

Roland Schwaiger

SAP PRESS

Inkl.
Downloads

Mit Syntax-
Highlighting!

Schrödinger programmiert ABAP

Das etwas andere Fachbuch

- ☛ Von den ersten Schritten bis zum kompletten Programm
- ☛ Topaktuell zu S/4HANA und Fiori: das beste ABAP aller Zeiten

- ☛ Umwerfende Beispiele, fantastisch illustriert

DRITTE AUFLAGE

Rheinwerk
Computing

Liebe(r) Leser(in),

oder Ihr Arbeitgeber

Sie haben gewählt:



ABAP

Wir gratulieren – und bedienen

Sie gern. Wo soll's denn hin?

Ins Hirn? – Gerne doch.

Die Windungen sehen gut aus, da lässt sich Fachwissen unterbringen.

JA, BITTE.
AM LIEBSTEN IM
SCHLAF. ODER
EINFACH MIT 'NEM
TRICHTER...

Apropos unterbringen:

Haben Sie schon mal versucht,
eine Vorratskammer mit ganz
glatten Wänden zu füllen? –
Geht nicht. Nicht ohne Regale
jedenfalls. Bretter, Nischen,
Vorratsdosen – damit geht's.

Was diesen **Trichter** angeht,
da müssen wir leider **passen**.
Aber seien Sie nicht allzu enttäuscht,
denn Sie halten mehr als einen Ersatz in
den Händen:

**Das etwas
andere Fachbuch.**

Mit Nischen, Behältern in allen Größen und belastbaren Fachböden. Wir haben keine Mühe gescheut.

Wir haben einen hervorragenden Ausbilder engagiert.

Ein Experten-Team hat die Wände behauen, den Code eingefärbt und die nötigen Umwege eingebaut, so dass Sie bequem goutieren können.

Wobei **bequem** nicht „passiv“ bedeutet, und **goutieren** nicht „schlafen“, aber das wäre ja ohnehin viel zu schade.

Falls Sie
bei „goutieren“
jetzt an Lauge, dünne
Nudeln denken –
vergessen Sie's.

Spaghetti-Cooking
und Copy-&-Paste-
Programmieren ist
hier nicht drin.

Gestatten:
Schrodingers,
Ihr Lesepartner.

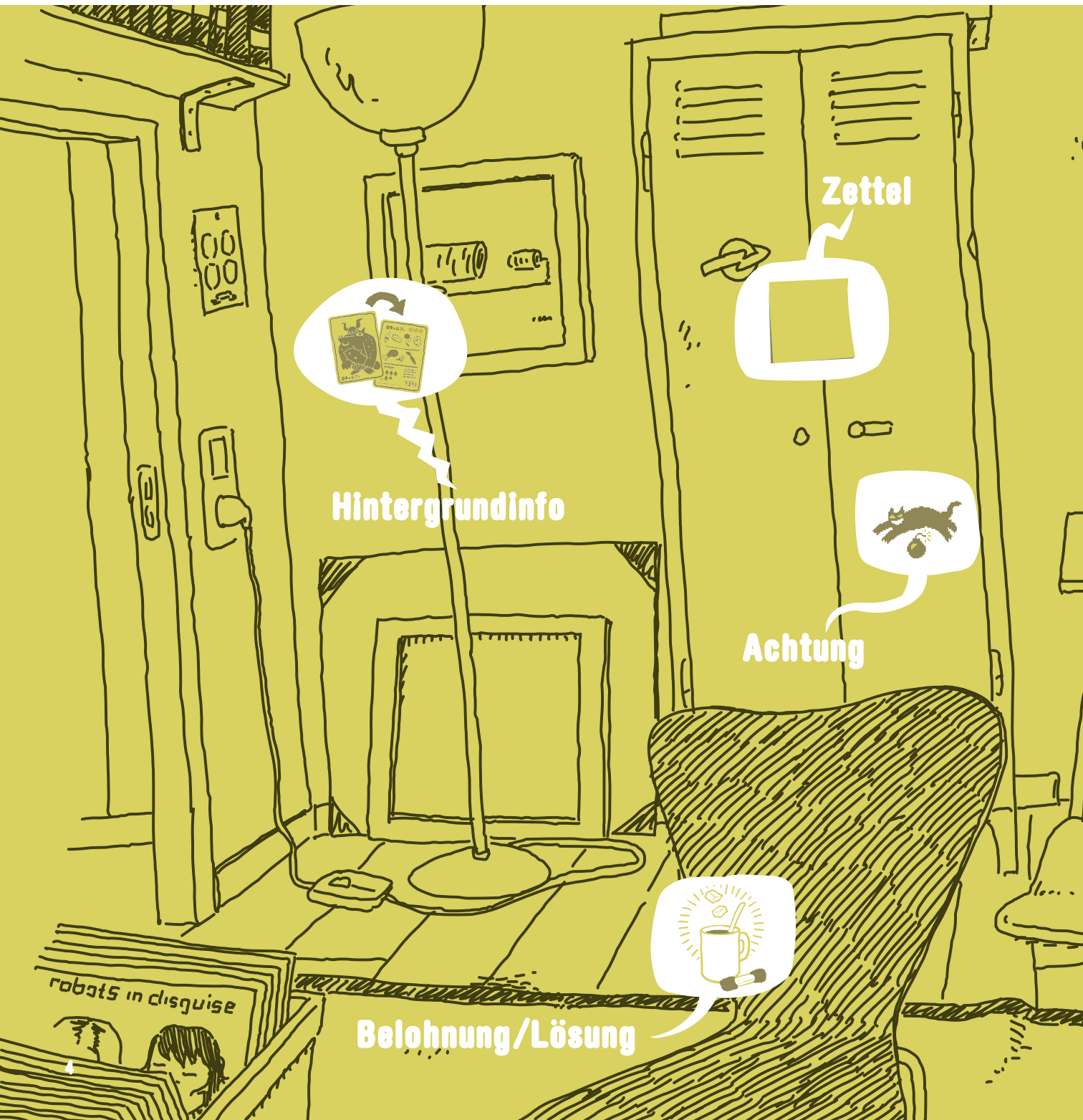


Bitte sehr. **Es ist angerichtet.**

Viel Spaß!

Ihr Verlag

Schrödingers Büro



Die nötige Theorie, viele Hinweise und Tipps



Begriffsdefinition

Ablage

Einfache Aufgabe

Schwierige Aufgabe

Schrödingers Werkstatt



Belohnung/Erledigt!

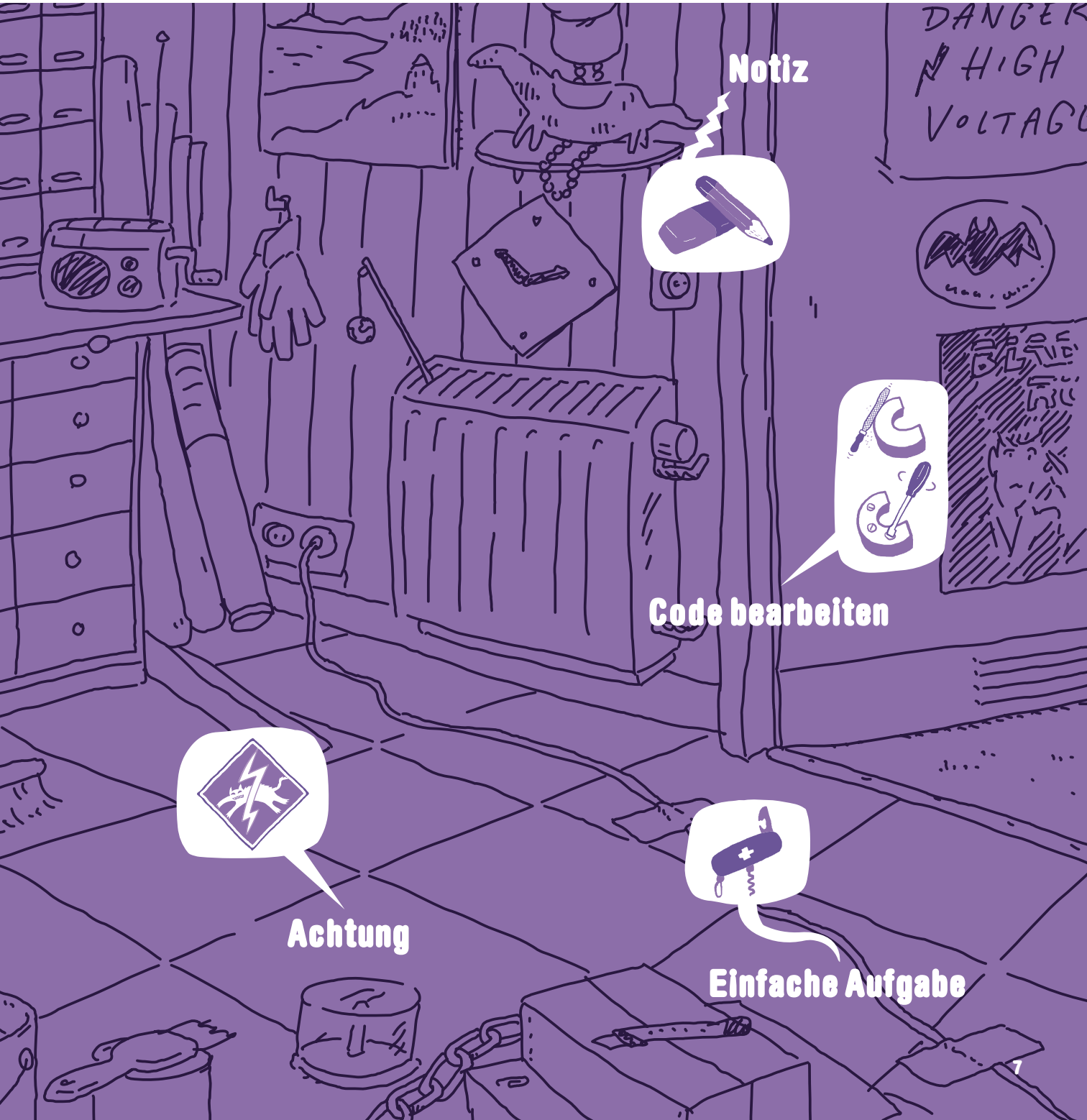


Fehler/Müll



Schwierige Aufgabe

Ummengen von Code, der ergänzt, verbessert
und repariert werden will



Notiz



Code bearbeiten



Achtung



Einfache Aufgabe

Schrödingers Wohnzimmer



Zettel



Belohnung/Lösung



Fehler



Übung

Mit viel Kaffee, Übungen und den verdienten Pausen



Schwierige Aufgabe

Einfache Aufgabe

Achtung



Janina liest auch in ihrer Freizeit und auf Reisen gerne und viel. An Ihren Urlaubsfotos kann man deswegen immer gleich erkennen, wo sie gewesen ist.

Als Kanutin umschifft Almut Stromschnellen mit notfalls nur einem Paddel. Eine Fähigkeit, die sie zur Fachbuchlektorin prädestiniert.



Janina Karrasch und Almut Poll,
LEKTORAT



Janina „Sherlock“ Brönner. Bei der Buchherstellung sind detektivische Kombinationsgabe und Finesse gefragt. Die Kollegen haben sich allerdings das Pfeiferauchen verboten.

Janina Brönner, HERSTELLUNG



Schon zu Schulzeiten zeichnete Leo am liebsten die Bücher voll, von denen er am wenigsten kapierte.

Seit er weiß, dass man das auch gegen Bezahlung machen kann, kapiert er gar nichts mehr.

Leo Leowald lebt und arbeitet in Köln als freiberuflicher Illustrator. Er veröffentlicht unter anderem in *titanic*, *jungle world* und bei *reprodukt* und zeichnet seit 2004 den Webcomic www.zwarwald.de.



Andreas zweite Leidenschaft neben der Buchgestaltung ist kochen. Wie auch immer: Hauptsache rare – VERY RARE!

Andreas Tetzlaff ist selbstständiger Buchgestalter in Köln. Er arbeitet normalerweise für Kunstbuchverlage – dass ausgerechnet ein IT-Fachbuch ihn vor künstlerische Herausforderungen stellt, hätte er sich vorher nicht träumen lassen ...



Mit Blick auf Schrödingers Katze wünschte sich der Verlag zunächst einen Autoren, der ausschließlich verkatert schreibt. Roland konnte uns aber davon überzeugen, dass die Fähigkeit, Bier zu brauen, bei einem Buchprojekt schnöder Trunksucht deutlich überlegen ist. Zudem lernt er Karate und macht ganz gerne eine Kata, was ja immerhin so ähnlich klingt wie Kater – zumindest in Österreich ☺

GESCHRIEBEN VON: Dr. Roland Schwaiger

Einbandgestaltung:
Leo und Andreas (s.o.)
(u. a. unter Verwendung des Fonts
WIMBY von Evert Ypma, Danke Evert!)

Korrekturat:
Osseline Fenner, Troisdorf
Annette Lennartz, Köln

FÜR DIE, DIE ES GENAU WISSEN WOLLEN

Dieses Buch wurde gesetzt aus unzähligen Schriften, Illustrationen und anderen komischen Zeichen, die den Setzer in den Wahnsinn trieben (s. Andreas Tetzlaff).

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-8362-8210-9

© Rheinwerk Verlag GmbH, Bonn 2019
3., korrigierte und erweiterte Auflage 2019



SCHRÖDINGER, DER
SYMPATHISCHE
KATZENPHOBISCHER
VON NEBENAN, IST
DER HELD DIESES
BUCHES.

FÜR IHN ER-
FÜLLT SICH EIN
TRAUM: ER DARF
SEINE QUEST IN
HAUSSCHUHEN
ANTRETEN.

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Alle Rechte vorbehalten, insbesondere das Recht der Übersetzung, des Vortrags, der Reproduktion, der Vervielfältigung auf fotomechanischen oder anderen Wegen und der Speicherung in elektronischen Medien. Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen. Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

INHALTSVERZEICHNIS

Vorwort 24

Kapitel 1: Eine Nudelmaschine für zu Hause

Die SAP, das SAP-System und ABAP

Seite 27

Den Schrödinger hat es mal wieder erwischt: Ein österreichischer Nudelhersteller hat seine alte Firma, Spaghetti Infinito, übernommen. Dabei war das doch bis jetzt so gemütlich in der Spaghetti-Informatik. Man konnte programmieren, wie und womit man wollte – Hauptsache, es lief. Wenn er zukünftig keinen Spaghetticode mehr programmieren darf, kann er sich gleich etwas Neues suchen: Zum Beispiel bei der ABAP-Fabrik, von der er schon so viel gehört hat. Klingt lustig, ABAP. Und wenn die auch Code produzieren ..., vielleicht kann er dann ja doch wieder ein bisschen Spaghetticode programmieren! An seinem ersten Tag lernt er den Schwaiger Roland kennen, seinen ABAP-Ausbilder. Der seine Vorliebe für Spaghetticode sofort entdeckt und sich zunutze macht. Aber anders, als Schrödinger denkt.

Servus Schrödinger!!!	28	Kompilieren/Interpretieren	41
Schrödingers SAP-System	31	ABAP allgemein	42
Die SAP – Das Unternehmen	33	ABAP-Syntax	43
SAP-System	34	Beispiel ABAP	45
Die Erfolgsgeschichte: Vom ABAP zum ABAP	37	Schrödingers ABAP	46

Kapitel 2: Rein in die Küche – raus aus der Küche

Starten mit dem SAP-System

Seite 47

Ein spannender Tag für den Schrödinger: Das erste Mal! Der Schwaiger Roland meint, das tut gar nicht so weh, wie alle sagen, der hat leicht reden. Und fängt gleich mit der Anmeldung am SAP-System an. Äh, wo eigentlich? Ach ja, SAP Logon. Geschafft, und jetzt das Anmeldebild. Wie? Ach ja, SAP GUI. Aber was soll das mit diesen Mandanten? Wir sind doch keine Anwälte, wir sind Entwickler!

Anmelden zum Tauchkurs: Eintauchen in die SAP-Welt	48	Transaktion starten	53
Alles ist Transaktion – Transaktionen als Shortcuts	50	Willkommen Schrödinger! – SAP-Willkommenstext	54
Melde dich an! – SAP-Anmeldung	51	Die Auslage des SAP-Systems – SAP GUI-Aufbau	57

Alles auf einen Blick – SAP Easy Access	59	Hüben und drüben, nichts als Üben – Rolands	
Modus Operandi – Zusätzliche Fenster öffnen	61	Folterkammer oder Übung macht den Meister ...	64
Das ist doch kein Thema	63	Hoppi-Galoppi ... SAP Fiori	66

Kapitel 3: Ciao a tutti! (Hallo Welt!)

Der 20-Minuten-Einstieg in die ABAP-Programmierung

Seite 69

Endlich, das erste ABAP-Programm! Schrödinger war sich bislang gar nicht sicher, ob der Schwaiger Roland überhaupt programmieren kann. Und dann legt er auf einmal richtig los: Pakete anlegen, Datenbankzugriffe, Ausnahmebehandlung, Schlüsselwortdokumentation. Dieses ABAP fängt an, Spaß zu machen, weil es so viel zu entdecken gibt.

Einsteigen und anschnallen!	70	Das ABAP-Einmaleins – Grundlegende Syntax	96
Object Navigator – Die integrierte		Der Durchblicker – Programm entwickeln	98
Entwicklungsumgebung	72	Her mit den Daten – DB-Zugriff	100
Entwicklung organisieren – Systemlandschaft,		Jetzt lese ich	101
Änderungsauftrag	73	Ohne Struktur läuft nichts – Grundlegende	
Ihr neuer Auftrag lautet: Auftragsanlage	75	Programmstruktur	102
Organisationskapsel Paket	77	Alles muss es auch nicht sein – Einfaches	
Vom Entwickler zum Pakettier – Paket anlegen ...	79	Selektionsbild	104
Einen Schritt zurück und bald zwei nach vorne ...	82	Layouten und Daten transportieren –	
Die Infos zum Aufbau einer Datenbanktabelle ...	83	Ein einfaches Dynpro	108
Das zweischichtige Domänenkonzept	84	Dekorieren der Auslage – Dynpro-Layout	111
Technisches Fundament eines Typs –		Ablauflogik ohne Ablaufdatum – Ablauflogik	
Domäne anlegen	86	programmieren	117
Lege die Bedeutung an – Datenelement		Ein Shortcut für den User – Transaktionscode	
anlegen	89	anlegen/ausführen	121
Kühlschrankdesign – DB-Tabelle anlegen	91	Aufbruch in eine neue Welt:	
Daten verarbeiten – Programm anlegen	95	ABAP Development Tools	122

Kapitel 4: AAA – Alles außer ABAP

ABAP-Infrastruktur und -Werkzeuge

Seite 125

Der Schrödinger hat es schon gemerkt: Zum ABAPen reicht es nicht aus, sich die Sprache ABAP selbst reinziehen. Denn wo liegen die Programme, welche Programme gibt es eigentlich, womit werden sie bearbeitet? Welche Speicherbereiche werden genutzt? Wie kommen die Programme zu den Anwendern? Und ungefähr weitere 1.000 Fragen. Aber der Schwaiger Roland ist ja (noch) geduldig.

Warum?	126	Object Navigator – Der Regisseur	141
Die Ablage der Entwicklungsobjekte –		Repository Browser ausnutzen	143
Repository	127	Repository Browser browsen	144
Wer sucht der findet – Suchen mit dem		Synchron oder aus dem Tritt –	
Repository Infosystem	132	Objekt-Browser-Synchronisation	147
Suchen und Finden – Repository Infosystem	134	ABAP Editor – Die Schreibmaschine	149
Geschichtet, aber nicht gefaltet –		Debugger – Der Kammerjäger	152
ABAP-Infrastruktur	135	Debugger entschleunigt	153
Wohin mit dem Anwender –		Verwendungsnachweis –	
Speicherorganisation extern und intern	138	Der Rückwärtssucher	157
Werkzeugkiste – Entwicklungswerkzeuge	140	Die ADT können das auch	160

Kapitel 5: Schräge Typen*

*also jetzt nicht der Schwaiger Roland

Datentypen und -objekte

Seite 165

Definieren und deklarieren: das tägliche Geschäft des ABAP-Programmierers. Das ist ja wie mit den unterschiedlichen Nudeltypen. Die eine Sorte ist lang und dünn, andere sehen aus wie Röhren oder wie Schmetterlinge. Jedoch, und das weiß der Schrödinger natürlich, wird für ein bestimmtes Gericht oder eine besondere Pastasoße ein bestimmter Nudeltyp benötigt. Und so ist es bei der Programmierung eben auch: Für bestimmte Aufgaben werden bestimmte Typen benötigt.

Vorspann	166	Dann elementar global – Globale	
Von 0 auf 100 in wenigen Seiten – Technik der		elementare Typen	180
Datenspeicherung im Hauptspeicher	166	Elementar global con domäne – Domäne	183
Variables Standgas – Datenobjekte	168	Strukturell lokal – Lokale strukturierte Typen	186
Variablen sind zum Angreifen	169	Struktur global – Globale strukturierte Typen	188
Variablendeklaration inline	170	Tabellarisch – typisch/intern –	
Ein typischer Blickwinkel – Datentypen	171	Lokaler Tabellentyp	192
Anders betrachtet – Datentypen	173	Her mit der internen Tabelle	193
Die Vermessung	174	Wie viel Spalten hätten Sie denn gerne –	
Ansprache – Datenobjekte verwenden	175	Zeilentyp	194
Dynamik pur oder Statik nur? – Statisch und		Normal, sortiert oder doch gehasht –	
dynamisch typisierte Variablen	176	Tabellenart	195
Mein lokaler Typ	177	Open the door please – Schlüssel	197
Zuerst mal elementar lokal – Lokale		Heute mal kopflos – Kopfzeile	198
elementare Typen	178	Tabellen für alle – Globaler Tabellentyp	199

Operation: Table – Tabellenoperationen	201	Ändern und Löschen zum Abrunden –	
Bibliothek einräumen	203	Noch mehr Tabellenoperationen	206
Einzeln geht es auch – Einzelverarbeitung von		Eine kleine Gutenachtgeschichte (über CDS)	207
Tabelleneinträgen	205		

Kapitel 6: Si parla ABAP? Echtes Küchenlatein

ABAP-Syntax I

Seite 213

Jetzt geht's ans Eingemachte, Schrödinger, oder sollten wir besser sagen ans Einkochte? Hier lernst du mal das Basisvokabular, so wie damals, als nur Spaghetti gemacht wurden und nicht auch dieses andere Zeugs.

Von Kontrollen und Schleifen	214	Notandor – Logische Ausdrücke	228
Zuweisungen, Operationen, Funktionen	214	Verzweige, falls ... – IF ... THEN ... ELSE	230
Bitte nicht abweisen – Zuweisungen	217	Cassis – CASE ... WHEN ... ENDCASE	232
Was du darfst und was du sollst –		Schleifen	233
Konvertierungsregeln	219	Bedingungslose Schleifen – DO ... ENDDO	233
Ketten ohne Perlen – Zeichenketten-Funktionen ...	221	Krasses Projekt für Hardcore-ABAPer	234
Von Segmenten und Offsetten –		Bedingte Schleife – Fußgesteuert	236
Zeichenketten-Operationen	225	Bedingte Schleife – Kopfgesteuert	237
Ohne Rechnung, ohne mich – Arithmetische		Jetzt aber mal systematisch – Systemfelder	238
Operationen	227	Großbaustelle Rezeptprogramm	240

Kapitel 7: Kleine Häppchen sind leichter wiederverdaut („Schluss mit Spaghetti!“)

ABAP-Syntax II

Seite 243

Schwaiger wirkt wild und entschlossen. Er hat endgültig genug vom Spaghetti-Coding, das Schrödinger fabriziert. „Heute mache ich aus dir einen Wiederverwender“, hallt die letzte Schwaiger-Drohung durch die Hallen, und Schrödinger versucht, sich mit Wehmut von seiner letzten Spaghetti zu trennen ...

Motivation durch Demotivieren	244	Ereignisse: Rekapitulation mit Selbstreflexion	247
Den richtigen Zeitpunkt finden –		Zur richtigen Zeit an irgendeinem Fleck?	249
Ereignisblöcke	245	Meine schönsten Ereignisse –	
Ereignisreiche Programme –		Rahmenprogramm	250
Ereignisschlüsselwörter	246	Dynpros mit Modul – PBO, PAI, HOV	252

Module, ganz kurz – Dynpro-Programmierung	253	Sichtbar oder nicht – Lokale Überdeckung	262
Aber nun mit Schnittstelle – Unterprogramme	254	Aufruf bitte – Unterprogramm-Aufruf	263
Definiere zuerst! Unterprogramm-Definition	255	Totalumbau	264
Übergeben und Übernehmen – Schnittstellenparameter für das Unterprogramm	256	Globale Wiederverwendung – Funktionsbausteine	268
Lesen von DB mit Kapsel – DB-Zugriff in Unterprogramm	258	Funktionsgruppe	270
Ein Typ muss es sein – Schnittstelle typisieren	259	Leg dir eine Funktionsgruppe an	273
Bitte mit Typ – Typisieren	260	Es ist so weit, global bereit – Funktionsbaustein	274
Warum in die Ferne schweifen? – Lokale Variablen	261	Her mit dem Rezept	276
		Haaalloooooo – Funktionsbaustein-Aufruf	279
		Dein Finale	281

Kapitel 8: Schrödinger zeigt Klasse

ABAP Objects

Seite 283

Schrödinger ist heute nicht gut aufgelegt, weil der Schwaiger Roland mit der Objektorientierung daherkommt. Es funktioniert doch alles bestens mit dem „prozeduralen ABAP“ (so nennt das der Schwaiger Roland). Was hat der bloß? Na okay, ein paar Mal haben sich schon Fehler eingeschlichen, die schwer zu finden waren. Mit der Objektorientierung soll ja alles besser werden mit der Ordnung. Und wo mehr Ordnung ist, ist alles leichter zu finden – sogar die Fehler. Und außerdem kann man mit ABAP Objects anscheinend tolle GUIs realisieren. Nach dem ersten Schockerlebnis ist Schrödinger also gleich überzeugt.

Motivation zur OO	284	Mit Methode – ran an den Code	316
Begriffe aus der Realität	285	Die Schnittstelle der Methode	320
Holistische Sichtweise	287	Ganz spezielle Methoden	323
Modellierung	288	Methoden mit Fülle	325
Erst denken, dann lenken: Einfache UML als Analysewerkzeug zur Visualisierung von Klassen/Objekten	292	Selbstreferenz	328
Klasse Kaffee(-vollautomat) auf dem Papier	294	Statisches Attribut	329
Ohne meine Kollaborateure bin ich nichts	296	Methodenaufruf	330
Klasse oder doch Objekt	298	Laufzeit sequenziert	332
Klasse Lokal	300	Methoden für den Wasserbehälter	334
Dein kleines Objektistan	303	Von lokal nach global	336
Kaffee für zwei	306	Going global – SE24	337
Datenablage – Attribute	308	Attribute und Methoden	338
Wasserstand und Pause	310	Kaffeebohnen im Behälter	342
Klasse Attribute, oder was?	311	Senden und Empfangen	346
Privat verkalkt/Private Attribute	314	Sender mit Sendungsbedürfnis	347
Ab jetzt mit Methode – Schrödinger frischt auf	315	Nächster Zubereitungsschritt: der Empfänger	348
		Objektorientiertes Verkuppeln	349
		Kaffee s'il vous plaît	351

Kapitel 9: Erben oder nicht, das ist hier die Frage: Pasta di Erbe

Vererbung

Seite 353

Hoffentlich ist der Schrödinger heute in der passenden Stimmung. Eine Vertiefung der Objektorientierung, puh, da wird er schwitzen. „Was, da geht noch mehr? Jetzt hab ich mich gerade erst vom ersten OO-Schock erholt!“ Schrödinger wird wohl erkennen, dass Spaghetti, Ravioli, Tortiglioni etc. alles Nudeln sind, oder?

Motivation	354	Redefinieren – Polymorphie von Methoden	371
Vererben und Erben von Klassen	357	Redefinieren als globale Herausforderung	374
Vererbung lokal	360	Statische und dynamische Typen von Klassen	
Meister der Pyramiden	362	mit ein wenig Casting	380
Globale Vererbung: Ein praktisches Beispiel		Festigung der Einsichten	382
im SAP-Standard	364	Down-Cast	385
Was wird wie und wo vererbt	367	Abstrakte Klassen	388

Kapitel 10: Keine Details bitte

Der Blick durch Interfaces

Seite 393

„Man kann auch noch einen Schritt weiter gehen und die Implementierung von Methoden von deren Schnittstellendefinition trennen“, verkündet Roland und legt eine Kommunikationspause ein. Schrödinger steht da, der Mund ist offen, und er ist sich noch nicht der Tragweite der Aussage bewusst.

Schizophrenie?	394	Von Suchmaschinen und Tilden	403
Lokal zum Einsteigen	397	Hast du gerufen?	404
Schnittig und definiert	398	Die vielen Gestalten der Methoden	406
Volles Risiko – alles oder nichts	399	Globalisierung mit positiven Auswirkungen	409
Klasse Zuordnung	400	Global klasse Implementierung	411
Klasse Definition	401	Singleton-Party	413
Klasse Implementierung	402		

Kapitel 11: Das Kapitel für Feiglinge

Ausnahmen

Seite 417

„Fehlerbehandlung ist für Feiglinge! Das ist ja so, als ob ich beim Nudelkochen die Nudeln koste, ob sie al dente sind oder nicht!“ Schrödinger hält nichts von Ausnahmebehandlungen. Bis zum ersten Zwischenfall: als seine Freundin sich über klebrige und zerkochte Nudeln beschwerte.

Ausnahmsweise behandelt	418	Ausnahmen melden (mit Nummern)	430
Eine kleine Geschichte der Ausnahmen	420	... und behandeln (nach Nummern)	431
Noch mal langsam: Mit SY-SUBRC.		Ausnahmslos Objekte	433
Ausnahmewert wird gesetzt	424	Definieren geht vor	435
... und von dir behandelt	425	Ausnahmen melden (mit Ausnahmeklassen)	437
It's RAISING man, hallelujah.	426	... und Ausnahmen behandeln, aber richtig!	
Ausnahmen melden	426	Mit Ausnahmeklassen.	440
... und behandeln	427	Das ausnahmengespickte Projekt	441
The Catcher in the System	430		

Kapitel 12: Spaghetti-Programmierung

Selektionsbilder, Listen und Nachrichten (GUI I)

Seite 447

Es ist so einfach, eine SAP-Oberfläche zu programmieren: Ein Selektionsbild erstellen, eine Liste ausgeben, vielleicht noch ein paar Nachrichten an den Anwender verschicken – und fertig ist das berühmt-berüchtigte SAP GUI! Das klingt nach Spaghetticode und ist damit ganz nach dem Geschmack von Schrödinger. Zum Einstieg in die GUI-Programmierung ist es der beste Weg, einfach mal ein paar Elemente auf dem Bildschirm auszugeben. Für das richtige Ambiente sollen dabei Klettererdbeeren und Vogerlsalat sorgen ..., sagt der Schwaiger Roland.

Vom kleinen Window-Farmer bis zum Groß-		Mehr, Mehrere, Mehreres (ohne Rauschen)	470
GUI-Besitzer: Jeder will ein Selektionsbild!	448	Zur Anwendung gebracht ... fast	474
Selektionsbild für Beschränkte mit Liste	450	Graue Theorie: Eingabehilfe, Prüftabelle	476
PARAMETERS: Das kleinere der		Entfessele den Künstler in dir:	
Selektionsbild-Ungeheuer	452	Screen-Gestaltung	478
Typ und Vorschlag	455	Endlich Texte!	482
Hey, Checker!	457	Wie im Hamsterrad	484
Rund wie ein Radiobutton	459	Ereignisreich geht's weiter	490
„Du darfst“ war gestern, heute ist „Du musst“	461	Tagesschau ... also eine Nachrichtensendung	493
Abflug in den Keller	462	L – Li – Lis – List – Liste	497
Aber satt war er noch immer nicht:		Keine Beichte notwendig:	
SELECT-OPTIONS	465	Interaktion mit einer Liste	504

Kapitel 13: Lasagne aufs Auge

Dynpro-Programmierung (GUI II)

Seite 507

„Wow, ein dynamisches Programm!“ Das gefällt Schrödinger sofort, dieses Dynpro. Dynamik ist einfach klasse, außer natürlich, es geht um Sport. Mit der Ablauflogik, Elementen zur Gestaltung der Darstellung und Eigenschaften, die man selbst programmieren kann: Super! Das ist schon richtig modular. (Ja, so langsam gewöhnt sich Schrödinger an die Feinheiten im ABAP-Vokabular.) Und es sind auch schon Schichten erkennbar, die Bausteine zur Dynpro-Programmierung sehen fast schon aus wie eine geschichtete MVC-Architektur. Und Schrödinger liebt Schichten, vor allem in Form von Lasagne!

Dynamisch programmieren	508	Die Gestaltung eines eigenen Menüs	531
Wer schreit hier so? Rahmenprogramme, die Dynpros rufen!	512	Kannst du mit dem Druck umgehen?	534
Dreieinigkeit	515	FFFFF	537
Alles ist im Fluss und manchmal geht es im Kreis	517	Weck den Künstler in dir	539
Wohin soll ich mich wenden?	521	Dein Dynpro	547
Wo Module bestimmen	522	Sammle Elemente	549
Mit welchem Titel darf ich dich ansprechen?	525	Wohin mit den Daten?	551
Über der Fläche steht die Oberfläche	528	Ablauflogik Reloaded	553
		Dynpro rufen mal anders	561

Kapitel 14: Ravioli

Web-Dynpro-ABAP-Programmierung (GUI III)

Seite 565

Heute wartet ein besonderer Leckerbissen auf Schrödinger: Web Dynpro ABAP. Webanwendungen erstellen ohne HTML-Kenntnisse. Das klingt gut, findet Schrödinger. Ist aber für den ABAP-Programmierer eher ein i-Tüpfelchen. Kann man gut online lesen. Also: Schau auf der Bonus-Seite unter <http://www.sap-press.de/4741>

**Materialien zum
Herunterladen:**
<http://www.sap-press.de/4741>

Kapitel 15: Wie Phönix aus der Asche

SAPUI5 oder wie baue ich eine moderne Webanwendung (aus Sicht von SAP ;-))

Seite 567

Schrödinger hat was Neues entdeckt: Webanwendungen mit HTML5, JavaScript, JQuery und CSS3. Wow, und das alles fürs SAP-System! Na, das kann ja lustig werden.

SAP Ju-Ai-Feif	568	Destinationen muss man haben	583
Der kleinstmögliche SAPUI5-Werkzeugkoffer	571	Was ist Ihre Destination, Herr Schödinger?	584
Hello OData-World!	575	Eine Architektur für SAPUI5-Anwendungen	586
Bei Aufruf Service (und dessen Analyse)	578	Wir generieren uns eine Anwendung	587
Jetzt aber: Daten anzeigen lassen	581		

Kapitel 16: Raus aus meiner Küche!

Berechtigungen

Seite 593

„Hör mal, Schwaiger Roland, zu viele Köche verderben doch bekanntlich die Nudelsoße. Gibt es Möglichkeiten in ABAP, dass nicht jeder alles darf? Also dass ich bestimmten Anwendern manche Aktionen verbieten oder erlauben darf? Wenigstens einschränken?“ Der Schwaiger Roland lächelt nur weise.

Berechtigungsrundumschlag – Überblick		Der Berechtigungs-Selbstchecker –	
Berechtigungen	594	S_TCODE prüfen	602
Am Anfang steht das Objekt mit Klasse –		Experimente mit AUTHORITY-CHECK	604
Berechtigungsobjekt	597	Warum mag mich keiner – SU53	605
Die Details zur Berechtigung	600	Finger weg von meinem View!	607



Kapitel 17: Vorratskammer einrichten mit ziemlich viel Schnickschnack

DB-Tabellen erstellen

Seite 611

Was geschieht, wenn Anwender alle Daten im GUI eingegeben haben? Wohin gehen die Daten dann? Die müssen doch gespeichert werden? Denn die Anwender werden wahrscheinlich nicht immer wieder die gleichen Daten eingeben wollen – außer wenn sie an einem schwachen Kurzzeitgedächtnis leiden. Das würde sich Schrödinger öfter wünschen, das mit dem Kurzzeitgedächtnis, denn dann könnten sich die Anwender nicht an seine Programmierfehler erinnern ..., aber der Schwaiger Roland erklärt ihm dann doch lieber, wie er eine SAP-Datenbank-tabelle erstellt.

Freiland-Datenhaltung – Daten persistieren	612	Ich will auch anders suchen – Sekundärindex	638
Warum einfach, wenn es mit Schnittstelle geht –		Ändern oder nicht, was geht –	
Die Datenbankschnittstelle	614	Erweiterungskategorie	641
Transparente Tabelle en Detail	617	Definieren und Realisieren –	
Spalten und der Rest – Tabellenfelder	620	Datenbankobjekt	643
Ohne Technik keine Tabelle – Technische		Mein erster Eintrag – Datenbanktabellen-	
Einstellungen	625	Einträge erzeugen	644
Mehr als eine Tabelle	630	Artenvielfalt im Dictionary –	
Welcher Wert ist möglich – Fremdschlüssel	633	Weitere Tabellenarten	646

Kapitel 18: Vorratskammer plündern

DB-Tabellen auslesen

Seite 647

Daten rein, Daten raus, und das möglichst einfach: So wünscht sich das der Schrödinger. Da kann ihm der Schwaiger Roland helfen. Und alleine die Begriffe „Open SQL“ und „ANSI SQL“ klingen wie Musik in seinen Ohren. Und er kann dann Daten aus einer oder sogar mehreren Datenbanktabellen lesen. Manchmal wünscht sich Schrödinger, er könnte den Schwaiger Roland in der Vorratskammer einsperren. Nur über Nacht.

Erster Takt – SQL	648	Open SQL grundiert – Basisbefehlssatz	
Zweiter Takt – SQL	648	aus Open SQL	651
Eingelagert und geplündert – Datenpufferung	649	Verklausuliert – SELECT-Klausel	653
Datenmanipulator nativ – DML mit Native SQL ...	650	Wohin damit – INTO target	656
Einfach definieren – Datendefinition mit		Woher – FROM source	657
dem ABAP Dictionary	650	Aber bitte nicht alles – WHERE-Klausel	659
Alles recht und schön – Berechtigungen		Da geht noch mehr – Weitere Klauseln	660
und Konsistenzprüfungen	651	Und jetzt alles kombiniert – SELECT-Beispiel	661

Mehr als eine Tabelle ist auch okay – Views	666	Mit Views programmieren	682
Schritt für Schritt zum View mit relationalen Operatoren	670	Weil du so brav warst – Programmieren mit Views	684
Da will ich auch nicht alles – Selektionsbedingungen	679	Die CDS-Beziehungskiste: Von Joins und Assoziationen	686
Zusammenfassung in Bildern – View-Definition ...	680	Mit CDS-Views programmieren	688

Kapitel 19: Vorratskammer in Schuss halten

DB-Daten pflegen

Seite 691

Daten pflegen – den Ausdruck mag Schrödinger. Und denkt an die Kräutersammlung auf seinem Fensterbrett. Da müssen manchmal auch neue Kräuter eingefügt werden, mal muss er das eine durch das andere austauschen. Und für das Löschen hat er auch schon eine leckere Soßenidee. Datenpflegeservice Schrödinger!

Lesen ist Silber, Schreiben ist Gold – DELETE, UPDATE, MODIFY und INSERT	692	Und noch eins und noch eins und ... – INSERT ...	706
Sichere Daten – Transaktionskonzept	692	Massendaten einfügen	708
Von einem Zustand zum nächsten – Datenbank-LUW	694	Ändern muss man auch ab und zu – UPDATE	711
Bist du konsistent?	697	Massendaten	712
Aktionen bündeln – SAP-LUW	698	Weg damit – DELETE	714
Und jetzt auch noch transaktional – SAP-Transaktion	699	Kombianweisung – MODIFY	717
Datenmanipulator-Entwicklungsrahmen – Das Programm für die Open-SQL-Anweisungen	701	Halt, Sperre! – Sperrkonzept	719
		Die fünf Gebote der performanten DB-Programmierung	720

Kapitel 20: Mit Schirm, Charme – und vielleicht noch einem Melonensorbet

Daten in Dateien, Datenablage ohne DB

Seite 721

„Warst du schon einmal bei einem richtig guten Italiener?“ Schrödinger ahnt nichts Gutes. „Also so einem, wo du bereits am Eingang dem Kellner deine Garderobe abgeben kannst.“ – „Ja, das Prinzip der Garderobe ist mir bekannt, wieso?“ – „Na ja, wäre es nicht klasse, wenn die Anwender in ABAP auch Dateien einlesen oder herunterladen könnten – also an der Garderobe abgeben könnten?“ Au Backe, das ist ja mal eine miese Metapher. Aber es ist vielleicht trotzdem wichtig zu wissen, wie man mit Daten umgeht, wenn man kein Datenbanksystem hat.

Daten ohne Datenbank	722	Download now!	732
GUI-Loads – Upload und Download	722	Einen hamma noch – Upload	735
Frontend Services	724	Daten auf dem Applikationsserver	737
Pfad ermitteln	727	OPEN House – OPEN DATASET	738
Download now, zumindest vorbereitet –		DatenTRANSFER – Daten schreiben	739
Download vorbereiten	729	READ DATASET – Daten lesen	741
Auch nett für Datenwiederverwendung –			
Clipboard	730		

Kapitel 21: Täglich wechselnde Speisekarten

Dynamische Programmierung

Seite 743

Morgens frische Lebensmittel auf dem Markt einkaufen, mittags daraus eine Speisekarte zaubern. Jeden Tag neu. Schrödinger bekommt Hunger, aber darum kann es dem Schwaiger Roland jetzt ja nicht gehen. „Was genau ...“ – „Wenn du erst während der Laufzeit eines Programms die Informationen erhältst, die für die Ausführung des Programms nötig sind, zum Beispiel den Namen einer Datenbanktabelle, dann ...“ – „Okay, okay, ein neues Konzept. Sag mir, wie es heißt, aber dann geht es in die Kantine.“

Dynamische Programmierung hat nichts		Dynamisches Feld	753
mit Beweglichkeit zu tun	744	Dynamischer Typ	753
Feldsymbole	744	Dynamische Komponente	754
Datenrefs	746	Dynamische Bedingung	754
RTTS = RTTI + RTTC	748	Dynamisches Unterprogramm	756
Überschrift finden	749	Programmerzeugung im Hauptspeicher	756
Dynamische Tokens	751	Programmerzeugung im Repository	757

Bildnachweis	760	Index	761
---------------------------	------------	--------------------	------------

Vorwort

Lieber Leser,
es hat einen Grund, warum du dieses Buch in deinen Händen hältst: Du solltest es kennenlernen! Ich kann dir versprechen, dass Schrödinger seit der ersten Auflage unheimlich dazugelernt hat. Die dritte Auflage ist noch mehr vollgepackt mit SAP-Programmierwissen. ABAP, SAPUI5, CDS und SAP HANA werden deine neuen Freunde werden.

Der Schrödinger ist bei mir eingezogen und lebt nun seit geraumer Zeit – seit Anfang 2011 – als achtes Familienmitglied bei uns. Nicht nur im Haus, sondern auch in unseren Köpfen. Die tägliche Frage: „Was wird Schrödinger heute wieder anstellen?“ beschäftigt uns schon beim Frühstück.

Dabei hat alles so harmlos angefangen. Stefan Proksch, mein geschätzter Lektor der Web-Dynpro-Bücher, fragte mich, ob ich denn nicht Lust auf eine Bekanntschaft mit Schrödinger hätte. Ein umgängliches Kerlchen, mit Ecken und Kanten, der bereits die C++-Schule bei Dieter Bär besuchen durfte. Das konnte ich mir natürlich nicht entgehen lassen. Naiv und enthusiastisch, wie ich bin.

Am Anfang war der Sturm und Drang, so wie in jeder frischen Beziehung, und der Schrödinger und ich hatten eine intensive Phase des „Aneinander Gewöhnens“.

Ja, ja, da waren wir noch jung. Mittlerweile ist daraus eine solide Beziehung entstanden, die darauf beruht, sich aufeinander verlassen zu können. Dazu hat sicher auch unsere Lektorin Almut Poll beigetragen. Sie hat unsere Beziehung gestärkt, geglättet, gehegt und gepflegt.

Aber es geht ja nicht um mich, sondern um dich – oh, du geschätzter Wegbegleiter. Schrödinger und ich werden dich mit in die ABAP-Fabrik nehmen, und du bekommst eine ausführliche Werkstour. Apropos Werkstour, diese beginnt mit einem Hinweisschild am Empfang. Der Text darauf lautet so:



Was darfst du nicht erwarten: eine verstaubte Betriebsführung. Also, wenn du bereits auf die „üblichen“ Betriebsführungen konditioniert bist, dann darfst du leider nicht eintreten.

Keine Chance. Der Sicherheitsdienst wird dich sofort enttarnen. Also **bitte umdrehen, sofort.**

Du bist noch da. Das freut uns aber sehr. Du hast die erste Hürde elegant genommen.

Was darfst du also erwarten: eine Einführung in die essenziellen Handgriffe im SAP-System, einen ABAP-Schnelldurchlauf, mit dem sogar ein Formel-1-Pilot überfordert wäre, dann alles außer ABAP und alles mit ABAP. Also Typen, Variablen, Syntax, Abzweigungen, äh, Verzeihung, Verzweigungen, (Schuld-)Zuweisungen nach Operationen, gepflegte Ausdrücke und Lichterketten. Routinierte Ereignisse vor klasse Objekten. Einige Bilder für den User, und das sogar im Web. Datenbank und Tabellen, schreiben und lesen, und nicht die Dynamik zu vergessen, die ABAP wandlungsfähig machen. Dazu noch einiges an Betriebssport und -küche und ein gelegentlicher Besuch im Kreativzentrum.

Viel Spaß!

Ups, da hätte ich ja fast etwas vergessen. Mein Dankeschön, und das betrifft dieses Mal nicht nur die üblichen Verdächtigen. Und wie immer weiß ich nicht, wie ich anfangen soll ... vielleicht eine kleine Geschichte zuerst: Stefan Proksch, mein Web-Dynpro-Lektor und Schreiblehrer, hat ihn mir persönlich vorgestellt und war sehr geheimnisvoll dabei. Offiziell durfte ich ihn – den Schrödinger – gar nicht kennen und musste sogar unter Folter seine Existenz leugnen. Das hab ich dem Stefan sogar unterschreiben müssen. Aber leugnen und foltern hin oder her, der Schrödinger hat sich bei mir immer mehr eingenistet. Und damit das so richtig nachhaltig war, hat Stefan mir auch unermüdlich immer wieder neue Facetten vom Schrödinger vorgestellt. Gott sei Dank war er geduldig genug mit mir. Der Schrödinger war mir am Anfang nicht geheuer. Und dann hat Stefan auch noch unsere Männer-WG verlassen, und da schauten der Schrödinger und ich mal kurz etwas dumm.

Doch welch Glück: Innerhalb kürzester Zeit ist ein neues WG-Mitglied eingezogen: die Almut Poll. Und da waren wir wieder drei: die Almut, der Schrö und der Roland. Von da an ging's aber hurtig dahin, und die Almut hat den Schrödinger und mich so richtig nach vorne gebracht. Danke Almut, dass du dem ganzen Projekt noch mal richtig Schwung gegeben hast. Deine Einfälle und Anmerkungen haben mir so manches Mal aus der Patsche geholfen. Danke!

Es gibt noch mehr, die mich begleitet haben:

Dr. Gerhard Rodé möchte ich für die informative Mail-Kommunikation zu der Entwicklung von ABAP danken.

Meinem geschätzten Referentenkollegen Dr. Stefan Ehret gebührt Dank für seine hoch qualitativen Schulungsunterlagen und die Erlaubnis zur Verwendung seiner dynamischen SE16.

Martin Schwaiger, mein Bruderherz, hat mir schönen Input für die Kaffeegeschichten geliefert. Danke dafür, und hoffentlich wird unser **Espresso Stout** ein Wintertraum.

Meinem Sohn Nico für die Schrödinger-Zeichnungen und Schrödinger-Ideen („Papa, was macht der Schrödinger heute wieder ...?“) und dafür, dass er Teile des Buches geschrieben hat. ☺

Meiner Tochter Marie für den Schrödinger-Gugelhupf, der wie immer auch dem Schrödinger schmeckte (der hat ihn alleine verputzt).

Meiner Tochter Elisa für die schicken Schrödinger-Kekse.

Dir, Ursula, weiß ich nicht, wie ich danken soll, denn zu viel wäre es, was ich schreiben dürfte. Danke für meine Freiräume und für den ganzen Rest. Und ich verspreche, dieses Jahr kein Buch mehr zu schreiben. ☺

Widmen möchte ich dieses Büchlein meinem Vater, von dem ich lernte, durchzuhalten.

**Herzlichst,
Roland**

—EINS—

Die SAP,
das SAP-System
und ABAP

Eine Nudelmaschine für zu Hause

Den Schrödinger hat es mal wieder erwischt: Ein österreichischer Nudelhersteller hat seine alte Firma, Spaghetti Infinito, übernommen. Dabei war das doch bis jetzt so gemütlich in der Spaghetti-Informatik. Man konnte programmieren, wie und womit man wollte – Hauptsache, es lief. Wenn er zukünftig keinen Spaghetticode mehr programmieren darf, kann er sich gleich etwas Neues suchen: Zum Beispiel bei der ABAP-Fabrik, von der er schon so viel gehört hat. Klingt lustig, ABAP. Und wenn die auch Code produzieren ..., vielleicht kann er dann ja doch wieder ein bisschen Spaghetticode programmieren! An seinem ersten Tag lernt er den Schwaiger Roland kennen, seinen ABAP-Ausbilder. Der seine Vorliebe für Spaghetticode sofort entdeckt und sich zunutze macht. Aber anders, als Schrödinger denkt.

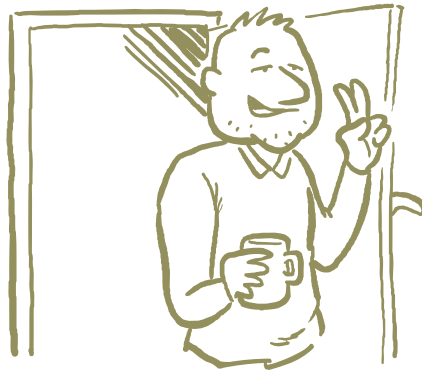
Servus Schrödinger!!!

Gestern hat Schrödingers Freundin den Kaffee über die Tastatur seines neuen Laptops geschüttet. Unglaublich! Und dann auch noch die Übernahme „seiner“ Firma Spaghetti Infinito durch einen österreichischen Nudelhersteller. Noch unglaublicher! Das Leben kann so nicht weitergehen. Veränderungen müssen her.

Zeit, sich etwas Neues zu suchen – „Eine neue Herausforderung“ dachte sich Schrödinger und – gedacht, getan – startete seinen neuen Lebensabschnitt bei der ABAP-Fabrik als ABAP-Entwickler.

Schrödingers erster Tag in der ABAP-Fabrik.

Servus Schrödinger!



Morsche
Roland!



Eines ist klar, Schrödinger, wir haben SAP im Einsatz, das wird etwas komplett anderes als deine bisherigen Spaghetti-Orgien.

Super.
Aktuelle
Program-
mierung.

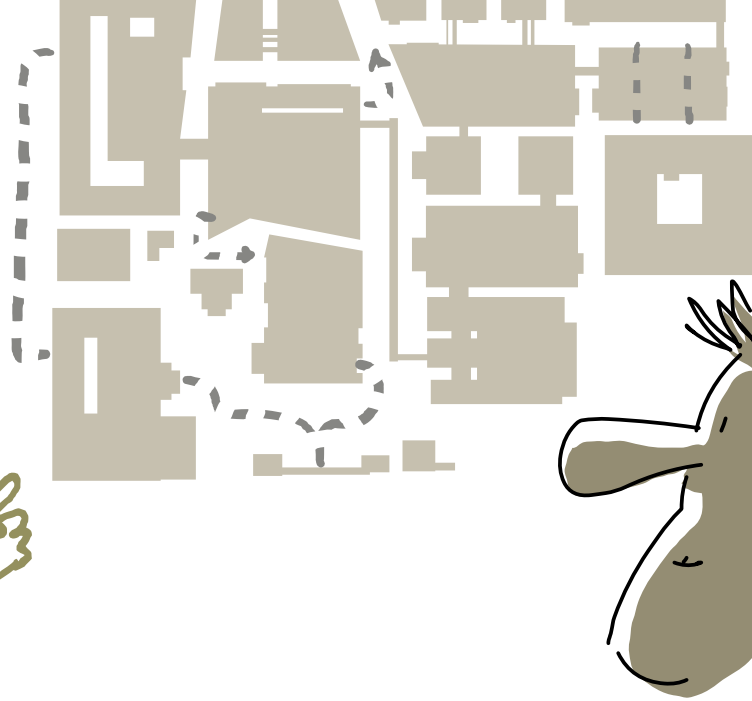


Siches, da hab ich ja schon
einiges von gehört ...



Sandwich-
Anzeige-
Programm

Stimmt! Ich auch, und genau aus diesem Grund werde ich dir mal zeigen, wie das Ding so wirklich funktioniert. Zuerst erklär ich dir mal das Gelände unserer ABAP-Fabrik, damit du dich zurechtfindest, wenn ich mal nicht an deiner Seite bin. Komm mit, wir gehen zum Empfang.



Also marschieren der Schrödinger und der Roland frohen Mutes zum Empfang der ABAP-Fabrik, und wie es sich gehört, wird der Schrödinger dort einem kräftigen und intensiven Syntaxcheck unterzogen.

Schon am Anfang ein Syntaxcheck. Das kann ja heiter werden.



Und dein Gehirn-Scan sieht auch gut aus. Ich erklär dir mal, wie ich mit dir das Firmengelände bereisen werde.



Zuerst marschieren wir zum **Besucherzentrum**, wo du eine schonende Einführung in SAP und ABAP bekommst. Da werde ich dein Gehirn noch nicht durchwaschen. Total motiviert, schreiten wir dann in das **Ausstellungszentrum**, wo du dich mit dem SAP GUI und SAP-Transaktionen anfreundest. Zwischenzeitlich sehen wir natürlich in der **Küche**, im **Sportzentrum** und im **Kreativzentrum** vorbei. Da kannst du dich so richtig austoben mit Spaß, Sport und Spiel – und das **Basteln** kommt auch nicht zu kurz.

Im **Erlebnispark** kannst du mal alle Abteilungen der ABAP-Fabrik durchspielen. Da bist du praktisch als Trainee unterwegs. Mit der soliden Grundausbildung werde ich dich dann durch alle Fachabteilungen lotsen.

Von der **Infrastruktur** (Applikationsserver, Transportwesen) zur **Produktion** (Typen, Schleifen, Verzweigungen, Verarbeitungsblöcke, ABAP OO, SAP GUI und Web), weiter durch die **Sicherheitsabteilung** (Berechtigungen) und dann zur **Lagerhalle** (Datenbank und Daten). Mit der Qualitätssicherung (QS) haben wir nicht gesprochen. Deren Gebäude war zu furchterregend (da sitzt das Lektorat).

*Gott sei Dank habe ich
meine Wanderschuhe mitgebracht.
Das klingt nach einer spannenden
und/oder langen Reise, quasi
ABAP- statt Jakobsweg.*

**Bevor es richtig losgehen
kann, solltest du aber
noch eines
erledigen, lieber**

**Schrödinger. Da bei uns in der
ABAP-Fabrik nicht jeder
Neuling sofort an das SAP-System
darf, solltest du dir DEIN SAP-
System installieren. Und dann
geht es richtig los.**

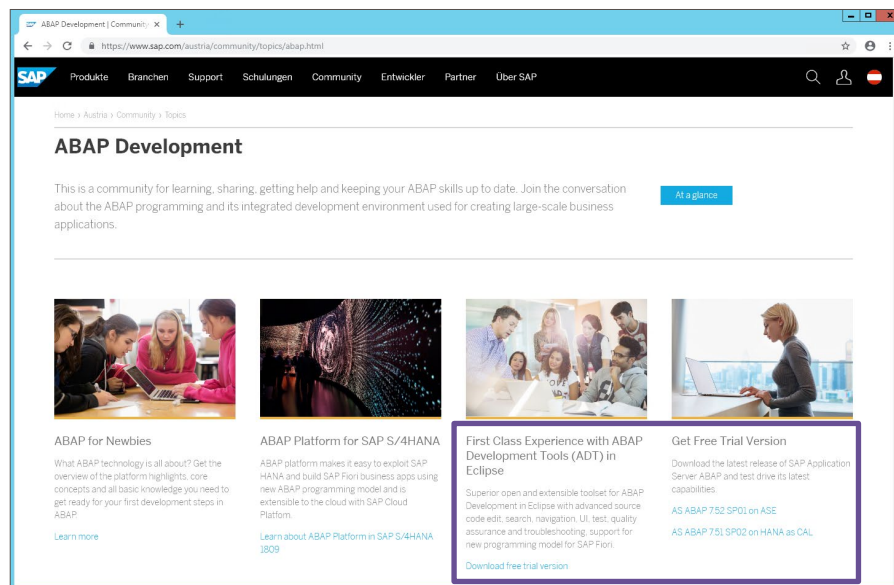
*Okay, kein
Problem. Äh, wie
mach ich das?*

Schrödingers SAP-System

Also du fragst dich, wie du zu einem SAP-System kommst? Ja, ja, früher war das nicht so einfach, aber heutzutage ist das total einfach.

Du brauchst ein Entwicklungswerkzeug und ein SAP-System. Besorge dir deshalb die folgenden Systeme, wenn du schon auf Gratis-Shoppingtour bist:

- ☛ Die **ABAP Development Tools (ADT)**, das sind die Entwicklungswerkzeuge von SAP für ABAP. Sie werden in Eclipse bereitgestellt. Wir brauchen sie erst später, aber installieren kannst du sie ja schon mal.
- ☛ Den **SAP NetWeaver Application Server ABAP (AS ABAP)**, den brauchen wir sofort. Ein kostenloses Trial-System benötigst du natürlich nur dann, wenn du nicht bereits ein SAP-System zur Verfügung hast.



Download your SAP system and the rest.

1. Registriere dich in der **SAP Community**. Rufe dazu die Seite <https://www.sap.com/germany/index.html> auf (gratis und hoch qualitativ).
2. Wähle den Menüeintrag **Community**.
3. Wähle den Bereich **ABAP-Entwicklung**. Du kannst auch nach dem Stichwort suchen, falls du den Link nicht gleich findest.
4. Folge den detaillierten Installationsanweisungen für die ADT und das ABAP-Trial-System AS ABAP 7.51 oder 7.52.

Na, dann los mal mit dem Installieren!

Das ist DEIN Einstieg in die große weite SAP-Welt: dein eigenes SAP-System für zu Hause.

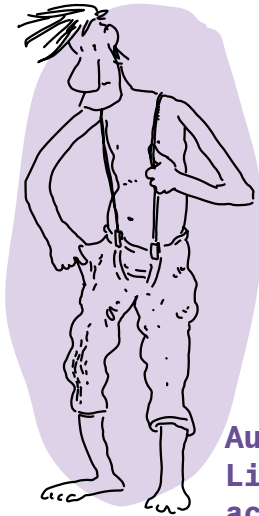
Okay, okay, super, super, super!

[Belohnung]

Weil du so tapfer bis jetzt installiert hast, bekommst du von mir eine Belohnung: die sechs Schrödingers.



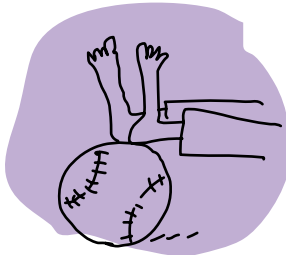
Viel trinken!



**Auf die
Linie
achten...**



**...Zeit für dich
finden!**



**Die Bälle nicht
vergessen!**



**...für genügend
Schlaf sorgen...**



...Obst!

Beherzige die Regeln, und du wirst das Buch bis zum Ende durchhalten.

Die SAP – Das Unternehmen

ABAP kannst du nicht ohne SAP verstehen. Beide sind zu stark miteinander verzahnt. Und wenn ich so von SAP schreibe, höre ich immer „Im SAP oder das SAP oder die SAP“. Was jetzt? Wovon spricht der Mann von der Straße, die Frau vorm Computer, wenn sie von SAP sprechen?

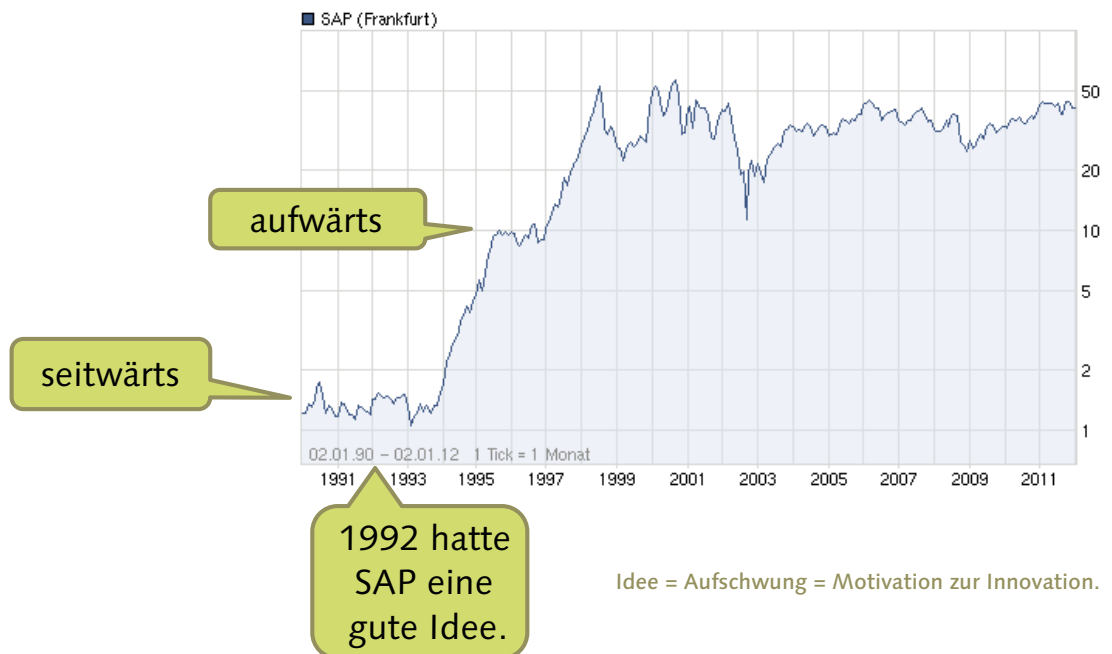
Ja, wovon denn eigentlich. SAP ist doch ein Unternehmen, und SAP ist doch ein System?

Genau, das ist das Bedeutungswirrwarr, das ich entwirren möchte.

Das Unternehmen **SAP Systemanalyse und Programmentwicklung** wurde 1972 von fünf ehemaligen IBM-Mitarbeitern in Weinheim (Deutschland) gegründet: Claus Wellenreuther, Hans-Werner Hector, Klaus Tschira, Dietmar Hopp und Hasso Plattner.

1976 wurde daraus die SAP GmbH, die dann 1977 in Walldorf (Deutschland) ihr Zuhause fand. 1988 erfolgte die Gründung der SAP Aktiengesellschaft Systeme, Anwendungen und Produkte in der Datenverarbeitung und wurde an der Frankfurter und Stuttgarter Börse gelistet.

Einen besonderen Knick und unglaublichen Wertzuwachs erlebte sie zwischen 1993 und 1999. Dieser Anstieg lag an einer Idee, die 1992 entwickelt wurde. Welche Idee das war, wirst du bald lesen.



Die SAP ist mittlerweile der größte europäische und weltweit viertgrößte Softwareproduzent. Sie hat mehr als 91.000 Mitarbeiter an Standorten in weltweit über 75 Ländern (Stand März 2018). Diese arbeiten verteilt über den ganzen Globus und erwirtschaften in 120 Subunternehmen mehr als 23 Mrd. € Umsatz. SAP betreut mehr als 335.000 Kunden, wobei über 12 Mio. Menschen in mehr als 120 Ländern SAP-Produkte verwenden.

Seh schöne Zahlen. Hilft mir das beim ABAPen?

Nicht direkt, jedoch ist besonders erwähnenswert, dass

- ABAP so aussieht, wie es aussieht, weil die SAP eine gute Idee hatte,
- ABAP permanent an aktuelle technische Entwicklungen angepasst wird,
- die ABAP-Gemeinschaft riesengroß ist und du mit ihr über dein ABAP sprechen kannst,
- ABAP international ist.

*Humm, okay, und was ist jetzt mit der „guten Idee“?
Die habe ich auch jeden Tag. Zum Beispiel habe ich gerade die gute Idee, einen Espresso und ein Croissant zu genießen.*

Lies weiter, munter und heiter...

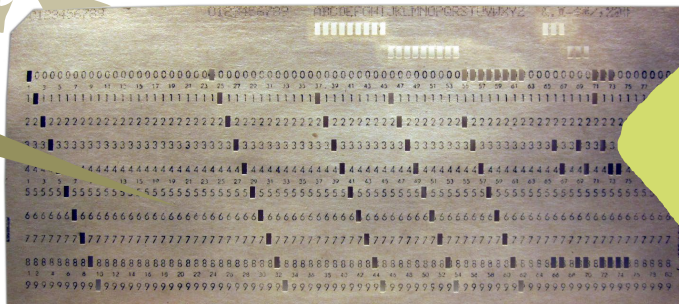
SAP-System

Etwas Softwarearchäologie zur Entwicklung des SAP-Systems. Die erste Version des SAP-Flaggschiffs „Enterprise Software“ war das **System RF** zum Einsatz in der Finanzbuchhaltung (1973).

Wow, gab es damals schon Computer?

Klar, kam direkt nach den programmierbaren Webstühlen. Das System RF ist auch unter dem Namen R/1 bekannt, wobei der Name erst nach der Entwicklung ersonnen wurde. Das R steht nicht für Release, sondern für „Real-time data processing“ und meint, dass, anstatt die Daten mechanisch auf Lochkarten zu speichern, die Eingabe per Bildschirm durchgeführt wird, also in real-time. Ohne Lochkarte.

80 Bytes
Speicher,
wow!



Das sind
Loch-
karten.

Hole in one.

*Man kann Löcher in Karten stanzen
und so mit einem Computer sprechen.
Was war da mit dem papierlosen Büro?*

Da wurden Daten draufgespeichert und mit Lochkartenlesemaschinen in den Speicher des Computers eingelesen. Nur damit du eine Idee der Kartenmenge bekommst: Wird pro Karte eine Bestellung erfasst und pro Tag gehen 35.000 Bestellungen ein, umfasst der Stapel der Tagesbestellungen 35.000 Lochkarten – ziemlich viel Papier. Übrigens wird ein Stapel im Englischen als **Batch** bezeichnet, klingt da was? Was könnte Batch-Job bedeuten?

*Als Lochkartenverantwortliches hat
man vor lauter Löchern das Papier
nicht mehr gesehen.*



[Hintergrundinfo]

Am Anfang der SAP-Entwicklung waren die SAP-Programme auf Lochkarten gespeichert. 1973 trug es sich zu, dass Hasso Plattner stolperte. Na und? Tja, er trug einen Metallkasten mit 2.000 Lochkarten mit Programmen, und der ging leider auf. Zwei Tage sortieren war das Resultat.

Das **R/1** bestand aus einer Menge von „Systemen“, diese wurden für IBM-Rechner mit dem DOS-Betriebssystem entwickelt.

MS-DOS?

Nö, IBM DOS.

1979 wurde R/1 durch **R/2** abgelöst. SAP R/2 ist eine mainframebasierte Business Suite, die sehr erfolgreich in den 1980ern und 1990ern war und immer noch vereinzelt – wie du vielleicht gehört hast – eingesetzt wird. Das Besondere an R/2 war, dass es mehrwährungs- und mehrsprachenfähig ist. Also wurden Beträge nicht automatisch (implizit) einer Währung zugeordnet, sondern explizit (das werde ich dir zeigen, wie das geht). Des Weiteren werden in Abhängigkeit von der **Anmeldesprache** des Anwenders Oberfläche und Texte dargestellt.

*Also wenn jemand sagt, dass er das R/2 kennt,
mit einem leichten Schimmern in den Augen,
dann macht der anscheinend schon ziemlich
lange mit SAP herum.*

Das kannst du laut sagen. Diese Leute können auch meistens noch **Assembler** programmieren. Na ja, und dann hatte SAP eine Idee, die SAP zum globalen Durchbruch

verhalf. Am 6. Juli 1992 wurde das SAP R/3 offiziell für den Markt freigegeben. Das R/3 wurde auf Basis einer sogenannten Client-Server-Architektur umgesetzt. Das 3 im R/3 steht nun nicht für die dritte Version, sondern für die Drei-Schichten-Architektur: Präsentationsschicht – Applikationsschicht – Datenbankschicht.

Der Anwender konnte mit einem Desktop-Client-UI arbeiten. Gleichzeitig wurde eine sogenannte **Basis** eingeführt, die die technische Basis des R/3 darstellte. Die Entwicklung wurde in **Modulen** organisiert, wie zum Beispiel in Modulen für **Finanzen** (FI), **Controlling** (CO), **Materialwirtschaft** (MM), **Vertrieb** (SD), **Produktion** (PP) oder **Personal** (HR). Durch eine einheitliche Benutzeroberfläche wurde die Effizienz beim Arbeiten mit dem System gesteigert. Eine Verteilung auf unterschiedliche Rechner wurde möglich.

Client-Server-Architektur ist ja nun ziemlich bekannt. Und wegen DER Idee ging es so aufwärts?

Die richtige Idee zum richtigen Zeitpunkt und ein Quäntchen Glück = **Erfolg**.

Mit der Client-Server-Architektur hast du zum Beispiel den Vorteil, dass du besser skalieren, also auf Änderungen in den Anforderungen, reagieren kannst.

[Hintergrundinfo]

Ursprünglich war das R/3 als Ergänzung zum R/2 gedacht und für die IBM AS/400 entwickelt worden. Diese Hardware war jedoch mit dem neuen System überfordert. SAP musste auf UNIX-Workstations mit Oracle-Datenbank umstellen, die nach dem Client-Server-Prinzip arbeiteten.

R/3 begann, R/2 abzulösen ...



R/3 machte den Weg frei für SAP. Eine unglaubliche Erfolgsgeschichte begann, und Umsätze und Mitarbeiterzahlen schnellten in die Höhe. Das lieben Anleger, und damit explodierte förmlich der Aktienkurs.

Ah, daraus der Knick im Aktienkurs. Damals reflektierte der Kurs noch den Erfolg, oder war das wie der Fall, hummm?

1996 wurde mit Microsoft gemeinsam die **SAP-Internetstrategie** vorgestellt, die offene Schnittstellen zur Kopplung von Internetanwendungen mit dem System R/3 präsentierte.

Dieser Ansatz wurde **Internet Transaction Server** (ITS) genannt (R/3-Version 3.1 G und H).

Und ich dachte, das Internet und World Wide Web (WWW) haben sich schon früher verbreitet.

Wenn man bedenkt, dass am 6. August 1991 Tim Berners-Lee seine Grundlagen zum **WWW** veröffentlichte, war die Initiative doch etwas zeitversetzt. ☺

Mit **SAP R/3 Enterprise** veränderte sich einiges an der Architektur des Applikationsservers. Aus der Basis wurde der **SAP Web Application Server (WAS)**. Das immer stärker werdende Internet erforderte eine neue Basisarchitektur. Das SAP-System war ab dem Zeitpunkt offen für Internetprotokolle. Die **SAP NetWeaver-Technologie** wurde **2003** präsentiert, ganz dem Zeitgeist entsprechend volle Ausrichtung auf **Prozesse**.

SAP Enterprise Resource Planning (ERP) wurde den Kunden zur Verfügung gestellt mit einem Blick in Richtung **service-orientierter Architektur (SOA)**. **SAP All-in-One** und **SAP Business One** wurden als Lösungen für mittelgroße und kleine Unternehmen eingeführt.

Basierend auf der Grundlage **R/3**, hat sich eine Vielzahl von SAP-Produkten entwickelt, die zur sogenannten **SAP Business Suite** zusammengefasst wurden.

2009 wurde die SAP **Business Suite 7** vorgestellt.

Und dann kam ein großer Sprung:

SAP stellte 2015 **SAP S/4HANA** vor, die nächste Generation ihres Kernprodukts. S für simple, 4 für vierte Generation und HANA für die SAP-eigene In-Memory-Datenbanktechnologie.

*Und was hat das jetzt alles mit
meinem ABAP zu tun?*

Na, alles. Erstens ist das SAP-System die Heimat deines ABAP. Ohne SAP-System kein ABAP. Zweitens steckt in den ganzen Produkten ganz viel **ABAP** drin!

Die Erfolgsgeschichte: Vom ABAP zum ABAP

Ich hätte den Abschnitt auch so anfangen können: vom **Allgemeinen-Berichts-Aufbereitungs-Prozessor** zum **Advanced Business Application Programming**.

Das zweite klingt besser.

Beides gleich teuer. ABAP ist die **Programmiersprache**, wenn es um die Entwicklung von **Geschäftsfunktionalität** (Business Functionality, yeah) im SAP-System geht. **Java** steht auch zur Verfügung, aber das ist eine andere Geschichte.



ABAP gibt es schon seit Anbeginn der Zeit

und hat sich kontinuierlich weiterentwickelt. SAP leistet sich ein eigenes Sprachenteam, das für die Weiterentwicklung von ABAP zuständig ist.

Begonnen hat ABAP als **Makro-Assembler** im R/2 (Allgemeiner-Berichts-Aufbereitungs-Prozessor) mit Spezialisierung auf die **Erstellung von Reports**. Aus dieser Zeit kommt auch die Bezeichnung **Report** für Programm in ABAP. Für Softwarehistoriker und die nächste Geek-Party interessant: Da es um Reports geht, ist **WRITE** der mächtigste Befehl in ABAP.

[Hintergrundinfo]

SAP R/1 und SAP R/2 waren in Assembler programmiert. Auswertungen waren eine Folge von Assembler-Makros und -Anweisungen.

Das Erzeugen der Auswertungen war sehr aufwendig, und sie waren online nicht lauffähig.

ABAP/3 besaß Befehle, die von ihrer Bezeichnung her bereits eine gewisse Ähnlichkeit mit ABAP/4-Anweisungen hatten, wie zum Beispiel **MOVE**, **COMPUTE**, **WRITE** und weitere.

Die ABAP/3-Programme wurden durch einen Präprozessor in Assembler-Makros und -Anweisungen konvertiert. Auch diese Programme waren nur im Batch lauffähig („zweiphasiges“ ABAP).

Später wurden Programme in ABAP/4 entwickelt („einphasiges“ ABAP). Diese Programme waren online und im Batch lauffähig, besaßen jedoch einen eingeschränkten Funktionsumfang.

Es konnten nur Listen und Extrakte erzeugt werden.

Erst ab Release 4.3 konnten in ABAP/4 Transaktionen mit Dynpros und Modul-Pools entwickelt werden.

Mitte der 1980er-Jahre wurde ABAP als **interpretierende Sprache** ausgeprägt, die der Entwicklung von betriebswirtschaftlichen Anwendungen genügte und die Entwicklung von **Dialogprogrammen** zuließ. Für ABAP gilt dasselbe wie zum Beispiel auch für Java: Der Quelltext wird in ein Metaformat, das sogenannte **Load**, übersetzt und von einem plattformabhängigen Interpreter ausgeführt.



R/2 5.0E

Audit Guidelines	Release	Date	Author	Page
	5.0E	04/01/95	AK REV	I-2

4.3 SAP facts

ABAP (Advanced Business Application Programming Language) is a fourth generation programming language developed by SAP. It is intended for use in an SAP environment, but can also be used in other constellations.

ABAP is an **Interpreter** programming language with language elements that contains all capabilities for:

- data selection
- sorting
- print editing
- data modification (INSERT, UPDATE, MODIFY, DELETE)
- data protection (AUTHORITY-CHECK)
- uploading/downloading (PC).

However, ABAP programs can also be compiled and then executed as a load phase from a load library.

The calling up and executing of ABAP programs is not logged. The user, however, has the option to write the corresponding instructions for logging into each ABAP.

ABAP programs which make changes to tables are not logged in all cases, unless this feature was written into the program itself.

The range of functionality for ABAP transactions **ABAP** and **TM38** can be restricted via the following authorization objects: PROGRAM, EDITOR, and SAP-SQL.

Given the authorizations for ABAP transactions (TM38, ABAP) or given rights of access to the datasets, programs can be generated and executed immediately online in the productive environment!

SAP systems are delivered with a great number of



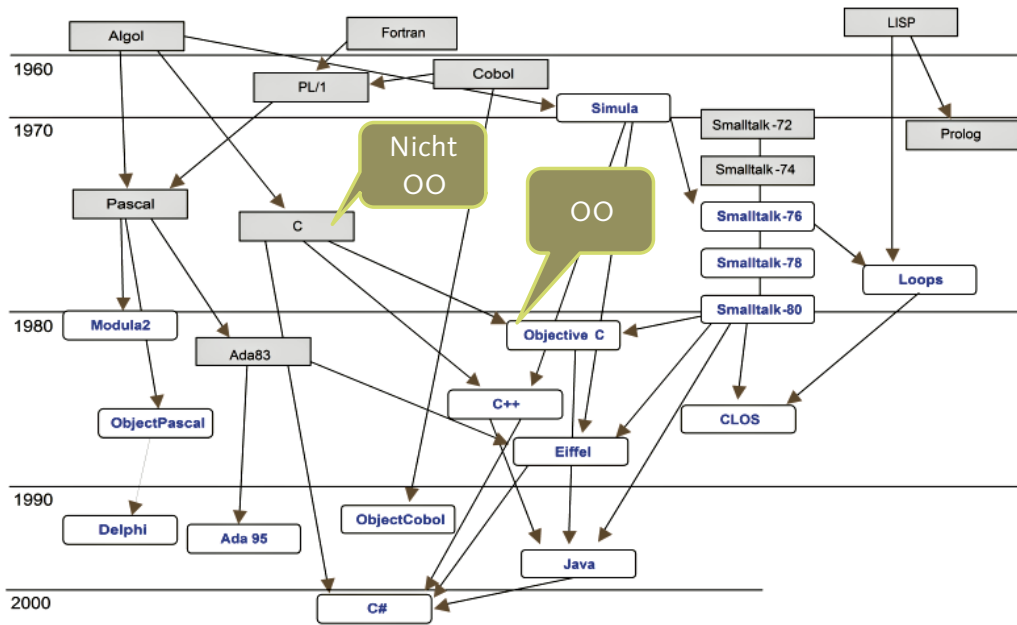
R/2 ABAP.
Bitte diese Abbildung ausschneiden, einrahmen und ins Museum stellen.

Anfang der 1990er wurde R/3 erschaffen und damit auch eine neue Variante von ABAP, genannt: ABAP/4. Das 4 kommt von 4. Generation, was das auch immer bedeuten möge. ABAP wurde zur Basis des gesamten R/3. Einzig der **Systemkern** ist in **C** geschrieben, alle anderen Komponenten, so zum Beispiel auch die Entwicklungsumgebung, sind in ABAP entwickelt. Das bedeutet für dich: Du kannst alle Programme ansehen und durchlesen!

Apfelkern?

Ganz ähnlich. Der Systemkern ist ein zentrales Modul des SAP-Systems, das auszuführende Aufgaben, zum Beispiel die Ausführung eines ABAP-Programms, auf freie Ressourcen verteilt, Speicher verwaltet und andere Basisdienste leistet. Also ist es die Schnittstelle zwischen dem Betriebssystem und den SAP-Anwendungen. In der SAP Management Console siehst du den Kern als eigenen Prozess mit der Bezeichnung disp+work (sieh doch mal nach).

Ende der 1990er-Jahre wurde ABAP endlich die **Objektorientierung** beigebracht, darum hört man statt ABAP/4 auch ABAP oder **ABAP OO** oder **ABAP Objects**. Mit diesen Erweiterungen zogen Kapselung, Vererbung und Polymorphie als objektorientierte Techniken in das SAP-System ein.



Da war ABAP noch ein kleiner Makro-Assembler.

Da war ABAP schon eine Interpretersprache.

Jetzt ist ABAP O(h)O.

Entwicklung der Programmiersprachen aus Wikipedia mit ABAP-Spezialitäten garniert.

[Hintergrundinfo]

UNICODE: ein internationaler Standard, der für Schriftzeichen einen digitalen Code festlegt.
Check out ISO 10646.

Danach kam noch die Einführung von Unicode, die eine Erweiterung und Semantikänderung des ABAP-Sprachumfangs mit sich brachte.
Im nächsten Schritt hat die SAP alles in den **SAP NetWeaver** verpackt und ABAP zur Programmierschnittstelle des **SAP NetWeaver Application Servers ABAP** (manche bevorzugen **AS ABAP** als Bezeichnung) erkoren und damit die **SAP Basis** – zwischenzeitlich auch **SAP Web Application Server** genannt – abgelöst.



Mit dem Einzug der Datenbank SAP HANA hat sich das Programmiermodell von ABAP noch mal erweitert. Es können jetzt Datenbankprozeduren in ABAP implementiert werden und per Code-Pushdown in die Datenbank geschickt werden. Das hat den riesen Vorteil, dass die Ausführungsgeschwindigkeit immens gesteigert werden kann.

Hot News 2018: Die ABAP Community geriet in Aufruhr: ABAP ist in den Wolken, äh ... auf der SAP Cloud Platform (SCP). Es gibt jetzt eine Entwicklungsumgebung in der Cloud, in der gestandene ABAP-Entwickler ihr hart erarbeitetes Wissen für Cloud-Entwicklung einsetzen können.

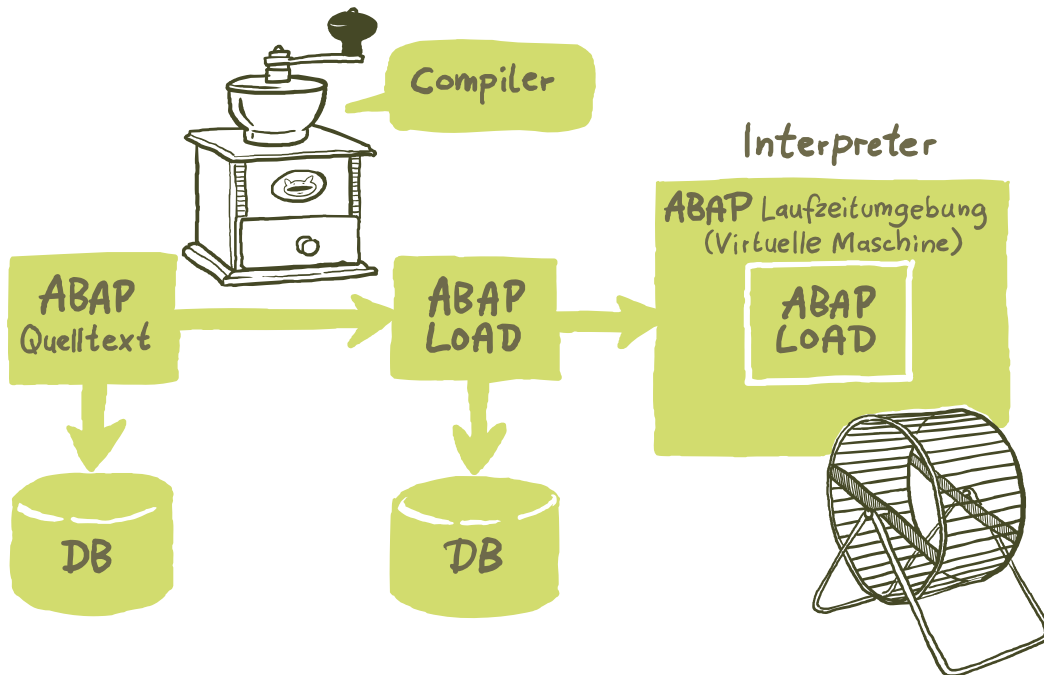
Genug der Geschichte, was ist jetzt mit ABAP, wirst du dir denken. Here you go!

*Ich denk mir nur: Geschichtsunterricht?
Ich will programmieren, Mann!*

„Wehret den Anfängen“, sag ich mit Ovid!

Kompilieren/Interpretieren

Wie du schon gelesen hast, wird **ABAP** (also das ABAP, das du lernen wirst, also das Neueste vom Neuen) von Quelltext in die Bytecode-Zwischenrepräsentation übersetzt. Das Ergebnis nennt SAP das **Load**. Dieses wird zusätzlich zum Quelltext in der Datenbank gespeichert.



Das ABAP-Programm wird in Bytecode kompiliert, der dann von der Laufzeitumgebung interpretiert wird (Compiler/Interpreter).

Das ABAP **Load** wird dann durch die ABAP-**Laufzeitumgebung** ausgeführt.
Bei Bedarf wird das Load neu generiert.

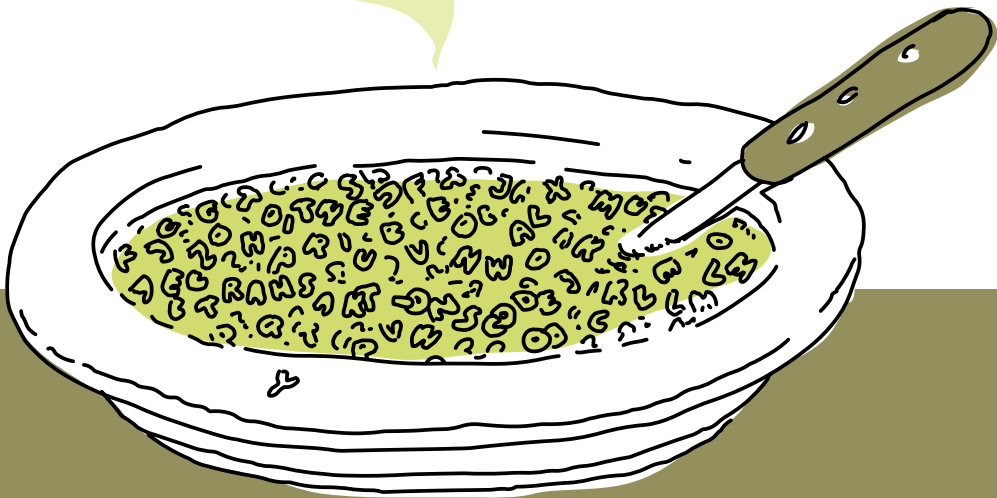
Genug des Rundherums. Was ist mit ABAP?

ABAP allgemein

Einige Schlüsseleigenschaften von ABAP, die du noch lieben lernen wirst.

- ☛ ABAP ist typisiert.
- ☛ ABAP ermöglicht mehrsprachige Anwendungen.
- ☛ ABAP ermöglicht SQL-Zugriffe auf die Datenbank, indem es dir ABAP-Befehle für Datenbankzugriffe zur Verfügung stellt.
- ☛ ABAP ist auch objektorientiert.
- ☛ ABAP ist plattformunabhängig (dafür ist der Interpreter plattformabhängig).
- ☛ ABAP ist aufwärtskompatibel (du kannst immer noch R/2-Sourcen in den neuesten Systemen finden).
- ☛ ABAP ist für die Entwicklung von betriebswirtschaftlichen Anwendungen konzipiert.
- ☛ Eine Zählung hat über 270 Schlüsselwörter in ABAP ergeben.
- ☛ Das Werkzeug zum Bearbeiten von ABAP-Quelltexten ist der **ABAP Editor**.

*Seh nett. Klingt schon mal gut für den Anfang.
Und wie sieht so ein ABAP aus?*



ABAP-Syntax

Die ABAP-Syntax ist nicht schwer zu erlernen. Es ist eher wie Geschichten zu schreiben – Programmgeschichten.



[Begriffsdefinition]

Der grundsätzliche Aufbau von ABAP ist:
ABAP-Schlüsselwort Zusätze und Operanden Punkt

*Ich kann mir das noch nicht ganz vorstellen.
Aber wie ich meinen Roland kenne, kommt da
noch was nachgeliefert.*

Obwohl wir uns noch nicht sehr lange kennen, kennst du mich schon ziemlich gut.

- ☛ Das erste Wort in einem Satz heißt **ABAP-Schlüsselwort**.
- ☛ Jede Anweisung in ABAP muss mit einem **Punkt** beendet werden, daher wird eine Anweisung auch als **Satz** bezeichnet.
- ☛ Wörter in einem Satz werden durch mindestens ein **Leerzeichen** getrennt!

[Achtung]

Ohren auf, Mund zu, Neuronen aktivieren:
Die Leerzeichen sind für den ABAP-Parser
total wichtig. Also nicht vergessen!

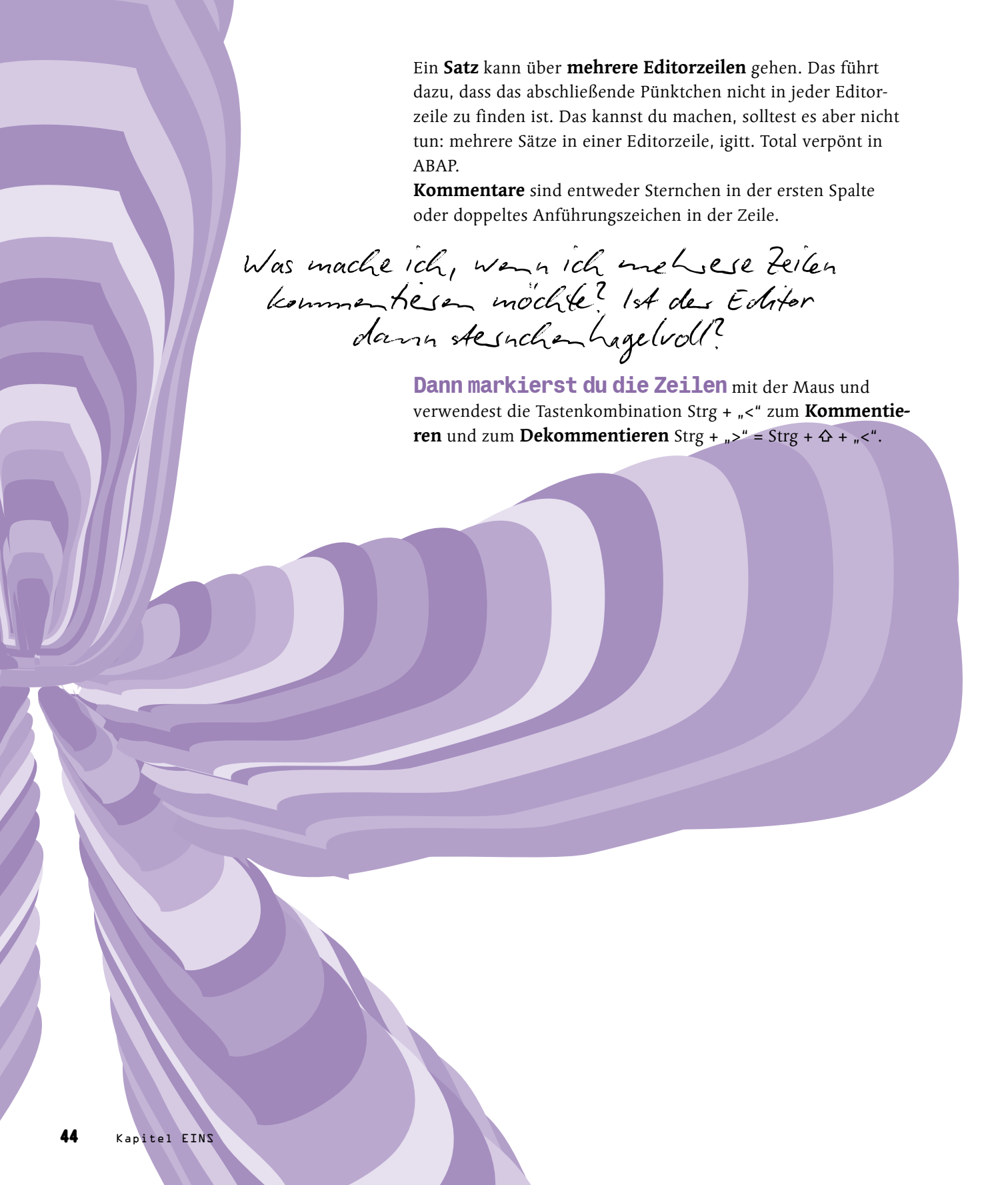


Wenn ein **Satz** aus mehreren **Teilen** besteht, dann muss jeder dieser Teile mit einem **Komma** abgetrennt werden.

Ein **Doppelpunkt** besagt, dass das Schlüsselwort vor dem Doppelpunkt auf alle Teile des Satzes anzuwenden ist. Das entspricht der Grundanforderung: Der Entwickler hat faul zu sein. Je weniger zu tippen ist, desto besser. Übrigens wird die Schlüsselwort-Doppelpunkt-Kombination als **Kettensatz** bezeichnet.

ABAP ist **nicht case sensitive**, das bedeutet, dass **Groß- und Kleinschreibung NICHT relevant** ist. Tipp vom Routinier: Ich stelle gerne den sogenannten **Pretty Printer** im ABAP Editor so ein, dass die Schlüsselwörter groß geschrieben werden. Das nennt man auch den **Poorman's Syntax Check**. Das lernst du dann beim **ABAP Editor**, wie man das einstellt.





Ein **Satz** kann über **mehrere Editorzeilen** gehen. Das führt dazu, dass das abschließende Pünktchen nicht in jeder Editorzeile zu finden ist. Das kannst du machen, solltest es aber nicht tun: mehrere Sätze in einer Editorzeile, igitt. Total verpönt in ABAP.

Kommentare sind entweder Sternchen in der ersten Spalte oder doppeltes Anführungszeichen in der Zeile.

Was mache ich, wenn ich mehrere Zeilen kommentieren möchte? Ist der Editor dann sterchenhagelvoll?

Dann markierst du die Zeilen mit der Maus und verwendest die Tastenkombination Strg + „<“ zum **Kommentieren** und zum **Dekommentieren** Strg + „>“ = Strg + ⇧ + „<“.

Beispiel ABAP



Und schon kommt der erste Test. Sieh dir mal das folgende Beispiel an.
Welche Elemente erkennst du bereits?

REPORT ZSCHROEDINGER01.

- * Ein Stern in der ersten Spalte kommentiert die ganze Zeile
- * Eine Einsprungsmarke für das Laufzeitsystem

START-OF-SELECTION. "Kommentar bis zum Ende der Zeile

- * Ausgabe vom Text auf der Liste, in einer neuen Zeile

WRITE: / 'You can do it'.

Also, welche Teile erkennst du schon?

Ich erkenne Sternchenkommentar, Schlüsselwörter und einen Kettensatz. Und das Programm sieht aus wie der Sternenhimmel.

Gut gemacht!

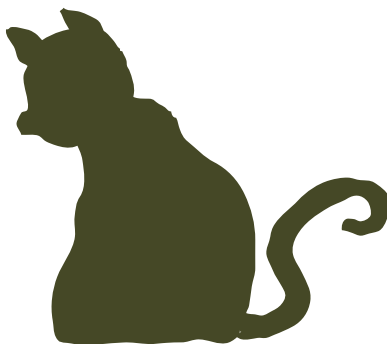


Schrödingers ABAP

Wenn du bis jetzt durchgehalten hast – Hut ab –, wirst du sicher schon festgestellt haben, dass ABAP nicht wie eine Standalone-Programmiersprache zu betrachten ist. ABAP ist immer abhängig vom technischen Kontext, in den es eingebettet ist. Darum hat ABAP auch keine Versionen, sondern immer nur die technische Basis und die Komponenten, die drum herum sind oder darauf aufbauen.

Was ist aus meiner Sicht wichtig für dich als angehenden ABAP-Jedi-Ritter?

- ☞ Du sollst ABAP schätzen, insbesondere sei respektvoll, da das Alter von ABAP beachtlich ist.
- ☞ Lerne das Rundherum um ABAP, mach dich mit der Infrastruktur vertraut.
- ☞ Verinnerliche die Kernkonzepte von ABAP, und mach deine Übungen.
- ☞ Hör auf den Roland, und lies das Buch fertig.
- ☞ Zuerst denken und dann tun.
- ☞ Möge **SAP_ALL** mit dir sein.



—ZWEI—

Starten
mit dem SAP-
System

Rein in die Küche — raus aus der Küche

Ein spannender Tag für den Schrödinger: Das erste Mal!
Der Schwaiger Roland meint, das tut
gar nicht so weh, wie alle sagen, der hat leicht reden.
Und fängt gleich mit der Anmeldung am
SAP-System an. Äh, wo eigentlich? Ach ja, SAP Logon.
Geschafft, und jetzt das Anmeldebild. Wie?
Ach ja, SAP GUI. Aber was soll das mit diesen Mandanten?
Wir sind doch keine Anwälte, wir sind Entwickler!

Anmelden zum Tauchkurs: Eintauchen in die SAP-Welt



Du hast dein SAP-System? Dann findest du auf dem **Desktop** wahrscheinlich das **SAP Logon**-Symbol.



[Hintergrundinfo]

Klugscheißer-Modus: **SAP Logon** ist ein Windows-Programm, das für die **Anmeldung** an SAP-Systemen auf Windows-PCs verwendet wird. Es dient als **Vermittler** zwischen dem SAP-System und der SAP GUI-Benutzeroberfläche.

Und du möchtest dich jetzt sicher **am SAP-System anmelden**. Dazu musst du das **SAP Logon starten**. Zwei Wege stehen dir zur Verfügung:

Du kannst das SAP Logon entweder

- mit einem Doppelklick auf das **Desktop-Symbol** oder
- durch die Auswahl aus der **Programmliste** starten.

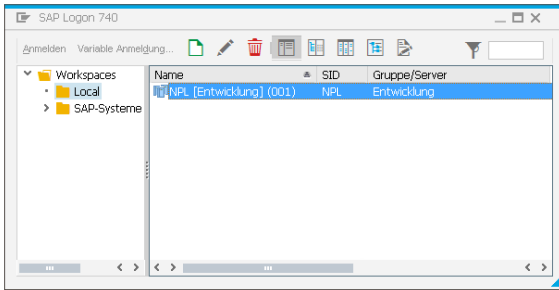


Das SAP Logon als Desktop-Icon.

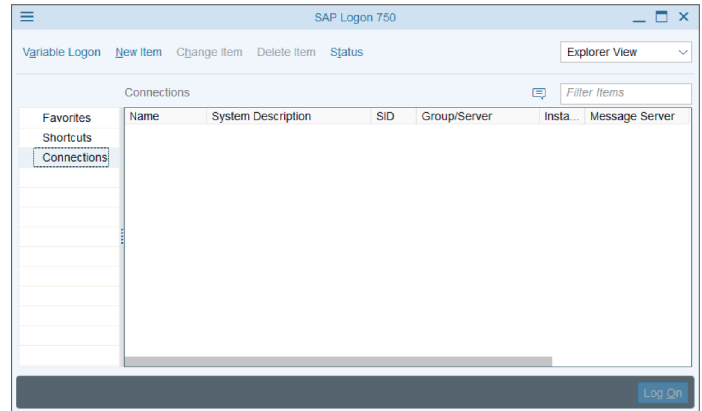
Im SAP Logon – Knoten **Verbindungen** – findest du eine Liste mit **SAP-Systemen**, an denen du dich anmelden kannst – einfach doppelklicken.

Da muss ich natürlich vorher das SAP Logon starten, eh klar!





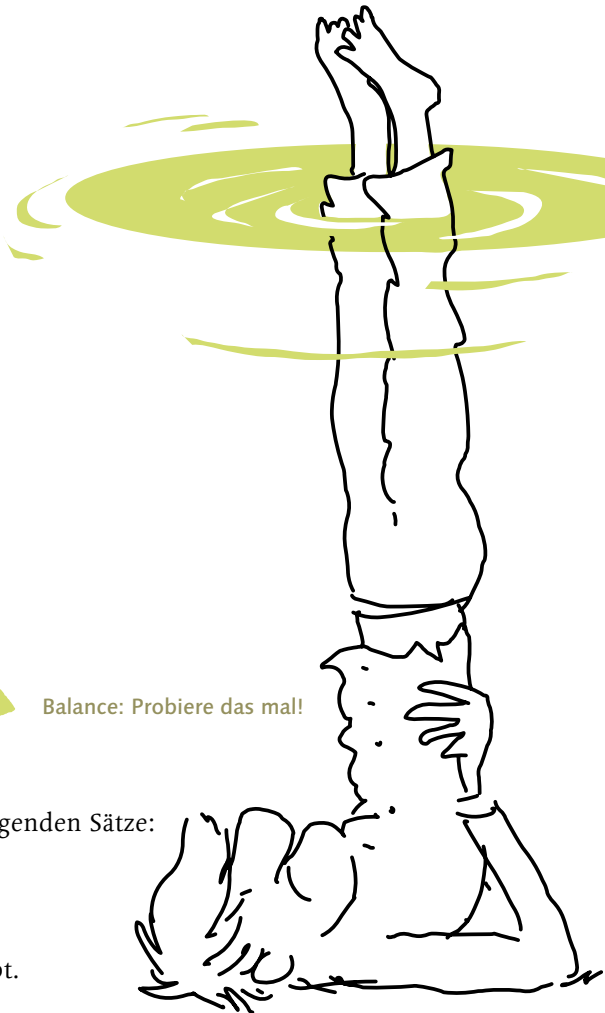
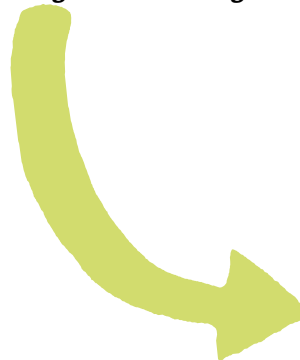
Das SAP Logon in zweifacher Pracht und Herrlichkeit. Einmal für Release 7.40 im SAP Blue Crystal Theme, einmal für Release 7.50 mit SAP Belize Theme. Bei den folgenden Screenshots habe ich mich für Blue Crystal entschieden.



[Hintergrundinfo]

Die Last-/Benutzerverteilung dient dazu, die SAP-Anwender auf den Applikationsserverinstanzen dynamisch zu verteilen. Wenn sich Anwender am System anmelden, werden sie automatisch am Server mit der derzeit besten Performancestatistik und/oder den wenigsten Anwendern angemeldet.

Du kannst die Liste der Systeme natürlich verändern. Aber vorher solltest du dich noch bei einer **Gleichgewichtsübung** entspannen.



Balance: Probiere das mal!

Parallel zur Übung, während du so locker in der Kerze stehst, sprich bitte leise, mit Inbrunst, aber immer wiederkehrend die folgenden Sätze:

- ☛ Ich darf Einträge in der Systemliste hinzufügen.
- ☛ Ich darf Einträge löschen.
- ☛ Ich darf Einträge ändern.
- ☛ Ich liebe meinen Systemadministrator, der mir das alles erlaubt.

Alles ist Transaktion – Transaktionen als Shortcuts

Im Knoten **Verknüpfungen** kannst du SAP-Transaktionen starten, Reports ausführen oder Systembefehle direkt durchführen.



[Begriffsdefinition]

SAP-Transaktionen sind Shortcuts zu Funktionen.

Funktionen? $f(x) = x$ Quadrat?



[Begriffsdefinition]

Funktionen sind Programme.

Und die Reports? Ist ein Report ein Bericht, oder wie?

[Begriffsdefinition]

Report ist eine SAP-Bezeichnung für ein Programm, um gaaaaanz genau zu sein, ein ausführbares Programm. Natürlich sind alle Programme ausführbar, aber dies ist eine spezielle Kategorie von Programm.

Und jetzt kommt's. Pass auf. Ich weihe dich in das große Geheimnis des SAP GUI ein. Nur noch wenige Sekunden, und du gehörst dazu:



[Begriffsdefinition]

In SAP sind Funktionen mit Shortcuts verbunden. SAP bezeichnet diese Abkürzungen als **Transaktionscodes**. Abkürzungen deswegen, weil du nicht über das Menü zu deiner Funktionalität navigieren musst, sondern in einem Schritt die Funktion starten kannst.

Egal, was du im GUI erreichen möchtest: Idealerweise kennst du den Transaktionscode. Eine Auswertung gefällig? Ein allumfassender Bericht? Her mit dem Code, und schon bist du da und dort.

Ich an deiner Stelle würde schon mal – ganz nach alter Sitte – Papier und Bleistift zurechtlegen, damit du die ganzen Transaktionscodes aufschreiben kannst, die dir noch bei der Lektüre begegnen werden.

Was sind Papier und Bleistift – ha?

Melde dich an! – SAP-Anmeldung

Das SAP-System baut seit den 1970er-Jahren auf einer **Client-Server-Architektur** auf – das hast du bestimmt schon gelesen, oder? Also ein Teil (**Client**) fordert Services an, und ein anderer Teil (**Server**) erledigt die Anfragen. Fast wie beim Hausputz. Deine Freundin fragt an, und du läufst herum.

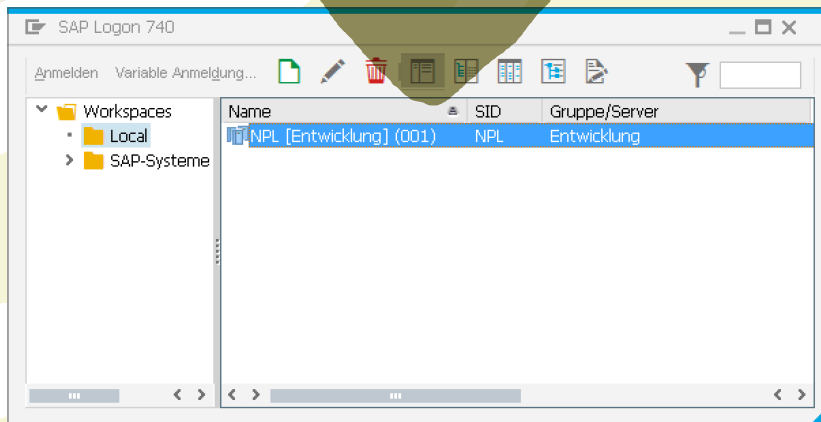
Um mit einem SAP-Server Verbindung aufnehmen zu können, musst du den Client (der wird **SAP GUI** genannt) starten – das funktioniert über das **SAP Logon**, das eine Liste von SAP-Systemen anbietet, an denen du dich anmelden kannst.

Irgendwie hatten wir das schon, oder?

Ja, ich bin noch nach!

Jaja, ich nerve, immer diese Wiederholungen – aber dann bleibt es besser hängen. ☺

„DOPPEL-
KRIECH“



Da willst du hin.



*Gähnen! Jetzt hab ich eine Frage:
Woher kommen die Systeme in der
SAP-Logon-Liste?*

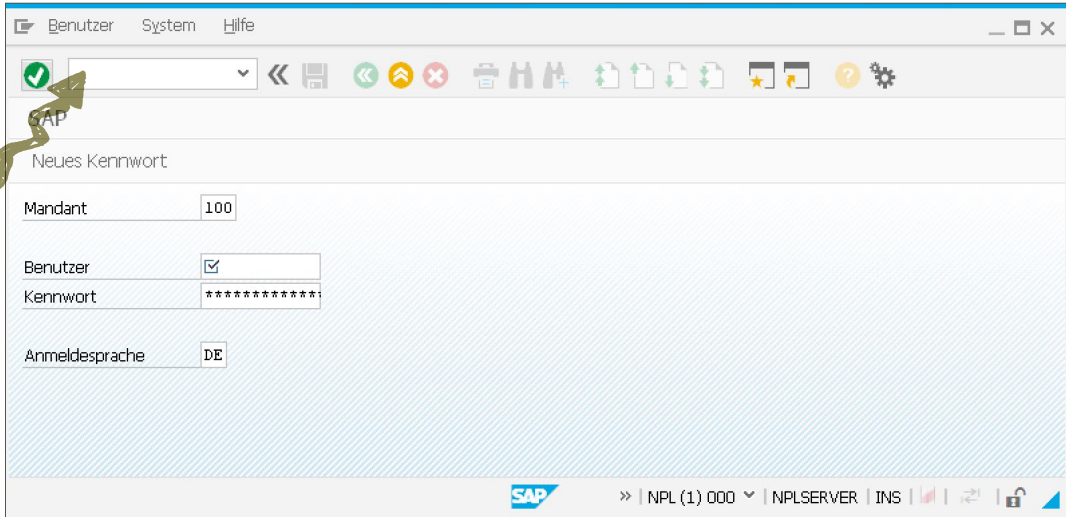
Die muss man von Hand eintragen!
Blätter mal zurück.



Also **Doppelklick** und ... **Schwups**, da ist das SAP GUI in voller Pracht und Blüte.

Einstieg links halten, bitte!

Befehlsfeld für Funktionscodes, Transaktionscodes und andere Kurzbefehle, z. B. „/h“



Zugegebenerweise schaut das jetzt nicht besonders überwältigend aus, das kommt noch.

Einige wichtige Aspekte von SAP kannst du bereits bei der **Anmeldemaske** erkennen. Das SAP-System kennt das **Mandantenkonzept** und sieht dafür ein Eingabefeld vor, damit sich der Anwender – also DU – an einem Mandanten in diesem SAP-System anmelden kann.

Hm, yuhu, und was ist ein Mandant?



[Begriffsdefinition]

Ein **Mandant** in SAP-Terminologie ist eine Gruppierung von rechtlichen, organisatorischen, administrativen und/oder betriebswirtschaftlichen Einheiten mit einem gemeinsamen Zweck.

Wie bitte?

Stell dir einfach einen **Konzern** als Beispiel vor.

Als Nächstes musst du deinen **Benutzer** und dein **Kennwort** angeben. Das erhältst du vom netten SAP-Administrator

oder, da du **Meister deines eigenen Systems** bist, legst du dir einen Benutzer mit Kennwort an. Einen Benutzer legt man mit Transaktion **SU01** an, aber versuche nicht, dir **SU01** zu merken, es ist zu früh, das bringt jetzt noch nichts.

Übrig bleibt dann eigentlich nur mehr das **Sprachen**-Eingabefeld, in das du die Anmeldesprache eingibst. Damit wird bestimmt, in welcher Sprache die Texte im SAP GUI angezeigt werden.

Transaktion starten

Um eine Funktion zu **starten**, musst du den **Transaktionscode** im **Befehlsfeld** eingeben – und zwar so:

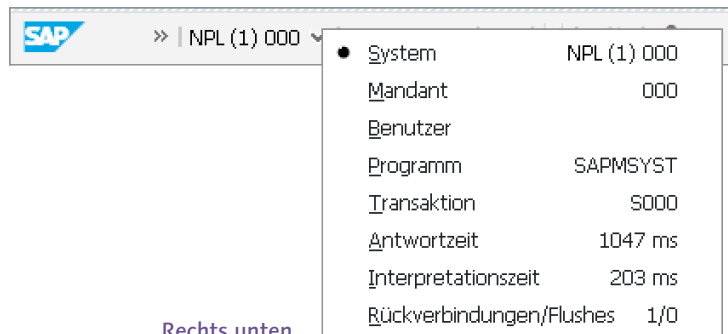
☞ /n<TA-Code>

Dadurch wird die aktuelle Anwendung in diesem Modus beendet und der neue TA-Code gestartet.

☞ /o<TA-Code>

Falls du die aktuelle Anwendung nicht beenden möchtest, verwende **/o**. Damit wird die Anwendung in einem neuen Modus gestartet.

Den Transaktionscode für die **aktuelle Anwendung findest** du rechts unten im Statusfeld, wenn du auf das **Dreieck** im ersten Statusfeld klickst.



The screenshot shows the SAP status bar with a dropdown menu open. The menu displays the following information:

System	NPL (1) 000
Mandant	000
Benutzer	
Programm	SAPMSYST
Transaktion	S000
Antwortzeit	1047 ms
Interpretationszeit	203 ms
Rückverbindungen/Flushes	1/0

Rechts unten.

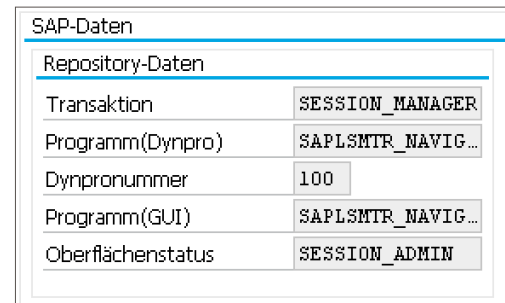
Mithilfe des Menüpfads **System • Status** erhältst du eine Menge von Informationen, unter anderem kannst du den Transaktionscode im Bereich der **Repository-Daten** auslesen.



[Einfache Aufgabe]

Probiere doch mal den Aufruf der Transaktion SU01 im selben/neuen Modus aus. Du kannst im Benutzereingabefeld deinen Benutzernamen eingeben und dann „Anzeigen“ wählen.

Rufe zu einem beliebigen Zeitpunkt über das Menü die Statusanzeige auf (**System • Status**), und ermittle die Transaktion und die Dynpro-Nummer.



The screenshot shows the 'SAP-Daten' window with the 'Repository-Daten' tab selected. The data is as follows:

SAP-Daten	
Repository-Daten	
Transaktion	SESSION_MANAGER
Programm(Dynpro)	SAPLSMTR_NAVIG...
Dynpronummer	100
Programm(GUI)	SAPLSMTR_NAVIG...
Oberflächenstatus	SESSION_ADMIN

Statusinfo.

Willkommen Schrödinger! – SAP-Willkommenstext

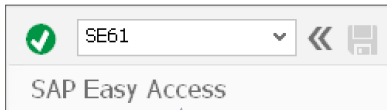
Setz dich mal auf deinen gemütlichen Computerstuhl, und stell dir vor, dass du im Einstiegsbild persönlich begrüßt wirst. Dein eigener Willkommenstext im SAP GUI! Aber bitte nur, wenn du an dein Testsystem denkst.

Natürlich, was denkst du denn?

Also, wie kannst du dich begrüßen?

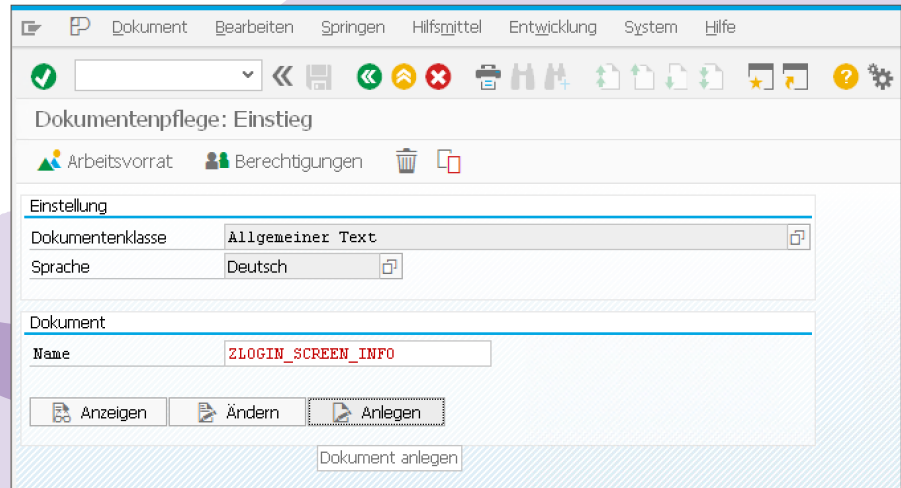
- Am SAP-System anmelden – Easy-Cheesy.
- Transaktionscode **SE61** aufrufen. Einfach **SE61** in das Eingabefeld links oben eingeben und mit der **Weiter/Enter/Return**-Taste bestätigen.

Transaktionscode-Eingabe mit Start im selben Modus.



SE61

SAP Easy Access



Dokumentenpflege: Einstieg

Arbeitsvorrat Berechtigungen

Einstellung

Dokumentenklasse	Allgemeiner Text
Sprache	Deutsch

Dokument

Name	ZLOGIN_SCREEN_INFO
------	--------------------

Anzeigen Ändern Anlegen

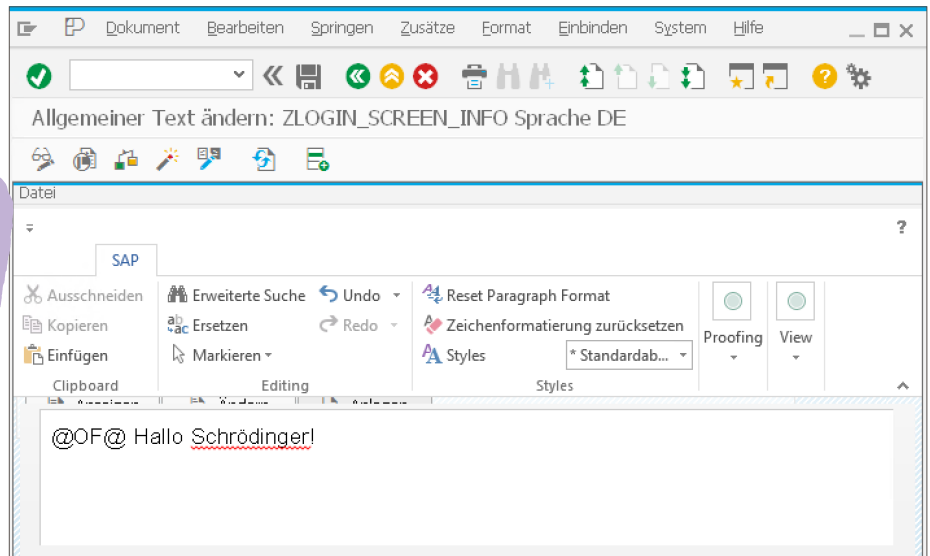
Dokument anlegen

Einstieg in die Dokumentenpflege.

- Beim Einstieg in die Pflege der Texte wählst du für die **Dokumentenklasse Allgemeiner Text**, und dann deine **Sprache**, zum Beispiel **Deutsch**. Dann verwende für das Feld **Name** den Wert **ZLOGIN_SCREEN_INFO**, und klicke auf den **Ändern**-Button. Ist das Dokument nicht vorhanden, klicke auf **Anlegen**. Und, hat es funktioniert?

Sicher!

Bravo, du bist
im Änderungs-
modus.



Dein Willkommenstext.

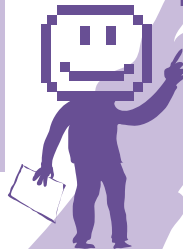
- Jetzt noch den Text im Editor mit einem **netten Icon** davor eintippen, also zum Beispiel **@OF@ Hallo Schrödinger!**



Mein Name ist @OF@.

- Du bist fertig, also mit der Tastenkombination **CTRL + S** **Sichern** und **Aktivieren** (**CTRL + F12**). Deine Änderung ist durchgeführt.

Jetzt kommt der Test:
Noch mal anmelden, und ...



Benutzer System Hilfe

✓ [Dropdown] << [Diskette] << [Up Arrow] [X] [Printer] [Folder] [Folder]

SAP

Neues Kennwort

Mandant	100	Information 💡 Hallo Schrödinger!
Benutzer	[Input Field]	
Kennwort	*****	
Anmeldesprache	DE	

Willkommen.

[Belohnung]

So, du bist drin! Tusch! Leg die Wassermusik auf, und mach eine Pause. Genieße die expressionistische Kraft des Einstiegsbildes.

Tröööt.

Die Auslage des SAP-Systems – SAP GUI-Aufbau

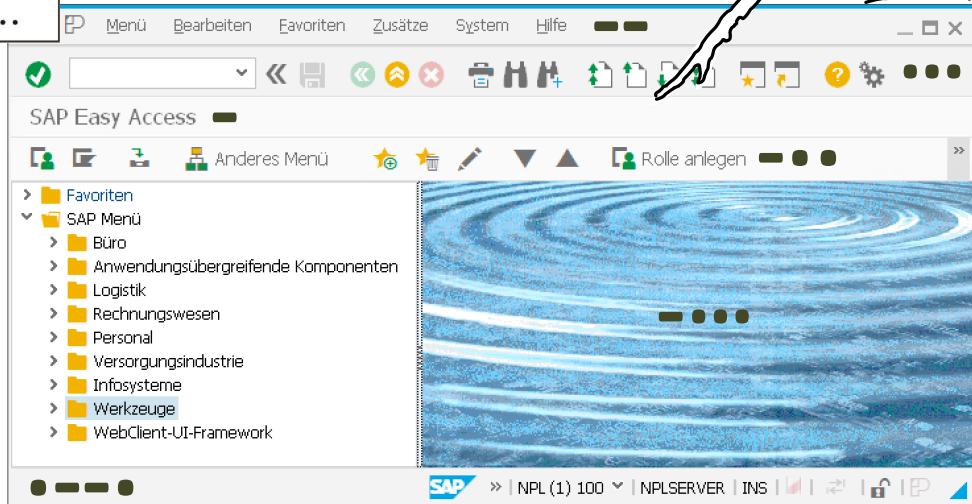
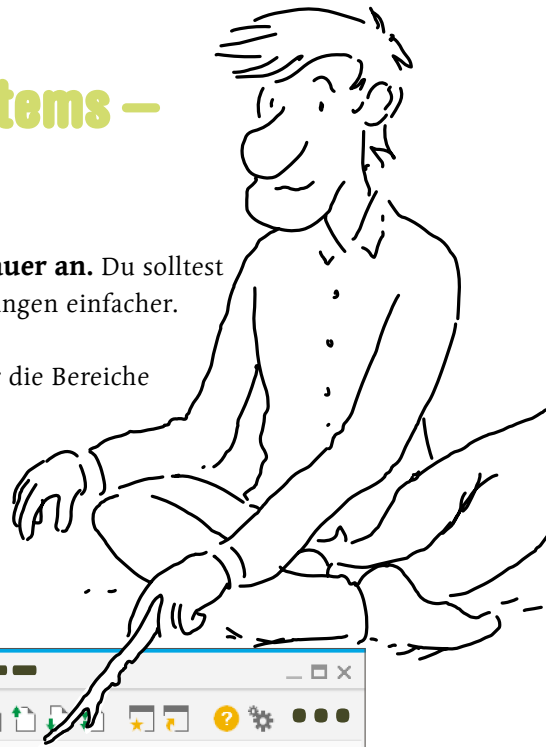
HUMMEL
HUMMEL

A	B	C	D
..
E	F	G	H
.
I	J	K	L
..
M	N	O	P
..
Q	R	S	T
..
U	V	W	X
..
Y		Z	
..

So, nun sehen wir uns das GUI genauer an. Du solltest dich anmelden, das macht die Erläuterungen einfacher. Okay? Auf geht's. Übrigens, als kleines Rätsel habe ich dir die Bereiche im Morsecode gekennzeichnet.

Das hilft dir hoffentlich.

Warum einfach, wenn's auch kompliziert geht?



SAP GUI.

Die unterschiedlichen Bereiche des SAP GUI bieten unterschiedliche Funktionen an.

- (M) Oben ist die **Menüleiste** mit den **Menüeinträgen**. Dabei sind die wichtigsten **Menüs** die, die immer vorhanden sind: **System** und **Hilfe**.
- (S) Dann kommt die **Symbolleiste** (Systemfunktionsleiste). Die wichtigsten **Funktionen** und Funktionstasten in der Symbolleiste sind: **Enter**, **Befehlsfeld** (für die **Transaktionscodes**), **Sichern**, **Zurück**, **Verlassen**, **Abbrechen** ... **Layout anpassen**.
- (T) Gefolgt von der **Titelleiste** (aktuelle SAP-Anwendung).
- (D) **Weiter geht's** mit der **Drucktastenleiste** (Anwendungsleiste), in der besonders wichtige oder oft verwendete Funktionen aus den Menüs hervorgehoben werden.
- (F) Und dann gibt es noch die **Funktionstastenzuordnung**, mit der du Funktionen durch Tastenkombinationen aufrufen kannst.

[Hintergrundinfo]

Hast du lange nach dem F gesucht? Die Funktionstastenzuordnung kann man nicht sehen, außer man öffnet ein Menü oder fährt mit der Maus über eine Drucktaste.

Dann erscheint die F-Tastenzuordnung in Klammern neben dem Funktionsnamen.

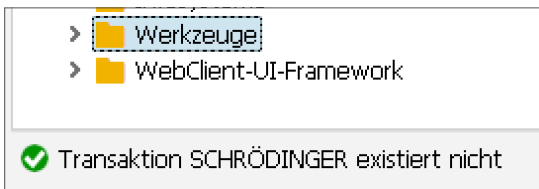


(B) Ein **Dynpro-Bereich**. Dieser ist der Platzhalter für die auszuführenden Programme. Genau hinsehen, dort spielt sich die Party ab.

Tüüt - tüü - tüü - tüü.

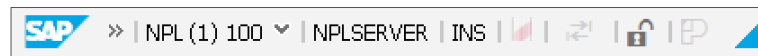
(P) Die **Statusleiste** liefert allgemeine Informationen über das SAP-System und die Transaktion oder Anwendung, mit der du gerade arbeitest.

Die **Systemnachrichten** werden auf der linken Seite der Statusleiste angezeigt, wie zum Beispiel **Informationen**, **Fehler** oder **Warnungen**.



Statusmeldung, bitte schön.

Die **drei Statusfelder auf der rechten Seite** der Statusleiste enthalten einerseits die **Informationen über den Server** und andererseits Statusinformationen über Server und Dateneingabe-Modus. Den Dateneingabe-Modus erreichst du durch die Tasten **Einf**/**Entf** oder durch Anklicken des Dateneingabe-Modus-Feldes.



Info, Info, Info.

Klick doch mal auf das Serverinfo-Feld. Was passiert?

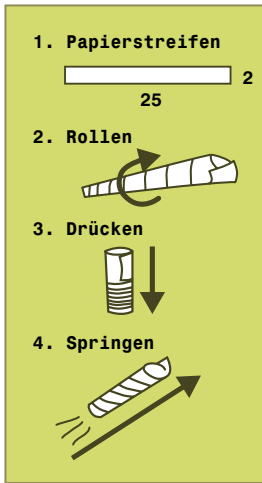
Es poppt!

• System	NPL (1) 100
Mandant	100
Benutzer	SCHWAIGER
Programm	SAPLSMTR_NAVIGATION
Transaktion	SESSION_MANAGER
Antwortzeit	172 ms
Interpretationszeit	93 ms
Rückverbindungen/Flushes	3/2
Flavor-Name	Original

Korrekt, es erscheint ein Drop-down mit zusätzlichen Informationen über das System, den Mandanten, den aktuellen Anwender, die Transaktion und das Programm sowie die Kommunikationsstatistiken.



Serverinfos, und noch viel mehr.



[Belohnung/Lösung]

Nun ist es Zeit für eine **Bastelpause**, probier mal das Springpapier.



Springpapier.

Alles auf einen Blick – SAP Easy Access

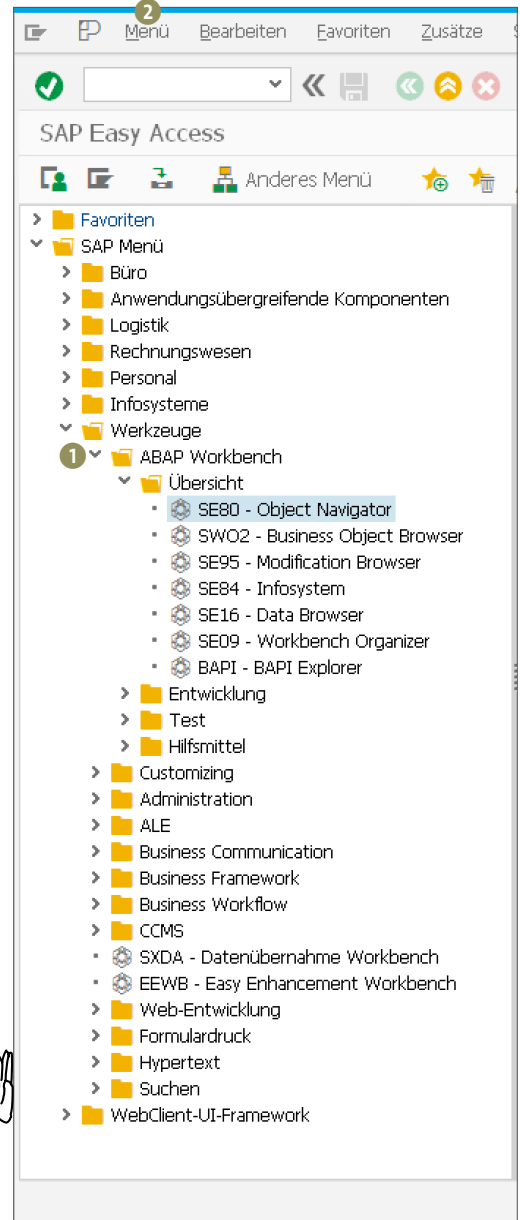
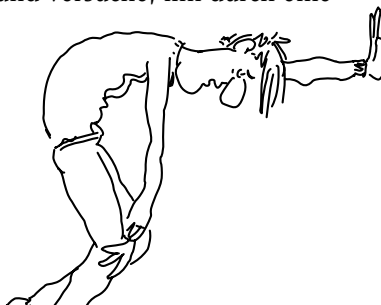
Nachdem du nun schon einiges in SAP ausprobiert hast, wird es wieder Zeit für etwas Theorie. Hast du im SAP GUI das SAP-Menü auf der linken Seite gesehen? Das nennt die SAP **SAP Easy Access** und ist dein **Benutzermenü**, das für dich automatisch erscheint. Schon aufgeklappt?

Das SAP-Menü bietet Zugriffe auf Funktionen, ich könnte jetzt auch **Transaktionen** sagen, die fein säuberlich in einem Baum nach Kategorien organisiert sind.

Herr Ober, das Menü, bitte!

Im **Navigationsbereich** kannst du die Menüs mit dem Drop-down-Pfeil ❶ links neben den Menüeinträgen **expandieren** und **komprimieren**.

Menüs kannst du über den Pfad **Menü • Benutzermenü** bzw. **Menü • SAP-Menü** auswählen ❷. Falls du mehr Funktionen als die angezeigten benötigst, rede einfach mit dem Systemadministrator deines Vertrauens, und versuche, ihn durch eine Schachtel Pralinen zu bestechen.



Menü	Bearbeiten	Favoriten	Zusätze
Benutzermenü			Strg+F10
SAP Menü			Strg+F11
Business Workplace			Strg+F12
Rolle anlegen			Strg+Umsch+F12
Menü einer Rolle anzeigen			Umsch+F5
Auffrischen			Strg+F1



Welches Menü soll es sein?

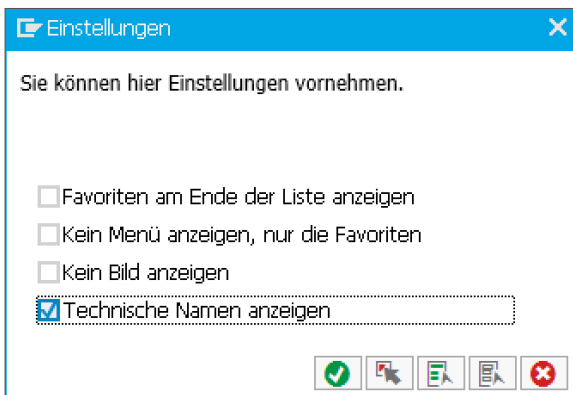
Das **Ausführen** von Programmen funktioniert am besten per Doppelklick auf den Menüeintrag, denn damit startest du die zugehörige Transaktion. Über das Kontextmenü der Menüleiste hast du Zugriff auf alle **wichtigen Funktionen**.

Hast du bemerkt, dass ich im SAP Easy Access eine spezielle Transaktion markiert habe? Ja? Stimmt, das ist die **SE80**, im Volksmund auch als **Object Navigator** bezeichnet. Ab heute deine **Lieblingstransaktion(scode)** zum Programmieren in SAP.



Ich liebe es!

Das Aussehen des SAP Easy Access kann verändert werden, um die Visualisierung deinen Bedürfnissen anzupassen. Mit dem Menüeintrag **Zusätze • Einstellungen** geht es in die Einstellungen.



Menü anpassen

Mit dem Setzen der Checkbox **Technische Namen anzeigen** kannst du die technischen Namen der Funktionen vor den Funktionstexten ein- und ausblenden.

Achtung, noch ein Geheimnis: Die technischen Namen sind nichts anderes als die **Transaktionscodes**.

Da gibt es noch
etwas Praktisches:

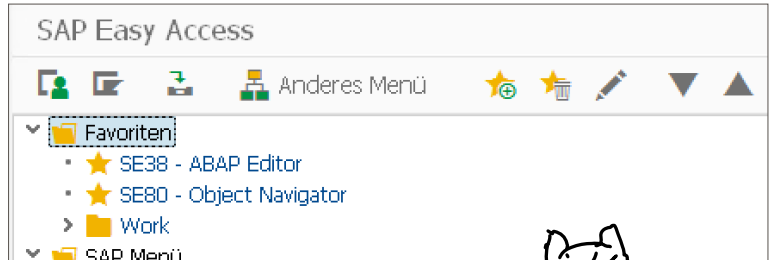


[Einfache Aufgabe]

Probiere mal, die **SE24** per Drag & Drop
auf den Favoritenknoten zu ziehen.

Favorit.

Der Knoten **Favoriten** ist deine persönliche Ablage für Transaktionen, Dateien, Webadressen.



Modus Operandi – Zusätzliche Fenster öffnen

Mach doch mal einen zweiten Modus auf!

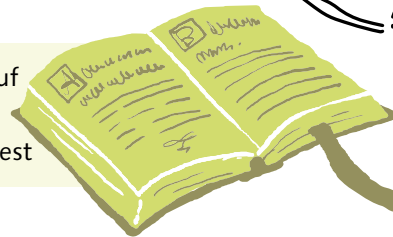
*Modus kenne ich
nicht!*

Wie – „Modus“ kennst du nicht?
Sorry, meine Schuld:

[Begriffsdefinition]

Ein **Modus** ist im Grunde genommen eine SAP GUI-Instanz auf
deinem Bildschirm, also ein weiteres **SAP GUI-Fenster**.

Damit kann man parallel an Anwendungen arbeiten – zumindest
Frauen. ☺



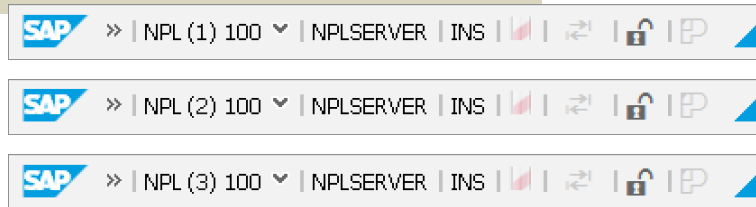
Der mächtige Systemadministrator kann zwischen zwei und sechs Modi
erlauben. Die Modi sind voneinander unabhängig, und auch wenn du einen
schließt, beeinflusst das die anderen nicht.

**Es gibt drei Möglichkeiten, einen neuen
Modus zu ÖFFNEN.**

- ☛ aus der Menüleiste heraus **System • Erzeugen Modus**
- ☛ mithilfe der Drucktaste **Neuer Modus** in der Symbolleiste
- ☛ **/o** + Transaktionscode im Befehlsfeld

Um einen Modus zu **SCHLIESSEN**, gehst du einen der folgenden Wege:

- ☛ Menü **System • Löschen Modus**
- ☛ das kleine „x“, rechts oben im Modusfenster
- ☛ **/i + <Nummer des Modus>**, den habe ich auch noch nicht gekannt. ☺
Woher kommen die Nummern? Lies weiter.



Nummer des Modus.

Jeder Modus erhält eine **eindeutige Nummer**, die als Teil der Systeminformation rechts neben dem Systemnamen dargestellt wird.

Eine **Liste aller Modi** erhältst du, wenn du **/o** im Befehlsfeld eingibst und mit der Drucktaste **Weiter** bestätigst.

Modi.

Backend-Sessions		
✕		
<input checked="" type="checkbox"/> Nr.	Transaktion	Zeit
2	Object Browser	20:12:42
3	ABAP Class Builder	20:12:54
4	ABAP Dictionary Pflege	20:13:26
<input checked="" type="checkbox"/> Erzeugen Session löschen ✕		

Um alle Modi auf einmal zu schließen und dich abzumelden, kannst du

- ☛ **/nend** in das Befehlsfeld eingeben oder
- ☛ **/nex** eintippen.
- ☛ **/nend** ist der **Freundliche** und fragt, ob du dir sicher bist, und **/nex** ist der **Bedingungslose**, der ohne nachzufragen alle Modi schließt.

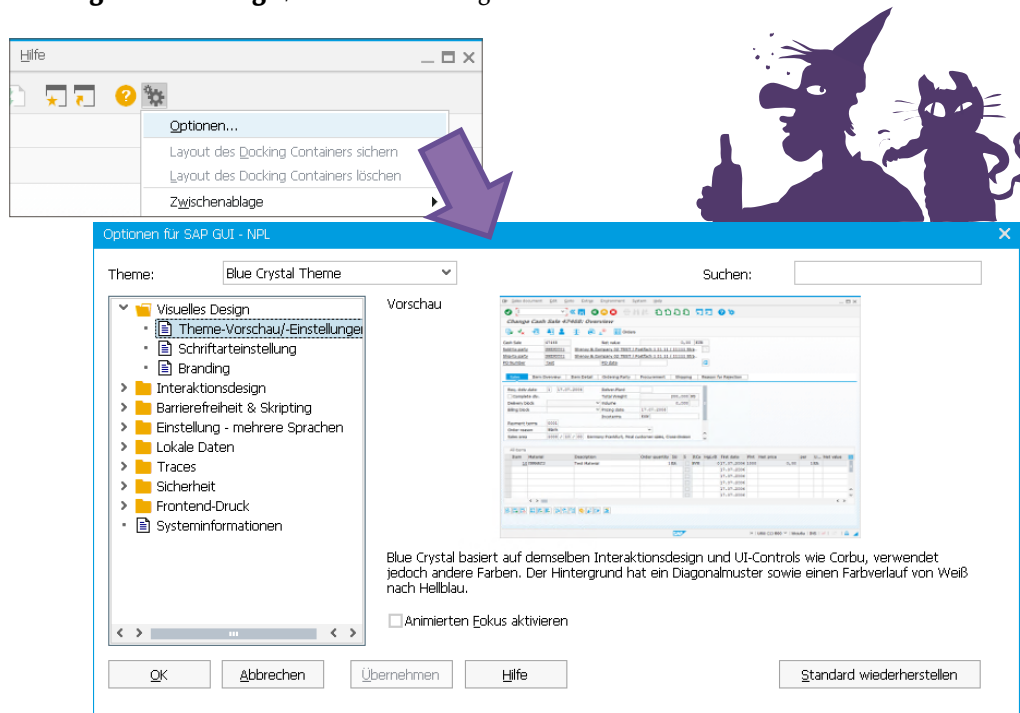


Das ist doch kein Thema

Die SAP möchte dich nicht festnageln.

Wie bitte???

Na, auf ein bestimmtes Aussehen deines SAP GUI natürlich. Du kannst dir selbst eines der sogenannten **Themes** aussuchen, um das optische Erscheinungsbild deines SAP GUI zu beeinflussen. Ich verwende zum Beispiel **SAP Blue Crystal**, weil das das Standardthema ist, das nach der Installation eingestellt ist, und außerdem ist es schön bunt. Sehr viele Entwickler verwenden zum Arbeiten aber das Theme **SAP Signature Design**, das ist ein wenig nüchterner. **Hicks.**



Das Thema wechseln.

Einstellen kannst du das Thema mit der Drucktaste **Optionen**, also dem Zahnradsymbol ganz rechts in der Symbolleiste. Wenn du da draufdrückst, erscheint ein Pop-up mit diversen Einstellungsmöglichkeiten, unter anderem einem Drop-down-Menü mit dem verräterischen Titel **Theme**. Dort wählst du deinen Liebling aus.

*Bei mir ändert sich aber nichts.
Hab ich da was falsch gemacht?*

Du doch nicht! Die neue Einstellung zieht erst, nachdem du SAP GUI und SAP Logon geschlossen und dann wieder neu geöffnet hat. DANN erst kommst du in den Genuss deines Themes.

Hüben und drüben, nichts als Üben – Rolands Folterkammer oder Übung macht den Meister

Frage mit möglicher Belohnung:

[Einfache Aufgabe]

Für welche **Anwendung** steht der Transaktionscode **SE11**?

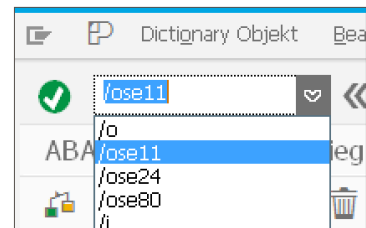
Tipp: Nutze die Macht der Symbolleiste.

Zur Belohnung: Potato-Wedges vom Feinsten!



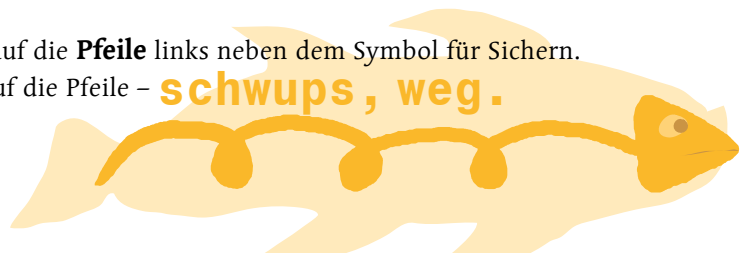
[Hintergrundinfo]

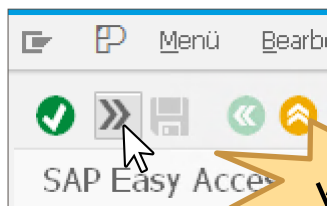
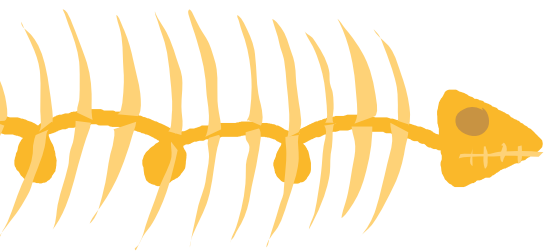
Tipp: Falls du schon Transaktionscodes im Befehlsfeld eingegeben hast, kannst du diese wieder aus dem Drop-down-Menü auswählen.



Wiederverwendung.

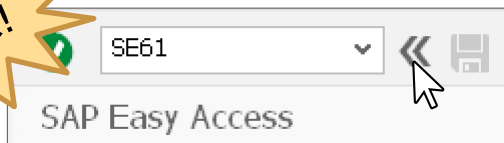
Ach ja: Falls sich das Befehlsfeld versteckt, **klicke** auf die **Pfeile** links neben dem Symbol für Sichern.
Falls du es **verstecken** möchtest, klicke ebenfalls auf die Pfeile – **schwups, weg.**



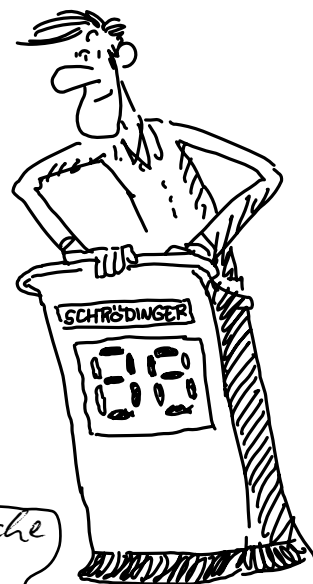


Einblenden.

Klick!



Ausblenden.



Fragen ohne Belohnung:

- ☛ Woran erkennst du, ob in den Einstellungen zum SAP Easy Access-Menü die Anzeige der technischen Namen aktiviert wurde?

- ☛ Kennst du einen anderen Namen für den technischen Namen?

Also ohne Belohnung geht bei mir gar nichts!
Aber die Fragen sind so babyisch, dass ich das blind mache
und in einem Satz beantworte.
Wenn die Einstellung vorgenommen ist, erscheint der
Transaktionscode links, vor dem Menüeintrag.

Frage zum Ende:

Heute scheint die Sonne, und die Vögel zwitschern ihr fröhliches Lied. Die Blumen duften blumig und fühlen sich blumig an. Welch ein harmonischer Tag. Du möchtest dich ebenso harmonisch vom SAP-System abmelden. Wie?

Na, das Fremdlische.

Eine kleine Übung noch:

In der Symbolleiste gibt es eine Drucktaste zum Öffnen eines neuen externen Modus. Findest du sie? Wie könntest du stattdessen einen neuen externen Modus öffnen.

Das kleine Kästchen
mit blauem Balken und
gelbem Pfeil.
Aber io ist mir lieber.



[Belohnung/Lösung]

Jetzt müssten die Wedges die richtige
Verzehrtemperatur haben.

Mahlzeit!

Hoppi-Galoppi ... SAP Fiori

Halt, da habe ich doch noch was entdeckt, das für dich interessant sein könnte, Schrödinger.

Noch was?

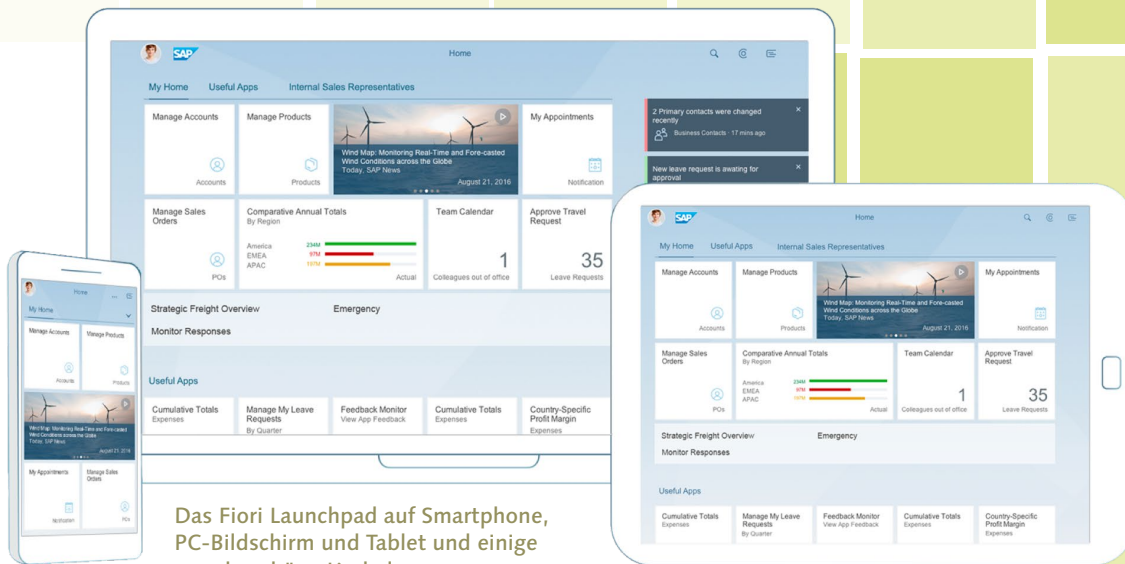
*Das war doch schon alles so interessant.
Außerdem werden die Wedges kalt.*

Das muss ich dir noch erklären, damit du UI-mäßig zukunftsfit bleibst:
Es nennt sich SAP Fiori.

*Klingt wie ein
kleines Pferdchen.*



Fast. Es geht um DIE zukünftige UI-Technologie der SAP, eine webbasierte Benutzeroberfläche, die mit einer Programmier Technik namens **SAPUI5** entwickelt wird. Das wirst du später selber implementieren, aber da muss ich dich noch richtig gut drauf vorbereiten. Für den Moment will ich dir nur zeigen, wie die neue Oberfläche aussieht. Die Fiori-Anwendungen rufst du mit dem **SAP Fiori Launchpad** auf. Das ist vergleichbar mit einem Portal, in dem die unterschiedlichsten Funktionen angeboten werden.



Das Fiori Launchpad auf Smartphone, PC-Bildschirm und Tablet und einige wunderschöne Kacheln.

*Lauter Kacheln,
sieht fast aus wie mein Küchenboden.*

Kacheln sind das neue Credo, alles wird gekachelt. Hinter jeder Kachel versteckt sich eine Anwendung (**App**). Eine Kachel ist also sowas wie eine Transaktion. Die SAP hat sich mit Fiori aber noch viel mehr vorgenommen: Es gibt ein eingebautes Berechtigungskonzept, und die Oberflächen passen sich automatisch an verschiedene Endgeräte an. Das ist schlüssig und flüssig und zu guter Letzt einfach zu bedienen.

*Und die Kacheln schwirren da alle so herum
wie fliegende Untertassen?*

Die Kacheln kannst du zur besseren Übersicht in Gruppen und Katalogen zusammenfassen. Für die werden dann Berechtigungen vergeben. Ein **Katalog** ist eine Sammlung von Fiori-Apps, die zu einer Rolle im Unternehmen oder zu einer Person gehören. Eine **Gruppe** ist eine Menge von Katalogen und damit von Apps, die einem User aufgrund seiner Rolle(n) zugeordnet sind.

[Einfache Aufgabe]

Besuche die Seite <https://www.sapfioritrial.com>
und experimentiere mit dem SAP Fiori Launchpad
und einigen der Kacheln.



*Und was ist mit den Wedge... rch ...
rrch ... RRRCH ...*



[Belohnung/Lösung]

Lass die neue Art der Benutzerinteraktion auf dich wirken
und atme dabei dreimal ein und aus ... und ein und aus ...
und ein und aus ... Gute Nacht, Schrödinger!



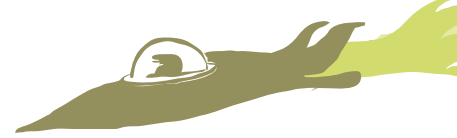
—DREI—

Der 20-Minuten-
Einstieg in
die ABAP-
Programmierung

Ciao a tutti!
(Hallo Welt!)

Endlich, das erste ABAP-Programm!
Schrödinger war sich bislang gar nicht sicher, ob der
Schwaiger Roland überhaupt programmieren kann.
Und dann legt er auf einmal richtig los:
Pakete anlegen, Datenbankzugriffe, Ausnahmebehandlung,
Schlüsselwortdokumentation. Dieses ABAP fängt an,
Spaß zu machen, weil es so viel zu entdecken gibt.

Einsteigen und anschnallen!



Es ist wahrscheinlich ratsam, dass du dir eine Tasse **Kaffee** (Betonung auf dem letzten e) oder ein **Erfrischungsgetränk** deiner Wahl zubereitest und bereitstellst.

Zur Auswahl steht alles mit Durchhaltefaktor > ein Tag:

- ☛ Kaffee, gemessen nicht in Tassen, sondern Kannen
- ☛ schwarzer Tee, direkt bezogen von der eigenen Teeplantage
- ☛ Energydrink, vorrätig in Paletten

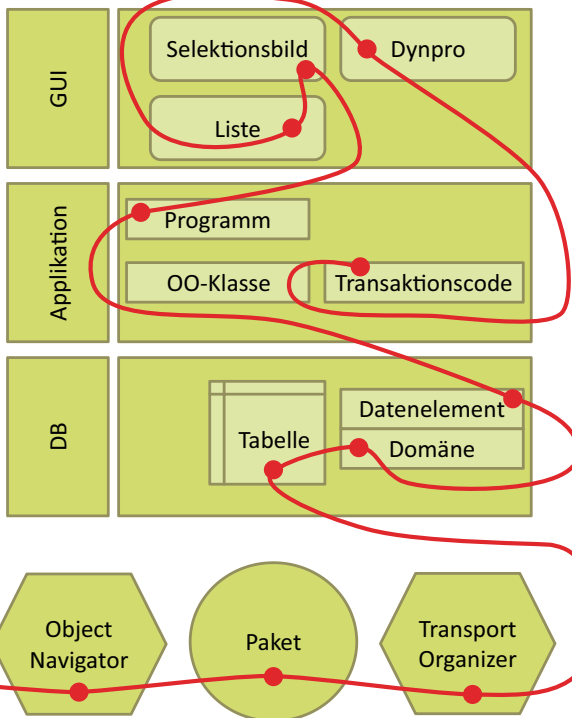
Dieses Kapitel benötigt deine volle Konzentration und Energie. FOKUS!

Ich werde dich mit **unglaublicher Geschwindigkeit** durch das **ABAP-Universum** ohne Pause („We break for nobody!“) navigieren, und wir werden alle Planeten des Inner Rims besuchen, wobei ich das Steuer unseres Raumschiffs übernehme.

Bereit für den Start? Noch schnell einen **Schluck**, und los geht's. Wie ich sehe, bist du ja bereits am SAP-System angemeldet.

Mein Plan für dieses Kapitel ist eigentlich nicht umsetzbar. Aber durch einen genialen Trick habe ich die Verständlichkeit extrem erhöht: Ich habe einen **didaktischen roten Faden** eingefädelt!

Mein Plan



Du wirst auf den folgenden Seiten mit Hilfe des Object Navigators die Entwicklungsobjekte anlegen und verwalten. Ein Paket hilft dir beim Gruppieren der Objekte, die du (im „wirklichen Leben“) mit dem Transport Organizer weiter zu einem anderen System schicken kannst.

Cooler Plan, oder? Ich habe dir Markierungen zum Ausschneiden eingefügt. Schneide doch einfach den Plan aus, damit du nicht immer hin und her blättern musst.

Scherzkeks!



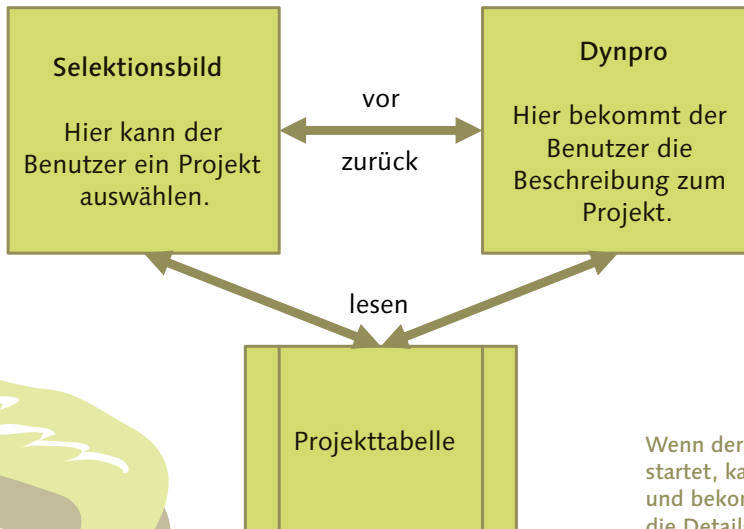
Nachdem du dann so gut organisiert bist, entwickelst du dich von unten (Datenbanktabelle, Datenelement und Domäne) über die Mitte (Programme, Klassen, ...) nach oben (zeig her, was du kannst). Details folgen!

Na gut, dann einige Details dazu:
Um dir einen Einblick und Durchblick zu verschaffen, möchte ich dir gerne beschreiben, wie du so ein Kunstwerk schaffen und in Form eines Projekts umsetzen kannst.

Sehr beeindruckend für Fensterbrett, Balkon und Garten. Durchblick und Einblick in einem.



Die Beschreibung werde ich natürlich im SAP-System in einer **Datenbanktabelle** hinterlegen und dir auf einem **Dynpro** anzeigen. Da das Programm und die Datenablage sehr allgemein ausfallen werden, kannst du sie auch für deine eigenen Projekte als Beschreibungsbasis verwenden.



Wenn der Anwender das Programm startet, kann er ein Projekt auswählen und bekommt nach der Bestätigung die Details zum Projekt angezeigt.

Der Anwender kann in einem **Selektionsbild** ein Projekt auswählen und Details dazu anzeigen lassen. Dafür wird ein Dynpro verwendet. Die Daten kommen natürlich, wie es sich gehört, aus einer Datenbanktabelle.

Falls dein Motor schon warmgelaufen ist, legen wir einen Le-Mans-Start hin, sprinten von der Couch zum Schreibtisch und werfen die Transaktion zum Programmieren an.

Object Navigator – Die integrierte Entwicklungsumgebung

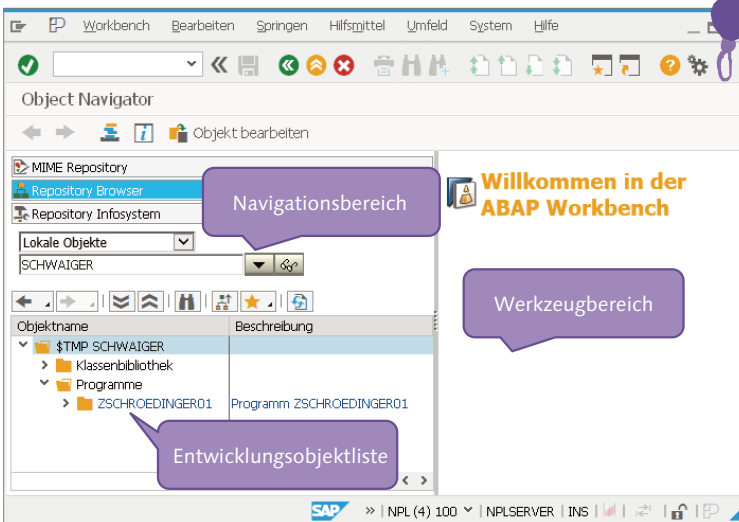
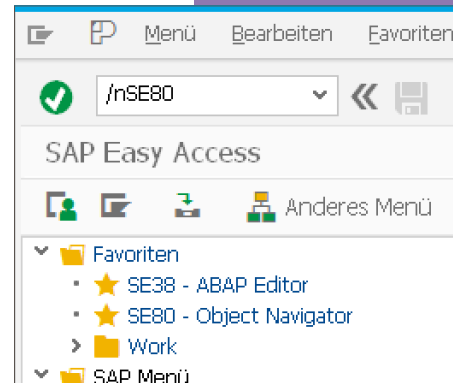
Das Werkzeug, von dem aus du alle deine **Entwicklungen** starten wirst, ist der **Object Navigator**, der den Transaktionscode **SE80** besitzt (Rührt sich da was im Kopf?). Damit hast du auf alle weiteren Transaktionen Zugriff, die für die **Bearbeitung** von **Entwicklungsobjekten** benötigt werden. Der Object Navigator ist deine **Schaltzentrale**: Anlegen, Ändern, Löschen, Umbenennen, Aktivieren, Kopieren, und vieles mehr wird dir zur Verfügung gestellt.

1. Tippe im Transaktionscode-Feld **/nse80** ein, und bestätige deine Eingabe.

Starten der Motoren mit der SE80.

2. Schau dich etwas um in der SE80. Für uns sind derzeit nur einige Teile wichtig.

- Der **Navigationsbereich** dient dir zum Auswählen von Entwicklungsobjekten.
- Die **Objektliste** zeigt dir die ausgewählten Entwicklungsobjekte an.
- Der **Werkzeugbereich** dient zum ...? Wozu wohl?



[Einfache Aufgabe]

Sieh dich in der SE80 um, mach dich mit den Bereichen vertraut, versuche hier und dort einen Doppelklick. Falls du dich verläufst, rufe einfach wieder **/nse80** auf. Das ist wie bei Hänsel und Gretel mit den Brotkrümel, wobei du nur einen hast, den du dafür aber auch sicher wiederfindest.

Im Navigationsbereich kannst du Objekte auswählen, in der Objektliste dann das Ergebnis der Auswahl bewundern und im Werkzeugbereich bearbeiten.

Entwicklung organisieren – Systemlandschaft, Änderungsauftrag

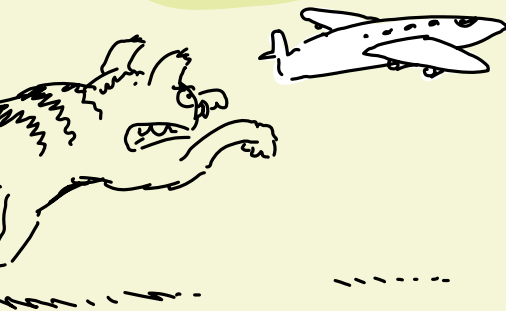
SAP empfiehlt dir und somit seinen Kunden eine **Drei-System-Landschaft**. Ein System für die **Entwicklung** – in dem werden wir uns bewegen –, ein System für die **Qualitätssicherung**, und eines für die **Produktion**, auch als **Produktivsystem** bezeichnet. Irgendwo müssen die Anwender deiner formidablen Programme ja auch arbeiten können.



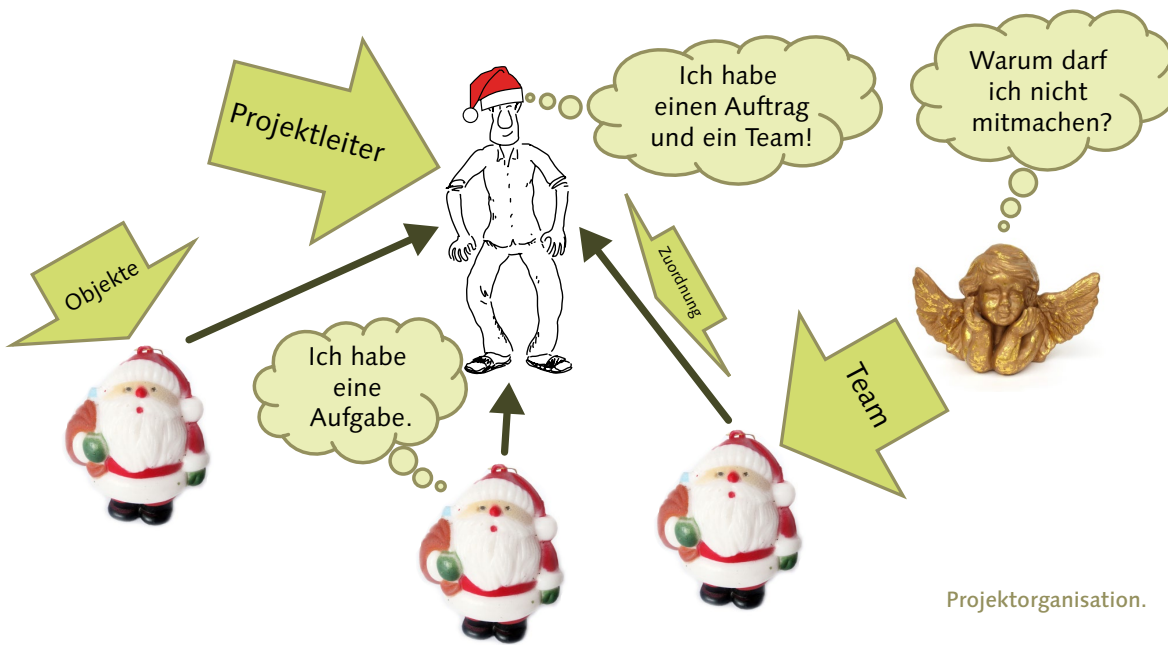
Bauen –
Testen –
Verwenden.



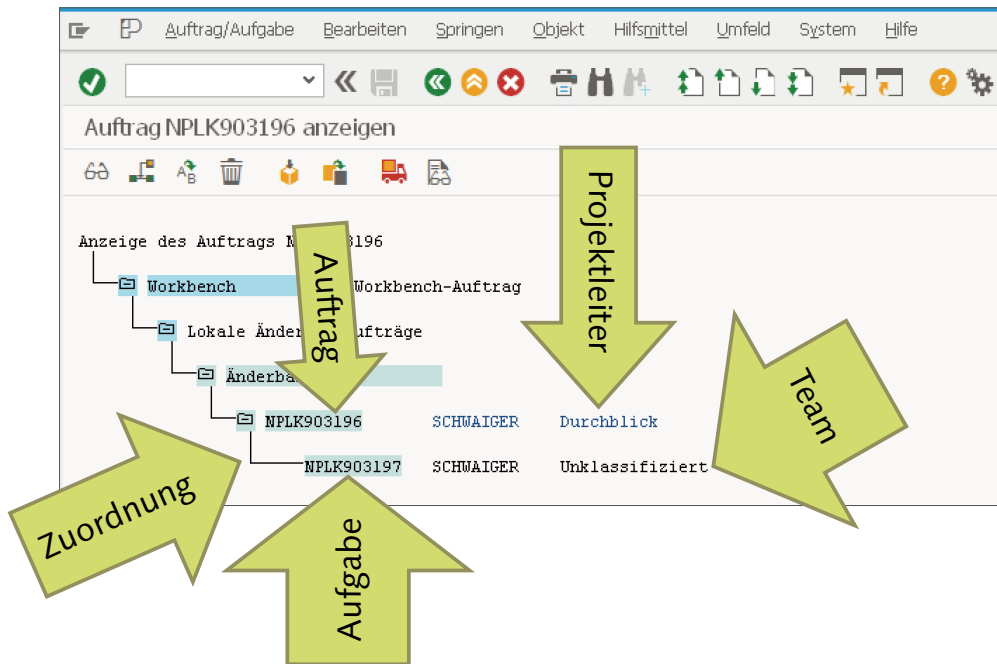
Oh Schreck, da wird jemand
damit arbeiten? Schluck.



Die Entwicklungsobjekte werden im Rahmen eines **Entwicklungsprojekts** im **Entwicklungssystem** angelegt. Am Ende der Entwicklung müssen die Objekte in das Folgesystem transportiert werden. Zu diesem Zweck legt der **Projektleiter** mit der Transaktion **SE09** einen Änderungsauftrag an und ordnet die Mitarbeiter dem Auftrag zu. Dadurch wird **pro Mitarbeiter** eine **Aufgabe** im **Auftrag** angelegt. Falls du nun ein Entwicklungsobjekt änderst oder anlegst, wird dieses Objekt der Aufgabe zugeordnet. Während der Abarbeitung der Aufgaben im Projekt werden alle Objekte im Auftrag gesammelt. Mitarbeiter, die nicht Teil des Projekts sind, können die Entwicklungsobjekte leider nicht ändern.

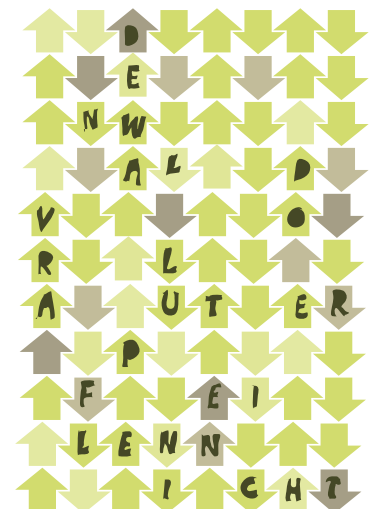


Falls dir das mit dem Engerl zu kitschig ist, kann ich dir auch eine eher formale, nüchterne Darstellung eines Änderungsauftrags (auch bekannt als **Workbench-Auftrag**) anbieten.



So sieht das im SAP-System aus. Gleiche Information, nur ohne W-Mann und Engerl. So einen Auftrag mit Aufgabe wirst du gleich anlegen.

Der W-Mann gefällt mir besser.

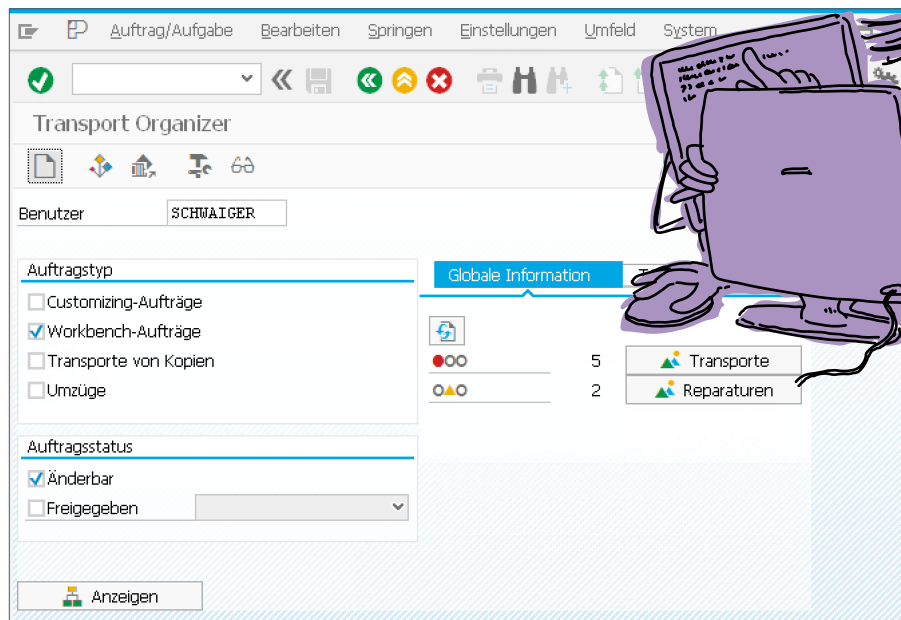


Ist das Projekt umgesetzt, kann jeder Entwickler seine **Aufgabe freigeben**. Dies führt aber noch nicht zur kompletten Freigabe des Auftrags. Die muss vom **Projektleiter** (dem Besitzer des Auftrags) erteilt werden. In welche Systeme transportiert wird, legt die **Transportschicht** fest, die dem Paket zugeordnet ist.

Ihr neuer Auftrag lautet: Auftragsanlage

Jetzt legen wir uns einen Änderungsauftrag für unser Projekt **Durchblick** an.

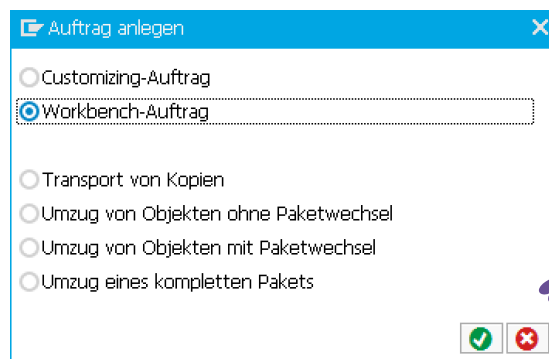
1. Ruf die Transaktion SE09 auf, am besten mit /ose09. Was geschieht beim /o noch mal?
Natürlich öffnet sich ein neues Modus. Zumindest solange noch Fenster geöffnet werden können.



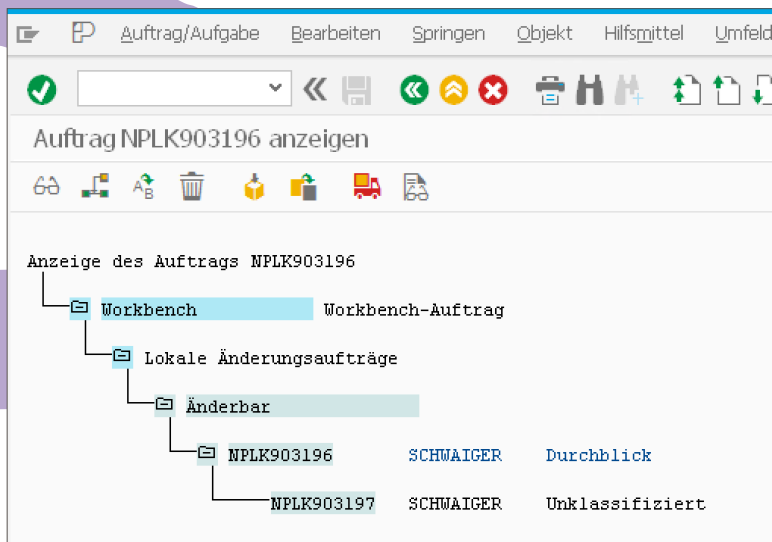
Dein erster Auftrag, den du als Projektleiter anlegst.
Die Anlegen-Drucktaste (F6) macht den Job.

2. Im folgenden Pop-up wählst du **Workbench-Auftrag**, da wir Entwicklungsobjekte anlegen werden. Bestätige dies mit dem grünen Häkchen.

Der Workbench-Auftrag ist dein Sammel-Container für alle Entwicklungsobjekte zu einem Projekt.



3. Wir kommen schon zum Ende. Im erscheinenden Pop-up gibst du noch die Bezeichnung des Auftrags ein – ich finde „Durchblick“ sehr passend –, und, da du schon automatisch als Bearbeiter zugeordnet bist, bestätigst du das mit der **Sichern**-Drucktaste (Enter).

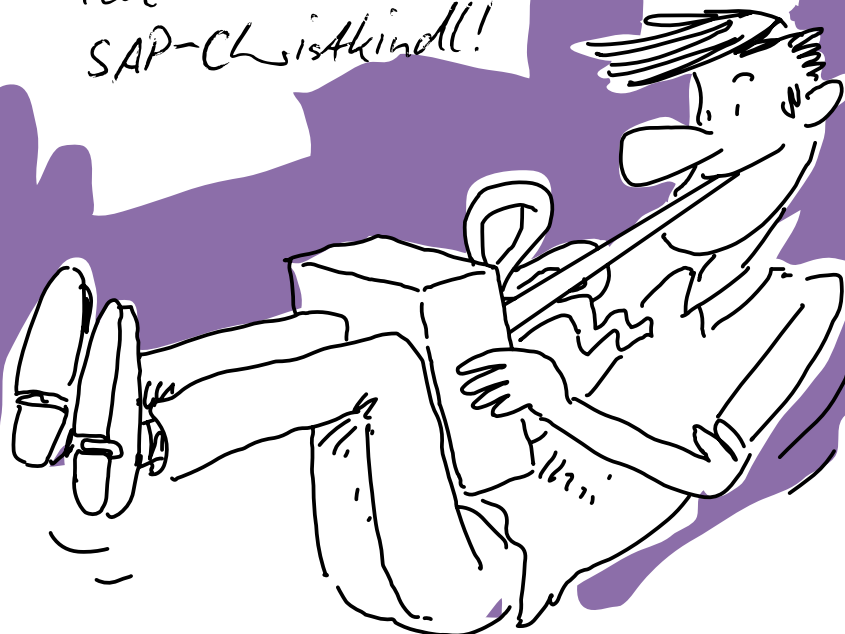


Dein Auftrag und deine Mission: Durchblick zu schaffen. Du bist Projektleiter und Mitarbeiter zugleich. Kannst du mit dieser Schizophrenie umgehen? Sei nicht zu hart zu dir, falls du den Termin nicht halten kannst.

Damit bist du für die Entwicklung gerüstet.

Auf in Windeseile zum nächsten Schritt. Das **Paket**!

*Das ist ja wie Weihnachten.
Ich bekomme ein Paket vom
SAP-Christkind!!*



Organisationskapsel Paket

Ein **Entwicklungsobjekt** kann einem **Auftrag** zugeordnet werden. Das währt aber nur kurze Zeit, nämlich so lange, bis der **Auftrag freigegeben** wird. Dann kommt schon der nächste Auftrag ... Man könnte auch sagen, dass das **Objekt einem Auftrag temporär zugeordnet** ist.

Sind wir noch im Plan?

Beim Kästchen Paket, ist das richtig?

Ja, alles noch im Plan. Es gibt noch eine andere Art der Zuordnung, nämlich eine **permanente Zuordnung** zu einem Paket. Wenn du Java kennst, wird dir der Begriff **Paket** bereits etwas sagen. Falls du Java nicht kennst, macht das auch nichts, und du darfst weiterlesen.



[Begriffsdefinition]

Ein Paket ist ein **Container**, in dem unter anderem Entwicklungsobjekte gesammelt werden. Dabei können Pakete **Schnittstellen** und **Verwendungserklärungen** besitzen. Darüber hinaus können Pakete **geschachtelt** werden.

Für die unterschiedlichen Granularitätsebenen gibt es unterschiedliche Arten von Paketen.

- **Standardpakete** („kein Hauptpaket“) nehmen die Entwicklungsobjekte auf.
- Standardpakete können in **Hauptpakete** geschachtelt und somit gruppiert werden.
- Hauptpakete wiederum können **Strukturpaketen** zugeordnet werden.
- **Strukturpakete** werden bei der Strukturierung großer Softwareprojekte eingesetzt. Die sind für dich als Einsteiger derzeit nicht wichtig. Ich würde vorschlagen, wir reden in einem Jahr wieder über das Thema.

Früher hat es andere Container gegeben, die als **Entwicklungsklassen** bezeichnet werden. Diese sind mit den Standardpaketen vergleichbar.

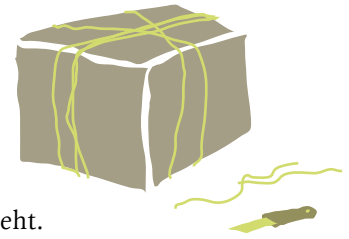


[Hintergrundinfo]

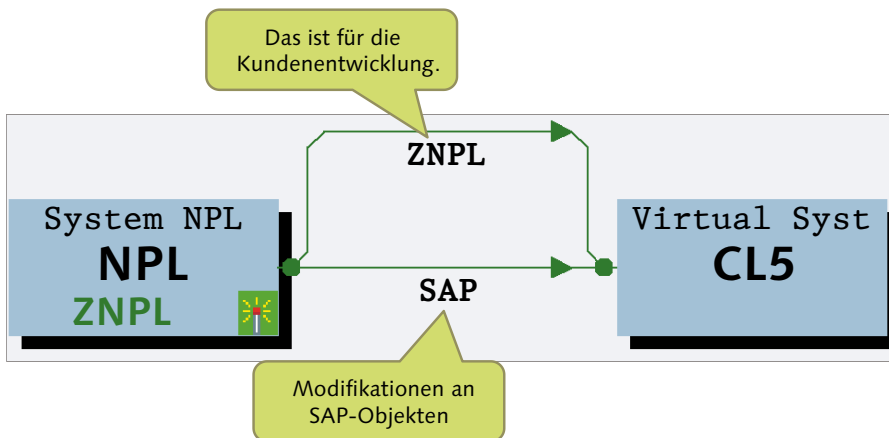
Im SAP-System gibt es reservierte **Namensräume** für Kundenobjekte und SAP-Objekte. Wenn du den Kundennamensbereich verwendest (die Namen der Entwicklungsobjekte beginnen mit Z oder Y bzw. mit einem bei SAP reservierten Namensraum, zum Beispiel /SCHROED/), werden deine Objekte während des Einspiels neuer SAP-Entwicklungsobjekte nicht überschrieben.

Safety first!

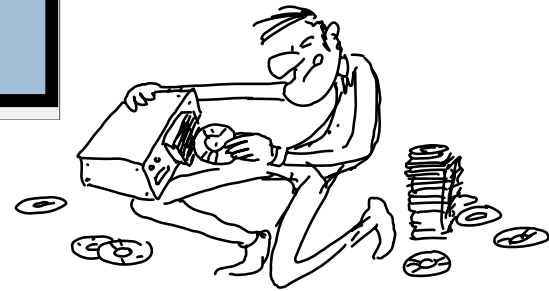
Die **EIGENSCHAFTEN**, die zu einem Paket festgelegt werden, sind:



- **Paketname**, natürlich im Kundennamensraum!
- **Kurzbeschreibung**: Hilft zu beschreiben, um was es thematisch in dem Paket geht.
- **Anwendungskomponente**: Dient der Einordnung des Pakets in die Gesamtmenge der Anwendungen – die auch als Anwendungshierarchie bezeichnet wird –, kann jedoch initial, das heißt leer bleiben.
- **Softwarekomponente**: Die Softwarekomponente beschreibt eine Menge von Paketen, die nur gemeinsam auslieferbar sind. Für Kundenprojekte wird **immer** die Softwarekomponente **HOME** verwendet.
- **Transportschicht** (ah, da ist sie): Für Kundenentwicklungen wird eine Transportschicht eingerichtet, die darüber entscheidet, ob und wenn ja, in welches Folgesystem die Objekte transportiert werden.
- **Pakettyp**: Es stehen drei verschiedene Pakettypen zur Verfügung: **Standardpaket**, **Hauptpaket** und **Strukturpaket**. Wie die Namen schon andeuten, sind die Typen für unterschiedliche Einsatzgebiete gedacht. Wir werden **Standardpakete** für die Entwicklung verwenden.



Die Wege des ABAP sind wunderbar.



[Hintergrundinfo]

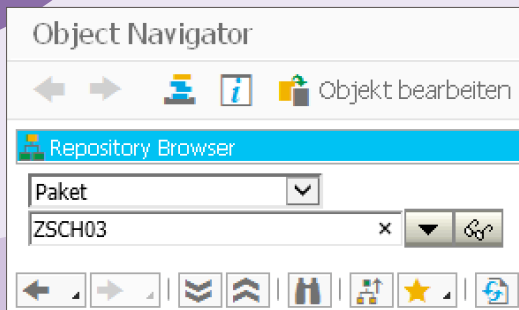
Kleiner Tipp am Rande: Die Transportwege und -schichten kannst du dir mit der Transaktion **STMS** ansehen. Falls du diesen Transportweg nicht siehst und daran interessiert bist (für das Verständnis von ABAP ist das nicht unbedingt nötig), findest du im Anhang eine detaillierte Beschreibung zur Einrichtung des Transportmanagementsystems (TMS).



Vom Entwickler zum Pakettier – Paket anlegen

Also legen wir ein Paket an.

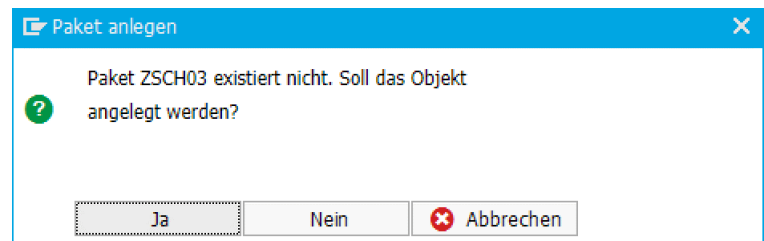
1. Wechsle in den Object Navigator. Das sollte ziemlich einfach sein, da der Modus mit dem Object Navigator immer noch vorhanden sein sollte. Wenn nicht: auch egal, einfach noch mal **SE80** aufrufen.
2. Im Navigationsbereich wählst du als Typ **Paket**, und ich schlage dir den Namen **ZSCH03** für dein Paket vor. Das ist aber nur ein Vorschlag. Was ist der einzige Aspekt, den du bei der Vergabe des Namens beachten musst?



Du willst also ein Paket anlegen. Bravo, und auch noch so ein schöner Name.

An der ersten Stelle muss ich ein Z oder Y verwenden oder mir einen Namensraum bei der SAP reservieren. Den wünsche ich mir zu meinem Geburtstag.

3. Nachdem du die Eingaben erledigt hast, hämmerst du einfach auf die **Enter**-Taste. Und das schlaue System stellt fest, dass es das Paket noch nicht gibt. Bestätige die Neuanlage mit der Drucktaste **Ja**.



Ja, ich will anlegen, bitte.

**DRAUF
DRÜCKE!**