

Dr. Boris Rubarth

Schnelleinstieg in ABAP®

- ▶ Schritt-für-Schritt-Anleitungen für Anfänger
- ▶ Nachvollziehbare Erläuterungen und Code-Beispiele
- ▶ Hilfen für Ihre erste eigene ABAP-Anwendung
- ▶ Tipps und Tricks für das Programmieren in ABAP

Inhaltsverzeichnis

1	Erste Schritte in ABAP	7
1.1	Hallo ABAP	7
1.2	Verwenden des ABAP Editor	8
1.3	Starten eines Reports	13
1.4	Verwenden der ABAP-Schlüsselwortdokumentation	15
1.5	Anzeigen von vorhandenem Quelltext	16
1.6	Versionieren eines Reports	19
1.7	Fazit	19
2	Verwenden von ABAP-Anweisungen	21
2.1	Verwenden von Parametern als Eingabevariablen	21
2.2	Verwenden von Bedingungschaltern	25
2.3	Verwenden von übersetzbaren Texten	26
2.4	Arbeiten mit String-Variablen	28
2.5	Verwenden von elementaren Datentypen	30
2.6	Debugging eines Reports	33
2.7	Fazit	36
3	Verwenden der ABAP Workbench	37
3.1	Starten der ABAP Workbench	37
3.2	Konfigurieren von ABAP Workbench-Optionen	39
3.3	Arbeiten mit Paketen und Transportaufträgen	41
3.4	Transaktionscode für einen Report	46
4	Verwenden von Datentypen aus dem ABAP Data Dictionary (DDIC)	51
4.1	Einstieg in komplexe Datentypen	51

4.2	Arbeiten mit dem ABAP Data Dictionary (DDIC)	55
4.3	Arbeiten mit internen Tabellen	60
5	Modularisierung und Wiederverwendung von Funktionen	67
5.1	Verwenden von FORM-Routinen	67
5.2	Verwenden von Funktionsbausteinen	71
5.3	Verwenden von ABAP-Klassen	90
5.4	Ausnahmekonzept für Klassen	101
6	Zugreifen auf die Datenbank	105
6.1	Verwenden von ABAP-Anweisungen für den Zugriff auf die Datenbank	105
7	Technologien für ABAP-Benutzungsoberflächen	115
7.1	Arbeiten mit Nachrichten	115
7.2	Arbeiten mit Listenverarbeitung	117
7.3	Arbeiten mit Bildern und Dynpros	127
7.4	Kurzer Überblick über SAP-UI-Technologien	144
8	Fazit	149
	Weitere Bücher von Espresso Tutorials	160

2 Verwenden von ABAP-Anweisungen

In diesem Kapitel werden wir ABAP-Anweisungen genauer betrachten und auch Variablen ins Blickfeld rücken. Sie erfahren mehr über die Definition von Variablen und wie der Benutzer eines Reports in die Lage versetzt wird, Werte für Variablen festzulegen.

Anhand des Debugging in einer Anwendung können Entwickler den Ablauf einer Programmausführung Schritt für Schritt nachvollziehen, um die Anwendung zu prüfen. Den ABAP-Debugger werden Sie ebenfalls in diesem Kapitel kennenlernen.

2.1 Verwenden von Parametern als Eingabevariablen

Die Anweisung PARAMETERS (immer im Plural) bietet eine einfache Möglichkeit für Benutzer, Eingaben für einen Report vorzunehmen, die in einer Variablen gespeichert werden. Geben Sie den Namen des Parameters an, den Typ und gegebenenfalls einen Vorschlagswert:

```
PARAMETERS pa_name type c length 12 DEFAULT 'HUGO'.
```

(In diesem Beispiel verwenden wir bereits den Typ *c* für Character (Zeichen). Dies wird in Kürze erläutert; es handelt sich dabei um den zu erwartenden Typ).

Wenn Sie den Report starten, zeigt das System ein einfaches Eingabefeld mit dem vorgegebenen Vorschlagswert an, den Sie überschreiben können, siehe Abbildung 2.1.



Abbildung 2.1: Ein Parameter als Eingabevariable

Der Parameter dient als Eingabefeld und als Variable.

Nun verwenden wir die Eingabevariable, um unseren Willkommensgruß etwas persönlicher zu gestalten. Dafür nutzen wir CHAIN-Anweisungen:

```
WRITE: 'Hello', pa_name.
```

CHAIN-Anweisungen



Eine CHAIN-Anweisung fasst mehrere Anweisungen mit demselben Schlüsselwort in einer Anweisung zusammen, die auf verschiedene Datenobjekte angewendet wird. Dafür ist nach jedem Schlüsselwort (hier: WRITE) ein Doppelpunkt erforderlich sowie ein Komma zwischen den Datenobjekten, auf die sich das Schlüsselwort bezieht.

Haben Sie versucht, Ihren Namen mit Kleinbuchstaben einzugeben und festgestellt, dass er in Großbuchstaben angezeigt wurde? Sie können für das Schlüsselwort PARAMETERS den *Schlüsselwortzusatz* »LOWER CASE« verwenden, damit Kleinbuchstaben verwendet werden. Beachten Sie, dass dies keine *Konvertierung* in Kleinbuchstaben ist, sondern die Erlaubnis zur Verwendung von Kleinbuchstaben (also eine Konvertierung in Großbuchstaben zu unterlassen).

Auf der Benutzungsoberfläche (UI: User Interface) in Abbildung wird weiterhin der Name der Eingabevariablen angezeigt, was für den Benutzer nicht immer verständlich ist. Wenn Sie bedenken, dass die Länge einer Parameterbezeichnung auf 8 Zeichen begrenzt ist, können Sie bei Bedarf auch einen aussagekräftigeren Text eingeben.

Sie können für diese Variable zusätzliche Informationen erfassen, indem Sie im Menü zu SPRINGEN • TEXTELEMENTE • SELEKTIONSTEXTE navigieren. Geben Sie als *Selektionstext* für PA_NAME »Dein Name« ein, wie in Abbildung 2.2 dargestellt.



Abbildung 2.2: Bearbeiten des Textes für eine Parametervariable

Beachten Sie, dass Sie den Selektionstext in dieser Ansicht aktivieren müssen, bevor Sie zum Quelltext Ihres Reports zurückkehren (mithilfe der Funktionstaste (F3)).

Falls sich mehrere Ihrer Objekte in einem inaktiven Zustand befinden, zeigt das System diese in Form einer Liste an. So können Sie einige der Objekte zur Aktivierung auswählen oder auch alle Objekte aktivieren. Im folgenden Beispiel ist der Report inaktiv. Wenn Sie den Selektionstext aktivieren, sind die beiden folgenden Objekte in der angezeigten Liste enthalten: das Objekt REPS (S für Source [Quelle]) und das Objekt REPT (T für Text), siehe Abbildung 2.3.

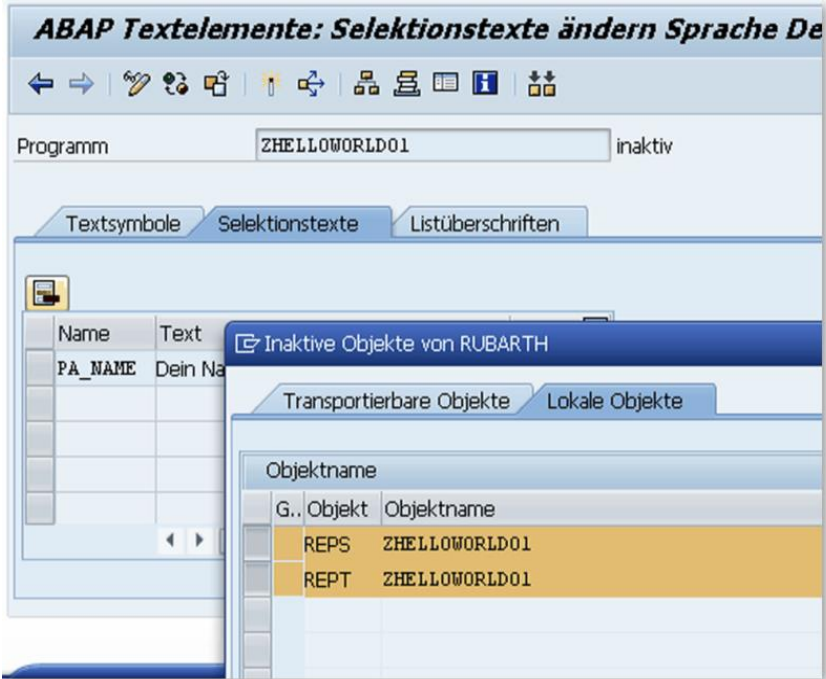


Abbildung 2.3: Liste inaktiver Objekte

Liste inaktiver Objekte



Wenn Sie über etwas mehr Praxis in der ABAP-Entwicklung verfügen, begegnet Ihnen möglicherweise der Fall, in dem auch die Liste inaktiver Objekte angezeigt wird, obwohl Sie dem Anschein nach lediglich ein inaktives Objekt haben. Öffnen Sie die Registerkarte »Transportierbare Objekte«, um diese auf weitere inaktive Objekte zu prüfen. Der *Transport* entwickelter Objekte wird später vorgestellt und erläutert.

Selektionstexte bieten den Vorteil, dass sie in andere Sprachen übersetzt werden können. Wenn Sie bei der Anmeldung eine andere Sprache auswählen, zeigt Ihr Report den Parametertext in dieser Sprache an (vorausgesetzt, der Text wurde in diese Sprache übersetzt).

Selektionstexte und Selektionsbilder



Der Parametertext wird als »Selektionstext« bezeichnet, da er die Datenselektion näher beschreibt. Ein fortgeschrittener Report, der Daten aus einer Datenbank liest, weist folgenden Ablauf auf: Zunächst muss der Benutzer die Selektionsbedingungen angeben (PARAMETERS-Anweisung), dann erfolgt der Datenbankzugriff anhand der Selektionsbedingungen und schließlich werden dem Benutzer die ausgewählten Daten angezeigt.

Die Anzeige der Selektionstexte wird als »Selektionsbild« bezeichnet. Das Selektionsbild ist der internen Bildnummer 1000 zugewiesen; darauf werden wir später noch näher eingehen.

Um die wichtigsten Punkte noch einmal zusammenzufassen:

- ▶ Eine PARAMETERS-Anweisung deklariert eine lokale Variable.
- ▶ Zu Beginn eines Reports zeigt das Selektionsbild ein Eingabefeld an, damit der Benutzer den Wert der Variablen festlegen kann.
- ▶ Das Eingabefeld auf dem Selektionsbild verfügt über einen Textbezeichner, der übersetzt werden kann.

2.2 Verwenden von Bedingungsschaltern

Lassen Sie uns nun die Verwendung einer IF-Anweisung betrachten, anhand der sich bestimmen lässt, ob der Benutzer den Standardwert für einen Parameter geändert hat. Die Verwendung von IF-

B Index

- /h 34
- ABAP Class Builder 92
- ABAP Data Dictionary 55
- ABAP Development Tools 148
- ABAP Editor 8
- ABAP in Eclipse 148
- ABAP Kommentare 12
- ABAP List Viewer 122
- ABAP Objects* 90
- ABAP Workbench 37
- ABAP Workbench-Optionen 39
- ABAP-Anweisung 7
- ABAP-Klasse 90
- ABAP-Schlüsselwort 7
- ALV 122
- Anweisungen für interne Tabellen 64
- Arbeitsbereich einer internen Tabelle 62
- Architekturmodell AS ABAP 61
- AS ABAP 8
- AT LINE-SELECTION 118, 119
- AT SELECTION SCREEN 118
- Aufrufen eines Funktionsbausteins 81
- Ausnahmeklassen 101
- Ausnahmekonzept für Klassen 101
- Ausnahmen eines Funktionsbausteins 79
- Ausnahmeobjekt 103
- Auswahlknöpfe 120
- AUTHORITY CHECK 114
- BAPI 73, 83
- BAPI_FLIGHT_GETLIST 84
- Bedingungsschalter 25
- Beispielanwendung 59
- Benutzungsoberflächen 115
- Bericht 8
- Bildschirm 127
- Breakpoint 33
- BSP 145
- BSP-Seite 145
- Business Server Page 145
- Business-Objekt 83
- CALL FUNCTION 81
- CALL METHOD 98
- CALL SCREEN 128
- CASE 32
- Class Builder 92
- CLIENT SPECIFIC 109
- COLOR 122
- COMMIT WORK 73
- CONSTRUCTOR 100

- CREATE OBJECT 99
- DATA 29
- Data Browser 59
- Data Dictionary 55
- Datenbanktabelle 55
- Datenelement 57
- Datenobjekttyp 28
- Datentyp c 29
- Datentyp D 30
- Datentyp decfloat 30
- Datentyp im DDIC 57
- DDIC 55
- Debugging 33
- DESCRIBE 124
- Detailliste 118
- Dialogtransaktion 127
- Domäne 57, 59
- Drag-and-drop 99
- Dynpro 127
- Dynpro-Ablauflogik 130, 131
- Dynpro-Programm 127
- Eclipse 148
- Eingabevariablen 21
- elementare Datentypen 30
- Elementliste für Dynpro 131
- Entwicklungssystem 43
- Ereignisse 35, 117
- ERM 60
- Erweiterungskonzept 18
- EXIT 32
- Floorplan Manager 147
- Folgedynpro 129
- FORM 68
- FORM-Routine 67
- FORM-Routine aufrufen 69
- FORM-Routinen definieren 68
- FPM 147
- Function Builder 71, 75
- Funktionsbausteine 71
- Funktionsgruppe* 71, 76
- Globale Strukturdefinition 55, 59
- grafische Screen Painter 131
- GROUP 120
- Grundliste 118
- HOTSPOT 121
- HTMLB 146
- ICON_LIST 121
- INCLUDE 126
- INITIALIZATION 118
- Instanzen einer Klasse 90
- Instanzmethode 91
- Intellisense* 13
- interne Tabelle 61
- Interne Tabelle 61
- internes Datenformat 31
- Internet Transaction Server 144
- itab 61
- ITS 144
- Klassenausnahmen 101
- Klassenkonstruktor 100
- Klassenmethode aufrufen 98
- Klassenmethode testen 96
- komplexe Datentypen 51
- Liste 118
- Listenformatierung 121
- Listenpuffer 118
- Listenverarbeitung 117
- Logical Unit of Work 114

- lokale Definition 52
- Lokale Objekte 45
- LOOP AT 62
- LOWER CASE 22
- LUW 114
- Mandant 39
- Meldungen 115
- Menu Painter 136
- Menüs und Drucktasten 136
- MESSAGE 115
- Message-Typ 115
- Modularisierung 67
- Modulpool 127, 128, 137, 138
- MOVE-CORRESPONDING 53
- Muster-Funktion 81, 98
- Nachrichten 115
- Nachrichtenklasse 116
- Namensgleichheit für Dynpro-Elemente 133
- Native SQL 105, 106
- Object Navigator 37
- Objektauswahl 46
- Objektliste 37, 38
- objektorientierte Entwicklung 90
- Objektorientierung 90
- öffentliche Methode 91
- OK_CODE 141, 142
- Open-SQL 105, 107
- PAI 131
- Paket 41, 42, 43
- PARAMETERS 21
- PBO 131
- Performance bei
 - Datenbankzugriffen 113
- Pretty Printer 40
- private Methode 91
- PROCESS AFTER INPUT 131
- PROCESS BEFORE OUTPUT 131
- Produktivsystem 43
- Quelltext anzeigen 16
- RADIOBUTTON 120
- RAISE EXCEPTION 102
- RDBMS 105
- Remotefähiger
 - Funktionsbaustein 73, 76
- Report 7
- Reporttransaktion 47
- Repository Browser 38
- RETURN-Parameter 83
- RFM 76, 83, 91
- SAP GUI für Windows 48
- SAP NetWeaver Application Server ABAP 8
- SAPUI5 147
- Schlüsselwortedokumentation 15
- Screen Painter 128
- Screen-Verarbeitung 127
- SELECT 107
- SELECTION-SCREEN 119
- Selektionsbild 119
- Selektionstext 23
- sflight 59, 110
- SKIP TO LINE 121
- Statische Methode 91

- String-Variable 28
- sy-lilli 119
- sy-listi 119
- sy-lsind 119
- SYST 55
- SY-SUBRC 58
- Tabellenschlüssel 64
- TABLE-Anweisung 143
- Table-Control 137
- Table-Control-Assistent 138
- Testen eines
 - Funktionsbausteins 79
- Testsystem 43
- Textpool 27
- Textsymbol 26
- TOP-Include 137, 138
- Transaktionscode 46
- transparente Tabelle 59
- Transportauftrag 41, 42, 43
- TRY/CATCH 102
- TYPE 52
- TYPE TABLE OF 61
- übersetzbare Texte 26
- Unterprogramm 67
- VALUE 29
- Verkapselung 91
- Versionieren 19
- Vorwärtsnavigation 78, 128
- Web Dynpro ABAP 147
- WebClient-UI-Framework 146
- work area 62
- Workprozess 105
- Zugriff auf ein Strukturfeld 53
- Zuweisungsoperator 26