLIEFERUNGSKLASSE und DATA BROWSER/TABELLENSICHT-PFLEGE gelten dabei dieselben Erläuterungen wie bei der Anlage einer Datenbanktabelle (siehe Abschnitt 3.5.3 und Abschnitt 3.5.4).

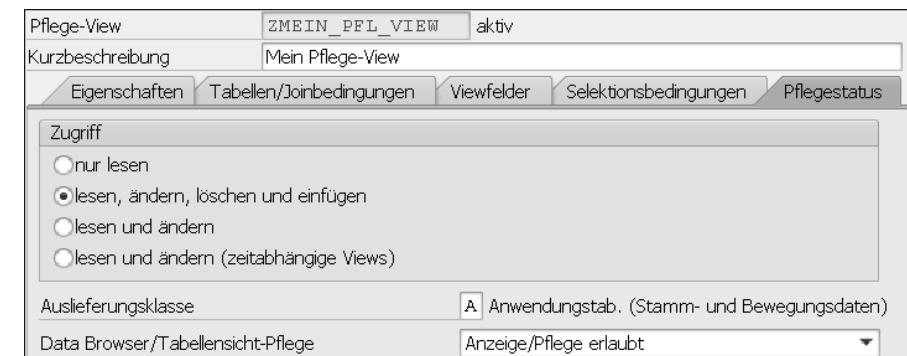


Abbildung 3.30 Pflegestatus eines Pflege-Views definieren

### 3.8.4 Help-View

Wenn Sie im Pop-up-Fenster zum Anlegen eines Views den Typ HELP-VIEW ausgewählt haben, gelangen Sie in den Help-View-Editor. Dieser ist wie der Editor zur Anlage eines Pflege-Views aufgebaut. Auch hier müssen Sie die einzelnen Tabellenverknüpfungen auf der Registerkarte TABELLEN/JOINBEDINGUNGEN über den Button BEZIEHUNGEN eintragen.

Anschließend können Sie diesen Help-View als Datenquelle für eine Suchhilfe angeben. Dieses Vorgehen ist in Abschnitt 3.10, »Suchhilfen«, beschrieben.

## 3.9 Pflegedialoge

Datenbanktabellen müssen mit Daten gefüllt werden. Damit dies nicht ausschließlich über SQL-Befehle (siehe Kapitel 9, »Zugriff auf Datenbanken«) geschehen kann, bietet SAP die Möglichkeit, einen sogenannten *Pflegedialog* für eine Datenbanktabelle anzulegen. Über einen solchen Pflegedialog können Sie den Inhalt einer Datenbanktabelle pflegen. Sie können Zeilen hinzufügen, löschen oder ändern. Häufig wird für Pflegedialoge auch der Begriff *SM30-Tabelle* verwendet, da diese über den Transaktionscode SM30 (Tabelbensicht-Pflege) zugänglich sind.

In den folgenden Abschnitten zeige ich Ihnen, wie Sie einen Pflegedialog anlegen (siehe Abschnitt 3.9.1) und die Eingabemaske verbreitern können (siehe Abschnitt 3.9.2).

### 3.9.1 Pflegedialog anlegen

Zur Anlage eines Pflegedialogs gehen Sie wie folgt vor:

1. Wechseln Sie in Transaktion SE11, und lassen Sie sich die Tabelle anzeigen, zu der Sie den Pflegedialog anlegen wollen (siehe Abschnitt 3.5.1, »Datenbanktabellen anzeigen«).
2. Aus der Tabellenanzeige können Sie nun über den Pfad HILFSMITTEL • TABELLENPFLEGEGENERATOR im Hauptmenü zur Anlage eines Pflegedialogs gelangen.
3. Folgende Angaben müssen Sie in der Anlageoberfläche machen (siehe Abbildung 3.31):
  - BERECHTIGUNGSGRUPPE: Wer darf später auf die Tabelle zugreifen, alle Benutzer oder nur bestimmte?
  - FUNKTIONSGRUPPE: die Funktionsgruppe, in der die Dynpros und die Ablauflogik für den Pflegedialog generiert werden sollen
  - PFLEGETYP: Möchten Sie eine Suchmaske als Einstiegsbild haben (ZWEISTUFIG), oder soll der Benutzer direkt zur Tabellenanzeige gelangen (EINSTUFIG)?
  - PFLEGEBILDNUMMER: Wenn Sie als PFLEGETYP EINSTUFIG ausgewählt haben, müssen Sie nur das Eingabefeld ÜBERSICHTSBILD füllen, bei einem zweistufigen Pflegedialog müssen Sie sowohl das Eingabefeld ÜBERSICHTSBILD als auch das Feld EINZELBILD füllen. Vergeben Sie hier jeweils eine Nummer (z. B. 1 oder 2), diese müssen sich lediglich unterscheiden.
  - AUFZEICHNUNGSROUTINE: Wählen Sie STANDARD AUFZEICHNUNGSROUTINE, fordert das System nach einer Pflege von Tabellenzeilen einen Transportauftrag an, damit ein Transport in andere Systeme ermöglicht wird. Bei der Auslieferungsklasse »C« ist diese Option standardmäßig eingestellt.
4. Klicken Sie anschließend in der Funktionsleiste auf ANLEGEN (□).

Wenn alles geklappt hat, können Sie Ihren Pflegedialog nun in Transaktion SM30 testen. Geben Sie hierzu den Namen der Tabelle ein, und klicken Sie auf PFLEGEN.

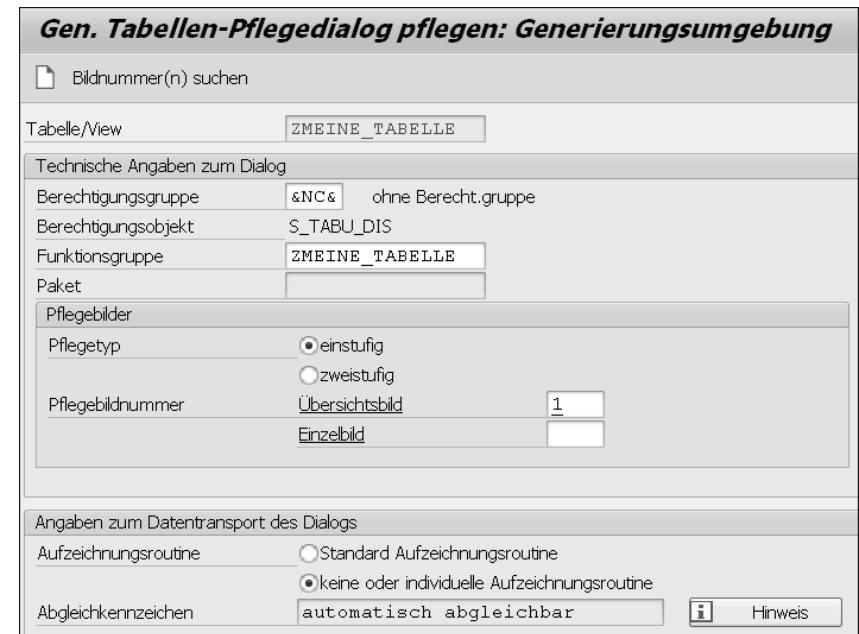


Abbildung 3.31 Generierung eines Pflegedialogs

### 3.9.2 Pflegedialog verbreitern

Wenn Sie Ihren Pflegedialog in Transaktion SM30 aufrufen, werden Sie feststellen, dass die Breite der Tabelle viel zu knapp bemessen ist. Die in Abbildung 3.32 markierte Fläche ist damit ungenutzt. Dies ist gerade bei vielen Spalten für die Anwender sehr unschön und umständlich. Daher gibt es einen Trick, um den Pflegedialog zu verbreitern.



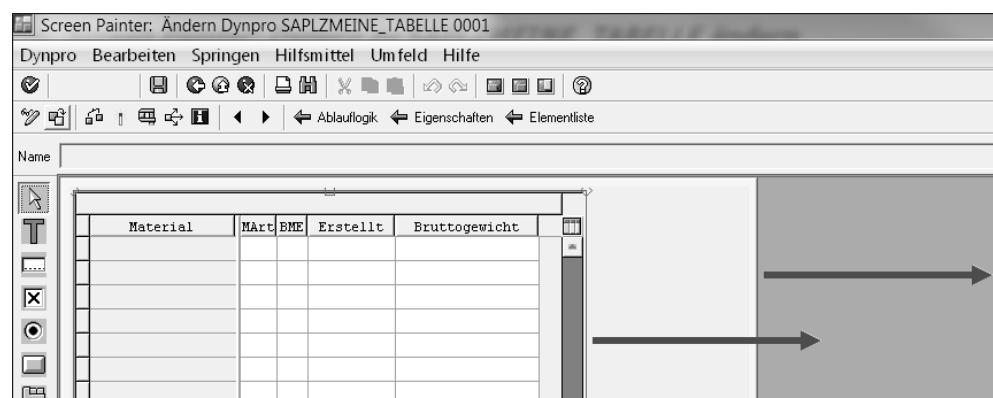
Abbildung 3.32 Ungenutzter Platz eines Pflegedialogs

#### Weitere Informationen zur Dynpro-Programmierung

Zum Verständnis der folgenden Erläuterungen ist es hilfreich, wenn Sie bereits grundlegende Kenntnisse in der Dynpro-Programmierung haben. Die Grundlagen lernen Sie in Kapitel 14.

Die Vorgehensweise ist überschaubar

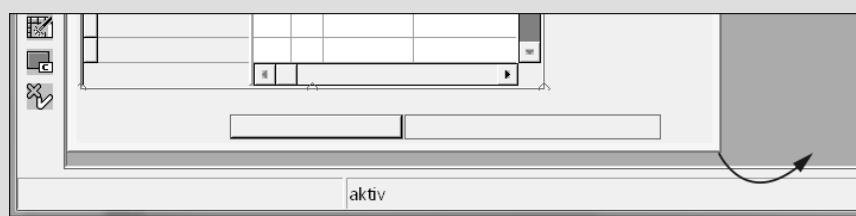
1. Öffnen Sie die Funktionsgruppe zum Pflegedialog, indem Sie in Transaktion SE80 die Dropdown-Liste FUNKTIONSGRUPPE nutzen.
  2. Öffnen Sie das Dynpro für das Übersichtsbild des Pflegedialogs (im Ordner DYNPROS, z. B. Dynpro 0001).
  3. Wechseln Sie in den Bearbeitungsmodus über den Button ANZEIGEN <-> ÄNDERN ().
  4. Öffnen Sie den Layout-Editor über den Button LAYOUT.
  5. Verbreitern Sie, wie in Abbildung 3.33 dargestellt, den Hauptbereich für das Fenster, indem Sie den Rahmen mit der Maus nach rechts ziehen.



**Abbildung 3.33** Verbreiterung eines Pflegedialogs im Screen Painter

[!] Dynpro-Größe ändern

Falls Ihnen dabei die Meldung »Element(e) außerhalb der neuen Grenzen (Dynpro-Größe nicht verändert)« angezeigt wird, müssen Sie die Dynpro-Größe vergrößern. Scrollen Sie dazu, wie in Abbildung 3.34 dargestellt, zum unteren Rand des Bildschirms im Screen Painter, und ziehen Sie die Ecke des Dynpros mit gedrückter linker Maustaste erst nach unten und dann nach rechts.



**Abbildung 3.34** Dynpros im Screen Painter verbreiter

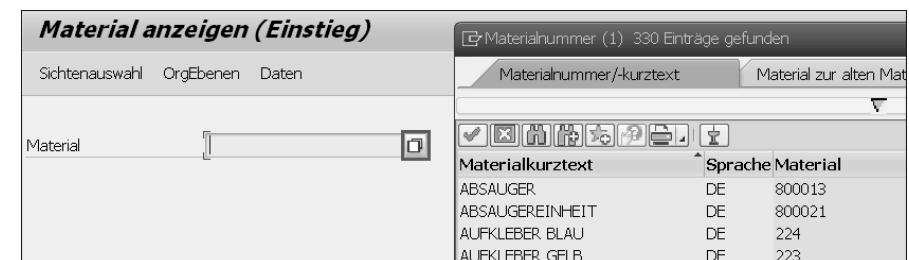
6. Aktivieren ( ) Sie anschließend, und testen Sie Ihre Änderungen in Transaktion SM30.

### **Neugenerierung bei Anpassung**

Falls Sie den Pflegedialog später in Transaktion SE11 bzw. SE54 nochmals anpassen, wird das Dynpro teilweise neu generiert und wieder auf die Standardgröße gebracht. In diesem Fall müssen Sie also die beschriebenen Änderungen noch einmal durchführen.

### 3.10 Suchhilfen

Suchhilfen stehen innerhalb des SAP-Systems als Eingabehilfen für den Anwender bereit. Sie zeigen dem Anwender eine Liste aller möglichen Eingabewerte für ein Eingabefeld. Die möglichen Eingabewerte können, wie in Abbildung 3.35 zu erkennen, durch weitere Informationen angereichert sein, damit der Anwender z. B. zu einer Materialnummer auch den zugehörigen Materialtext sieht und ihm so die Auswahl leichter fällt. Durch einen Doppelklick auf eine Zeile in der Suchhilfe wird der ausgewählte Wert in das Eingabefeld übernommen.



**Abbildung 3.35** Eine Suchhilfe für die Materialnummer

Innerhalb des ABAP Dictionarys können Sie eigene Suchhilfen anlegen. Geben Sie dazu den gewünschten Namen der Suchhilfe auf dem Hauptübersichtsbildschirm der Transaktion SE11 im Feld **SUCHHILFE** ein, und klicken Sie auf **ANLEGEN**. Im sich öffnenden Pop-up-Fenster haben Sie folgende Arten von Suchhilfen zur Auswahl:

- ▶ *Elementare Suchhilfe* als Standardeingabehilfe (siehe Abschnitt 3.10.1)
  - ▶ *Sammelsuchhilfe* als Zusammenfassung von mehreren elementaren Suchhilfen, wobei für jede Suchhilfe eine Registerkarte angezeigt wird (siehe Abschnitt 3.10.2)

Innerhalb von Suchhilfen bietet Ihnen das System darüber hinaus mit *Append-Suchhilfen* die Möglichkeit, Sammelsuchhilfen modifikationsfrei zu erweitern. Dieses Vorgehen ist in Abschnitt 21.4.8, »Suchhilfen-Erweiterungen«, beschrieben.

### 3.10.1 Elementare Suchhilfe

Wenn Sie im Pop-up-Fenster zur Anlage einer Suchhilfe den Typ ELEMENTARE SUCHHILFE ausgewählt haben, gelangen Sie in den Suchhilfen-Editor:

1. Zur Definition einer Suchhilfe müssen Sie zunächst folgende Eigenschaften definieren (siehe Abbildung 3.36):

- **SELEKTIONSMETHODE:** Wählen Sie, ob die Daten von einer Datenbank oder einem View gelesen werden sollen.
- **DIALOGVERHALTEN:** Wählen Sie, ob ein Fenster zur Werteeinschränkung angezeigt werden soll oder ob die Ergebnisliste direkt angezeigt werden soll.

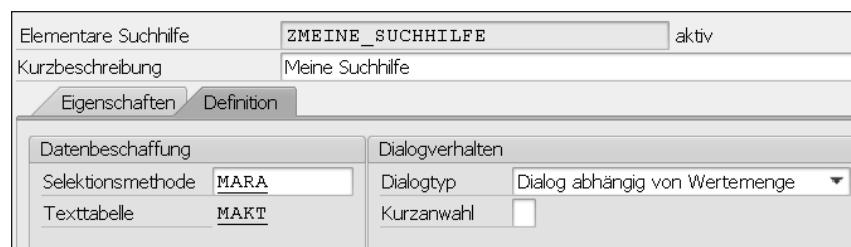


Abbildung 3.36 Selektionsmethode und Dialogverhalten einer Suchhilfe

2. Darüber hinaus müssen Sie die SUCHHILFEPARAMETER bestimmen (siehe Abbildung 3.37). Das sind die Felder, die im Suchhilfefenster angezeigt werden sollen. Dabei müssen Sie auch bestimmen, welche Felder bei der Auswahl eines Eintrags aus der Ergebnisliste übernommen werden.

Parameter	Suchhilfeparameter	IMP	EXP	LPos	SPos	SAnz	Datenelement	M...	Defaultwert
	MATNR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	1	<input type="checkbox"/>	MATNR	<input type="checkbox"/>	
	MAKTX	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2	2	<input type="checkbox"/>	MAKTX	<input type="checkbox"/>	

Abbildung 3.37 Definition von Suchhilfeparametern

3. Über die Spalte IMP definieren Sie, welche Werte aus Ihrer Datenquelle übernommen werden, also in der Wertetabelle angezeigt werden können.

4. Über die Spalte EXP definieren Sie, welche Werte in die Maske übernommen werden.

5. Über die Spalte LPos definieren Sie, ob das Feld in der Trefferliste auch wirklich dargestellt werden soll. Mit der Nummer geben Sie die Reihenfolge der Spalten an (z. B. zuerst das Feld MATNR für das MATERIAL ❶ und anschließend das Feld MAKTX für den MATERIALKURZTEXT ❷, siehe Abbildung 3.38). Soll der Parameter nicht auf der Trefferliste erscheinen, lassen Sie dieses Feld frei.

6. Über die Spalte SPos definieren Sie, ob das Feld in der Suchmaske dargestellt werden soll. Mit der Nummer geben Sie die Reihenfolge der Eingabefelder an (z. B. erst das Feld MATNR für das MATERIAL ❶ und dann das Feld MAKTX für den MATERIALKURZTEXT ❷, siehe Abbildung 3.38). Soll der Parameter nicht auf der Suchmaske erscheinen, so lassen Sie dieses Feld frei.

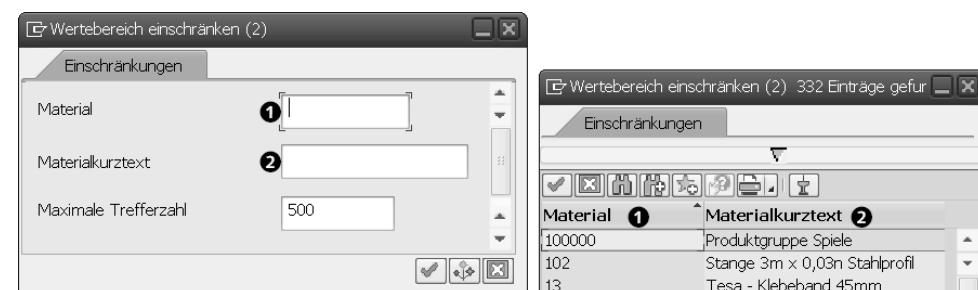


Abbildung 3.38 Unterschied zwischen den Angaben in den Spalten »LPos« und »SPos«

7. Zum Schluss AKTIVIEREN (aktivieren Sie Ihre Suchhilfe.

Nun können Sie die Suchhilfe an ein Datenelement hängen (siehe auch Abschnitt 3.2) oder sie direkt in einem Selektionsbildschirm durch die Angabe eines Parameters mit dem Zusatz MATCHCODE OBJECT verwenden:

PARAMETERS: p\_matnr TYPE matnr MATCHCODE OBJECT zmeine\_suchhilfe.

### 3.10.2 Sammelsuchhilfe

Wenn Sie im Pop-up-Fenster zur Anlage einer Suchhilfe den Typ SAMMEL-SUCHHILFE ausgewählt haben, gelangen Sie zum Suchhilfen-Editor:

1. Zur Definition einer Sammelsuchhilfe müssen Sie zuerst die Suchhilfeparameter definieren (d. h. bestimmen, welche Werte übernommen und welche zurückgegeben werden). Diese Suchhilfeparameter tragen Sie auf der Registerkarte DEFINITION ein (siehe Abbildung 3.39).

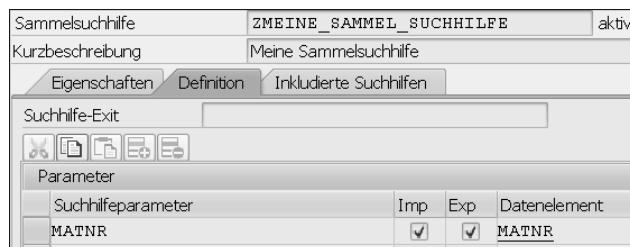


Abbildung 3.39 Suchhilfeparameter einer Sammelsuchhilfe

2. Anschließend geben Sie auf der Registerkarte INKLUDIERTE SUCHHILFEN die elementaren Suchhilfen an, die zur Sammelsuchhilfe zusammengefasst werden sollen.
3. Nun müssen Sie die auf der Registerkarte DEFINITION definierten Suchhilfeparameter den Suchhilfeparametern der inkludierten Suchhilfen zuordnen. Klicken Sie hierzu, wie in Abbildung 3.40 dargestellt, auf die Zeile der Suchhilfe ① und anschließend auf den Button PARAMETERZUORDNUNG ②.
4. Tragen Sie einen BEZUGSPARAMETER ein, und bestätigen Sie Ihre Zuordnung mit dem Button ÜBERNEHMEN.

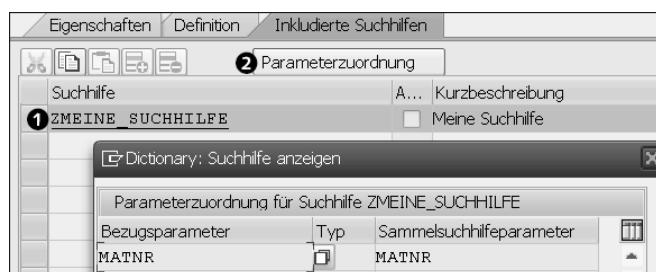


Abbildung 3.40 Parameterzuordnung in einer Sammelsuchhilfe

5. AKTIVIEREN ( aktivieren Sie anschließend Ihre Sammelsuchhilfe.

Nun können Sie die Sammelsuchhilfe an ein Datenelement hängen (siehe Abschnitt 3.2) oder wie die elementare Suchhilfe in einem Selektionsbildschirm verwenden (siehe Abschnitt 3.10.1).

## 3.11 Datenbank-Utility-Tool

Das Datenbank-Utility-Tool brauchen Sie vor allem dann, wenn Ihre Datenbanktabelle bereits Daten beinhaltet, Sie aber aufgrund einer neuen Anforderung z. B. neue Schlüsselfelder hinzufügen müssen und beim Aktivieren

der Tabelle dadurch folgende Fehlermeldung erhalten: »Strukturänderung auf Feldebene (Bitte Tabelle umsetzen)«.

Wählen Sie im Hauptmenü HILFSMITTEL • DATENBANKOBJEKT • DATENBANK-UTILITY, um das Tool zu öffnen. Sie können nun drei grundlegende Datenbankoperationen (siehe Abbildung 3.41) durchführen:

- DATENBANKTABELLE ANLEGEN
- DATENBANKTABELLE LÖSCHEN
- AKTIVIEREN UND DATENBANK ANPASSEN

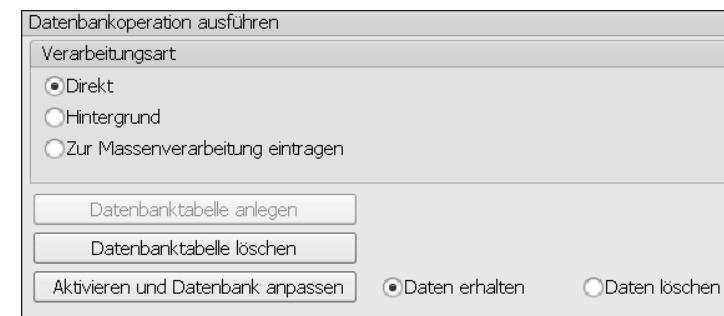


Abbildung 3.41 Datenbankoperationen mit dem Datenbank-Utility-Tool

Um die geänderte Datenbanktabelle zu aktivieren, wählen Sie AKTIVIEREN UND DATENBANK ANPASSEN. Bestätigen Sie das Pop-up-Fenster AUFTRAG: ANPASSEN mit JA.

## 3.12 Das Sperrkonzept

Über das Sperrkonzept soll verhindert werden, dass durch gleichzeitige Änderungen Datenschieflstände in einer Datenbanktabelle entstehen. Vergleichen könnte man diesen Mechanismus mit dem Sperrmechanismus in Microsoft Excel: Nur ein Benutzer darf mit einer Excel-Tabelle arbeiten, ein anderer Benutzer darf derweil auf die Daten nur lesend zugreifen. Das Sperrkonzept im ABAP Dictionary geht hier noch einen Schritt weiter und ermöglicht nicht nur das Sperren einer kompletten Datenbanktabelle, sondern auch das Sperren einzelner Einträge einer Datenbanktabelle. In einem Sperrobjekt werden die Tabellen mit ihren Schlüsselfeldern angegeben, in denen bestimmte Datensätze gesperrt werden sollen. Das Setzen bzw. Freigeben von Sperren erfolgt durch den Aufruf von Funktionsbausteinen, die automatisch bei der Definition eines Sperrobjekts generiert werden.

## Ein Sperrobject anlegen

Um ein Sperrobject anzulegen, gehen Sie wie folgt vor:

1. Geben Sie den gewünschten Namen des Sperrobjekts auf dem Hauptübersichtsbildschirm der Transaktion SE11 im Feld SPERROBJEKT ein. Der Name muss dabei mit einem E anfangen, gefolgt vom Kundennamensraum (z. B. Z oder Y). Beispielsweise könnte Ihr Sperrobject EZ\_MEIN\_SPEROBJ heißen.
2. Klicken Sie anschließend auf ANLEGEN.
3. Es öffnet sich der Sperrobject-Editor, in dem Sie nun auf der Registerkarte TABELLEN die zu sperrende Tabelle eintragen und einen Sperrmodus auswählen (siehe Abbildung 3.42). Folgende (wichtige) Sperrmodi stehen Ihnen dazu zur Verfügung:
  - SCHREIBSPERRE: Nur ein Benutzer kann die gesperrten Daten lesen bzw. bearbeiten.
  - LESESPERRE: Mehrere Benutzer können auf dieselben Daten (lesend) zugreifen. Sobald ein Benutzer jedoch die Daten bearbeitet, hat ein zweiter Benutzer keinen Zugang auf diese Daten.
  - ERWEITERTE SCHREIBSPERRE: Eine Transaktion kann nur eine Sperre anfragen, jede weitere Anforderung einer Sperre wird abgewiesen.

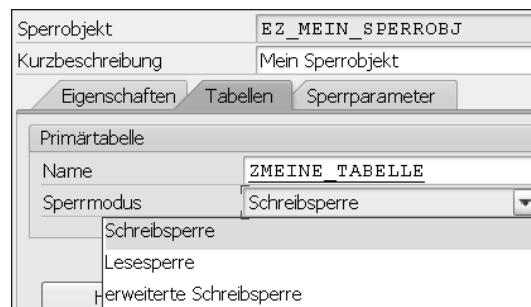


Abbildung 3.42 Sperrmodi eines Sperrobjekts

4. Auf der Registerkarte SPERRPARAMETER können Sie zusätzlich die Sperrparameter (in Form der Primärschlüssel) bearbeiten und gegebenenfalls einen Primärschlüssel deaktivieren.
5. Zuletzt AKTIVIEREN (aktivieren Sie Ihr Sperrobject.

Wenn Sie nun im Hauptmenü den Pfad SPRINGEN • SPERRBAUSTEINE wählen, werden Ihnen zwei Funktionsbausteine präsentiert (siehe Abbildung 3.43).

Funktionsgruppe	Kurztext	Funktionsgruppe
Name des Funktionsbausteins	Kurztext zum Funktionsbaustein	
/1BCDWBEN/TEN0000	Enqueue und Dequeue Funktionen	
DEQUEUE_EZ_MEIN_SPEROBJ	Enqueue-Sperre für Objekt EZ_MEIN_SPEROBJ freigeben	
ENQUEUE_EZ_MEIN_SPEROBJ	Enqueue-Sperre für Objekt EZ_MEIN_SPEROBJ anfordern	

Abbildung 3.43 Sperrbausteine eines Sperrobjekts

Diese können Sie nun verwenden, um die Datenbanktabelle zu sperren bzw. zu entsperren. Wie Sie Funktionsbausteine mit ABAP generell aufrufen, ist in Abschnitt 13.4.2, »Aufruf von Funktionsbausteinen«, beschrieben. Sie müssen beim Aufruf lediglich die Primärschlüssel der zu sperrenden Datensätze angeben.

## Einleitung

ABAP ist die von der SAP entwickelte Programmiersprache für die Erstellung kundenspezifischer Anwendungsprogramme. Die Abkürzung steht für Advanced Business Application Programming. Darüber hinaus nutzt SAP selbst ABAP auch als Basis für SAP ERP, die SAP Business Suite und die SAP-NetWeaver-Plattform. All diese Systeme basieren auf Millionen von in ABAP geschriebenen Zeilen.

Die Abkürzung ABAP bedeutete ursprünglich Allgemeiner Berichtsaufbereitungsprozessor und war, wie es der Name vermuten lässt, dafür gedacht, kundeneigene Berichte aufzubereiten. ABAP orientierte sich in seiner ursprünglichen Form an der Syntax der Programmiersprache COBOL. Über die Jahrzehnte wurde ABAP kontinuierlich weiterentwickelt und verbessert. So hat sich der Sprachumfang mittlerweile vervielfacht, da immer wieder neue Anweisungen und Funktionen hinzukamen, wobei ältere aus Kompatibilitätsgründen beibehalten wurden.

Die aktuelle Version von ABAP (7.51, Stand Ende 2016) steht in ihrer Funktionalität anderen modernen Programmiersprachen kaum noch nach. Bereits 2004 wurde der Sprachumfang um die objektorientierten Elemente von *ABAP Objects* erweitert. Dennoch findet man aufgrund der erwähnten Kompatibilitätssicherung in ABAP auch heute noch viel ältere Funktionalität, die sich mit den neuen, modernen Konstrukten vermischt hat. In der Praxis hat der Entwickler daher häufig mit einer Mischform aus »alter« und »neuer« Welt zu tun. Eine einseitige Betrachtung ergibt daher keinen Sinn.

Dennoch ist die Zukunft von ABAP bereits angebrochen. ABAP hat durch die letzten beiden Releases 7.40 und 7.50/7.51 sehr viele neue Funktionen gewonnen, die in die Kapitel dieses Buches integriert sind. Besonders wichtig in Bezug auf dieses neue ABAP sind die neue Datenbank SAP HANA und die neue Oberflächentechnologie SAPUI5 bzw. die von SAP bereitgestellten SAP-Fiori-Apps. Diese neuen Technologien sind das Aushängeschild des SAP-ERP-Nachfolgers SAP S/4HANA. Doch auch wenn die SAP Business Suite für SAP S/4HANA (auf Vorlage der alten Systeme) komplett überarbeitet wurde, bleibt ABAP die Kernprogrammiersprache auch für dieses System.

Betrachtet man die durchschnittliche Halbwertszeit (d. h. die Einsatzzeit beim Kunden) eines ERP-Systems von ca. 20 Jahren (SAP R/3 kam 1993, SAP ECC 2004), wird uns ABAP durch SAP S/4HANA noch lange erhalten bleiben – auch wenn dies in den letzten Jahren (z. B. im Rahmen der Ankündigung der Java-Strategie) nicht immer als Selbstverständlichkeit galt.

In anderen Worten: ABAP ist die Programmiersprache von SAP und wird dies bleiben. Herzlich willkommen seien daher alle Leser, die sich entschieden haben, sich intensiver mit ABAP im Allgemeinen und ABAP Objects im Speziellen zu beschäftigen.

## Aufbau dieses Buches

Zuallererst möchte ich Ihnen einen Überblick über das vor Ihnen liegende Buch geben. Im Anschluss habe ich Ihnen verschiedene Lesepläne zusammengestellt – abhängig davon, welche Vorkenntnisse Sie mitbringen.

- ▶ **Teil I, »Die Werkzeugkiste des ABAP-Entwicklers«,** beschäftigt sich mit den Entwicklungswerkzeugen des SAP GUI wie der eingebauten ABAP Workbench (Transaktion SE80). Außerdem stelle ich Ihnen hier die neuen ABAP Development Tools for SAP NetWeaver (ADT) für die Entwicklungs-umgebung Eclipse vor.
- ▶ **Teil II, »Der Kern der Sprache ABAP«,** beschäftigt sich mit dem ABAP-Sprachkern. Von absolut grundlegenden Anweisungen über den Zugriff auf Datenbanktabellen, die Verarbeitung der Daten und ABAP-Objects-Anweisungen bis hin zur Report- und Dynpro-Programmierung ist hier alles vertreten. So hat hier auch der Zugriff auf die neue SAP-HANA-Datenbank mit ABAP (auch *ABAP für HANA* genannt) mit Eingang gefunden. Insbesondere finden Sie in diesem Teil die mit den Releases 7.40 und 7.50/1 neu eingeführten Sprachbefehle und Funktionen.
- ▶ **Teil III, »Techniken zur Qualitätssicherung«,** dreht sich um den Test und die Qualitätskontrolle Ihrer geschriebenen Anweisungen. Hier werden die gezielte Verwendung des ABAP Debuggers sowie die zur Verfügung stehenden Analysewerkzeuge erläutert.
- ▶ **Teil IV, »Fortgeschrittene Programmiertechniken«,** behandelt weitere Themen aus dem Umfeld von ABAP. Dazu gehören die Entwicklung von Schnittstellen, die Erweiterung von SAP-Standardprogrammen, die Entwicklung von Formularen und Tabellenanzeigen sowie auch fortgeschritte-

tene Techniken wie die dynamische Programmierung oder die Anwendung objektorientierter Frameworks wie der Object Services oder Shared Objects.

- ▶ **Teil V, »Objektorientierte Programmierung«,** vermittelt in erster Linie die Grundlagen der objektorientierten Programmierung mit ABAP und zeigt konkrete Methoden für den Entwicklungsprozess. Insbesondere werden hier die objektorientierte Modellierung sowie Hilfsmittel wie UML und Entwurfsmuster besprochen.
- ▶ Im letzten Teil des Buches, **Teil VI, »Ein Blick über den Tellerrand: Was Sie als ABAP-Entwickler sonst noch kennen sollten«,** möchte ich Ihnen den Grund für die vielen Änderungen der letzten Releases etwas genauer vorstellen: SAP HANA als neue Datenbankplattform sowie SAPUI5 als neue Web-Frontend-Technologie, die eine regelrechte Innovationswelle angestoßen haben. In diesem Teil gehe ich insbesondere sowohl auf die vielen neuen Begriffe im Umfeld dieser Technologien als auch auf die Entwicklung von OData-Services ein.

Der Anhang fasst alle wichtigen ABAP-Anweisungen, obsoleten ABAP-Anweisungen, Systemfelder, eingebauten Datentypen, nützliche Funktionsbausteine und Klassen, Transaktionscodes sowie technische Tabellen zur Programmierung mit ABAP zusammen.

## Ihr Weg durch das Buch

Da sich dieses Buch an mehrere Zielgruppen richtet (ABAP-Einsteiger, -Kenner und -Experten) und gleichermaßen als Nachschlagewerk für den Alltag dienen soll, finden Sie im Folgenden einige Hinweise, wie Sie aus dem Buch für sich am meisten Gewinn erzielen können.

Aus folgenden Leseplänen können Sie wählen:

- ▶ Einsteiger in ABAP
- ▶ Einsteiger in ABAP Objects
- ▶ Kenner von ABAP Objects (vor allem der Releases vor und bis 7.31)
- ▶ Interessenten an den neuen Technologien
- ▶ Experten

Die Lesepläne bauen aufeinander auf, vom Einsteiger bis zum Experten. Versuchen Sie, wann immer es geht, so viel wie möglich am System nachzuvoll-

ziehen, da sich Programmierung am besten durch die konkrete Anwendung erlernen lässt.

In hervorgehobenen Informationskästen sind Inhalte zu finden, die wissenswert und hilfreich sind, aber etwas außerhalb der eigentlichen Erläuterung stehen. Damit Sie die Informationen in den Kästen sofort einordnen können, haben wir die Kästen mit Symbolen gekennzeichnet:

- [+] Die mit diesem Symbol gekennzeichneten *Tipps* geben Ihnen spezielle Empfehlungen, die Ihnen die Arbeit erleichtern können.
- [»] In Kästen, die mit diesem Symbol gekennzeichnet sind, finden Sie Informationen zu *weiterführenden Themen* oder wichtigen Inhalten, die Sie sich merken sollten.
- [!] Dieses Symbol weist Sie auf *Besonderheiten* hin, die Sie beachten sollten. Es *warnt Sie* außerdem vor häufig gemachten Fehlern oder Problemen, die auftreten können.

### **Leseplan für Einsteiger in ABAP**

Fangen Sie mit der Lektüre des Kapitel 1 an, um sich mit der Entwicklungsumgebung vertraut zu machen. Insbesondere ist hier Abschnitt 1.4.2, »Einen Report anlegen«, von Bedeutung, der Ihnen zeigt, wie Sie einen neuen Report in Transaktion SE80 anlegen und ausführen können.

Bevor Sie sich dann mit den Anweisungen von ABAP auseinandersetzen, ist es wichtig, die Rolle des in Kapitel 3 beschriebenen ABAP Dictionarys zu verstehen. Lesen Sie sich hier alle Abschnitte bis Abschnitt 3.4, »Tabellentypen«, durch. Die folgenden Abschnitte können Sie erstmal außer Acht lassen.

Mit Ihrem ersten angelegten Report und dem Grundwissen über das ABAP Dictionary können Sie nun die in Kapitel 7, »Die ABAP-Grundbefehle«, beschriebenen grundlegenden ABAP-Anweisungen ausprobieren. Versuchen Sie, die vielen Beispiele direkt bei sich im System auszuprobieren. Überspringen Sie erstmal die als fortgeschritten markierten Abschnitte.

Nach einer kurzen Verschnaufpause können Sie sich direkt intensiv mit Kapitel 8 auseinandersetzen, das die internen Tabellen von ABAP intensiv behandelt. Kapitel 9 zeigt Ihnen im Anschluss, wie Sie Daten von der Datenbank in interne Tabellen laden und weiterverarbeiten können.

Lesen Sie nun die restlichen Abschnitte (ab Abschnitt 3.5) von Kapitel 3, »Das ABAP Dictionary«. Insbesondere ist hier Abschnitt 3.5, »Datenbank-

tabellen«, von Bedeutung, der Ihnen zeigt, wie Sie selbst eine Tabelle auf der Datenbank anlegen können. So eine Tabelle können Sie mit den in Kapitel 9, »Zugriff auf Datenbanken«, gelernten Open-SQL-Anweisungen bzw. den in Abschnitt 3.9 beschriebenen Pflegedialogen bearbeiten.

Haben Sie eine SAP-HANA-Datenbank zur Verfügung, empfehle ich Ihnen nun die Lektüre von Kapitel 28, »SAP HANA«, sowie anschließend von Kapitel 10, »Zugriff auf SAP-HANA-Entwicklungsobjekte«. Sollten Sie keine HANA-Datenbank zur Verfügung haben, so ist dies nicht weiter schlimm, da Sie diese beiden Kapitel jederzeit nachträglich ohne Probleme durcharbeiten können.

Kapitel 12 beschäftigt sich dann mit der Erstellung von Selektionsbildschirmen in Reports. Dies hilft Ihnen dabei, Eingabemasken für Ihre Anwender bereitzustellen. Kapitel 4 zeigt Ihnen, wie Sie es schaffen, Ihren Report mit einem Transaktionscode zu verknüpfen, damit Ihre Anwender den von Ihnen entwickelten Report direkt aufrufen können.

Da die Programme durch die vielen Anweisungen immer größer werden, finden Sie in Kapitel 13 die in ABAP zur Verfügung stehenden Strukturierungselemente, die Ihnen dabei helfen, Ihre Programme in kleinere, voneinander abgetrennte Bereiche zu unterteilen. Insbesondere ist hier der Abschnitt 13.4, »Funktionsbausteine«, von Bedeutung. Lesen Sie sich dazu auch Kapitel 5, »Der Function Builder«, durch, um zu verstehen, wie Sie eigene Funktionsbausteine anlegen und verwenden können. Natürlich können Sie auch direkt Klassen und Interfaces zur objektorientierten Strukturierung Ihrer Programme nutzen, wie in Kapitel 11, »Die ABAP-Objects-Syntax«, beschrieben.

Anschließend können Sie dem Leseplan für Einsteiger in ABAP Objects folgen. Haben Sie bis hierhin alle Kapitel gemeistert, können Sie sich dem Umfeldwissen zu ABAP widmen, hierzu gehören die Kapitel in Teil III, Teil IV und Teil VI. Heben Sie sich Kapitel 23, »Fortgeschrittene Programmietechniken«, bis ganz zum Schluss auf.

### **Leseplan für Einsteiger in ABAP Objects**

Dieser Leseplan ist für all jene gedacht, die sich mit dem Sprachkern von ABAP bestens auskennen, aber noch nicht die Chance hatten, sich mit ABAP Objects auseinanderzusetzen, oder bei denen der letzte Einsatz der objektorientierten Konzepte schon eine Weile her ist.

Lesen Sie zunächst Kapitel 24, »Grundlagen der Objektorientierung«, sowie Kapitel 11, »Die ABAP-Objects-Syntax«. Üben Sie nach der Lektüre dieser beiden Kapitel erst einmal intensiv die Anwendung von Klassen und Instanzen am Beispiel des SAP List Viewers (ALV) mithilfe von Kapitel 19. Konzentrieren Sie sich hier insbesondere auf die neue ALV-Anzeige auf Basis der Klasse `CL_SALV_TABLE`, die in Abschnitt 19.2 beschrieben ist.

Legen Sie wie in Kapitel 6, »Der Class Builder«, beschrieben, eine eigene Klasse an, die Sie anschließend in Ihrem Report instanziieren und verwenden können. Hinweise zur objektorientierten Gestaltung von Reports gibt Abschnitt 27.6, »Objektorientierte Reports«.

Arbeiten Sie nun schrittweise die restlichen Kapitel in Teil V durch. Am Ende in der Einleitung zu Teil V finden Sie Verweise auf die konkrete Anwendung bereits in SAP existierender objektorientierter Frameworks.

### **Leseplan für Kenner von ABAP Objects**

Sie kennen bereits ABAP und ABAP Objects und sind vor allem an den Neuerungen bis Release 7.51 interessiert? Dann ist dieser Leseplan richtig für Sie.

Wenn Sie die ABAP Development Tools (ADT) für Eclipse noch nicht kennen, sollten Sie direkt mit Kapitel 2 einsteigen. Anschließend sind für Sie insbesondere die Übersichten in den folgenden Abschnitten interessant, die Auskunft über die ABAP-Neuerungen geben und jeweils auf die Abschnitte im Buch verweisen, in denen diese ausführlicher behandelt werden:

- ▶ Abschnitt 7.15, »Änderungen und Neuerungen bis ABAP 7.51«
- ▶ Abschnitt 8.12, »Änderungen und Neuerungen im Umfeld von internen Tabellen bis ABAP 7.51«
- ▶ Abschnitt 9.9, »Änderungen und Neuerungen im Umfeld von Open SQL bis ABAP 7.51«
- ▶ Abschnitt 10.4, »Änderungen und Neuerungen beim Zugriff auf SAP-HANA-Entwicklungsobjekte bis ABAP 7.51«
- ▶ Abschnitt 11.15, »Änderungen und Neuerungen in ABAP Objects bis ABAP 7.51«

Insbesondere sollten Sie nach der Lektüre folgende Begriffe verstanden haben und anwenden können:

- ▶ Inline-Deklaration
- ▶ Tabellenausdrücke

- ▶ Änderungen an Open SQL, z. B. die neuen Open-SQL-Ausdrücke
- ▶ Konstruktorausdrücke, insbesondere die Verwendung der Zusätze von `BASE` und `LET`
- ▶ den Zusatz `GROUP BY` der Anweisung `LOOP AT`
- ▶ ABAP Managed Database Procedures (AMDP) zur Programmierung von Stored Procedures in Klassen (für SAP HANA)
- ▶ AMDP BAdIs
- ▶ ABAP-CDS-Views
- ▶ Mesh-Typen

### **Leseplan für Interessenten an den neuen Technologien**

Um einen Überblick über SAP HANA und die neue Web-Frontend-Technologie zu erhalten, ist insbesondere Teil VI mit Kapitel 28, »SAP HANA«, und Kapitel 29, »SAPUI5, SAP Fiori und SAP Gateway«, von Bedeutung. In Kapitel 29 wird der Fokus auf die Entwicklung von OData-Services mit ABAP gelegt. Kapitel 10, »Zugriff auf SAP-HANA-Entwicklungsobjekte«, fasst die ABAP-Anweisungen zusammen, die Sie für den Zugriff auf die SAP-HANA-Datenbank benötigen. Vor allem aber mit den in Kapitel 2, »Die ABAP Development Tools«, beschriebenen ADT als Nachfolger der ABAP Workbench sollten Sie sich vertraut machen.

### **Leseplan für Experten**

Sie wissen bereits, welche sprachlichen Neuerungen es bis Release 7.51 gibt, und haben diese Anweisungen auch bereits verinnerlicht? Dann sind Sie hier richtig!

Informieren Sie sich z. B. über den neuen SAP List Viewer mit integriertem Datenzugriff (ALV with Integrated Data Access, IDA; siehe Abschnitt 19.3). Informationen zur Anwendung von Mesh-Typen finden Sie in Abschnitt 8.11, »Meshes«.

Kapitel 23 behandelt Themen der fortgeschrittenen Programmierung, insbesondere Parallelisierung, dynamische Programmierung und dynamisches SQL. Außerdem finden Sie hier Informationen zu persistenten Klassen und Shared Objects sowie der Arbeit mit XML und JSON. Kapitel 27 zeigt Ihnen, wie Sie Entwurfsmuster auf ABAP Objects anwenden können. Gerade die Implementierung des Model-View-Controllers und des Observer-Musters wird Sie hier interessieren.

### Ihr individueller Leseweg

Abweichend von den hier vorgestellten Leseplänen können Sie sich dem Buch natürlich auf vielen unterschiedlichen Wegen nähern. Da das Buch eine breite Zielgruppe ansprechen und gerade für den Alltag nützlich sein soll, habe ich versucht, mehrere Zugänge zum Buch zu ermöglichen. Stöbern Sie doch einfach einmal durch das Inhaltsverzeichnis, oder suchen Sie nach den Themen, die Sie interessieren im Index. Im Anhang finden Sie außerdem ein Glossar mit den wichtigsten Fachbegriffen sowie nützliche Übersichten mit ABAP-Anweisungen, wichtigen Klassen und Funktionsbausteinen, Transaktionscodes, technischen Tabellen u. v. m.

Bleibt mir nur noch, Ihnen viel Erfolg und Spaß bei der Arbeit mit dem Buch zu wünschen. Ich hoffe, es hilft Ihnen dabei, die kleinen und großen Herausforderungen des Alltags zu meistern. Die Zusammenstellung des Inhalts basiert vor allem auf meiner Tätigkeit als ABAP-Entwickler und meinen Erfahrungen als Trainer. Mein Dank gilt daher insbesondere denjenigen Teilnehmern meiner Kurse, die immer eine Frage mehr gestellt und mir auf diese Weise neue Sichtweisen eröffnet haben. So gesehen ist das vorliegende Buch ein Gemeinschaftswerk von und für (angehende) ABAPer. Ich freue mich daher auch über jeden Kommentar, der dazu beiträgt, das Buch weiter zu verbessern und abzurunden. Schreiben Sie mir gerne unter der E-Mail-Adresse *abap.felix.roth@gmail.com*.

In diesem Sinne: COMMIT WORK!

**Felix Roth**

# Auf einen Blick

## TEIL I Die Werkzeugkiste des ABAP-Entwicklers

1	Die ABAP Workbench .....	37
2	Die ABAP Development Tools .....	61
3	Das ABAP Dictionary .....	89
4	Transaktionen .....	129
5	Der Function Builder .....	139
6	Der Class Builder .....	147

## TEIL II Der Kern der Sprache ABAP

7	Die ABAP-Grundbefehle .....	165
8	Mit internen Tabellen arbeiten .....	231
9	Zugriff auf Datenbanken .....	285
10	Zugriff auf SAP-HANA-Entwicklungsobjekte .....	339
11	Die ABAP-Objects-Syntax .....	353
12	Reports und Selektionsbildschirme .....	393
13	Strukturierungselemente in ABAP .....	425
14	Die Dynpro-Programmierung .....	447

## TEIL III Techniken zur Qualitätssicherung

15	Tests und Qualitätskontrolle .....	473
16	Werkzeuge und Tipps zur Performanceanalyse .....	519
17	Das Transportwesen .....	547
18	Die Jobverwaltung .....	563

## TEIL IV Fortgeschrittene Programmiertechniken

19	Tabellenanzeige mit dem SAP List Viewer (ALV) .....	577
20	SAP-Schnittstellen .....	633
21	SAP-Erweiterungen .....	681
22	SAP-Formularentwicklung .....	731
23	Fortgeschrittene Programmiertechniken .....	763

## TEIL V Objektorientierte Programmierung

24	Grundlagen der Objektorientierung .....	809
25	Unified Modeling Language (UML) .....	843
26	Anwendungsentwicklung – wo fange ich an? .....	861
27	Entwurfsmuster .....	877

## TEIL VI Ein Blick über den Tellerrand:

### Was Sie als ABAP-Entwickler sonst noch kennen sollten

28	SAP HANA .....	907
29	SAPUI5, SAP Fiori und SAP Gateway .....	927
30	Andere SAP-Webtechnologien .....	963

# Inhalt

Einleitung .....	27
------------------	----

## TEIL I Die Werkzeugkiste des ABAP-Entwicklers

<b>1 Die ABAP Workbench .....</b>	<b>37</b>
-----------------------------------	-----------

1.1 Die Werkzeuge der ABAP Workbench .....	38
1.2 Der Object Navigator .....	39
1.3 Der Repository Browser .....	42
1.3.1 Übergeordnete Objektliste .....	43
1.3.2 Favoriten anlegen .....	44
1.4 Der ABAP Editor .....	44
1.4.1 Den neuen ABAP Editor aktivieren .....	45
1.4.2 Einen Report anlegen .....	46
1.4.3 Die Funktionsleiste .....	49
1.4.4 Die (Auto-)Vervollständigung .....	51
1.4.5 Der Pretty Printer .....	54
1.4.6 Die Musterfunktion .....	54
1.4.7 Der Package Builder .....	57
1.4.8 ABAP-Beispiele .....	59

<b>2 Die ABAP Development Tools .....</b>	<b>61</b>
---	-----------

2.1 SAPs Eclipse-Strategie .....	61
2.2 Installation und Konfiguration .....	64
2.2.1 Installation von Eclipse .....	64
2.2.2 Konfiguration des Backend-Systems .....	66
2.3 Ein System anbinden .....	67
2.4 Einen Report anlegen .....	68
2.5 Die Menüleiste und wichtige Tastaturkürzel .....	70
2.6 Der Pretty Printer und weitere Quellcodefunktionen .....	73
2.7 Die Musterfunktion .....	74
2.7.1 Muster für Funktionsbausteine .....	74
2.7.2 Muster für den Aufruf von Methoden .....	74
2.8 Der Debugger .....	75
2.8.1 Die Menüleiste des Debuggers .....	77
2.8.2 Die Variablen- und Outline-Anzeige .....	78
2.8.3 Breakpoints .....	78

2.8.4	Watchpoints .....	79
2.8.5	Interne Tabellen .....	80
2.9	Dokumentation mit ABAP Doc .....	82
2.9.1	Quick Fix .....	84
2.9.2	Dokumentation exportieren .....	84
2.9.3	Verlinkung zu Kurztexten des SAP GUI .....	85
2.10	Refactoring-Funktionen .....	85
2.10.1	Umbenennung .....	86
2.10.2	Eine Methode extrahieren .....	86
2.10.3	Quick Fix .....	87

### 3 Das ABAP Dictionary ..... 89

3.1	Domänen .....	91
3.1.1	Das zweistufige Domänenprinzip .....	91
3.1.2	Domänen anlegen .....	92
3.1.3	Konvertierungsroutinen .....	93
3.1.4	Wertebereich einer Domäne .....	95
3.2	Datenelemente .....	96
3.2.1	Feldbezeichner .....	97
3.2.2	Übersetzung .....	98
3.2.3	Zusatzeigenschaften .....	99
3.3	Strukturen .....	100
3.3.1	Erweiterungskategorie .....	100
3.3.2	Referenztabellen und das Referenzfeld .....	101
3.4	Tabellentypen .....	102
3.5	Datenbanktabellen .....	104
3.5.1	Datenbanktabellen anzeigen .....	104
3.5.2	Datenbanktabellen anlegen .....	105
3.5.3	Auslieferungsklasse .....	106
3.5.4	Tabellsicht-Pflege .....	106
3.5.5	Datenart .....	107
3.5.6	Größenkategorie .....	108
3.5.7	Pufferung .....	108
3.5.8	Felder ausprägen .....	109
3.5.9	Speicherart von Datenbanken mit SAP HANA .....	110
3.6	Indizes .....	110
3.7	Typgruppen .....	112

3.8	Views .....	113
3.8.1	Datenbank-View .....	113
3.8.2	Projektions-View .....	115
3.8.3	Pflege-View .....	116
3.8.4	Help-View .....	117
3.9	Pflegedialoge .....	117
3.9.1	Pflegedialog anlegen .....	118
3.9.2	Pflegedialog verbreitern .....	119
3.10	Suchhilfen .....	121
3.10.1	Elementare Suchhilfe .....	122
3.10.2	Sammelsuchhilfe .....	123
3.11	Datenbank-Utility-Tool .....	124
3.12	Das Sperrkonzept .....	125

### 4 Transaktionen ..... 129

4.1	Transaktionen anlegen .....	130
4.1.1	Dialogtransaktion .....	131
4.1.2	Reporttransaktion .....	131
4.1.3	ABAP-Objects-Transaktion .....	132
4.1.4	Variantentransaktion .....	133
4.1.5	Parametertransaktion .....	133
4.2	Transaktionsvarianten .....	134
4.2.1	Eine Screenvariante anlegen .....	135
4.2.2	Eine Transaktionsvariante anlegen .....	136
4.3	Transaktionen mit ABAP aufrufen .....	137

### 5 Der Function Builder ..... 139

5.1	Der Aufbau eines Funktionsbausteins .....	139
5.1.1	Eigenschaften .....	140
5.1.2	Die Schnittstelle .....	140
5.1.3	Ausnahmen .....	142
5.1.4	Der Quellcode .....	142
5.2	Einen Funktionsbaustein anlegen .....	142
5.3	Funktionsbausteine testen .....	143
5.4	Funktionsgruppen .....	144
5.4.1	Funktionsgruppe anlegen .....	144
5.4.2	Aufbau einer Funktionsgruppe .....	145
5.4.3	Lebensdauer einer Funktionsgruppe .....	146

## **6 Der Class Builder ..... 147**

6.1 Klassen anlegen .....	147
6.1.1 Vererbung .....	149
6.1.2 Interfaces .....	150
6.1.3 Freunde .....	151
6.1.4 Attribute .....	151
6.1.5 Methoden .....	152
6.1.6 Ereignisse .....	156
6.1.7 Typen .....	156
6.1.8 Aliases .....	157
6.1.9 Konstruktoren anlegen .....	157
6.1.10 Eine Klasse testen .....	158
6.1.11 Klassen direkt bearbeiten .....	159
6.2 Ausnahmeklassen anlegen .....	160
6.3 Interfaces anlegen .....	161

## **TEIL II Der Kern der Sprache ABAP**

### **7 Die ABAP-Grundbefehle ..... 165**

7.1 Syntaxregeln .....	166
7.2 Kommentare .....	167
7.3 Die SAP-Hilfe .....	168
7.4 Datendeklaration .....	169
7.4.1 Felder .....	170
7.4.2 Konstanten .....	173
7.4.3 Strukturen .....	174
7.4.4 Aufzählungstypen .....	176
7.4.5 Feldsymbole .....	179
7.4.6 Unterschied zwischen TYPE und LIKE .....	181
7.5 Inline-Deklarationen .....	182
7.6 Typdefinitionen .....	183
7.6.1 Felder .....	184
7.6.2 Strukturen .....	185
7.7 Initialisierung .....	186
7.7.1 Felder initialisieren .....	187
7.7.2 Speicherbereich freigeben .....	187
7.8 Steueranweisungen .....	187
7.8.1 Die IF-Abfrage .....	188
7.8.2 Logische Ausdrücke .....	189
7.8.3 Die CASE-Anweisung .....	190

### 7.8.4 Die Anweisung CASE TYPE OF .....

191  
7.8.5 Die DO-Schleife .....

192  
7.8.6 Die WHILE-Schleife .....

193  
7.8.7 Die CHECK-Anweisung .....

193  
7.8.8 Die EXIT-Anweisung .....

194  
7.8.9 Die CONTINUE-Anweisung .....

194  
7.9 Rechenoperationen .....

195  
7.10 Ausgabeanweisungen .....

197  
    7.10.1 Die Anweisung WRITE .....

198  
    7.10.2 Das Muster für die Listenausgabe .....

199  
7.11 Meldungen .....

199  
7.12 Mit Zeichenketten arbeiten .....

201  
    7.12.1 Vergleich von Zeichenketten .....

201  
    7.12.2 Verkettungsoperatoren .....

203  
    7.12.3 Teilfeldzugriff .....

204  
    7.12.4 Teilzeichenketten finden .....

206  
    7.12.5 Teilzeichenketten ersetzen .....

207  
    7.12.6 Eingebaute Funktionen für Zeichenketten .....

209  
    7.12.7 Zeichenketten-Templates .....

211  
7.13 Konstruktorausdrücke .....

214  
    7.13.1 VALUE: Erzeugung von Werten .....

216  
    7.13.2 REF: Referenzen besorgen .....

218  
    7.13.3 EXACT: Verlustfreie Zuweisung/Berechnung .....

218  
    7.13.4 CONV: Konvertierung von Werten .....

219  
    7.13.5 COND: Bedingte Ausdrücke .....

220  
    7.13.6 SWITCH: Bedingte Ausdrücke .....

221  
    7.13.7 NEW: Instanzierung .....

222  
    7.13.8 CORRESPONDING: Mapping von Strukturen  
        und internen Tabellen .....

223  
    7.13.9 Der Zusatz LET .....

226  
    7.13.10 Der Zusatz BASE .....

226  
7.14 Operandenpositionen .....

226  
    7.14.1 Funktionen und Ausdrücke für Lesepositionen .....

227  
    7.14.2 Ausdrücke für Schreibpositionen .....

228  
7.15 Änderungen und Neuerungen bis ABAP 7.51 .....

### **8 Mit internen Tabellen arbeiten ..... 231**

#### 8.1 Tabellenarten .....

233  
    8.1.1 Standardtabellen .....

233  
    8.1.2 Sortierte Tabellen .....

234  
    8.1.3 Hash-Tabellen .....

8.2	Interne Tabellen definieren .....	234
8.2.1	Schlüssel definieren .....	235
8.2.2	Obsolete: Deklaration einer internen Tabelle mit Kopfzeile .....	236
8.2.3	Ranges-Tabellen definieren .....	237
8.3	Interne Tabellen initialisieren .....	239
8.4	Zeilen hinzufügen .....	239
8.4.1	Daten mit SELECT hinzufügen .....	240
8.4.2	Zeilen mit APPEND anhängen .....	240
8.4.3	Zeilen mit INSERT hinzufügen .....	241
8.4.4	Werte mit VALUE hinzufügen .....	242
8.4.5	Der Zusatz FOR .....	244
8.4.6	Gruppierungen mit FOR .....	246
8.4.7	Der Zusatz LINES OF .....	250
8.4.8	Hinzufügen mit NEW .....	251
8.5	Inhalt auslesen .....	251
8.5.1	Tabellen mit READ TABLE auslesen .....	251
8.5.2	Tabellenausdrücke .....	253
8.5.3	Tabellen mit LOOP AT auslesen .....	257
8.5.4	Gruppieren mit dem Zusatz GROUP BY .....	259
8.6	Einträge löschen .....	264
8.7	Inhalt ändern .....	265
8.7.1	Tabelle mit READ TABLE ändern .....	266
8.7.2	Tabelle mit Tabellenausdrücken ändern .....	266
8.7.3	Tabelle mit MODIFY ändern .....	267
8.7.4	Tabelle mit CORRESPONDING anreichern .....	268
8.8	Interne Tabellen kopieren .....	271
8.8.1	Strukturgleiche interne Tabellen kopieren .....	271
8.8.2	Strukturfremde interne Tabellen kopieren .....	272
8.9	Interne Tabellen aufbereiten .....	273
8.9.1	Sortieren mit SORT .....	273
8.9.2	Zusammenfassung mit COLLECT .....	274
8.9.3	Reduzierungen mit REDUCE .....	274
8.9.4	Filterungen mit FILTER .....	276
8.10	Eingebaute Funktionen für interne Tabellen .....	277
8.11	Meshes .....	279
8.12	Änderungen und Neuerungen im Umfeld von internen Tabellen bis ABAP 7.51 .....	283

<b>9</b>	<b>Zugriff auf Datenbanken .....</b>	<b>285</b>
9.1	Die fünf goldenen Regeln .....	286
9.2	Die Open-SQL-Anweisung SELECT .....	288
9.2.1	Einträge lesen .....	288
9.2.2	Gelesene Spalten einschränken .....	291
9.2.3	Die WHERE-Klausel .....	293
9.2.4	Ranges-Tabellen und Selektionsoptionen .....	295
9.2.5	FOR ALL ENTRIES IN: Einschränkung durch interne Tabellen .....	296
9.2.6	Gruppierung und Sortierung der Ergebnisse .....	297
9.2.7	Syntax .....	297
9.2.8	Die FIELDS-Klausel .....	299
9.2.9	Host-Variablen und -ausdrücke .....	300
9.2.10	Inline-Deklaration .....	302
9.2.11	Begrenzung der Ergebnismenge mit OFFSET .....	303
9.2.12	Syntax .....	304
9.2.13	JOIN: Verknüpfung .....	304
9.2.14	WITH: Allgemeine Tabellenausdrücke .....	309
9.2.15	UNION: Vereinigung .....	310
9.3	Open-SQL-Ausdrücke .....	311
9.3.1	CASE-Anweisungen .....	311
9.3.2	Verknüpfungen von Zeichenketten mit && .....	312
9.3.3	Arithmetische Ausdrücke .....	313
9.3.4	Typumwandlungen mit CAST .....	314
9.3.5	Elementare Werte .....	315
9.4	Open-SQL-Funktionen .....	315
9.4.1	Aggregatfunktionen .....	316
9.4.2	Zeichenkettenfunktionen .....	317
9.4.3	Numerische Funktionen .....	319
9.4.4	Datumsfunktionen .....	320
9.4.5	Coalesce-Funktion: Nullwerte ersetzen .....	320
9.5	Ändernde Open-SQL-Anweisungen .....	321
9.5.1	DELETE: Löschen von Einträgen .....	322
9.5.2	INSERT: Einträge einfügen .....	324
9.5.3	UPDATE: Einträge ändern .....	326
9.5.4	MODIFY: Einfügen oder Ändern .....	328
9.6	Sekundäre Datenbankverbindungen .....	330
9.7	Natives SQL .....	330
9.7.1	EXEC-SQL .....	331
9.7.2	ABAP Database Connectivity .....	332

9.8	ABAP Core Data Services (CDS) .....	333
9.9	Änderungen und Neuerungen im Umfeld von Open SQL bis ABAP 7.51 .....	336
<b>10</b>	<b>Zugriff auf SAP-HANA-Entwicklungsobjekte .....</b>	<b>339</b>
10.1	Aufruf von SAP-HANA-Views .....	339
10.1.1	Aufruf mit nativem SQL .....	340
10.1.2	Aufruf über externe Views .....	340
10.1.3	Externe Views anlegen .....	341
10.2	Aufruf von Datenbankprozeduren .....	342
10.2.1	Aufruf mit nativem SQL .....	342
10.2.2	Aufruf mit einem Datenbankprozedur-Proxy .....	344
10.2.3	Datenbankprozedur-Proxy anlegen .....	345
10.3	ABAP Managed Database Procedures (AMDP) .....	346
10.3.1	ABAP Managed Database Procedures anlegen .....	347
10.3.2	Datenbankfunktionen anlegen .....	349
10.3.3	BAdIs für ABAP Managed Database Procedures .....	350
10.4	Änderungen und Neuerungen beim Zugriff auf SAP-HANA- Entwicklungsobjekte bis ABAP 7.51 .....	351
<b>11</b>	<b>Die ABAP-Objects-Syntax .....</b>	<b>353</b>
11.1	Grundaufbau einer Klasse .....	354
11.2	Sichtbarkeiten .....	357
11.3	Datentypen und Attribute .....	358
11.4	Methoden .....	360
11.4.1	Methoden implementieren .....	362
11.4.2	Methoden aufrufen .....	363
11.5	Konstruktoren .....	366
11.5.1	Instanzkonstruktor .....	367
11.5.2	Statischer Konstruktor .....	368
11.6	Ereignisse .....	369
11.6.1	Definition von Ereignissen .....	369
11.6.2	Ereignisse auslösen .....	370
11.6.3	Definition eines Ereignisbehälters .....	370
11.6.4	Ereignisbehandler registrieren .....	370
11.6.5	Beispiel für die Definition, das Auslösen und die Behandlung eines Ereignisses .....	371
11.7	Vererbung .....	373
11.8	Klassenarten .....	375
11.8.1	Abstrakte und finale Klassen .....	375

11.8.2	Statische Klassen .....	375
11.8.3	Ausnahmeklassen .....	376
11.9	Ausnahmen für Methoden .....	377
11.9.1	Klassenbasierte Ausnahmen .....	378
11.9.2	Lokale Ausnahmen .....	381
11.10	Freunde .....	383
11.11	Interfaces .....	383
11.11.1	Implementierung eines Interface .....	384
11.11.2	Verwendung von Interfaces .....	386
11.12	Das ABAP-Objects-Muster .....	387
11.13	Casting .....	388
11.13.1	Casting mit dem Zuweisungsoperator .....	388
11.13.2	Casting mit dem Casting-Operator .....	388
11.13.3	Casting mit der Anweisung CAST .....	389
11.14	Objekttyp überprüfen .....	389
11.14.1	Die Anweisung IS INSTANCE OF .....	389
11.14.2	Die Anweisung CASE TYPE OF .....	390
11.15	Änderungen und Neuerungen in ABAP Objects bis ABAP 7.51 .....	391

<b>12</b>	<b>Reports und Selektionsbildschirme .....</b>	<b>393</b>
12.1	Ereignisse eines Reports .....	394
12.2	Eingabeelemente .....	395
12.2.1	Parameter .....	396
12.2.2	Checkboxen .....	397
12.2.3	Radiobuttons .....	398
12.2.4	Dropdown-Liste .....	398
12.2.5	Selektionsoptionen .....	400
12.2.6	Buttons .....	401
12.2.7	Buttons auf der Funktionsleiste .....	403
12.2.8	Der Zusatz USER-COMMAND .....	404
12.3	Strukturierungselemente für den Selektionsbildschirm .....	405
12.3.1	Blöcke .....	405
12.3.2	Leerzeilen .....	406
12.3.3	Horizontale Linien .....	406
12.3.4	Textausgaben .....	407
12.3.5	Tabstrips .....	407
12.3.6	Modifikationsgruppen .....	408
12.4	Ereignisse eines Selektionsbildschirms .....	410

12.5	Textelemente .....	413
12.5.1	Zugriff auf Textelemente .....	414
12.5.2	Textsymbole .....	414
12.5.3	Selektionstexte .....	416
12.5.4	Listenüberschriften .....	417
12.6	Nachrichtenklassen .....	417
12.6.1	Nachrichtenklasse anlegen .....	418
12.6.2	Nachricht aufrufen .....	419
12.6.3	Parametrisierte Nachrichten .....	419
12.7	Einen Report mit ABAP aufrufen .....	420
12.8	SPA-/GPA-Parameter .....	422
12.8.1	SPA-/GPA-Parameter anlegen und setzen .....	423
12.8.2	SPA-/GPA-Parameter auslesen .....	424

## 13 Strukturierungselemente in ABAP ..... 425

13.1	Unterprogramme .....	426
13.1.1	Unterprogramm definieren .....	427
13.1.2	Sichtbarkeitsbereiche von Datendeklarationen .....	427
13.1.3	Aufruf eines Unterprogramms .....	428
13.1.4	Parameterübergabe .....	430
13.2	Makros .....	433
13.2.1	Makros definieren .....	433
13.2.2	Makros aufrufen .....	434
13.3	Includes .....	434
13.3.1	Include einbinden .....	435
13.3.2	Top-Include anlegen .....	435
13.3.3	Include anlegen .....	436
13.4	Funktionsbausteine .....	437
13.4.1	Arten von Funktionsbausteinen .....	438
13.4.2	Aufruf von Funktionsbausteinen .....	439
13.4.3	Funktionsbausteine finden .....	442
13.5	Datenkonsistenz .....	443

## 14 Die Dynpro-Programmierung ..... 447

14.1	Dynpros anlegen .....	448
14.1.1	Dynpro gestalten .....	449
14.1.2	Dynpro aufrufen .....	451
14.1.3	Zugriff auf Dynpro-Elemente .....	452
14.2	Ablauflogik eines Dynpros .....	452
14.2.1	Process Before Output (PBO) .....	453

14.2.2	PBO-/PAI-Module anlegen .....	454
14.2.3	GUI-Status .....	456
14.2.4	GUI-Titel .....	458
14.2.5	Process After Input (PAI) .....	460
14.3	SAP Control Framework .....	461
14.4	Pop-up-Fenster .....	464
14.4.1	Entscheidungen .....	465
14.4.2	Textanzeige .....	466
14.4.3	Werteabfrage .....	467

## TEIL III Techniken zur Qualitätssicherung

### 15 Tests und Qualitätskontrolle ..... 473

15.1	Der ABAP Debugger .....	473
15.1.1	Den neuen ABAP Debugger aktivieren .....	474
15.1.2	Den Debugger starten und beenden .....	474
15.1.3	Die Oberfläche .....	477
15.1.4	Die Werkzeuge .....	478
15.1.5	Steuerung des Debuggers .....	480
15.1.6	Schnellanzeige der Variablen .....	481
15.1.7	Vergleichstool .....	483
15.1.8	Aufrufstack .....	484
15.1.9	Pop-up-Fenster debuggen .....	485
15.1.10	Interne Tabellen .....	486
15.1.11	Debugger-Breakpoints .....	487
15.1.12	Watchpoints .....	491
15.1.13	Speicherverwaltung .....	492
15.1.14	Ausnahmen .....	494
15.2	Das Debugging-Skript .....	494
15.2.1	Einen Trigger für das Skript definieren .....	496
15.2.2	Ein Skript schreiben .....	497
15.2.3	Das Skript starten und beenden .....	499
15.3	Der Code Inspector .....	501
15.3.1	Ad-hoc-Prüfung über Transaktion SE80 .....	502
15.3.2	Prüfvariante .....	502
15.3.3	Objektmenge .....	504
15.3.4	Inspektion .....	504
15.3.5	Ergebnisliste .....	505
15.4	ABAP Unit .....	506
15.4.1	Grundsätzlicher Aufbau einer Testklasse .....	506

15.4.2	Systemeinstellungen .....	509
15.4.3	Assertions .....	510
15.4.4	Assistent für die Testklassengenerierung .....	511
15.4.5	Ausführen eines ABAP-Unit-Tests .....	513
15.4.6	Die Ergebnisanzeige .....	513
15.4.7	Der ABAP Unit Browser .....	514
15.5	Das ABAP Test Cockpit .....	515
15.5.1	Ausführung eines ATC-Tests .....	515
15.5.2	Die Transaktion ATC .....	516
15.5.3	Der ATC-Ergebnis-Browser .....	517

## 16 Werkzeuge und Tipps zur Performanceanalyse ..... 519

16.1	Richtlinien für die ABAP-Entwicklung .....	520
16.2	Transaktion SAT: Laufzeitanalyse .....	523
16.2.1	Laufzeitmessung durchführen .....	524
16.2.2	Laufzeitmessung auswerten .....	525
16.2.3	Anzeige der Messungen .....	525
16.3	Transaktion SE30: Die alte Laufzeitanalyse .....	527
16.4	SQL-Monitor .....	529
16.4.1	Transaktion SQLM: Den SQL-Monitor administrieren .....	530
16.4.2	Transaktion SQLMD: Analyse der Daten .....	532
16.5	SQL Performance Tuning Worklist .....	533
16.6	Transaktion ST05 .....	535
16.6.1	SQL-Trace .....	536
16.6.2	Analyse einer SQL-Anweisung .....	538
16.7	Laufzeitanalyse mithilfe der ABAP-Programmierung .....	538
16.7.1	Zeitmessung .....	539
16.7.2	Fortschrittsanzeige implementieren .....	539
16.8	Application Log .....	541
16.8.1	Transaktion SLG0: Ein Application Log anlegen .....	542
16.8.2	Log mit Nachrichten befüllen .....	542
16.8.3	Log als Pop-up-Fenster darstellen .....	544

## 17 Das Transportwesen ..... 547

17.1	Die SAP-Systemlandschaft .....	548
17.2	Transportaufträge .....	551
17.2.1	Transportauftrag anlegen .....	552
17.2.2	Transportauftrag freigeben und importieren .....	553
17.2.3	Aufgabe anlegen .....	557

17.2.4	Zuordnung des Transportauftrags ändern .....	557
17.2.5	Transportaufträge und Aufgaben löschen .....	559
17.2.6	Objekte in einen Transportauftrag aufnehmen .....	559
17.2.7	Transportaufträge verschmelzen .....	560
17.2.8	Transportauftrag oder Aufgabe finden .....	561
17.2.9	Freigabe eines Transportauftrags zurücknehmen .....	561

## 18 Die Jobverwaltung ..... 563

18.1	Transaktion SM36: Jobs definieren .....	563
18.1.1	Allgemeine Angaben .....	564
18.1.2	Startbedingung .....	565
18.1.3	Schritt (Step) definieren .....	569
18.2	Transaktion SM37: Jobs überwachen und freigeben .....	570
18.3	Ereignisse für Jobs .....	572
18.3.1	Ereignis definieren .....	572
18.3.2	Ereignis auslösen .....	572
18.4	Jobs mit ABAP definieren .....	573

## TEIL IV Fortgeschrittene Programmietechniken

### 19 Tabellenanzeige mit dem SAP List Viewer (ALV) ..... 577

19.1	Die alte ALV-Anzeige .....	579
19.1.1	Aufbau des Grundgerüsts .....	580
19.1.2	Eingabefähigkeit .....	582
19.1.3	Funktionen .....	586
19.1.4	Ereignisse .....	587
19.1.5	Spalten bearbeiten .....	588
19.1.6	Zellentypen .....	591
19.1.7	Farbige Hervorhebung .....	593
19.1.8	Icons .....	595
19.1.9	ALV-Tabellen sortieren und gruppieren .....	596
19.1.10	Aggregation .....	597
19.1.11	Layout speichern .....	598
19.2	Die neue ALV-Anzeige .....	599
19.2.1	Aufbau des Grundgerüsts .....	599
19.2.2	Funktionen .....	600
19.2.3	Ereignisse .....	602
19.2.4	Spalten bearbeiten .....	603
19.2.5	Zellentypen .....	607
19.2.6	Farbige Hervorhebung .....	608

19.2.7	Icons .....	609
19.2.8	ALV-Tabellen sortieren und gruppieren .....	610
19.2.9	Aggregation .....	611
19.2.10	Layout speichern .....	612
19.2.11	Filter .....	612
19.3	SAP List Viewer mit integriertem Datenzugriff (IDA) .....	613
19.3.1	Aufbau des Grundgerüsts .....	614
19.3.2	Funktionen .....	615
19.3.3	Ereignisse .....	616
19.3.4	Spalten bearbeiten .....	617
19.3.5	Zellentypen .....	622
19.3.6	Icons .....	623
19.3.7	ALV-Tabelle sortieren und gruppieren .....	623
19.3.8	Aggregation .....	623
19.3.9	Layout .....	624
19.3.10	Filter .....	625
19.3.11	Textsuche .....	627
19.4	Anzeigemodi .....	628
19.4.1	Ausgabe auf Selektionsbildschirm .....	628
19.4.2	Mehrere ALV-Tabellen in einem Split-Container ausgeben .....	630

<b>20</b>	<b>SAP-Schnittstellen .....</b>	<b>633</b>
20.1	RFC-Funktionsbausteine .....	634
20.2	Business-Objekte und BAPIs .....	635
20.2.1	Business Object Repository und BAPI Explorer .....	639
20.2.2	BAPIs .....	640
20.3	Flat Files .....	646
20.3.1	Dateien schreiben .....	647
20.3.2	Dateien einlesen .....	649
20.3.3	Weitere nützliche Funktionen .....	651
20.3.4	Beispiel .....	652
20.4	Webservices (SOAP) .....	654
20.4.1	WSDL-Dokument .....	656
20.4.2	Webservices anlegen und finden .....	657
20.4.3	Webservice konsumieren .....	664
20.5	Batch Input .....	670
20.5.1	Aufzeichnung der Transaktion .....	671
20.5.2	Direkte Ausführung .....	674
20.5.3	Mappe erstellen .....	677

20.6	Einführung in die Legacy System Migration Workbench (LSMW) .....	678
------	--	-----

## **21 SAP-Erweiterungen .....** **681**

21.1	User-Exits .....	681
21.2	Customer-Exits .....	684
21.3	Klassische Business Add-ins (BAdIs) .....	688
21.4	Enhancement Framework .....	692
21.4.1	Architektur .....	694
21.4.2	Explizite Erweiterungspunkte .....	697
21.4.3	Implizite Erweiterungspunkte .....	699
21.4.4	Klassen-Erweiterungen .....	701
21.4.5	Funktionsbaustein-Erweiterungen .....	703
21.4.6	Erweiterungssektionen .....	704
21.4.7	Struktur-Erweiterungen .....	706
21.4.8	Suchhilfen-Erweiterungen .....	708
21.4.9	Index-Erweiterungen .....	708
21.4.10	Einzelwert-Erweiterungen .....	709
21.4.11	Debugging von Erweiterungen .....	710
21.4.12	Transaktion SPAU_ENH: Abgleich von Erweiterungen im Rahmen von Updates .....	712
21.5	Neue Business Add-ins (BAdIs) .....	714
21.5.1	Aufruf .....	715
21.5.2	Definition aufrufen .....	715
21.5.3	Implementierung anlegen .....	716
21.5.4	Filterwerte .....	716
21.5.5	Menü-Exit .....	717
21.5.6	Dynpro-Exit .....	718
21.6	Switch Framework .....	727
21.6.1	Architektur .....	728
21.6.2	Schaltbare Objekte .....	729
21.7	Suche nach Erweiterungen .....	730

## **22 SAP-Formularentwicklung .....** **731**

22.1	Der Druckdialog .....	733
22.2	SAPscript .....	735
22.2.1	Formular erstellen .....	737
22.2.2	Druckprogramm erstellen .....	742
22.3	SAP Smart Forms .....	743
22.3.1	Formular erstellen .....	745

22.3.2 Druckprogramm erstellen .....	751
22.4 SAP Interactive Forms by Adobe .....	753
22.4.1 Formular erstellen .....	754
22.4.2 Druckprogramm erstellen .....	757
22.5 Generierung von PDFs .....	759
22.5.1 Generierung der internen Tabelle in SAPscript .....	759
22.5.2 Generierung der internen Tabelle SAP Smart Forms .....	760
22.5.3 Generierung der internen Tabelle in SAP Interactive Forms by Adobe .....	761
22.5.4 OTF in PDF konvertieren .....	762

## 23 Fortgeschrittene Programmietechniken ..... 763

23.1 Object Services .....	764
23.1.1 Persistente Klasse anlegen .....	764
23.1.2 Datenbanktabelle lesen .....	768
23.1.3 Datenbanktabelle aktualisieren .....	769
23.1.4 Query anlegen .....	770
23.1.5 Neuen Eintrag in der Datenbanktabelle anlegen .....	771
23.1.6 Löschen eines neuen Eintrags .....	772
23.2 Mit XML und JSON arbeiten .....	772
23.2.1 Exkurs: XML und JSON .....	773
23.2.2 Konvertierung von ABAP in JSON/XML .....	774
23.2.3 Konvertierung von JSON/XML in ABAP .....	775
23.2.4 Erzeugung eines XML-Dokuments .....	775
23.2.5 Objekte serialisieren .....	776
23.3 Daten im Memory ablegen .....	777
23.3.1 Die Anweisungen EXPORT und IMPORT .....	778
23.3.2 Shared Objects .....	780
23.4 Parallelisierung .....	785
23.5 Dynamische Erzeugung von Datenobjekten .....	786
23.5.1 Anonymes Datenobjekt mit CREATE DATA erzeugen .....	786
23.5.2 Anonymes Datenobjekt mit NEW erzeugen .....	788
23.5.3 Beispiel: Dynamische ALV-Tabelle erzeugen .....	789
23.6 Runtime Type Services (RTTS) .....	792
23.6.1 Strukturen: Klasse CL_ABAP_STRUCTDESCR .....	795
23.6.2 Interne Tabellen: Klasse CL_ABAP_TABLEDESCR .....	795
23.6.3 Referenzdatentypen: Klasse CL_ABAP_REFDESCR .....	797
23.6.4 Klassen: Klasse CL_ABAP_CLASSDESCR .....	797
23.6.5 Interfaces: Klasse CL_ABAP_INTFDESCR .....	798

23.6.6 Elementare Datentypen: Klasse CL_ABAP_ELEMDESCR .....	799
23.6.7 Beispiel: Eine interne Tabelle nach Microsoft Excel exportieren .....	799
23.7 Dynamisches SQL .....	802
23.7.1 Dynamische Selektionsliste .....	803
23.7.2 Dynamische FROM-Klausel .....	803
23.7.3 Dynamische WHERE-Klausel .....	805

## TEIL V Objektorientierte Programmierung

### 24 Grundlagen der Objektorientierung ..... 809

24.1 Einführung für ABAP-Entwickler .....	810
24.2 Klassen und Objekte .....	816
24.3 Instanziierung .....	820
24.4 Datenkapselung .....	822
24.5 Ereignisse .....	825
24.6 Vererbung .....	828
24.6.1 Redefinition .....	830
24.6.2 Klassenhierarchien .....	831
24.7 Klassenarten .....	833
24.7.1 Abstrakte Klassen .....	834
24.7.2 Finale Klassen .....	834
24.8 Interfaces .....	834
24.9 Polymorphie .....	839
24.10 Zusammenfassung .....	840

### 25 Unified Modeling Language (UML) ..... 843

25.1 Anwendungsfalldiagramm .....	844
25.1.1 Akteure .....	845
25.1.2 Anwendungsfälle .....	846
25.1.3 Beziehungen .....	847
25.1.4 Beispieldiagramm .....	849
25.1.5 Textuelle Beschreibung .....	849
25.2 Klassendiagramm .....	850
25.2.1 Attribute .....	853
25.2.2 Operationen .....	853
25.2.3 Beziehungen zwischen Klassen .....	854
25.2.4 Vom Anwendungsfall zum Klassendiagramm .....	857
25.2.5 Beispieldiagramm .....	858

**26 Anwendungsentwicklung – wo fange ich an? ..... 861**

26.1	Anforderungsermittlung .....	864
26.1.1	Ermittlung .....	864
26.1.2	Spezifikation .....	866
26.1.3	Verhaltensmodellierung .....	867
26.1.4	Validierung .....	869
26.2	Analyse .....	869
26.2.1	Die Klassenmodellierung .....	870
26.2.2	Verhaltensmodellierung .....	872
26.2.3	Verifikation .....	873
26.3	Entwurf .....	873

**27 Entwurfsmuster ..... 877**

27.1	Singleton .....	879
27.2	Fabrikmethode .....	881
27.3	Model View Controller .....	885
27.3.1	Model-Klasse .....	889
27.3.2	View-Klasse .....	889
27.3.3	Controller-Klasse .....	890
27.3.4	Hauptprogramm .....	891
27.3.5	Ersetzung des Views .....	892
27.4	Fassade .....	894
27.5	Observer .....	895
27.5.1	Haupttabelle .....	897
27.5.2	Abstrakte Detailtabelle .....	898
27.5.3	Konkrete Detailtabelle .....	899
27.5.4	Hauptprogramm .....	900
27.6	Objektorientierte Reports .....	901

**TEIL VI Ein Blick über den Tellerrand:  
Was Sie als ABAP-Entwickler sonst noch kennen sollten****28 SAP HANA ..... 907**

28.1	Überblick .....	908
28.2	Architektur .....	909
28.2.1	Die In-Memory-Technologie .....	910
28.2.2	Spaltenorientierte Speicherung .....	911
28.2.3	Wertekomprimierung .....	913
28.3	Migration auf SAP HANA .....	913

28.4	SAP-HANA-Objekte .....	915
28.4.1	Attribute View .....	916
28.4.2	Analytic View .....	917
28.4.3	Calculation View .....	917
28.4.4	Stored Procedures .....	919
28.5	Volltextsuchen .....	919
28.5.1	Volltextindex anlegen .....	920
28.5.2	Fuzzy-Suche .....	922
28.5.3	Linguistische Suche .....	923
28.5.4	Eingabefelder für die Vorschlagssuche .....	924

**29 SAPUI5, SAP Fiori und SAP Gateway ..... 927**

29.1	SAPUI5 .....	928
29.2	SAP Web IDE .....	930
29.3	SAP Fiori .....	931
29.4	OData .....	933
29.4.1	Metadatendokument .....	936
29.4.2	Aufbau eines OData-Service .....	936
29.4.3	Abfrageoptionen .....	938
29.4.4	Beispiel .....	938
29.5	SAP Gateway .....	939
29.5.1	Embedded Deployment .....	940
29.5.2	Zentrales Deployment ohne Entwicklung .....	940
29.5.3	Zentrales Deployment mit Entwicklung .....	941
29.6	Entwicklung eines OData-Service .....	942
29.6.1	Entwicklung .....	942
29.6.2	Veröffentlichung .....	949
29.6.3	Testen .....	951
29.6.4	Fehleranalyse .....	953
29.7	Überblick über die Implementierung der CRUDQ-Methoden .....	953
29.7.1	Auslesen der Schlüsselfelder .....	954
29.7.2	Auslesen des HTTP-Request-Bodys .....	954
29.7.3	Abfrageoptionen \$skip und \$top .....	956
29.7.4	Abfrageoption \$count .....	956
29.7.5	Abfrageoption \$inline-count .....	957
29.7.6	Abfrageoption \$filter .....	958
29.7.7	Abfrageoption \$select .....	959
29.7.8	Abfrageoption \$orderby .....	959
29.7.9	Meldungen ausgeben .....	960

**30 Andere SAP-Webtechnologien ..... 963**

30.1 Business Server Pages (BSP) .....	964
30.1.1 BSP-Anwendung mit HTML anlegen .....	966
30.1.2 BSP-Anwendung mit HTM/LB anlegen .....	969
30.2 Web Dynpro ABAP .....	972
30.2.1 Web-Dynpro-Component anlegen .....	973
30.2.2 Ausgabetabelle definieren .....	976
30.2.3 Methode zur Datenselektion implementieren .....	979
30.2.4 Context mit View verbinden .....	983
30.2.5 Ergebnistabelle anlegen .....	986
30.2.6 Logik für den Button implementieren .....	987
30.2.7 Web-Dynpro-Anwendung anlegen .....	989
30.3 Internet Communication Framework .....	991

**Anhang ..... 995**

A Das SAP-Flugdatenmodell .....	997
B Übersicht der ABAP-Anweisungen .....	999
C Eingebaute Datentypen .....	1010
D Transaktionscodes .....	1012
E Wichtige Systemfelder .....	1016
F Technische Tabellen .....	1017
G Nützliche Funktionsbausteine .....	1019
H Klassen .....	1023
I Namenskonventionen für die Programmierung .....	1025
J Systemglossar und Suche nach fremdsprachigen SAP-Begriffen .....	1029
K Der Autor .....	1031
L Glossar .....	1032
Index .....	1039

# Index

- ?= (Casting-Operator) 388  
{ } (eingebetteter Ausdruck) 211  
\* (Kommentarzeichen) 167  
/h (Systembefehl) 477  
&& (Verkettungsoperator) 204, 312  
= (Zuweisungsoperator) 388  
=> (Klassenkomponentenselektor) 363  
-> (Klassenkomponentenselektor) 363  
|| (Zeichenketten-Tempalte) 211  
\$metadata 936
- ABAP Dictionary 89  
ABAP Dictionary, Datenelement 171  
ABAP Doc 82  
ABAP Editor 44  
*Aktivierung* 45  
*Funktionsleiste* 49  
*neuer* 45
- ABAP für HANA 90  
ABAP List Viewer → ALV  
ABAP Memory 778  
ABAP Object 27  
ABAP Objects 809  
*Muster* 387  
*Neuerungen* 391  
*Syntax* 353
- ABAP Profiler 523  
ABAP Test Cockpit 515, 534
- A**
- ATC-Ergebnis-Browser 41, 517  
*Ausführung* 515
- ABAP Unit 506  
*Assertion* 510  
*Aufrufreihenfolge* 508  
*ausführen* 513  
Browser 514  
Modultest 513
- Systemeinstellungen 509  
Testklasse 506
- Testklassengenerierung 511  
Testmethode 507
- ABAP Unit Browser 41
- ABAP Unit Test  
Runner 513
- ABAP Workbench 37
- ABAP-Anweisung  
*DELETE* 264  
*obsolete* 1008  
*Übersicht* 999
- ABAP-Beispiel 59
- ABAP-Objects-Transaktion 132
- ABAP-Programm 166
- ABAP-Schlüsselwörter 166
- ABAP-Verbindung  
→ Systemverbindung
- ABAP-Wort 166  
Abbruchbedingung 244  
Abbruchmeldung 200  
Abfrageoption 956  
Abfrageoptionen 938  
Absatzformat 737, 741  
ABSTRACT (Zusatz) 360  
ABSTRACT METHODS  
(Zusatz)  
*INTERFACE* 384  
Abstraktion 840  
ACCEPTING DUPLICATE  
KEYS (Zusatz)  
*INSERT* 325  
ACTIVATION (Zusatz)  
*SET HANDLER* 371
- ADBC 285, 332  
Addition 195  
Adobe LiveCycle Designer 753  
ADT 61, 353  
*CDS View* 334  
*Installation* 66  
*Menüleiste* 70  
*Projekt* 67
- Agent 768  
Aggregatausdruck 316  
Aggregation, ALV 623  
agile Programmierung 870  
Akteur 844  
Alias 157, 305  
ALIAS (Zusatz), DATA 235  
ALL (Zusatz), UNION 310  
ALL INSTANCES (Zusatz),  
*SET HANDLER* 371
- ALL METHODS (Zusatz),  
*INTERFACE* 384
- ALL OCCURRENCES  
(Zusatz), FIND 206
- allgemeine Tabellenanzeige 107
- allgemeiner Tabellenausdruck 309
- ALV 577  
*Aggregation* 597  
*alt* 578  
*Eingabefähigkeit* 582

ALV (Forts.)  
*farbige Hervorhebung* 593, 608  
*Filter* 612  
*Gruppierung* 596  
*IDA* 578, 619  
*Klasse* 579, 599, 612  
*Layout* 598, 624  
*mit integriertem Datenzugriff* 613  
*neu* 578  
*Selektionsbildschirm* 628  
*Selektionsoptionen* 626  
*Sortierung* 596  
*Textsuche* 627  
*Tree* 579  
*Zellentypen* 607, 622  
AMC-Trace 535  
AMDP 346  
*anlegen* 347  
*BAdI* 350  
*Methode* 348  
Ampel-Button 457  
Analyse  
*dynamische* 519  
*objektorientierte* 862  
Analysephase 869  
Analysespezifikation 870  
Analytic View 339, 915, 917  
AND RETURN (Zusatz), SUBMIT 421  
AND WAIT (Zusatz), COMMIT WORK 445, 769  
Anforderungsermittlung 844, 864  
Anforderungsspezifikation 864, 869  
Ankreuzfeld → Checkbox  
Ankreuzstruktur 641  
Anweisung 166  
Anwendungsfall 846  
Anwendungsfalldiagramm 844, 866  
Anzeigemode 50  
APC-Trace 535  
App, transaktionale 932  
APPEND (ABAP-Anweisung) 240, 521  
APPEND LINES OF 521  
APPENDING (Zusatz), SELECT 290  
Append-Struktur 706  
Append-Suchhilfe 708  
Application Log 541  
*anlegen* 542  
*befüllen* 542  
*initialisieren* 542  
*speichern* 544  
Arbeitsbereich 174  
AREA HANDLE (Zusatz), CREATE OBJECT 783  
Array 291, 773  
AS (Zusatz), SELECT 291, 305  
AS CHECKBOX (Zusatz), PARAMETERS 397  
AS LISTBOX (Zusatz), PARAMETERS 398  
AS WINDOW (Zusatz), SELECTION-SCREEN  
BEGIN OF SCREEN 469  
ASCENDING (Zusatz)  
FOR GROUPS OF 247  
SELECT 297  
SORT 273  
Assert-Methode 510  
ASSIGN (ABAP-Anweisung) 180, 266, 786  
ASSIGN COMPONENT (ABAP-Anweisung) 787  
ASSIGNING (Zusatz), LOOP AT 258  
ASSOCIATION (Zusatz), TYPES 279  
Assoziation 855, 937  
AT SELECTION-SCREEN (Ereignis) 395, 410  
ATC → ABAP Test Cockpit  
Attribut 151, 358, 813, 816  
*Sichtbarkeit* 824  
*statisches* 819  
Attribute View 339, 916  
Attributvektor 913  
Aufgabe 547, 557  
Aufrufhierarchie 527  
Aufrufparameter 430  
Aufrufstack 484  
Aufzählungskonstante 176  
Aufzählungsstruktur 178  
Aufzählungstyp 176

Ausdruck  
*arithmetischer* 313  
*eingebetteter* 211  
*Konstruktorausdruck* 214, 229  
*logischer* 189  
*mathematischer* 196  
Open-SQL-Ausdruck 311  
*Rechenausdruck* 195  
Ausdrucksposition, allgemeine 227  
Ausführungsplan 537  
Ausführungsziel 564  
Ausgabeanweisung 197  
Ausgabeauftrag 734  
Ausgabegerät 742  
Auslieferungsklasse 106  
Ausnahme 155, 160, 376  
*abfangen* 441  
*COMMUNICATION\_FAILURE* 653  
*klassenbasierte* 376  
*lokale* 381, 441  
*SYSTEM\_FAILURE* 653  
Ausnahmekategorie 376  
Ausnahmekette 494  
Ausnahmeklasse 160, 376  
*anlegen* 160  
*CX\_DYNAMIC\_CHECK* 376  
*CX\_NO\_CHECK* 376  
*CX\_OS\_OBJECT\_EXISTING* 771  
*CX\_OS\_OBJECT\_NOT\_EXISTING* 769  
*CX\_ROOT* 376  
*CX\_STATIC\_CHECK* 376  
*CX\_SY\_ITAB\_LINE\_NOT\_FOUND* 254  
*Funktionsbaustein* 441  
*lokale* 376  
Ausnahmeobjekt 160, 376, 494  
Äußerer Join 307

**B**

BACK (Funktionscode) 460  
Backend-System 66  
BAdI 350, 688  
AMDP 350

BAdI (Forts.)  
*Filterwerte* 717  
*kernelbasiertes* 714  
*klassischer* 688  
*neues* 714  
BAdI-BUILDER 691, 722  
BAPI 522, 636, 640  
*erweitern* 641  
*Erweiterungsstruktur* 642  
Explorer 639  
*Übergabeparameter* 641  
BASE (Zusatz), CORRESPONDING 223, 226  
BASE TYPE (Zusatz), TYPES 177  
Batch Input 670  
*Aufzeichnung* 671  
*direkte Ausführung* 674  
*Mappe* 673, 677  
*Modus* 675  
Bearbeitungsmodus 50, 583  
BEGIN OF (ABAP-Anweisung) 279  
BEGIN OF ENUM (Zusatz), TYPES 177  
BEGIN OF SCREEN ... AS SUBSCREEN (Zusatz), SELECTION-SCREEN 407  
Benutzer, technischer 667  
Benutzereinstellungen 422, 1029  
Berechtigungsobjekt 130  
Betriebsart 567  
BETWEEN (Zeichenketten-Operator) 202  
Bewegungsdaten 107  
Beziehung 847  
Black-Box-Test 506  
Block 405  
BOR 636  
Breakpoint 51, 75, 475  
*anlegen* 78, 488  
*bearbeiten* 488  
*Debugger-Breakpoint* 487  
*externer* 79, 476  
*Session-Breakpoints* 475  
BSP 964  
BSP-Extension 965, 969  
Business Add-In → BAdI

**C**

Calculation View 339, 915, 917  
CALL (Open-SQL-Anweisung) 342  
CALL BADI (ABAP-Anweisung) 715  
Call by Reference 430  
Call by Value 430  
CALL CUSTOMER-FUNCTION (ABAP-Anweisung) 684  
CALL DATABASE PROCEDURE (ABAP-Anweisung) 344  
CALL FUNCTION (ABAP-Anweisung) 439, 653  
CALL METHOD (ABAP-Anweisung) 363, 366  
CALL SCREEN (ABAP-Anweisung) 451  
CALL SELECTION-SCREEN (ABAP-Anweisung) 469  
CALL TRANSACTION (ABAP-Anweisung) 137, 674  
CLOSE DATASET (ABAP-Anweisung) 647  
coalesce() (eingebaute Funktion) 320  
COBOL 27  
Code Inspector 501, 534, 914  
*Ad-hoc-Prüfung* 502  
*Inspektion* 501

Code Inspector (Forts.)  
*Objektmenge* 501  
*Prüfvariante* 501  
*Prüfvariante DEFAULT* 502  
*Prüfvariante TRANS-PORT* 502  
 Code-to-Data 285, 614, 908  
 COLLECT (ABAP-Anweisung) 274  
 COMMENT (Zusatz), SELECTION-SCREEN 407  
 Commit 321  
 COMMIT WORK (ABAP-Anweisung) 321, 443, 445, 769  
 Common Table Expression 309  
 Component 973  
 Component-Controller 975  
 COMPONENTS (Zusatz), DATA 235  
 COMPUTE (ABAP-Anweisung) 218  
 concat\_lines\_of() (eingebaute Funktion) 278  
 CONCATENATE (ABAP-Anweisung) 203, 312  
 COND (ABAP-Anweisung) 220  
 Condition Factory 626  
 CONNECTION (Open-SQL-Anweisung) 330  
 CONSTANTS (ABAP-Anweisung) 173  
 Consumer-Proxy 662  
 Container 461, 580  
*Docking-Container* 628, 631  
*Root-UI-Element-Container* 984  
 Context 973, 975  
 CONTINUE (ABAP-Anweisung) 194, 259  
 Control 461  
 Controller 885, 975  
 CONV (ABAP-Anweisung) 219  
 Core Data Services → CDS

---

**D**

Dankbankfunktion, Aufruf 350  
 DATA (ABAP-Anweisung) 175, 182  
 Data Definition Language 334  
 Data Provider Class 946  
 DATA VALUES (Zusatz), INTERFACE 384  
 Data-to-Code 287, 908  
 Datenart 107  
 Datenbank 89  
 Datenbank-Commit 321  
 Datenbankfunktion 342, 349  
 Datenbank-LUW 444  
 Datenbankprozedur 342, 908, 915, 919  
*Aufruf* 342  
*Proxy* 344  
 Datenbank-Rollback 322, 445  
 Datenbankschema 340  
 Datenbankschnittstelle 286  
 Datenbanktabelle 89, 104  
*ändern* 321  
*anlegen* 105  
*anzeigen* 104  
*Berechtigungen* 106

CORRESPONDING (ABAP-Anweisung) 176, 223, 268, 272, 522  
 CREATE DATA (ABAP-Anweisung) 222, 786  
 CREATE OBJECT (ABAP-Anweisung) 222, 355  
 CREATE PUBLIC|PROTECTED|PRIVATE (Zusatz) 357  
 Cross Join 305, 308  
 CRUD-Methode 934  
 CRUDQ 934  
 CSS 928  
 CSV 799  
 CTE 309  
 Custom Control 451, 461  
 Customer-Exit 684  
 Customizing-Mandant 548

**Datenbanktabelle (Forts.)**

*Datenart* 107  
*Größenkategorie* 108  
*Pflegedialog* 117  
*Pufferung* 108  
*Speicherart* 110  
*technische Einstellungen* 107  
 Datenbank-Utility-Tool 124  
 Datenbankverbindung, sekundäre 330  
 Datenbank-View 113, 335  
 Datenbankzugriff 285  
 Datenbankzugriff, Object Service 764  
 Datenelement 91, 96, 141, 171  
*anlegen* 96  
*Zusatzeigenschaften* 99  
 Datenexport 633  
 Datenhaltungsschicht 885  
 Datenimport 633  
 Datenkapselung 822, 840  
 Datenkonsistenz 321, 443  
 Datenobjekt 169, 786, 816  
 Datenreferenzvariable 218  
 Datentyp 170, 314, 358, 818  
*definieren* 183  
*eigener* 172, 183  
*eingebauter* 170  
*gebundener* 181  
*globaler* 181  
*lokaler* 181  
 Datums-/Zeittyp 314  
 Datumsfunktion 320  
 DB-spezifische Eigenschaften 110  
 DDL 334  
 Debugger 71  
 Debugger, ADT 75  
 Debugger-Breakpoint 487  
 Debugging 473  
*Hintergrundjob* 477  
*Pop-up-Fenster* 485  
 Debugging-Skript 494  
*erstellen* 497  
*starten* 499  
*Trigger* 496  
 DEEP (Zusatz), CORRESPONDING 223

DEFAULT (Zusatz)  
*PARAMETERS* 396  
*SELECT-OPTIONS* 401  
*Tabellenausdruck* 254  
 DEFAULT KEY (Zusatz), DATA 236  
 Default-Prüfvariante 502  
 DEFINE ... END-OF-DEFINITION (ABAP-Anweisung) 433  
 DEFINITION (Zusatz), CLASS 354  
 Definitionsteil 354  
 Deklaration 170  
 Deklaration, Inline 182  
 DELETE (Open-SQL-Anweisung) 322  
 demilitarisierte Zone 941  
 DESCENDING (Zusatz)  
*FOR GROUPS OF* 247  
*SELECT* 297  
*SORT* 273  
 Deserialisierung 773, 776  
 Design Patterns → Design Patterns  
 Desktop-Verknüpfung 485  
 DESTINATION (Zusatz), CALL FUNCTION 653  
 Destruktor 158, 822  
 Dezimalzahl 314  
 Dialogtransaktion 131  
 DISTINCT (Zusatz)  
*Aggregatausdruck* 316  
*SELECT* 290  
*UNION* 310  
 Division 195  
 DMZ 941  
 DO (ABAP-Anweisung) 192  
 Docking-Container 461, 631  
 Dokumentation 82  
 Domäne  
*anlegen* 92  
*Einzelwerte erweitern* 709  
*semantische* 91  
*technische* 91  
*Wertebereich* 95  
 Doppelpunkt 167  
 Down Cast 388  
 Drei-Systeme-Landschaft 549

**E**

Eclipse 61, 64  
 Eingabefähigkeit 582  
 Eingabefeld 450  
 Eingabefeld,  
*Web Dynpro* 984  
 Eingabehilfe 95, 121  
 Eingabeprüfung 95  
 eingebaute Funktion,  
*interne Tabellen* 277  
 Einheit 314  
 Einzelwert erweitern 709  
 EL 334  
 elementarer Wert 315  
 ELSE (ABAP-Anweisung) 188  
 ELSEIF (ABAP-Anweisung) 188  
 Doppelstrich 167  
 Down Cast 388  
 END OF (ABAP-Anweisung) 279

**END OF TASK (Zusatz), CALL FUNCTION** 785  
 ENDFORM (ABAP-Anweisung) 427  
 ENDING AT (Zusatz), CALL SCREEN 451  
 ENDLOOP (ABAP-Anweisung) 257  
 ENDMETHOD (ABAP-Anweisung) 362  
 ENDSELECT (Open-SQL-Anweisung) 291  
 Enhancement Framework 692  
 Enhancement Package 693  
 Enhancement  
→ Erweiterung  
 ENHANCEMENT-POINT (ABAP-Anweisung) 697  
 ENHANCEMENT-SECTION (ABAP-Anweisung) 704  
 Enqueue-Trace 535  
 Enterprise Service  
Browser 41  
Entität 933  
Entitätsklasse 871  
Entitätset 933  
Entitätstyp 936  
Entwicklungs paket 58  
Entwicklungssystem 549  
Entwicklungstestmandant 550  
Entwurf 874  
Entwurfsmuster 877  
Entwurfsspezifikation 875  
Ereignis 156, 369, 587, 636, 826, 896  
*ALV* 587  
*auslösen* 370, 572  
*definieren* 369, 572  
*Jobs* 572  
*Report* 394  
*Selektionsbildschirm* 410  
*statisches* 369  
 Ereignisbehandler 369, 826, 896  
*definieren* 370  
*registrieren* 370  
 Ereignisblock 410  
 Erweiterung 681, 695  
*Abgleich* 712  
*Debugging* 710

Erweiterung (Forts.)  
*Einzelwerte* 709  
*Funktionsbaustein* 703  
*Include* 711  
*Index* 708  
*Klassen* 701  
*Konflikt* 713  
*Methode* 702  
*Struktur* 706  
*suchen* 686  
*Suchhilfe* 708  
Erweiterungsimplementierung 696  
Erweiterungsimplementierungselement 696  
Erweiterungs-Infosystem 41  
Erweiterungskategorie 100, 706  
Erweiterungsmodus 50  
Erweiterungsoption 695  
Erweiterungspunkt  
  expliziter 697  
  impliziter 682, 699  
Erweiterungssektion 704  
Erweiterungsspot 696  
Erweiterungsstruktur 642  
Erzeugungsmuster 878  
Event → Ereignis  
EXACT (ABAP-Anweisung) 218  
EXCEPT (Zusatz)  
  CORRESPONDING 224  
  FILTER 276  
EXCEPTIONS (Zusatz) 360  
  CALL FUNCTION 441  
  METHODS 381  
EXCLUDING (Zusatz), SET  
  PF-STATUS 458  
EXEC SQL (ABAP-Anweisung) 331  
EXIT (ABAP-Anweisung) 194, 259  
EXIT (Funktionscode) 460  
Exit-Meldung 200  
Explain 537  
EXPORT(ABAP-Anweisung) 777  
EXPORTING (Zusatz)  
  CREATE OBJECT 355  
  SET PARAMETER 423  
Expression Language 334  
Extension-Index 708

externer Breakpoint 476  
externer View 340

---

**F**

F4-Hilfe erweitern 710  
Fabrikmethode 881  
Facade 894  
Fallunterscheidung 189, 311  
Farocode 593  
Fassade 894  
Favorit 44  
Fehler 376  
Fehlertext 376  
Feinentwurf 874  
Feld  
  Deklaration 170  
  sperren 584  
Feldbezeichner 97  
Felddefinition 643  
Feldgruppe 644  
Feldkatalog  
  ALV 581  
  bearbeiten 588  
  neue ALV-Anzeige 604  
Feldleiste → Struktur  
Feldselektor 175  
Feldsymbol 179, 258, 786  
  bearbeiten 482  
  Definition 179  
  Zuweisung 180  
Fenster  
  Formular 735  
  grafisches 737  
  Hauptfenster 736, 743  
  konstantes 737  
  Kopienfenster 744  
  Nebenfenster 743  
  rücksetzendes 737  
Festwert 95  
Festwert-Append 709  
Festwertintervall 95  
Festwertintervall,  
  Erweiterung 709  
FIELD (Zusatz)  
  GET PARAMETER 424  
  SET PARAMETER 423  
FIELDS (Zusatz),  
  SELECT 299

FIELD-SYMBOLS (ABAP-Anweisung) 179, 182  
File Transfer Protocol 646  
FILTER (ABAP-Anweisung) 276, 522  
Filter, ALV 612  
FILTERS (Zusatz),  
  GET BADI 715  
FINAL (Zusatz) 360  
FINAL METHODS (Zusatz),  
  INTERFACE 384  
FIND (ABAP-Anweisung) 206  
find() (Suchfunktion) 207  
Fiori → SAP Fiori  
FIRST OCCURRENCE (Zusatz), FIND 206  
Flat File 646  
Fließkommazahl 314  
FOR ... IN (Zusatz) 247  
FOR (Zusatz), SET HANDLER 370  
FOR ALL ENTRIES IN (Zusatz), SELECT 296, 521  
FOR EVENT (Zusatz), METHODS 370  
FOR GROUPS ... OF (Zusatz) 246  
FOR TESTING (Zusatz), CLASS 507  
FORM (ABAP-Anweisung) 427  
Form Builder 743  
Form Painter 735, 743  
Formalparameter 430  
Form-Include 146  
Formroutine  
  → Unterprogramm  
Formular 731  
  Fenster 735  
  Funktionsbaustein 752  
  Offline 753  
  Seite 735  
  Stil 745  
Formulkopf 737  
Fortschrittsanzeige 538  
FREE (ABAP-Anweisung) 187, 239  
Fremdschlüssel 95  
Fremdschlüsselbeziehung 95, 116, 279

Freund 151, 383, 825  
FRIENDS (ABAP-Anweisung) 383  
FROM (Zusatz)  
  DELETE 323  
  INSERT 325  
  LOOP AT 257  
  MODIFY 329  
  SELECT 289, 299  
  UPDATE 326  
FROM MEMORY ID (Zusatz), IMPORT 779  
FTP 646  
FuBa → Funktionsbaustein  
Function Builder 139, 438, 442  
FUNCTION KEY (Zusatz), SELECTION-SCREEN 403  
Function Module  
  → Funktionsbaustein  
Funktion  
  arithmetische 319  
  eingebaute 210, 277  
  mathematische 196  
  Zeichenketten 210, 317  
Funktionsbaustein 437, 810  
  ABAP4\_CALL\_TRANSACTION 137  
Ablaufart 140  
ALV 581  
anlegen 142  
APPL\_LOG\_INIT 542, 1019  
Arten 438  
Aufbau 139  
Aufruf 439  
Ausnahmen 142  
BAL\_DB\_SAVE 544, 1019  
BAL\_DSP\_LOG\_DISPLAY 545, 1019  
BAL\_DSP\_PROFILE\_POPUP\_GET 545  
BAL\_LOG\_CREATE 543, 1019  
BAL\_LOG\_MSG\_ADD 543  
BAPI\_MATERIAL\_DISPLAY 639  
Funktionsbaustein (Forts.)  
  BAPI\_MATERIAL\_SAVEDATA 641  
  BAPI\_TRANSACTION\_COMMIT 445, 1020  
  BAPI\_TRANSACTION\_ROLLBACK 445, 1022  
  BAPI\_USER\_GET\_DETAIL 658  
  BDC\_CLOSE\_GROUP 677, 1019  
  BDC\_INSERT 677  
  BDC\_OPEN\_GROUP 677  
  BP\_EVENT\_RAISE 572, 1019  
  CLOSE\_FORM 742  
  CONVERT\_OTF 762, 1020  
  DD\_POPUP\_WITH\_INFOTEXT 466, 1021  
  DYNP\_VALUES\_READ 399, 1020  
  Erweiterung 703  
  finden 442  
  FP\_FUNCTION\_MODULE\_NAME 757, 1020  
  FP\_JOB\_CLOSE 757  
  FP\_JOB\_OPEN 757, 761  
  ICON\_CREATE 402  
  JOB\_CLOSE 573  
  JOB\_OPEN 573  
  JOB\_SUBMIT 573, 1019  
  lokale Ausnahme 441  
  LVC\_FIELDCATALOG\_MERGE 581  
  OPEN\_FORM 742  
  POPUP\_GET\_VALUES 467, 1021  
  POPUP\_TO\_CONFIRM 465, 1021  
  Remote-fähiger 438, 634  
  SAPGUI\_PROGRESS\_INDICATOR 540  
  Schnittstelle 140  
  SCMS\_XSTRING\_TO\_BINARY 761  
  SSF\_FUNCTION\_MODULE\_NAME 752, 1020  
  SWO\_CREATE 638  
  SWO\_INVOKE 638

Funktionsbaustein (Forts.)  
  testen 143  
  Übersicht 1019  
  VRM\_SET\_VALUES 399, 1020  
Funktionscode 404, 456, 460  
Funktionsgruppe 120, 144, 818  
anlegen 144, 720  
aufrufen 442  
Lebensdauer 146  
Rahmenprogramm 145  
Funktionsleiste 456  
Funktionstaste 457  
Funktionstastenbelegung 456  
FUZZY (Zusatz), SELECT 922  
Fuzzy-Suche 627, 919, 922

---

**G**

Ganzzahl 314  
Ganzzahldivision 195  
Garbage Collector 493, 820  
Gebiet  
  anlegen 782  
  Shared Memory 780  
Gebietsklasse 780  
Geheimnisprinzip 874  
Generalisierung 829, 856  
GET BADI (ABAP-Anweisung) 715  
GET PARAMETER (ABAP-Anweisung) 424  
GET RUN TIME FIELD 539  
Get-Methode 824  
Glossar 1029  
goldene Regeln 286  
Groß- und Kleinschreibung 166  
Größenkategorie 108  
GROUP BY (Zusatz)  
  LOOP AT 259, 522  
  SELECT 298  
GROUP INDEX (Zusatz), LOOP AT 263

GROUP SIZE (Zusatz),  
LOOP AT 263  
Gruppenschlüssel-  
bindung 262  
Gruppierung 246  
GUI-Control 461  
GUI-Status 456  
    ALV 601  
    anlegen 456  
    Drucktastenleiste 457  
    Funktionstasten 457  
    kopieren 602  
    Menüleiste 457  
    SALV\_TABLE\_  
        STANDARD 601  
    setzen 458  
GUI-Titel 458  
    anlegen 459  
    Platzhalter 459  
    setzen 459

**H**

HANA → SAP HANA  
HANDLE (Zusatz),  
CREATE DATA 787  
HASHED TABLE (Zusatz),  
DATA 235  
Hash-Tabelle 234, 240  
Hauptpaket 58  
Hauptspeicher 910  
HAVING (Zusatz),  
SELECT 298  
HCP → SAP HANA Cloud  
Platform  
HELP-REQUEST (Zusatz),  
AT SELECTION-SCREEN  
411  
Help-View 117  
Hilfsvariable 226  
Hintergrundjob debuggen  
477  
Hintergrundverarbeitung  
563  
Host-Ausdruck 301  
Host-Variable 300  
Hotspot 591, 607, 622  
HTML5 928  
HTMLB 965, 969  
HTTP- und RFC-Trace 535  
HTTPS 659

**I**  
ICF 655, 991  
ICF-Service 991  
Icon 402  
    ALV 595, 609  
    Infotext 596  
ID (Zusatz),  
    SET PARAMETER 423  
IDA 578, 613  
IDE 37, 62  
Identitätstransformation  
772  
IF (ABAP-Anweisung) 188,  
220  
IF FOUND (Zusatz), PER-  
FORM 429  
IGNORING CASE (Zusatz),  
FIND 206  
IMPLEMENTATION  
(Zusatz), CLASS 355  
Implementierungsteil 354  
IMPORT (ABAP-Anwei-  
sung) 777  
IMPORTING (Zusatz) 360  
Importqueue 554  
IN (Zusatz), FILTER 276  
IN PROGRAM (Zusatz),  
PERFORM 429  
IN UPDATE TASK (Zusatz),  
CALL FUNCTION 444  
Include 434  
    anlegen 436  
    einbinden 435  
    Top-Include 434  
INCLUDE (ABAP-Anwei-  
sung) 435  
INCLUDE STRUCTURE  
(Zusatz), TYPES 185, 584  
Include-Beziehung 847  
Index 110  
    anlegen 111  
    Extension-Index 708  
    verwendeter 537  
    Volltext 919  
INDEX (Zusatz)  
    INSERT 241  
    MODIFY 267  
Information View → SAP-  
HANA-View  
Informationsnachricht 200

INHERITING FROM  
(Zusatz), CLASS 373  
INIT (Zusatz), REDUCE 275  
Initialisierung 186  
INITIALIZATION (ABAP-  
Anweisung) 394  
Inline-Count 957  
Inline-Deklaration 182,  
302, 614  
Inline-Kommentar 167  
In-Memory-Technologie  
910  
Inner Join 307  
INNER JOIN (Zusatz),  
SELECT 307  
INSERT (ABAP-Anwei-  
sung) 241  
INSERT (Open-SQL-Anwei-  
sung) 324, 521  
Inspektion 501  
Instanz 813, 817  
Instanziieren 817, 820  
Instanzkonstruktor 158  
Instanzmethode 636  
Integrated Data Access  
578, 613  
Integrated Development  
Environment 37, 62  
Interaktion, externe 867  
Interface 150, 835  
    Alias 157  
    ALV 624  
    ALV mit IDA 619  
    anlegen 161  
    definieren 383  
    IF\_MESSAGE 379  
    IF\_OS\_CA\_  
        PERSISTENCY 768  
    IF\_OS\_FACTORY 768,  
772  
    IF\_OS\_QUERY 771  
    IF\_OS\_QUERY\_  
        MANAGER 771  
    IF\_SALV\_C\_CELL\_TYPE  
        607  
    IF\_SALV\_GUI\_TABLE\_  
        DISPLAY\_OPT 617  
    IF\_SALV\_GUI\_TYPES\_IDA  
        618  
    IF\_SALV\_IDA\_CALC\_  
        FIELD\_HANDLER 620

Interface (Forts.)  
    IF\_SALV\_SERVICE\_TYPES  
        624  
    IF\_SERIALIZABLE\_  
        OBJECT 776  
    IF\_SHM\_BUILD\_  
        INSTANCE 781  
    IF\_T100\_DYN\_MSG 381  
    implementieren 384  
    Komponenten 383  
    Verwendung 386  
INTERFACE ... ENDINTER-  
FACE (ABAP-Anwei-  
sung) 383  
INTERFACES (ABAP-Anwei-  
sung) 384  
interne Tabelle 80, 91, 231  
    ändern 265, 486  
    Arten 233  
    aufbereiten 273  
    auslesen 251  
    Debugging 486  
    dynamisch erzeugen 787  
    eingebaute Funktion 277  
    füllen 239  
    gehaschte 521  
    initialisieren 239  
    Kopfzeile 236  
    kopieren 271  
    löschen 264  
    mit Kopfzeile 239  
    sortieren 273  
    sortierte 521  
    strukturfremde 271  
    strukturgleiche 271  
    Summen bilden 274  
    Verkettung 278  
    Zuweisung 223  
Internet Communication  
Framework 655, 991  
INTO (Zusatz)  
    CASE 191  
    CASE TYPE OF 390  
    CATCH 379  
    CONCATENATE 203  
    INSERT 325  
    MESSAGE 419  
    SELECT 289, 299  
INTO CORRESPONDING  
FIELDS OF (Zusatz),  
SELECT 292, 803

INTO TABLE (Zusatz),  
SELECT 240  
ipow() (eingebaute  
Funktion) 245  
IS INITIAL (Zusatz),  
TYPES 177  
IS INSTANCE OF (ABAP-  
Anweisung) 389  
Iteration, bedingte 244  
Iterationsausdruck 244  
Iterationsvariable 244  
**J**  
JavaScript 928  
Job 563  
    allgemeine Angaben 564  
    Betriebsart 567  
    definieren 563  
    einplanen 570  
    Ereignis 567, 572  
    freigeben 570  
    Jobklassen 564  
    Ketten 567  
    Klasse 564  
    mit ABAP definieren 573  
    periodische Ausführung  
        565  
    Schritt (Step) 569  
    Startbedingung 565  
    Übersicht 570  
    Wizard 564  
Jobverwaltung 563  
Join 304  
JOIN (Zusatz), SELECT 304  
JSON 773  
**K**  
Kachel 932  
Kardinalität 977  
KEEPING TARGET LINES  
(Zusatz), MOVE-CORRE-  
S PONDING 272  
KEEPING TASK (Zusatz),  
RECEIVE RESULTS  
FROM FUNCTION 785  
Kettensatz 167  
Kindklasse → Unterklasse  
    602

Klasse (Forts.)  
*CL\_SALV\_FUNCTIONS\_LIST* 601  
*CL\_SALV\_GUI\_TABLE\_IDA* 614  
*CL\_SALV\_GUI\_TOOLBAR\_IDA* 615  
*CL\_SALV\_LAYOUT* 612  
*CL\_SALV\_RANGE\_TAB\_COLLECTOR* 626  
*CL\_SALV\_SORTS* 610  
*CL\_SALV\_TABLE* 599  
*CL\_SXML\_STRING\_WRITER* 774  
*CL\_XML\_DOCUMENT* 775  
*CX\_SY\_AUTHORIZATION\_ERROR* 137  
*Datentypen* 358  
*Definitionsteil* 354  
*Entitätsklasse* 871  
*Ereignis* 156  
*erweitern* 701  
*finale* 375, 834  
*Freund* 151, 383  
*globale* 353, 810  
*Hierarchie* 831  
*Implementierungsteil* 354  
*instanzieren* 355  
*Interface* 150  
*Klasseneignis* 826  
*Komponente* 357  
*Komponentensichtbarkeit* 357  
*Konstruktor* 822  
*Konstruktorklasse* 781  
*Kontrollklasse* 871  
*lokale* 353, 810  
*Oberklasse* 830  
*persistente* 764  
*Schnittstellenklasse* 871  
*Shared Objects* 781  
*Sichtbarkeit* 357  
*statische* 375, 818  
*testen* 158  
*Unterklassen* 830  
*Vererbung* 149, 373  
*Wurzelklasse* 781  
*Klassenakteur* 767  
*Klassenattribut* 819  
*Klassendiagramm* 843, 850, 866

Klassenkomponente  
*dynamische* 375  
*statische* 375  
Klassenkomponentenselektor 359  
Klassenkonstruktor 158, 368  
Klassenmethode 819  
Klassifikation 816  
COLLECTOR 626  
Komponentenselektor 52  
Komprimierung 913  
Konsistenz 443  
Konstante 173  
Konstruktor 157, 366, 821  
*Instanz* 367, 821  
*Klassen* 368, 821  
*Redefinition* 840  
*statischer* 368  
Konstruktorausdruck 229, 356, 522, 614  
Konstruktorklasse 781  
Kontrollklasse 871  
Konvertierungs-Exit 94  
Konvertierungsroutine 93  
Kopfzeile 236  
Kopiermodus 155

**L**

Lastenheft 865  
Laufzeitanalyse 519, 523  
*ABAP Debugger* 528  
*auswerten* 525  
*Messung* 524  
*mit ABAP* 538  
*Transaktion SE30* 527  
Layout 598  
LEAVE PROGRAM (ABAP-Anweisung) 460  
LEAVE TO SCREEN (ABAP-Anweisung) 460  
Leerzeichen 166  
Leerzeilen 406  
LEFT [OUTER] JOIN (Zusatz), SELECT 308  
LENGTH (Zusatz), PARAMETERS 397  
Klassendiagramm 843, 850, 866

**M**

Lesesperre 126  
LET (Zusatz) 226  
COND 220  
CONV 219  
EXACT 219  
SWITCH 222  
LIKE (Zusatz)  
DATA 181  
FORMS 432  
LIKE LINE OF (Zusatz) 235  
line\_exists() (eingebaute Funktion) 279, 522  
line\_index() (eingebaute Funktion) 278  
LINES OF (Zusatz)  
APPEND 240  
INSERT 242  
NEW 250  
VALUE 250  
lines() (eingebaute Funktion) 277  
linguistische Suche 919  
Linie 406  
Liste, hierarchische 579  
Listenausgabe 197  
Listenüberschrift 417  
Literal 174, 211  
Logger 911  
Log-Handle 543  
Logical Unit of Work 444  
logischer Ausdruck 189  
Look-up-Variante 268  
LOOP AT ... ASSIGNING 521  
LOOP AT (ABAP-Anweisung) 257, 521  
LOWER CASE (Zusatz), PARAMETERS 396  
LSMW 678  
LUW 444

**M**

Makro 433  
aufrufen 434  
definieren 433  
Mandant 314, 548  
CUST 548  
Customizing-Mandant 549

Mandant (Forts.)  
*Entwicklungstestmandant* 550  
*Mandantentyp* 548  
PROD 548  
*Produktivmandant* 549  
*Prototypmandant* 551  
QTST 548  
SAND 551  
*Schulungsmandant* 551  
TEST 550  
TRNG 551  
Mandantenabhängigkeit 110  
Mapping 983  
MAPPING (Zusatz), CORRESPONDING 224  
Mapping-Vorschrift 224  
Master-Include 146  
MATCHCODE OBJECT (Zusatz), PARAMETERS 396, 925  
mathematische Funktion 196  
me-> (Selbstreferenz) 365  
Mechanismus 857, 872  
Meldung, Typen 200  
Memory 778  
MEMORY ID (Zusatz)  
PARAMETERS 396, 422  
SELECT-OPTIONS 422  
Memory Object Explorer 492  
Mengenfeld 314  
Menü-Exit 717  
Menüleiste 457  
Mesh 279  
Mesh-Pfad 280  
Mesh-Typ 279  
MESSAGE (ABAP-Anweisung) 200, 419  
MESSAGE RAISING (ABAP-Anweisung) 382  
Metadatendokument 936  
METHOD (ABAP-Anweisung) 362  
Methode 813, 817  
abstrakte 360  
anlegen 152  
aufrufen 363  
Ausnahmen 377  
definieren 360

Methode (Forts.)  
*erweitern* 702  
*finale* 360  
*implementieren* 362  
*Over-Write-Methode* 702  
Parameter 153  
*polymorphe* 839  
*Post-Methode* 702  
*Pre-Methode* 702  
*Redefinition* 155, 373  
*Signatur* 154  
*statische* 819  
*Verkettung* 365  
Methode, statische 819  
METHODS (ABAP-Anweisung) 360  
MIME Repository 40  
MODE (Zusatz), CALL TRANSACTION 675  
Model 885  
Model Provider Class 946  
Model View Controller 885  
Modell 862  
*strukturelles* 843  
*verhaltensorientiertes* 843  
Modellierung 862  
MODIF-ID (Zusatz) 408  
Modifikation 681  
Modifikationsgruppe 408  
Modifikator 824  
MODIFY (ABAP-Anweisung) 267, 521  
MODIFY (Open-SQL-Anweisung) 328  
MODIFY SCREEN (ABAP-Anweisung) 409  
Modul 454  
MOVE (ABAP-Anweisung) 218  
MOVE-CORRESPONDING (ABAP-Anweisung) 176, 223, 272, 522  
Multiplikation 195  
Muster 51, 54  
*ABAP Objects* 387  
ADT 74  
Funktionsbaustein 74  
Methodenaufruf 74  
MVC 885

**N**

Nachricht  
*anlegen* 418  
*aufrufen* 419  
*parametrisierte* 419  
Nachrichten-Header 543  
Nachrichtenklasse 417, 543  
*anlegen* 418  
*aufrufen* 419  
*Platzhalter* 419  
Namenskonvention 1025  
natives SQL 285, 330, 342  
Navigation 49, 937  
Navigations-Button 457  
Navigationsstapel 50  
NEW (ABAP-Anweisung) 222, 251, 356, 788  
NEXT (Zusatz), REDUCE 275  
NO INTERVALS (Zusatz), SELECT-OPTIONS 401  
NO-DISPLAY (Zusatz), PARAMETERS 397  
NO-EXTENSION (Zusatz), SELECT-OPTIONS 401  
Nullwert ersetzen 320

**O**

Oberklasse 149, 374, 829  
Object Navigator 39  
Object Services 764  
Objekt 813, 816  
*Klassifikation* 816  
*persistentes* 764  
Objektidentität 820  
Objektliste 43, 50  
Objektmenge 501  
objektorientierte Analyse 843  
Objektorientierung 809  
Objektreferenzvariable Fallunterscheidung 191  
*Typ prüfen* 389  
Objektsperre 547  
OBLIGATORY (Zusatz), PARAMETERS 396  
Observer 895

OData 933  
 OData-Service 927  
   entwickeln 942  
   Fehleranalyse 953  
   lesender 942  
   Metadaten 936  
   Methode 933  
   Navigation 935  
   testen 951  
   Veröffentlichung 949  
 OF PROGRAM (Zusatz)  
   SET PF-STATUS 458  
   SET TITLEBAR 459  
 Offset 204  
 OFFSET (Zusatz)  
   DELETE 322  
   SELECT 303, 956  
 ON ... FOR (Zusatz), AT  
   SELECTION-SCREEN  
   411  
 ON (Zusatz), SELECT 304  
 ON BLOCK (Ereignis) 411  
 ON COMMIT (Zusatz),  
   PERFORM 444  
 ON END OF (Ereignis) 411  
 ON EXIT-COMMAND  
   (Ereignis) 411  
 ON HELP-REQUEST FOR  
   (Ereignis) 411  
 ON RADIOBUTTON  
   GROUP (Ereignis) 411  
 OOA 843  
 OO-Transaktionsmodell  
   132  
 OPEN DATASET (ABAP-  
   Anweisung) 647  
 Open SQL 285  
   Ausdruck 311  
   dynamisches 802  
   Funktionen 315  
   Neuerungen 336  
   Performance 286  
   Syntaxänderungen 300  
   WHERE ... IN 295  
   Wildcards 295  
 Operandenposition 226  
 Operation 872  
 OPTION (Zusatz), SELECT-  
   OPTIONS 401  
 OPTIONAL (Zusatz), Tabel-  
   lenausdruck 254  
 ORDER BY (Zusatz)  
   DELETE 322  
   SELECT 297, 304  
 OTF 762  
 OTF-Tabelle 759  
 OTHERS (ABAP-  
   Anweisung) 441  
 Outer Join 307  
 OUTER JOIN (Zusatz),  
   SELECT 307  
 Outline-Anzeige 78  
 OUTPUT (Zusatz), AT  
   SELECTION-SCREEN  
   410

**P**

Package Builder 57  
 PACKAGE SECTION (Sicht-  
   barkeitsbereich) 358  
 Page Management 910  
 Paging 303  
 PAI 452, 460  
 Paket 57  
   anlegen 57  
   Hierarchie 58  
   Transport 551  
   Typ 58  
 Parallelisierung 785  
 Parameter 153, 396, 818  
 PARAMETERS (ABAP-  
   Anweisung) 396  
 Parametertransaktion 133  
 PBO 452  
 PDF 733, 759  
 PERFORM (ABAP-Anwei-  
   sung) 428  
 Performance 286, 519  
 Performance-Trace 523,  
   528  
 PERFORMING (Zusatz),  
   CALL FUNCTION 785  
 Persistenzdienst 764  
 Persistenzschicht 911  
 Pfad 340  
 Pflegedialog 117  
   anlegen 118  
   Ansicht verbreitern 119  
 Pflege-View 116  
 Pflichtfeld 467  
 Pointer 179

Polymorphie 839, 841  
 Pop-up-Fenster 451,  
   464, 485  
 Port, logischer 666  
 Potenzierung 195  
 Prädikatfunktion 279  
 Pretty Printer 54, 73  
 Primärindex 110  
 Primärschlüssel 110  
 Primärschlüsselfeld 584  
 Private 357  
 PRIVATE SECTION (Sicht-  
   barkeitsbereich) 358  
 Process After Input 452,  
   460  
 Process Before Output 452  
 Produktivmandant 548  
 Produktivsystem 549  
 Programm 636  
   ausführbares 47, 394  
   dynamisches 447  
   Typen 47  
 Project Explorer 68  
 Projekt, ADT 67  
 Projektion 115, 291  
 Projektions-View 115  
 Property 936, 937  
 Protected 357  
 PROTECTED SECTION  
   (Sichtbarkeitsbereich)  
   358  
 Prototypmandant 551  
 prozedurale Programmier-  
   sprache 425  
 Prüfkriterium 312  
 Prüfvariante 501  
   anlegen 502  
   FUNCTIONAL\_DB 914  
   FUNCTIONAL\_DB\_-  
     ADDITION 914  
   importieren 914  
 Public 357  
 PUBLIC SECTION (Sichtbar-  
   keitsbereich) 358  
 Puffer 108  
 Puffer-Trace 535  
 Pufferung 108  
 Punkt 166  
 PUSHBUTTON (Zusatz),  
   SELECTION-SCREEN  
   401

**Q**

QL 334  
 Qualitätssicherungs-  
   mandant 548  
 Qualitätssicherungs-  
   system 549  
 Query Language 334  
 Query-Dienst 764  
 Query-Manager 770  
 Query-Objekt 770  
 Quick Fix 84, 87  
 Quickinfo 402

**R**

Radiobutton 398, 451  
 RADIobutton GROUP  
   (Zusatz), PARAMETERS  
   398  
 Rahmen 451  
 Rahmenprogramm 144,  
   720  
 RAISE (ABAP-Anweisung)  
   382  
 RAISE EVENT (ABAP-  
   Anweisung) 370  
 RAISING (Zusatz),  
   METHODS 360, 378  
 Ranges-Tabelle 295  
 Ranges-Tabellentyp 103  
 READ DATASET (ABAP-  
   Anweisung) 649  
 READ TABLE (ABAP-Anwei-  
   sung) 251, 521  
 READ-ONLY (Zusatz),  
   DATA 359  
 RECEIVE RESULTS FROM  
   FUNCTION (ABAP-  
   Anweisung) 785  
 Rechenausdruck 195  
 Rechenoperation 195  
 Redefinition 830, 839  
 REDEFINITION (Zusatz),  
   CLASS 374  
 REDUCE (ABAP-Anwei-  
   sung) 274, 522  
 Redundanz 815  
 Refactoring 85  
 Referenzfeld 101

Referenztabelle 101  
 Referenzübergabe 430  
 Referenzvariable 279, 356,  
   786, 820  
 Registerkarte 407, 451  
 Remote Function Call 634  
 Rename 86  
 REPLACE (ABAP-Anwei-  
   sung) 208  
 replace() (eingebaute Funk-  
   tion) 208  
 Report 47  
   anlegen 46  
   aufrufen 420  
   Ereignisse 394  
   objektorientierter 901  
 RDDIT076 561  
 RS\_AUCV\_RUNNER 513  
 Selektionsbildschirm 393  
 Reporttransaktion 131  
 Repository Browser 40, 42  
 Repository-Infosystem 40,  
   442  
 Repräsentant 260  
 Repräsentantenbindung  
   260  
 RESPECTING CASE (Zusatz),  
   FIND 206  
 Rest (Rechenoperation)  
   195  
 RESULT XML (Zusatz),  
   CALL TRANSFORMA-  
   TION 774  
 RESUMABLE (Zusatz),  
   METHODS 378  
 Return-Code 1016  
 RETURNING (Zusatz),  
   METHODS 360  
 RFC 634  
 RIGHT [OUTER] JOIN  
   (Zusatz), SELECT 308  
 RISK LEVEL HARMLESS  
   (Zusatz), CLASS 507  
 Robustheit 823  
 Rollback 443, 445  
 ROLLBACK WORK (ABAP-  
   Anweisung) 322, 445  
 Rolle 845  
 RTTC 792  
 RTTI 792  
 RTTS 792

Rückgabe 182  
 Runtime Type Services 792

**S**

Sammelsuchhilfe 121  
 SAP .NET Connector 635  
 SAP Community Network  
   442  
 SAP Control Framework  
   461  
 SAP Fiori 927, 931  
   Apps 927  
   Launchpad 927, 932  
   UX 927  
 SAP Gateway 927, 939  
 SAP GUI,  
   Programmierung 447  
 SAP HANA 90, 285, 907  
   Datenbankschema 340  
   goldene Regeln 287  
   Migration 913  
 SAP HANA Cloud Platform  
   930  
 SAP HANA Studio 908,  
   915  
 SAP Interactive Forms by  
   Adobe 753  
 SAP Java Connector 635  
 SAP List Viewer → ALV  
 SAP S/4HANA 928  
 SAP Smart Forms 731, 743  
 SAP User Interface Develop-  
   ment Toolkit for HTML5  
   → SAPUI5  
 SAP Web IDE 930  
 SAP-Business-Objekt 635  
 SAP-HANA-Objekt 908,  
   915  
 SAP-HANA-View 339, 908,  
   915  
 SAP-Hilfe 168  
 SAP-Komponente 636  
 SAP-LUW 444  
 SAPscript 731, 735  
 SAPUI5 303, 927, 928  
 SAP-Verknüpfung 486  
 Satz 166  
 Schalter 728  
 Schleife 187

Schlüssel 314  
*definieren* 235  
*interne Tabelle* 235  
 Schlüsselfeld 110  
 Schlüsselwort, Dokumentation 168  
 Schnellanzeige der Variablen 481  
 Schnittstelle 633  
 Schnittstellenklasse 871  
 Schreibposition 226, 228  
 Schreibsperrre 126  
 Schulungsmandant 551  
 SCN 442  
 SCREEN (Tabelle) 412  
 Screen Painter 449  
 Screenvariante 135  
 Script Wizard 497  
 Script Wrapper 499  
 Searched Case 312  
 Seite  
*Folgeseite* 736, 743  
*Formular* 735  
 Seitenattribut 968  
 Seitenfenster 735  
 Sekundärindex 111  
 Selbstdreferenz 365  
 SELECT (ABAP-Anweisung) 240  
 SELECT (Open-SQL-Anweisung) 288, 520  
 SELECT \* (Open-SQL-Anweisung) 520  
 SELECTION-SCREEN  
*BEGIN OF BLOCK* (ABAP-Anweisung) 405  
*Set-/Get-Parameter*  
*BEGIN OF SCREEN* (ABAP-Anweisung) 469  
 SELECT-OPTIONS (ABAP-Anweisung) 400, 521  
 Selektion 293  
 Selektionsbedingung 114, 237  
 Selektionsbildschirm 393  
*Blöcke* 405  
*dynamisches Ausblenden* 412  
*Eingabeelemente* 395  
*Ergebnisse* 410  
*Funktionscode* 404  
*Leerzeilen* 406

Selektionsbildschirm  
*(Forts.)*  
*Linie* 406  
*Parameter* 396  
*Pop-up-Fenster* 468  
*Strukturierungselemente*  
*405*  
*Tabelle SCREEN* 412  
*Textausgaben* 407  
*Texte* 416  
*Variante* 132

Selektionsliste 291, 299, 803  
 Selektionsoptionen 295, 400  
 Selektionsparameter 421  
 Selektionstabelle 421  
 Selektionstext 416  
 Selektor 824  
 SEPARATED BY (Zusatz),  
*CONCATENATE* 203  
 Serialisierung 773, 776  
 Service-Consumer 664  
 Session-Breakpoint 475  
 SET (Zusatz), UPDATE 328  
 SET HANDLER (ABAP-Anweisung) 370, 588  
 SET PARAMETER (ABAP-Anweisung) 423  
 SET PF-STATUS (ABAP-Anweisung) 458  
 SET TITLEBAR (ABAP-Anweisung) 459  
 Set-/Get-Parameter  
*BEGIN OF BLOCK* (ABAP-Anweisung) 405  
*Shared Memory* 778, 780  
*SHARED MEMORY*  
*ENABLED* (Zusatz), CLASS 781  
*Shared Objects* 778, 780  
*Sichtbarkeit* 357, 823  
*package* 824  
*private* 823  
*protected* 823  
*public* 823  
 SIGN (Zusatz), SELECT-OPTIONS 401  
 SINGLE (Zusatz), SELECT 289  
 Singleton 879

SKIP (Zusatz), SELECTION-SCREEN 406  
 SM30-Tabelle 117  
 Smart Forms 731, 743  
 Smart Style 745  
 Snapshot 532  
 SOA-Manager 655, 661  
 SOAP 654  
 Softwarekomponente 58  
 SORT (ABAP-Anweisung) 273  
 SORTED TABLE (Zusatz), DATA 235  
 sortierte Tabelle 234, 240  
 SOURCE XML (Zusatz), CALL TRANSFORMATION 775  
 SPA-/GPA-Parameter 422  
*anlegen* 423  
*auslesen* 424  
*setzen* 423  
 Spalte 104  
*ausblenden* 588, 604, 618  
*Breite* 590, 606  
*Datenelement* 622  
*einfärben* 608  
*entfernen* 604  
*hinzufügen* 589, 604, 619  
*löschen* 589, 618  
*Position* 590, 605, 621  
*umbenennen* 590, 605, 619  
 Spaltenorientierung 911  
 Speicherabzug 493  
 Speicherart 110  
 Speichern, Implementierung 583  
 Speicherverwaltung 492  
 Sperrbaustein 126  
 Sperrkonzept 125  
 Sperrmodus 126  
 Sperrobject 125  
 Spezialisierung 829, 856  
 SPLIT ... AT (ABAP-Anweisung) 650  
 Split-Container 631  
 Spool-Auftrag 734  
 SPOTS (Zusatz)  
*ENHANCEMENT-POINT* 697

SPOTS (Zusatz) (Forts.)  
*ENHANCEMENT-SECTION* 705  
 Sprache 314, 1029  
 SQL  
*natives* 285, 330, 342  
*Open SQL* 285  
 SQL Performance Tuning  
*Worklist* 520, 529, 533  
 SQL-Monitor 519, 529  
*Administration* 530  
*Snapshots* 532  
 SQLScript 346, 908, 918  
 SQL-Trace 536  
 SSCRFIELDS (Struktur) 403, 405  
 Stammdaten 108  
 STANDARD TABLE (Zusatz), DATA 235  
 Standardtabelle 233  
 Startbedingung 564  
 STARTING AT (Zusatz), CALL SCREEN 451  
 STARTING NEW TASK (Zusatz), CALL FUNCTION 785  
 START-OF-SELECTION (ABAP-Anweisung) 356, 394  
 statischer Konstruktor  
*→ Klassenkonstruktor*  
 Status-Icon 451  
 Statusmeldung 200  
 Stepliste 569  
 Steuerkommando 740  
 Stored Procedure  
*→ Datenbankprozedur*  
 String-Template 211  
 strlen() (Zeichenkettenfunktion) 205  
 STRUCTURE (Zusatz), TYPES 178  
 Struktur 100, 174  
*ALV* 581  
*anlegen* 100  
*Append-Struktur* 706  
*definieren* 185  
*erweitern* 706  
*tief* 100, 174  
*zu interner Tabelle* 235  
 Zuweisung 223

Strukturierungselement 405  
 Strukturmuster 878  
 Strukturpaket 58  
 Subklasse → Unterklasse  
 SUBMIT (ABAP-Anweisung) 420  
 Subscreen 407, 449, 721  
 substring() (Teilfeldfunktion) 205  
 Subtraktion 195  
 Suche  
*Fuzzy-Suche* 922  
*linguistische* 923  
*Type-Ahead* 924  
*Vorschlagssuche* 924  
 Suchhilfe 121, 396  
*anlegen* 121  
*Append-Suchhilfe* 708  
*elementare* 121  
*erweitern* 708  
*Vorschlagssuche* 926  
 SUPER (Referenz) 374  
 super-> (Pseudoreferenz) 830  
 Superklasse → Oberklasse  
 SWITCH (ABAP-Anweisung) 221  
 Switch Framework 693, 727  
 Table Control 451  
 TABLE OF (Zusatz), SELECT 290  
 TABLES (Zusatz), FORMS 432  
*PERFORM* 430  
 Tabstrip 407, 451  
 Tag Browser 40, 966, 970  
 technische Einstellungen 107  
 Technische Info 423, 536  
 Teilfeldfunktion 205  
 Teilfeldzugriff 204, 208  
 Test Repository 41  
 Testklasse 506, 511  
 Testmethode 507  
 Text 607  
*Feldbezeichner* 97  
*Selektionsbildschirm* 407  
*Selektionstext* 416  
*Übersetzung* 98  
 Textanalyse 923  
 Textdatei 646

---

**T**

Tabelle  
*anlegen* 105  
*Anzeige* 577  
*Arten* 233

Text-Editor 461  
 Textelement 413, 417, 739  
 Textfeld 451, 750  
 Textliteral 174  
 Textsuche 627, 923  
 Textsymbol 402, 413  
 TIMES (Zusatz), DO 192  
 TMS 548  
 TO (Zusatz), LOOP AT 257  
 TO MEMORY ID (Zusatz),  
   EXPORT 779  
 Token 166  
 Toolbar Control 463  
 Top-down-Ansatz 334  
 Top-Include 47, 146, 434  
 Trace 535  
   *im Debugger* 535  
   *Performance-Trace* 523  
 Transaktion 129, 636  
   /IWFND/ERROR\_LOG  
     942, 953  
   /IWFND/GW\_CLIENT  
     942  
   /IWFND/MAINT\_  
     SERVICE 942,  
     949, 951  
   AL11 647  
   ATC 516  
   BAPI 639  
   CMOD 685, 686  
   DBACOCKPIT 330  
   ICON 595  
   mit Parametern 133  
   mit Variante 133, 134  
   OMSR 644  
   PFCG 67  
   SAT 519, 523  
   SAUNIT\_CLIENT\_SETUP  
     509  
   SCI 501, 502, 914  
   SCOV 534  
   SE01 40, 548  
   SE03 561  
   SE10 40, 548  
   SE11 89, 920  
   SE16 107  
   SE16N 107  
   SE18 722  
   SE19 689, 690  
   SE20 41, 694  
   SE21 57, 131  
   SE24 147, 765  
 Transaktion (Forts.)  
   SE30 523, 527  
   SE37 442  
   SE38 46  
   SE71 735  
   SE80 39, 46, 966  
   SE84 40  
   SE91 418  
   SE93 130  
   SEGW 942  
   SFP 753  
   SFW1 729  
   SFW2 729  
   SFW3 729  
   SFW5 729  
   SHD0 135  
   SHDB 672  
   SHMA 782  
   SICF 662, 991  
   SICK 347  
   SLG0 542  
   SLG1 541  
   SM13 445  
   SM30 117  
   SM35 671  
   SM36 563  
   SM37 563, 570  
   SM59 653  
   SM62 572  
   SM64 572  
   SM69 570  
   SMARTFORMS 743  
   SMOD 685  
   SOAMANAGER 661  
   SP01 734  
   SPACKAGE 57  
   SPAU 712  
   SPAU\_ENH 712  
   SPDD 712  
   SPROXY 657  
   SQLM 529  
   SQLMD 529, 532  
   ST05 535  
   STERM 1029  
   STMS 548, 554  
   STWBM 41  
   SU01 422, 734  
   SWLT 529, 533  
   SWO1 636  
   WSCONFIG 661  
 Transaktionscode 129  
   anlegen 130

Transaktionscode (Forts.)  
   SE37 139  
   Übersicht 1012  
 Transaktionsdienst 132,  
   764  
 Transaktionsrecorder 671  
 Transaktionsvariante 134  
 TRANSFER DATASET  
   (ABAP-Anweisung) 647  
 Transformation 772  
 Transport  
   Art 551  
   Aufgabe 557  
   von Kopie 551  
 Transport Organizer 40,  
   548, 553  
 Transport Organizer Tools  
   561  
 Transportauftrag 547, 551  
   anlegen 552  
   Aufgabe 547  
   finden 561  
   Freigabe zurücknehmen  
   561  
   freigeben 553  
   importieren 554  
   löschten 559  
   Objekte aufnehmen 559  
   Objektsperre 547  
   schreiben 558  
   verschmelzen 560  
   Zuordnung ändern 557  
 TRANSPORTING (Zusatz),  
   MODIFY 267  
 TRANSPORTING NO-  
   FIELDS (Zusatz), READ  
   TABLE 252, 522  
 Transport-Management-  
   System 548  
 Transport-Prüfvariante 502  
 Transportschicht 58  
 Transportwesen 547  
 Trigger 497  
 TRY ... CATCH (ABAP-  
   Anweisung) 803  
 TRY ... ENDTRY (ABAP-  
   Anweisung) 379, 441  
 Typ  
   *eigener* 786  
   *umwandeln* 388  
 Typbeschreibungsobjekt  
   799

TYPE (Zusatz)  
   DATA 102, 172, 181,  
   235  
   FORMS 432  
   MESSAGE 200  
   PARAMETERS 397  
   TYPES 184  
 TYPE RANGE OF (Zusatz),  
   DATA 238  
 TYPE TABLE OF (Zusatz),  
   DATA 102, 235  
 Type-Ahead-Suche 924  
 TYPES (ABAP-Anweisung)  
   184, 279  
 Typgruppe 112  
 Typkonvertierung 173  
 Typumwandlung 314

**U**

UI-Element 449, 451, 965,  
   984  
 ULINE (Zusatz), SELEC-  
   TION-SCREEN 406  
 Umbenennung 86  
 UML 843  
 UNION (Open-SQL-  
   Anweisung) 299  
 UNION (Zusatz), SELECT  
   310  
 Unterklasse 374, 829  
 Unterpaket 58  
 Unterprogramm 426, 810  
   Aufruf 428  
   Aufruf aus anderem Pro-  
     gramm 428  
   definieren 427  
   Parameterübergabe 430  
 UNTIL (Zusatz) 244  
 Up Cast 388  
 UP TO ... ROWS  
   (Zusatz) 522  
   DELETE 322  
   SELECT 290  
 UPDATE (Open-SQL-  
   Anweisung) 326  
 Use Case Diagram 844, 866  
 User Experience 927  
 User-Breakpoint 476  
 USER-COMMAND (Zusatz)  
   PARAMETERS 404

USER-COMMAND (Zusatz)  
   (Forts.)  
   SELECTION-SCREEN 404  
 User-Exit 681  
 USING (Zusatz)  
   CALL TRANSACTION  
     674  
   PERFORM 430  
 USING KEY (Zusatz)  
   APPEND 241  
   FILTER 276  
   MODIFY 267  
   TYPES 279  
 USING SELECTION-SET  
   (Zusatz), SUBMIT 421  
 VIA SELECTION-SCREEN  
   (Zusatz), SUBMIT 420  
 View 113, 885  
   Analytic 917  
   Attribute 916  
   Calculation 917  
   CDS 333  
   externer 340  
 VISIBLE LENGTH (Zusatz),  
   PARAMETERS 397  
 Vollbildmodus 50  
 Volltextindex 919  
 Vorschlagshilfe 51  
 Vorschlagsliste 51  
 Vorschlagssuche 924  
 Vorwärtsnavigation 49

**V**

VALUE (ABAP-Anwei-  
   sung) 216, 242, 522  
 VALUE (Zusatz)  
   CONSTANTS 173  
   DATA 172  
   Tabellenausdruck 256  
 VALUE CHECK (Zusatz),  
   PARAMETERS 396  
 VALUE-REQUEST (Zusatz),  
   AT SELECTION-SCREEN  
     411  
 VALUES (Zusatz), INSERT  
   325  
 Variable 169  
   Anzeige 78  
   bearbeiten 482  
   vergleichen 482  
 Variante 132  
 Variantentransaktion 133  
 Verantwortlichkeit 871  
 Verbuchungsbaustein 438,  
   444  
 Vererbung 149, 373, 815,  
   828, 841  
 Vergleich 189  
 Vergleichstool 483  
 Verhaltensmuster 878  
 Verkettungsfunktion 278  
 Verkettungsoperator 203,  
   312  
 Verknüpfung 304  
 Verwendungsnnachweis 50  
 Verzweigung 187

**W**

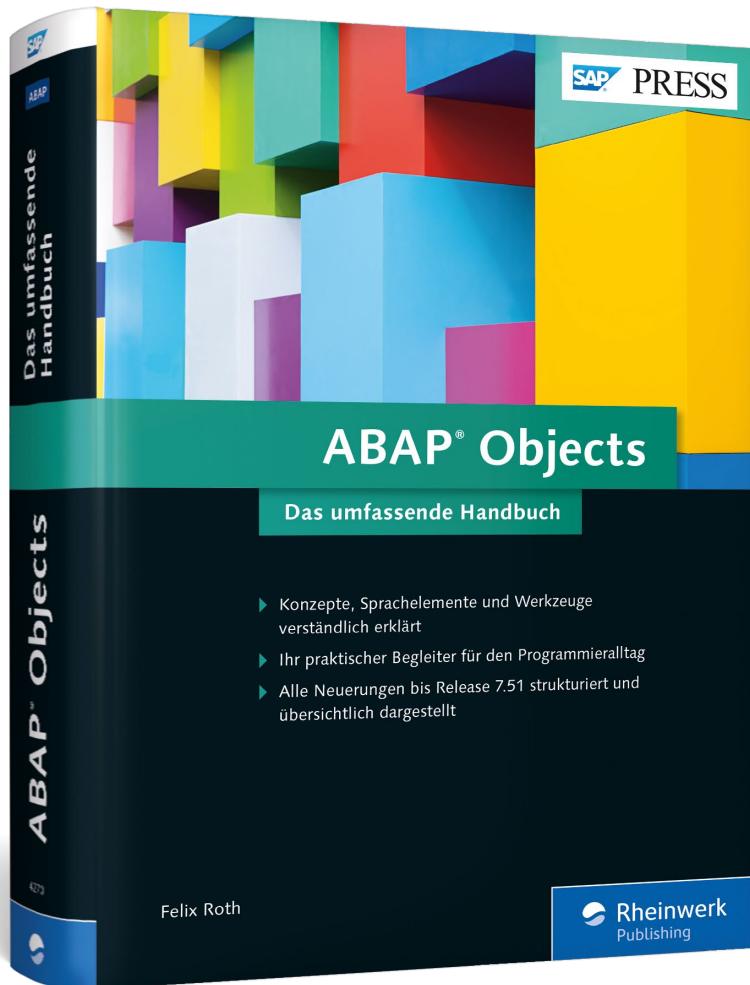
Währungsfeld 314  
 Währungsschlüssel 314  
 WAIT UNTIL (ABAP-Anwei-  
   sung) 785  
 Warnung 200  
 Wartbarkeit 815  
 Watchpoint 79, 491  
 Web Dynpro  
   Anwendung 974  
   Component 974  
   Knoten 976  
   Mapping 983  
   Methode 979  
 Web Dynpro ABAP 963,  
   972  
 Web Dynpro ABAP, SAP  
   List Viewer 886  
 Web Dynpro Text Browser  
   41  
 Web-Dynpro-Anwendung  
   aufrufen 990  
 Weboberfläche 963  
 Webservice  
   anlegen 658  
   Aufruf mit ABAP 669  
   Freigabe 661  
   logischer Port 666  
   Service-Definition 658  
   testen 669  
   Wizard 657  
 Webservice konsumieren  
   664

Webservice-Framework	SCREEN BEGIN OF BLOCK 405
655	
Webservices	WITH NON-UNIQUE (Zusatz), DATA 235
öffnen 657	
Sicherheitsprofil 659	WITH OVERVIEW (Zusatz), CALL 343
Wert, elementarer 315	WITH SELECTION-TABLE (Zusatz), SUBMIT 421
Wertebereich 95	WITH TABLE KEY 521
Wertetabelle 95	WITH UNIQUE (Zusatz), DATA 235
Werteübergabe 430	WITHOUT AUTHORITY- CHECK (Zusatz), CALL TRANSFORMATION 137
Wertparameter 369	WITHOUT MEMBERS (Zusatz)
Wertüber- und -rückgabe 430	FOR GROUPS OF 247 LOOP AT 262
WHERE (Zusatz) 521	Workarea 174
DELETE 322	Workprozess 444
FILTER 276	Workspace 65
MODIFY 267	WRITE (ABAP-Anwei- sung) 198
UPDATE 328	WSDL 656
WHERE-Klausel (OpenSQL) 293	Wurzelklasse 781
WHILE (ABAP-Anwei- sung) 193	
WHILE (Zusatz) 244	XINX 709
Wiederverwendung 815	XML 772
Window 974	XML-Dokument 775
WITH (ABAP-Anweisung) 309	xsdbool() (Funktion) 202
WITH (Zusatz)	
MESSAGE 420	
SET TITLEBAR 459	
SUBMIT 421	
WITH AUTHORITY-CHECK (Zusatz), CALL TRANS- FORMATION 137	
WITH FRAME ... TITLE (Zusatz), SELECTION-	

**Z**

Zahlenliteral 174
zeichenartiger Typ 314
Zeichenformat 737
Zeichenkette 201
ersetzen 207
finden 206
Operator 201
Vergleich 201
Verkettung 203
Zeichenkettenfunktion 317
Zeichenketten-Template 211
Zeichenliteral 174
Zeigervariable 820
Zeile 104
Zeile einfärben 594
Zeilenkommentar 167
Zeilentyp 747
Zeilentyp, strukturierter 103
Zelle
einfärben 594
Typ 607, 622
Z-Include 684
Zusatz 166
Zuweisung 172, 820
Zuweisungsoperator 172, 388
zweistufiges Domänenprin- zip 91
Zwei-Systeme-Landschaft 550

**X**



Felix Roth

## ABAP Objects – Das umfassende Handbuch

1.056 Seiten, gebunden, Dezember 2016  
69,90 Euro, ISBN 978-3-8362-4270-7

 [www.sap-press.de/4201](http://www.sap-press.de/4201)



**Felix Roth** ist Trainer, Entwickler und Berater mit Herzblut und im kompletten ABAP-Technologie-Umfeld zu Hause. Seit 2013 arbeitet er als SAP Consultant und Software Engineer bei der X-CASE GmbH. In seiner Position berät er Kunden in den verschiedensten Entwicklungsprojekten und beschäftigt sich maßgeblich mit den neuesten SAP-Technologien. Darüber hinaus kümmert er sich hauptverantwortlich um die Nachwuchsgewinnung und die Weiterbildung der Werkstudenten und Praktikanten. Seit 2014 hält Felix Roth regelmäßig SAP-Schulungen für die Integrata AG und erstellte dafür auch mehrere Schulungsunterlagen.

*Wir hoffen sehr, dass Ihnen diese Leseprobe gefallen hat. Sie dürfen sie gerne empfehlen und weitergeben, allerdings nur vollständig mit allen Seiten. Bitte beachten Sie, dass der Funktionsumfang dieser Leseprobe sowie ihre Darstellung von der E-Book-Fassung des vorgestellten Buches abweichen können. Diese Leseprobe ist in all ihren Teilen urheberrechtlich geschützt. Alle Nutzungs- und Verwertungsrechte liegen beim Autor und beim Verlag.*

*Teilen Sie Ihre Leseerfahrung mit uns!*

