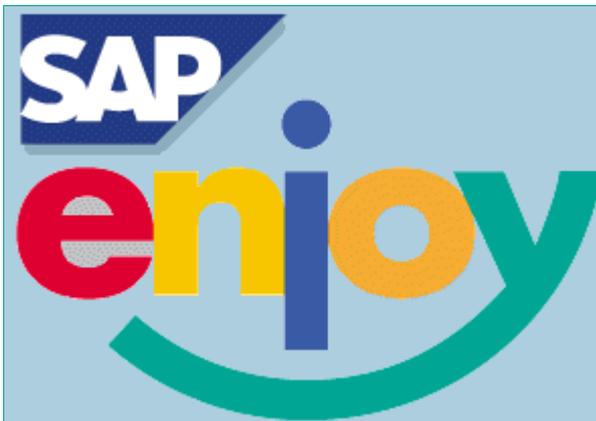


# 1 Vorwort

Dieses Buch wendet sich an ABAP-Programmierer, die die Möglichkeiten der objektorientierten Programmierung in ihren Anwendungen nutzen wollen.

Doch warum sollten ABAP-Programme nun objektorientiert entwickelt werden? Hierfür gibt es mehrere wichtige Gründe.

Seit Release 4.6 hat SAP die Benutzeroberfläche des R/3-Systems durch den vermehrten Einsatz grafischer Elemente deutlich ergonomischer gestaltet. Mit Unterstützung dieser **EnjoySAP**-Initiative soll die tägliche Arbeit am SAP-System dem Anwender Spaß machen. Allerdings sind die grafischen Elemente – Controls genannt – vollständig in objektorientierter Technologie aufgebaut. Die zahlreichen neuen Enjoy-Transaktionen mit Einsatz von Controls haben das Erscheinungsbild der SAP-Anwendungen tiefgreifend verändert. Um nun diese Controls auch in eigenen Programmen zu nutzen, muss man ABAP Objects beherrschen.



**Abb. 1-1**  
EnjoySAP-Logo

Auch die mächtigen Anwendungen der MS-Office-Reihe lassen sich über Controls mit Hilfe der OLE2-Technologie in SAP-Transaktionen integrieren. Stellen Sie sich eine SAP-Transaktion vor, in der ein Teilbereich des Dynpros z.B. mit einem Word-Dokument gefüllt ist. Innerhalb dieses Word Controls können Sie fast alle Aktionen vornehmen, die MS-Word beherrscht – sogar Makros können ausgeführt werden. Man nennt diese Technologie **Office-Integration**; sie zu realisieren ist nur in objektorientierter Programmierung möglich.

In diesem Zusammenhang muss der **Business Document Service** (BDS) erwähnt werden: Über diesen Dienst lassen sich beliebige Dokumente als Binärdaten auf der Datenbank des SAP-Systems ablegen. Die dahinter stehende Philosophie hat weit reichende Folgen für die Datenorganisation eines jeden Unternehmens, welches SAP für seine Betriebswirtschaft verwendet. Die Möglichkeiten des BDS selbst können ebenfalls nur in objektorientierter Programmierung genutzt werden.

Ein wesentlicher Vorteil von objektorientierter Programmierung ist die größere Code-Unabhängigkeit der Software-Komponenten gegenüber der klassischen Programmierweise. So stellt die objektorientierte Programmierung auch die Basis für die softwaretechnische Öffnung des SAP-Systems dar. Die objektorientierte Programmierung in ABAP ist somit die Schlüsseltechnologie, um moderne **E-Commerce-Lösungen** aus dem ABAP-Applikationsserver zu ermöglichen.

Ein Beispiel hierfür ist der ab Web Application Server Release 6.20 zur Verfügung stehende native **XML-Support**, der es ermöglicht, offene und von der Datenstruktur unabhängige Systemschnittstellen aus der ABAP-Welt ohne jegliche Fremd-Software zu realisieren.

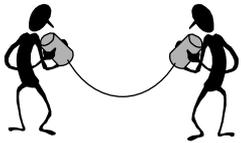


Abb. 1-2

mySAP.com-Logo



Doch die bisher vorgestellten pragmatischen Gründe für eine objektorientierte ABAP-Programmierung verdecken die viel weiter reichenden strategischen Überlegungen: Nur unter objektorientiertem Design und Programmierung lassen sich komplexe Software-Systeme mit einem hohen Qualitätsstandard herstellen. Die Objektorientierung ist hier der Schlüssel für einen hohen Wiederverwendbarkeitsgrad und eine bestmögliche Stabilität der entstandenen Software-Komponenten.

Gerade diese Möglichkeiten führen dazu, die Entwicklungsaufwände für komplexe Aufgaben drastisch zu senken, indem die Gesamt-



aufgabe geschickt in wiederverwendbare Teilkomponenten aufgeteilt wird und diese bei Folgeprojekten benutzt werden. Um eine solche Software-Architektur aufzubauen, ist freilich einige Erfahrung notwendig, ohne Objektorientierung aber nahezu unmöglich. Im diesem Buch sind solche Konzepte im Rahmen des Teils zur Dialogprogrammierung mit Controls ausführlich dargelegt. Die dort gezeigten **Templates, Basis-Interfaces und -klassen** sind allesamt praxiserprobt. Vor Kurzem konnten wir ein von einem der größten SAP-Beratungshäuser auf 35 Tage geschätztes Projekt dank unserer Architektur mit nur elf Tagen Aufwand realisieren.

Die eben beschriebenen Entwicklungen lassen mich folgendes klare Fazit ziehen:

**Nur wer auch die objektorientierten Sprachelemente in ABAP beherrscht, kann zukünftig in SAP R/3 Software herstellen, die den hohen qualitativen und funktionalen Anforderungen der Zukunft genügt.**

## 1.1 Release-Stand und Software-Beispiele

Die zahlreichen Navigationshilfen und Screenshots basieren auf dem SAP-Release ERP 2005, die beschriebenen Konzepte und Verfahren funktionieren jedoch ebenso auf älteren SAP-Release-Ständen ab Release 4.6. An denjenigen Stellen, wo es Unterschiede in dieser auch für SAP neuartigen Technologie gibt, wird gesondert darauf hingewiesen.

Sämtliche Beispielprogramme stehen auf [www.xaption.de](http://www.xaption.de) zum Download bereit. Fragen werden im Forum [www.abaps.de](http://www.abaps.de) schnellstmöglich beantwortet.

## 1.2 Notwendige Vorkenntnisse

Da die Verwendung von EnjoySAP Controls auf einer dialogorientierten Verarbeitung beruht, benötigen Sie nicht nur das Wissen um die Bedienung der Werkzeuge der ABAP Workbench (mit Ausnahme des Class Builders), sondern auch Kenntnisse zur Dialogprogrammierung.



## 1.3 Warenzeichen

Die in diesem Buch verwendeten Begriffe SAP, NetWeaver, mySAP.com, R/3, R/3 Enterprise und ABAP sind eingetragene Warenzeichen der SAP AG, Walldorf. Aufgrund der besseren Lesbarkeit wurde auf eine Kennzeichnung dieser Wörter durch das entsprechende Symbol (®) verzichtet.

Die Strichmännchen an den Seitenrändern heißen »Screen Beans«; sie stammen ebenso wie die verwendeten Fotos aus der Clipart-Gallery von Microsoft Office. Wir bedanken uns für die Erlaubnis, diese hier verwenden zu dürfen.

## 1.4 Zum Inhalt



Da die meisten der ABAP-Programmierer die objektorientierte Technologie nicht beherrschen, werden im **ersten Teil** zunächst die **Grundlagen der objektorientierten Programmierung** vermittelt. Dies ist eines der wichtigsten Kapitel, da hier gegenüber der klassischen Programmierweise eine völlig neue Denkwelt erschlossen wird. Entwickler, die diese bereits beherrschen, können das betreffende Kapitel recht schnell überfliegen.

Im dritten Kapitel werden die **OO-Sprachelemente der ABAP-Sprache** im Detail vorgestellt. In einer praktischen Übung werden diese Konstrukte verwendet und gleichzeitig die Beherrschung der »neuen Denke« in ihrer praktischen Anwendung überprüft. Zum Abschluss gibt es noch einige Überlegungen zum **objektorientierten Design**.

Lokal deklarierte OO-Konstrukte gelten analog zu lokalen Datendeklarationen nur im jeweiligen Programm. Ähnlich wie das Data Dictionary für Datendeklarationen wirkt der **Class Builder** und stellt die OO-Elemente allen Programmen zur Nutzung zur Verfügung.

Nachdem wir eigene OO-Elemente deklarieren und verwenden können, betrachten wir die von SAP bereitgestellten OO-Klassen.

Es folgt im **zweiten Teil** ein weit reichender Überblick über die wichtigsten **EnjoySAP Controls**. Zu jedem Control gibt es ein eigenes Kapitel mit Beispielprogramm. Da die Controls intelligente Software-Komponenten des Präsentationsservers sind, werden die Problemstellungen der verteilten Datenverarbeitung anhand der **Synchronisation der Automation Queue** ausführlich besprochen. Ebenso wichtig sind die Verfahren, wie man in ABAP auf **Ereignisse**, die ein solches Control ausgelöst hat, reagieren kann.

Die Krönung der Control-Verarbeitung stellt die **Office-Integration** dar, welche im **dritten Teil** ausführlich vorgestellt wird. Beispiele zeigen die Verwendung von MS-Word und MS-Excel als integrierte Komponenten innerhalb von SAP-Transaktionen.

Die Business-Object-bezogene Speicherung von solchen Office-Dokumenten auf der SAP-Datenbank hat weit reichende Konsequenzen. Dem **Business Document Service** ist daher ein eigenes Kapitel zugeordnet.

In einem abschließenden Kapitel zur Office-Integration werden die Möglichkeiten des **Datenaustauschs** und der weiteren Integration von Office-Anwendungen kurz vorgestellt: Die Office-Dokumente können Makros (in Visual Basic) enthalten und ihrerseits auf Ressourcen zugreifen, die innerhalb des SAP-Systems liegen. Hier haben wir dann eine komplexe Client/Server-Anwendung mit interaktiver Verarbeitung in ABAP Objects und Visual Basic.

Im **vierten Teil** geht es schließlich um die ab R/3 Enterprise in der ABAP-Welt zur Verfügung stehenden **XML-Technologien**, die es ermöglichen, komplexe Datenschnittstellen bis hin zu kompletten EAI-Prozessen über Webservices mit geringsten Mitteln herzustellen.

Nach einer kurzen **Einführung zu XML** werden die Möglichkeiten des **Parsens und Renderns von XML-Dokumenten** mit ABAP gezeigt. SAP unterstützt in der **iXML-Bibliothek** die offenen Standards **DOM** und **SAX** zur Verarbeitung von XML-Daten. Sogar die automatische Prüfung mit Hilfe von **DTDs** wird unterstützt.

**XSL-Transformationen** ermöglichen die Konvertierung von Datenstrukturen auch aus ABAP-Programmen heraus. Mit Hilfe der **ABAP Serialization XML** und **Simple Transformations** lassen sich ABAP-Datenstrukturen direkt nach XML und wieder zurück umwandeln. Nach Integration dieser Verfahren in BSP-Seiten oder Webservices lassen sich so komplette **XML-EAI-Szenarien** aufbauen – einfach nur mit ABAP und ein wenig XSLT.

Neben XML fanden inzwischen weitere moderne Technologien Einzug in die ABAP-Welt, die mit Hilfe von objektorientierter Programmierung zu realisieren sind. Es handelt sich jeweils um Technologien zur Gestaltung von Dialoganwendungen mit Web-Frontend. Mit Hilfe von **Business Server Pages (BSP)** lassen sich auf einfache Weise Web-Anwendungen in der ABAP-Welt realisieren. Die jüngste Technologie ist das **Web Dynpro for ABAP**, welches konsequent das MVC-Paradigma zum Design von Dialoganwendungen – ganz nach dem Vorbild des Web Dynpro for Java – umsetzt. Beide Themenfelder sind leider zu umfangreich, um vollständig in diesem Buch behandelt zu werden, obgleich sie auf OO-Technologien basieren. Zum Thema BSP existiert ein separates Buch und dem Web Dynpro for ABAP wurde ein Kapitel gewidmet, worin ein einfaches Praxisbeispiel gezeigt wird.

Zu einigen Themen dieses Buches – insbesondere zu den unverzichtbaren Grundlagen – habe ich **Übungsaufgaben** vorgesehen. Bei den anderen sind die jeweils wichtigen Programmcodings direkt im Kapitel wiedergegeben, so dass Sie keine Schwierigkeiten haben sollten, eigene Anwendungen zu erstellen. Mit Hilfe der Übungsaufgaben erreichen Sie einen besseren Lerneffekt und erlangen etwas Praxiserfahrung in der objektorientierten Arbeit unter ABAP. Ich empfehle

Ihnen daher, auf jeden Fall diese Übungen durchzuführen. Bei Schwierigkeiten finden Sie die entsprechenden Musterlösungen auf der Webseite zum Buch. Doch beachten Sie: Man lernt besser durch eigene Fehler als durch das bloße Nachvollziehen von Musterlösungen.

## 1.5 Grenzen?

Die Summe der Möglichkeiten von ABAP Objects lässt nur noch eine einzige Schranke für die Gestaltungsmöglichkeiten von SAP-Programmen zu: Horizont und Qualifikation des Programmierers. Ansonsten ist gemäß einem ehemaligen Werbeslogan »nichts mehr unmöglich«.



## 1.6 Danksagung

Es gibt eine Reihe von Personen, ohne die dieses Buch nicht hätte entstehen können. Ich bedanke mich bei Ingrid, Thorina und Thorsten für die zahlreichen Wochenenden, an denen sie auf ihren Ehemann und Papa verzichten mussten.

Christian Johann hat mir bei einigen XML-Themen wertvolle Unterstützung geboten, Frank Münker mir den Einstieg in's Web-Dynpro erleichtert, beiden gebührt mein Dank.

Bei meinem Kompagnon Elmar Schlecker bedanke ich mich dafür, dass ich mir neben meinem wichtigen Job Zeit für das Buch nehmen durfte.

Zuletzt gilt auch großer Dank Dr. Michael Barabas, meinem Verleger, den ich mehrmals vertrösten musste, für sein Verständnis und seine Unterstützung bei diesem Buchprojekt.