Peter Stender

Webprojekte realisieren nach neuesten OOP-Kriterien

Ein Workshop über die Kooperation von PHP/XSL/JavaScript

► Mit Online-Service

PRAXIS

VIEWEG+
TEUBNER

Peter Stender

Webprojekte realisieren nach neuesten OOP-Kriterien

Peter Stender

Webprojekte realisieren nach neuesten OOP-Kriterien

Ein Workshop über die Kooperation von PHP/XSL/JavaScript

Mit 45 Abbildungen

PRAXIS

VIEWEG+
TEUBNER

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.

Höchste inhaltliche und technische Qualität unserer Produkte ist unser Ziel. Bei der Produktion und Auslieferung unserer Bücher wollen wir die Umwelt schonen: Dieses Buch ist auf säurefreiem und chlorfrei gebleichtem Papier gedruckt. Die Einschweißfolie besteht aus Polyäthylen und damit aus organischen Grundstoffen, die weder bei der Herstellung noch bei der Verbrennung Schadstoffe freisetzen.

1. Auflage 2011

Lektorat: Christel Roß | Walburga Himmel

Vieweg+Teubner Verlag ist eine Marke von Springer Fachmedien.
Springer Fachmedien ist Teil der Fachverlagsgruppe Springer Science+Business Media.
www.viewegteubner.de

Umschlaggestaltung: KünkelLopka Medienentwicklung, Heidelberg
Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

ISBN 978-3-8348-1430-2

Vorwort

Dieses Buch ist entstanden, um dem Umstand gerecht zu werden, dass es zwar viele gute Bücher über die Grundlagen der PHP Programmierung gibt, aber keins welches die Erstellung eines kompletten Projektes von Anfang bis Ende beschreibt.

Leider ist dieses Buch nicht für blutige Anfänger in der Webprogrammierung gedacht, da ich um den Rahmen des Buches nicht zu sprengen, darauf verzichte die Sprachkonstrukte und Referenzen der einzelnen zum Einsatz kommenden Scriptsprachen zu erklären. Hierfür gibt es genügend andere Fachliteratur die sich speziell an Anfänger richten. Sollten Sie also diesen Workshop durcharbeiten wollen und noch relativ frisch in der Programmierung dynamischer Websites sein, ist es unumgänglich sich auch mit solchen Fachbüchern zu bewaffnen um erstmal die wichtigsten Grundlagen der einzelnen Sprachen zu erlernen. Am Ende des Buches finden Sie eine Reihe von Buchvorschlägen, sowie Webadressen für die einzelnen hier behandelten Sprachen, die sich speziell an Anfänger richten.

Da ein solches komplexes Projekt nicht allein mit PHP zu realisieren ist, kommen hier auch andere Sprachen wie CSS, JavaScript, XSL, SQL (MySQL) und die JS Framework JQuery und Prototyp zum Einsatz. XSL daher, weil wir hier mit einer Art virtuellem XML Datentransfer arbeiten werden, welcher XSL/XSLT als Layoutgrundlage benötigt. Dazu aber später mehr.

Weiterhin wird sehr viel Wert auf die objektorientierte Programmierung gelegt, also gekapselte Klassen definiert die im gesamten Projekt wiederholt Anwendung finden. Gekapselt deshalb um einen Zugriff von Außen zu unterbinden. Ich werde auf den Begriff der Kapselung im Kapitel *Kurze Einführung in die OO-Programmierung* noch näher eingehen. Der Grundgedanke oder ein gewisses Verständnis für die OOP Programmierung sollte daher schon vorhanden sein.

Die einzelnen Klassen werden wir selbst erstellen, also nicht auf vorgegebene Dateien aus z.B. dem Internet zugreifen. Diese Klassen entsprechen den neuesten Standards der OOP Programmierung mit PHP und können jederzeit in andere Projekte integriert und verwendet werden. Des Weiteren behandelt dieser Workshop die strikte Trennung von Datenverarbeitung und Layout, wie es in bekannten CMS Systemen Gang und Gebe ist. Die einzelnen Abschnitte gliedern sich daher

auch in Planung, Konzeption und Programmierung, wobei der Planungsbaum von Oben herab abgearbeitet wird.

Ein Vorteil dieser Vorgehensweise wird sein, dass Sie sich innerhalb des Projektes frei bewegen und schnell auf bereits fertige Teilabschnitte zugreifen können, ohne den Live Ablauf des Projektes im Internet nachhaltig zu stören.

Im ersten Teil werden wir ein CM-System erstellen, welches es uns erlaubt, die Textinhalte der späteren Websitedateien dynamisch zu bearbeiten ohne in den fertigen Quellcode eingreifen zu müssen. Auch wird aus diesem CMS heraus eine Möglichkeit eingebunden, die eigentlichen Websitedateien (XSL, PHP) mit ihrem hauptsächlichen Seitenaufbau ohne großen Aufwand zu erstellen und in den jeweiligen Ordnern zu speichern, was uns im späteren Verlauf des Projektes eine Menge an Programmierarbeit erspart.

Natürlich werden wir auch auf SEO (Suchmaschinenoptimierung) Rücksicht nehmen, sprich Möglichkeiten schaffen, die z.B. eine SE-Optimierte Linkmaskierung beinhaltet oder auch Änderungen der gesamten Textstruktur, die für das Indexieren auf gängigen Suchmaschinen wichtig sind (Alt-Tags, Descriptions, Title usw.).

Am Ende dieses Workshops werden Sie in der Lage sein, ein vollständiges Internetprojekt, das den Ansprüchen jeder Internetagentur gerecht wird, selbst zu erstellen. Ihre hierbei erarbeiteten Grundscripte können ohne Änderungen so auch für andere Projekte übernommen werden. Allerdings gebe ich zu bedenken, auch wenn Sie das gesamte Buch durchgearbeitet haben und mit den Bausteinen hieraus immer neue Projekte anlegen können, Sie sich nicht hier drauf auszuruhen, sondern sich weiterhin mit allen Neuerungen bezüglich der Webprogrammierung befassen sollten, um nicht irgendwann den Anschluss zu verlieren. Auch sollten Sie darauf achten, die einzelnen Abschnitte dieses Workshops zu verstehen, um später eventuelle Änderungen an Ihren Grundgerüsten ohne Schwierigkeiten vornehmen zu können. Gehen Sie immer davon aus, dass Sie nicht jeden Tag ein neues Projekt anlegen und es Monate dauern kann, bis Ihre Arbeit zur Zufriedenheit aller fertig gestellt ist. Hierbei werden Sie höchstwahrscheinlich im Livebetrieb permanent in Ihre eigene Programmierung eingreifen müssen, wenn Neuerungen oder Änderungen von Nöten sind. Des Weiteren kann es sein, dass Sie nicht alleine an einem Projekt tätig sein werden. Ihre Mitarbeiter/Kollegen werden dann von Ihnen wissen wollen, wie Ihre Programmierung funktioniert oder was beim Ablauf genau passiert. Aber keine Angst, der Abschnitt ‚Pflichtenheft' wird Sie hierbei unterstützen.

Selbstverständlich können Sie alle im Buch vorgestellten Scripte auf der zum Buch gehörenden Website herunterladen, z.B. wenn Sie keine Lust haben die gesamten Codes eigenhändig zu schreiben.

Doch bedenken Sie, dass nur selbst geschriebene Programme richtig verstanden werden. Der Lerneffekt ist immer ein wesentlich besserer, wenn man etwas mitschreibt, als es nur zu lesen.

Nun wünsche ich Ihnen viel Spaß bei der Reise von der Idee bis zum fertigen Produkt eines gepflegten Internetauftritts.

Kurz etwas über mich….

Meine Programmiererlaufbahn begann im Jahre 1979. Als die ersten Heimcomputer aufkamen war es ein Atari 600 XL, der meine ganze Aufmerksamkeit auf sich zog. Da ich von Anfang an ein reges Interesse daran hatte zu erfahren, wie solch ein Wunderwerk arbeitet und dazu natürlich auch die Programmierung gehörte, war es selbstverständlich mit dem damaligen Atari Basic alles aus einem Homecomputer herauszuholen, was dieser hergab. Da es zu dieser Zeit so gut wie keinerlei Fachliteratur für die Programmierung solcher Maschinen gab, brachte ich mir diese Sprache unter Mithilfe eines damals angebotenen Zeitschriftenkurses selbst bei.

Im Laufe der Zeit und mit der Hilfe anderer ‚Programmierer' gelang es mir schon damals, recht komplexe Programme z.B. für die Lagerhaltung zu schreiben.

Nach und nach wurden die Systeme größer und leistungsstärker und verfügten über immer neue Programmiersprachen für die unterschiedlichsten Bereiche.

Nach einiger Zeit war allerdings mein Wissensdurst gestillt, und ich sah keine neuen Herausforderungen in der Programmierung auf diesen doch recht einfachen Plattformen. Selbst das damals genutzte Windows 3.1 brachte keine neuen Anreize. Erst das Aufkommen des Internets weckte meine alte Leidenschaft und führte mich stückweise in die Programmierung von Webanwendungen.

1999 legte ich an der Volkshochschule Pinneberg ein Zertifikat für Internetprofis ab, welches ich im Anschluss als Einstieg sah, mein Hobby zum Beruf zu machen.

Somit begann ich im Jahre 2000 eine Ausbildung zum Fachinformatiker für Anwendungsentwicklung, welche ich 2002 mit Erfolg abschloss.

In einer Hamburger Werbeagentur machte ich erste Erfahrungen mit VBA für Access, PHP, Perl und der komplexen Programmierung dynamischer Webseiten mit Datenbankanschluss. Mein Wissen über PHP 4/5 sowie CSS, den Datenbanken MSSQL; MySQL und Postgres sowie JavaScript als clientseitige Scriptsprache vertiefte ich dann in einem Aachener Unternehmen, welches sich auf die Online Ticketbuchung, Messe und Kongressplanung sowie Erstellung von komplexen Internetprojekten für diverse Kunden spezialisiert hatte.

Hier war ich auch für die Netzwerkbetreuung auf Basis von Windows Server 2003 und Linux, (nicht gerade mein Steckenpferd) sowie für die einwandfreie Funktionalität der einzelnen Arbeitsplätze (26 an der Zahl) zuständig. Diverse eigene Internetprojekte, die teilweise auch heute noch recht zahlreich besucht werden, rundeten meine Arbeiten mit den genannten Sprachen ab. Dabei wurde das Coden mit PHP immer mehr vom OOP-Gedanken geprägt und vereinfachte meine Arbeiten ungemein. Auch heute noch lasse ich es mir nicht nehmen, verschiedenste Lösungen zu den unterschiedlichsten Problemen für das Web oder den heimischen PC zu programmieren.

Meine Java-Zertifikate mit anhängigem Studium von Java 5/6 für Applikationen und Applets legte ich als bisherigen Abschluss meiner Karriere im Jahre 2010 in Hamburg ab.

Zurzeit arbeite ich in einer Hamburger Internetagentur als Programmierer für diverse Onlineprojekte, zum Teil auf Basic von Templatesystemen, die überwiegend mit PHP, XML/XSL/XSL, JavaScript sowie das JavaScript Framework Prototyp erstellt werden.

To be continued……

Inhaltverzeichnis

1 Verwendete Sprachen

1.1 Wir programmieren mit...

Dass komplexe Intenetprojekte mit sogenannten Scriptsprachen realisiert werden, ist nicht erst seit gestern bekannt. Da HTML alleine nicht über die Funktionalität verfügt, z.B. Formulareingaben zu verarbeiten oder dynamisch Inhalte aus Datenbanken nachzuladen und anzuzeigen, gibt es Sprachen die das in Koorperation mit HTML bewerkstelligen können. Allen voran Perl, der Vorläufer von PHP und eben PHP, welches sich einer immer größer werdenden Beliebtheit erfreut, sind hier federführend.

PHP wird auf Grund der großen Beliebtheit ständig weiterentwickelt (derzeit Version 5) und knüpft mittlerweile an rein objektorientierte Sprachen wie JAVA an.

XHTML als Vermischung von HTML und XML ermöglicht es, eigene Elemente in die Programmierung von Webseiten einzubinden und erweitert somit die Anwendungspalette von HTML.

XSL (Extensible Stylesheet Language) ist eine in XML notierte Familie von Transformationssprachen zur Definition von Layouts für XML-Dokumente.

Die XSL-Subsprache XSLT wird außerdem zur Übersetzung/Transformation eines XML-Formats in ein anderes XML- oder Textformat genutzt.

Nicht zu vergessen ist auch JavaScript, als reine Clientsprache, die mittlerweile auch auf externe Quellen zugreifen kann.

AJAX (*Asynchronous JavaScript and XML)* ist hierbei ein Konzept der asynchronen Datenüberragung zwischen Browser und Server. Dieses ermöglicht es, HTTP-Anfragen durchzuführen, während eine HTML-Seite angezeigt wird, und die Seite zu verändern, ohne sie komplett neu zu laden. Des Weiteren lassen sich mit JavaScript komfortabel z.B. Formulare testen, ebenfalls ohne die Webseite ständig neu zu laden.

All diese neuen Techniken geben dem Entwickler Instrumentarien an die Hand, die es ihm ermöglichen, mit wenig Aufwand anspruchsvolle Webauftritte zu generieren. Wir werden uns diese Sprachen und Techniken ebenfalls zu nutze machen, um ein an Funktionalität und Dynamik anspruchsvolles Projekt zu erstellen.

P. Stender, *Webprojekte realisieren nach neuesten OOP-Kriterien,* DOI 10.1007/978-3-8348-8114-4_1,

Neben PHP werden wir hauptsächlich auf XSL/XSLT zurückgreifen und JavaScript respektive AJAX nutzen. Im Anschluss möchte ich kurz die einzelnen Sprachen, die wir verwenden, vorstellen.

1.1.1 (X)HTML

Die Sprache bedarf eigentlich keiner genaueren Erklärung, da sie Grundlagen aller Websites darstellt, ohne sie also nichts im Internet laufen würde. Ich möchte daher hier auch nur auf die spezielle Codierung als Standard für unser Projekt eingehen.
Der eigentliche Kopfbereich unserer HTML-Seiten, wenn ich sie mal so nennen darf, besteht im Wesentlichen aus der Einbindung der HTML Doctypes in ein XSL Stylesheet. Die reine HTML-Codierung wird später über die Klasse Template implementiert.

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns="http://www.w3.org/1999/xhtml"
        xmlns:fn="http://www.w3.org/2005/xpath-functions"
        xmlns:php="http://php.net/xsl">
<xsl:output
        method="xml"
        encoding="utf-8"
        indent="yes"
        doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-
        transitional.dtd"
        doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
        omit-xml-declaration="yes"/>
<!-- Templatebereich -->
</xsl:stylesheet>
```

Innerhalb des XSL-Stylesheets können natürlich alle HTML-Tags eingesetzt werden, allerdings mit der Einschränkung, dass sie zwingend (falls geöffnet) auch wieder geschlossen werden müssen.

1.1.2 CSS

Mit Hilfe vom Cascading Style Sheet (CSS) lassen sich HTML-Elemente exakt positionieren und Webseiten grafisch aufbessern, was durch reines HTML selbst nicht möglich ist. CSS ist hierbei eine Ergänzung zu HTML und basiert auf dem Klassenprinzip.

Ein Vorteil des CSS liegt in der Tatsache begründet, dass HTML-Elemente zentral bestimmten Layoutregeln vorgegeben werden können, die im gesamten Projekt Anwendung finden, ohne direkt dem Element zugeordnet werden zu müssen.

Beispiel:

```
TD {
   border:1px solid #000;
}
```

Dem HTML-Element TD, welches Bestandteil einer Tabelle ist, wird die Eigenschaft Border zugeordnet. Ergebnis ist, dass alle mit TD notierten Tags innerhalb Ihres Projektes nun über einen Rahmen verfügen, der 1 Pixel stark ist und die Farbe Schwarz hat. Weiterhin können Styleklassen und Styles generiert werden die über class="" oder id="" direkt in den HTML-Tags notiert aufgerufen werden.
Beispiel:

```
#TD {
   border:1px solid #000;
}
```

Notiert den Tag TD für den Aufruf mit id.

Beispiel:

```
.TD {
   border:1px solid #000;
}
```

Notiert den Tag TD für den Aufruf als `class`.

1.1.3 PHP

Serverseitige Scriptsprache, die Daten verarbeitet und geparst als HTML an den Browser zurückgibt. Sie wird überwiegend für dynamische Webseiten und Webanwendungen eingesetzt. PHP wird, wie die Bezeichnung schon sagt, auf dem Server geparst, sprich übersetzt, wobei als Ergebnis reiner HTML-Code zurückgegeben wird. Seit Sommer 2004 wurde die Version 5 als erweiterte Entwicklungsstufe veröffentlicht und ständig verbessert. Sie zeichnet sich mittlerweile durch ein verbessertes Objektmodell aus und führt somit objektorientierte Anwendungen effizienter aus. Sprachkonstrukte wie Überladung werden genauso ermöglicht wie Exceptions, Reflections, die Integration der SQLite-Datenbank sowie Erweiterungen bei XML- und DOM-Handhabung.

1.1.4 XSL

XSL dient einer XML-Datei als Layoutgrundlage. Sie ist in der XML-Familie eine notierte Transformationssprache zur Definition von Layouts für übergebene XML-Dokumente. Da XML-Daten zwar angezeigt werden, aber ihre Inhalte nicht grafisch dargestellt werden können, benötigt man hier eine Art Layoutcontainer, der die

XML-Daten in ein grafisches Gerüst verpackt. Dieses bewerkstelligt XSL, welches als Ausführungssprache diverse Befehle mitbringt, die eine gezielte Abfrage der XML-Daten ermöglicht.

Beispiel, Auflösung eines Arrays aus einem XML-Wurzelknoten:

```
<xsl:for-each test="bla">
   <xsl:value-of select="bla1" />
</xsl:for-each>
```

1.1.5 JavaScript

Programmiersprache. die clientseitig direkt im Browser ausgeführt wird. Mit ihrer Hilfe lassen sich diverse Ansichtsoptionen kreieren und Daten im Hintergrund vorladen, um diese später recht komfortabel in den von Server übergebenen HTML-Code einzubinden, ohne die Seiten ständig neu laden zu müssen. Auch unterstützt JavaScript als Bestandteil die AJAX-Technologie.

1.1.6 jQuery

jQuery ist ein JavaScript Framework welches komfortable Funktionen zur DOM-Manipulation und -Navigation zur Verfügung stellt. *JQuery* ist die meistverwendete JavaScript-Bibliothek. Andere Bibliotheken sind z.B. *Prototyp*, die wir auch bedingt einsetzen werden.

1.1.7 XML (wird nicht direkt programmiert)

Die Extensible Markup Language (XML) ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten. Die Strukturelemente von XML lassen sich frei definieren und werden als Knotenelemente angegeben. Die einzelnen Knotenelemente beinhalten die eigentlichen Daten, die in Knoten oder Unterknoten dargestellt werden können.

Die allgemeinen XML-Regeln sehen wie folgt aus:

1. Das Dokument besitzt genau ein Wurzelelement. Als Wurzelelement wird dabei das jeweils äußerste Element bezeichnet, z.B. <html> in XHTML.
2. Alle Elemente mit Inhalt besitzen eine Beginn- und eine End-Kennung z. B. <data>Data</data>. Elemente ohne Inhalt können auch in sich geschlossen sein, sofern sie über keinen Inhalt verfügen <data />.
3. Alle Elemente die geöffnet sind, müssen auch wieder geschlossen werden.
4. Ein Element darf nicht mehrere Attribute mit demselben Namen besitzen.

Da wir XML-Daten nicht direkt erstellen, sind diese Informationen nur von geringer Bedeutung für uns. Die Generierung der einzelnen Knotenelemente wird nachher recht komfortabel durch unsere Templateklasse erledigt.

2 Kurze Einführung in die OO-Programmierung

Die objektorientierte Programmierung (OOP) ist ein Programmierstil auf Basis des Konzeptes der Objektorientierung Die Idee dabei ist, alle Bestandteile eines Objektes in eine logische Struktur zu bringen und zusammenzufassen. Des Weiteren werden Daten und Funktionen gekapselt (Datenkapselung), was heißt, dass versehentliche Manipulationen von außen weitestgehend unterbunden werden. Unter Kapselung oder Datenkapselung versteht man somit das Verbergen von Daten nach außen, um sie vor direktem Zugriff zu schützen.

Stellen wir uns vor, wir haben eine Klasse *Mensch*. Die Klasse ist dabei das Objekt. Eine Methode dieses Objektes könnte sein, die Augenfarbe zu bestimmen. Eine andere bestimmt die Haarfarbe, die Hautfarbe usw. Weiterhin haben wir eine Datenbank, die diverse Informationen zu bestimmten Völkergruppen beinhaltet. Möchte ich nun alle dunkelhäutigen Menschen aus der Datenbank herausfiltern, gebe ich der Methode Hautfarbe des Objektes Mensch die Farbe Braun mit und bekomme als Ergebnis alle dunkelhäutigen Menschen aus der Datenbank geliefert.

Ein weiterer Vorteil neben der Kapselung ist hierbei, dass ich den für das Ergebnis verwendeten Programmabschnitt (Methode) nur einmal komplett ausprogrammieren und dann in jeder weiteren Datei nur noch die zum Suchen benötigten Parameter an das Objekt übergeben muss. Die gesamte Programmierung wird dadurch verkleinert und übersichtlicher in ihrem Aufbau. Durch weitere *Kapselung* und *Vererbung* ist es sogar möglich, einzelne Methoden eines Objektes weiter zu verfeinern. Ein solches Beispiel wäre eine Klasse *Säugetiere* (zu denen zweifelsohne auch der Mensch gehört). Die Klasse *Mensch* würde nun durch Vererbung von der Klasse *Säugetiere* alle Eigenschaften dieses Objektes und seiner Methoden übernehmen (erben). Suchen wir also alle dunkelhäutigen Menschen aus unserer Datenbank, würde das Objekt *Säugetiere* seine Methode zum Finden der Gruppe Mensch aus seiner Tabelle *Säugetiere* an die Klasse Mensch weitergeben, wobei das Objekt *Säugetiere* nicht direkt angesprochen werden muss. Klingt recht verwirrend, wird aber, wenn Sie weiter lesen deutlicher.

Die Planung solcher Datenmodelle ist extrem wichtig und vergleichbar mit der Normalisierung von Datenbanken. Für unseren Fall heißt das, dass wir z.B. eine Klasse *User* und eine Klasse *Userlogin* erstellen werden, die beide unterschiedliche Aufgaben haben, aber teilweise dieselben Methoden benötigen. Die Klasse *User* fragt

P. Stender, *Webprojekte realisieren nach neuesten OOP-Kriterien*, DOI 10.1007/978-3-8348-8114-4_2,

die Parameter Name, Username, Passwort usw. ab, während die Klasse *Userlogin* den Usernamen und das Passwort zum Abfragen der Logineingaben des Users benötigt. Da wir die Methoden Username und Passwort in der Klasse *User* angelegt haben, wäre es unsauber und unnötig, diese auch noch in die Klasse *Userlogin* zu schreiben, weil durch Vererbung die Klasse *Userlogin* auf die Methoden Username und Passwort der Klasse User zugreifen und diese in vollem Umfang nutzen kann. Nun habe ich den Begriff der *Vererbung* bereits mehrfach aufgezeigt. Die Funktionsweise der Vererbung beinhaltet nichts anderes, als dass bestimmte Methoden an andere Objekte abgegeben werden, damit diese darauf zugreifen und damit arbeiten können. Weitere Begriffe in der Objektorientierten Programmierung sind folgende:

Abstraktion
Als Abstraktion bezeichnet man in der Programmierung alle Objekte, die Aufgaben erledigen, Zustände weitergeben oder mit anderen Objekten kommunizieren. Hierbei müssen nicht zwingend die Implementierungen der einzelnen Fähigkeiten offen gelegt werden. Solche Abstraktionen sind z.B. Klassen.

Datenkapselung
Als Datenkapselung bezeichnet man in der Programmierung das Verbergen von Daten für einen direkten Zugriff von außen. Der direkte Zugriff auf Datenstrukturen wird unterbunden und durch definierte Schnittstellen geregelt.

Polymorphie
Ist ein Konzept in der Programmierung, das die Fähigkeit eines Bezeichners beschreibt, abhängig von seiner Verwendung, unterschiedliche Datentypen anzunehmen.

Vererbung
Haben wir bereits ausführlich erläutert.

Persistenz
Solange Objekte vorhanden sind, existieren auch deren Objektvariablen. Sie verfallen nicht nach Abarbeitung der Methoden.

Klassen
Zur besseren Verwaltung gleicher Objekte nutzen Programmiersprachen, die Objektorientierung unterstützen, so genannte Klasse. Klassen sind Vorlagen, aus denen Instanzen zur Laufzeit erzeugt werden. In der Softwareprogrammierung werden dann keine einzelne Objekte, sondern eine Klasse gleicher Objekte definiert. Die Klasse entspricht dabei einem komplexen Datentyp. Sie legt die Datentypen fest, aus denen mit Hilfe der Klassen erzeugte Objekte bestehen. Gleichzeitig definiert sie die

Algorithmen, die auf diesen Daten operieren. Zur Laufzeit interagieren einzelne Objekte miteinander, so dass, das Grundmuster dieser Interaktion durch die Definition der einzelnen Klassen festgelegt wird.

Programmiersprachen, die dieses Konzept am besten umsetzen, sind z.B. Java und C#. PHP steckt hier noch in den Kinderschuhen, was deutlich wird, wenn man sich die in Java komplett ausgereizten Fehlerfälle (exceptions) betrachtet. Definieren Sie in Java einen Try-Catch-Block und wollen damit bestimmte Fehlerfälle abfangen, so hat Java für jeden nur erdenklichen Fehlerfall die richtige Antwort parat. PHP hingegen verfügt nicht über solche speziellen Fehlerausgaben, die in einer Funktion oder Methode verwendet werden können. Fehlermeldungen, die der Parser an den Browser zurückgibt, sind meist recht umständlich zu lesen und nur schwer zu verstehen wenn man nicht genau weiß, wo der Fehler tatsächlich aufgetreten ist. Hier muss man zwingend eigene Exceptions von Hand in eine Klasse schreiben, um sie innerhalb einer Try-Catch-Anweisung zu nutzen. Das bedeutet im Umkehrschluss, dass die vom Parser zurückgegebenen Fehlermeldungen analysiert und für eigene Zwecke angepasst werden müssen. Dies aber nur am Rande.

Sie sehen, dass die OO-Programmierung recht umfangreich ist, wenn man aber deren Sinn verstanden hat, diverse Vorteile liefert. Im weiteren Verlauf unseres Projektes werden Sie verstehen, was diese Vorteile sind. Sollten Sie sich tiefer mit dieser Materie auseinandersetzen wollen, empfehle ich hier einschlägige Literatur im Internet. Für unsere Zwecke reicht das bis hier aufgezeigte erst einmal vollkommen aus.

Kurz anmerken möchte ich noch, wie eine Klasse in PHP aufgebaut ist. Diese kurze Einführung soll Ihnen einen Einblick in die Quelltexte der Klassen vermitteln, die Grundlage dieses Projektes sind. Sie werden im gesamten Verlauf des Buches immer wieder auf das Aufgezeigte treffen und selbst damit arbeiten.

Eine Klasse in PHP besitzt Methoden zur Datenmanipulation und wird mit Schlüsselwort class definiert. Jede Klasse hat einen Konstruktor, der wie eine Funktion deklariert ist und den gleichen Namen wie die Klasse selbst haben muss. Um eine neue Instanz einer Klasse anzulegen, wird das Schlüsselwort new verwendet, das den Konstruktor der jeweiligen Klasse aufruft. Das Schlüsselwort this ist ein Zeiger auf sich selbst und wird innerhalb von Objekten eingesetzt. Methoden sind interne Funktionen eines Objektes und dienen zum Zugriff und zur Manipulation von Feldern. Membervariablen so genannte Eigenschaften und Felder, die außerhalb des Konstruktor definiert werden, können keine Ausdrücke innehaben, dies geht nur innerhalb eines Konstruktor, allerdings muss hier der Zeiger this mit eingebracht werden. Die einzelnen Methoden und auch die Membervariablen können bei der