Objektorientierte Programmierung im Anfangsunterricht am Beispiel von PHP

Henry Becker

23rd October 2002

Contents

0.1 Fragestellung 2
0.2 Umfang 2
0.3 Vorwort 3

1 Grundlagen **4**
1.1 Programmierung 4
1.2 Objektorientierte Programmierung (OOP) 5
1.3 PHP 6

2 Richtziele **7**
2.1 Abstraktion 7
2.2 Modellbildung 7
2.3 Entwurf 8
2.4 Implementierung 8

3 Grobziele **9**
3.1 Grundlagen PHP 9
3.1.1 Begründung / Lernziel 9
3.1.2 fachliche Fundierung 9
3.2 Objekte - Modellierung und Implementierung 10
3.3 Kontrollstrukturen PHP 11
3.3.1 Begründung 11
3.3.2 Lernziele 11

3.3.3 Fachliche Fundierung (Syntax) 11
3.4 Projekt mit PHP 12

4 Stoffverteilungsplan **14**

5 Stundenentwürfe **16**
5.1 aus Lernabschnitt III - Objekte 16
5.1.1 Stundenthema 16
5.1.2 Dauer 16
5.1.3 Vorkenntnisse 16
5.1.4 geplanter Unterrichtsverlauf 17
5.1.5 Lernziele 18

6 Einschätzung von PHP und OOP im Unterricht **19**
6.1 Didaktische Reduktion und Motivation 19
6.2 Objektorientierung 20

0.1 Fragestellung

In diesem Dokument gehe ich der Frage nach, was objektorientierte Programmierung “von Anfang an” bedeutet. Wie sinnvoll ist OOP im Unterricht? Wie kann sie mit der Scriptsprache PHP eingesetzt werden?

0.2 Umfang

Dieser Entwurf ist ausgerichtet auf einen Kurs im Anfangsunterricht Informatik. Das kann je nach Bundesland variieren.

Der Stoffverteilungsplan richtet sich nach 2 Unterrichtsstunden pro Woche. Es stehen etwa 34 Stunden zur Verfügung.

0.3 Vorwort

Mir geht es in dieser Semesterarbeit nicht vorrangig um einen ausgereiften Lerplanentwurf. Schwerpunktmässig soll diskutiert werden, ob eine Sprache wie PHP im Schulunterricht eine Berechtigung hat.

Chapter 1

Grundlagen

1.1 Programmierung

Ein Diskussionspunkt in Informnatikerkreisen wird von mir wie folgt definiert:
Programierung ist

- das Begreifen eines Sachverhaltes
- die Analyse des selbigen
- die algorithmische Lösung
- sowie die Implementierung dieser Lösung.

Es wird betont, dass alle einzelnen Punkte voneinander zu trennen sind. D.h. wenn ein Problem analysiert wird, darf noch nicht an dessen Implementierung gedacht werden, bzw. die Implementierung darf die Analyse nicht konditionieren.

Zu jedem der von mir aufgestellten Punkte gibt es Mittel, diese zu bewältigen. Um einen Sachverhalt zu verstehen, bedarf es der menschlichen Intelligenz und konkret allgemeiner Bildung. Die Analyse kann ebenfalls auf meheren Wegen beschritten werden. Aus der Informatik ist hierfür zum Beispiel UML als Modellierungsprache hervorgegangen. Nun ist es wichtig einen Algoritmus zu finden

bzw. zu beschreiben. Im Idealfall darf erst jetzt eine Programmiersprache gesucht werden. Der eigentliche Lösungsprozess, die Algorithmisierung, wird häufig gar nicht als eigener Prozess betrachtet oder wahr genommen. Oft ist es so, dass erst wärend der Implementierung ein “passender” Algorithmus gefunden wird.

1.2 Objektorientierte Programmierung (OOP)

Nun, es gibt etliche Definitionen zur objektorientierten Programmierung. Ich möchte micht nicht für eine entscheiden. Larry Wall hat in seinem Buch “Programmieren mit Perl” folgendes geschrieben:“Ein Objekt ist eine Datenstruktur in Kombination mit einer Reihe von bestimmten Verhaltensmustern, die mit der Datenstruktur möglich sind. Üblicherweise reden wir davon, daß diese Verhaltensweisen von den Objekten direkt ausgeführt werden, was manchmal zu einer Vermenschlichung des Objektes führen kann. Zum Beispiel sagen wir, daß ein Rechteck ’weiss’, wie es sich selbst auf dem Bildschirm auszugeben hat, oder daß es ’weiss’, wie es eine Fläche zu berechnen hat.”

Ich fand diese Definition deshalb so schön, weil er es damit umgeht, einen wesentlichen Unterschied zwischen Attribut und Methode zu machen. Attribute, welche ein Objekt beschreiben (z.B. Farbe, Grösse) und Methoden, die ein Objekt zu Aktionen bewegen (z.B. laufen, blinken) sind doch eigentlich nur, wie er so schön sagt, “Verhaltensweisen”. Er schreibt weiter:„Jedes Objekt erhält seine Verhaltensweisen, indem es eine Instanz einer Klasse wird.”

Auch wurde erwähnt, dass es sich um eine Datenstruktur handelt. Das heisst, ein Objekt ist neben seinen Funktionen und Attributen immer mit seinen eigenen, speziellen und konketen Daten verbunden.

Somit haben wir den Kern der Objektorientierung erfasst. Objekte sind immer Kopien einer ganzen Klasse von Objekten. Verebung ist das Übernehmen von Methoden bzw. Attributen übergeordneter Objekte. Ein Objekt quasi von seinem Mutterobjekt Verhaltensweisen. Letztendlich sind alle Objekte immer Unterobjekt von einem anderen.

Haben mehrere bzw. alle Objekte eine gleiche Eigenschaft, handelt es sich um Polymorphie (Gleichgestaltigkeit) . Des weiteren ist die Kapselung von fundamentaler Bedeutung in der OOP. Kapselung bedeutet, dass es zu jedem Objekt eine öffentliche Schnittstelle gibt (Methodenbezeichner, Variablen) und eine private verdeckte Implementierung.

1.3 PHP

PHP (Hypertext Preprocessor) ist eine serverseitige Scriptsprache. Mit serverseitig werden im Internet viele Vorteile verbunden. Das eigentliche Programm läuft in der Regel zunächst nicht auf dem eigenen Rechner. Eine Tasache, die nicht zu unberechtigten Dateizugriffen auf den eigenen Rechner führen kann. Darüber hinaus wird auch keine Rechenkapazität vom Internetclient erfordert. Ein möglicher Nachteil ist die permernete Internetanbindung bei der Ausführung des Programmes.

PHP wird in HTML eingebettet. An dieser Stelle möchte ich nicht das weltberühmte “Hello World” Programm in PHP verheimlichen. HTML-Code: *<? echo “Hello Word”; ?>* Dies als *.php Datei abgespeichert und als Website aufgerufen[1] schon ein vollständiges PHP-Programm. Wobei *<?* und *?>* gerade noch HTML-Tags sind. Die PHP-Programme werden mit einem gewöhnlichem Editor geschrieben. Eine Entwicklungsumgebung gibt es für PHP meines Wissens nicht. Eventuell wird es Editoren geben, die Syntax-Highlighting für PHP beherrschen. Die Syntax von PHP ähnelt der von C, Java und Perl.

[1] Gemeint ist ein http-Zugriff.

Kapitel 2

Richtziele

2.1 Abstraktion

Die Schüler sollen lernen jedes im Rahmen des Informatikunterrichtes gestellte Problem zu abstrahieren. Bereits hier sollen Aspekte der Objektorientierung greifen. Durch den Abstraktionsprozess soll die Frage beantwortet werden, was ein Objekt ist und wie verhält es sich?

2.2 Modellbildung

Spätestens hier sollen die Schüler lernen informatische Probleme vollständig zu modellieren. Dabei ist die objektorientierten Vorgehensweise konkreten Charakters. und nicht nur die Benennung von Objekten sondern auch deren Klassifizierung. Dies gilt nicht nur für die Benennung von Objekten sondern auch für deren Klassifizierung. Es sind Beziehungen zwischen Objekten herzustellen. Ganze in sich geschlossene System sollen selbstständig modelliert werden können.

2.3 Entwurf

Hier zeigen die Schüler die Fähigkeit spezifische Probleme in dem modellierten System zu lösen. Dabei kommt es darauf an, für die Lösung einen Algorithmus zu finden.

2.4 Implementierung

Mit einer konkreten Programiersprache sollen die Schüler den von ihnen gefundenen Algorithmus zu einem lauffähigen Programm zu entwickeln. Dabei kommt es hier nicht hauptsächlich darauf an alle Details der verwendeten Programmiersprache zu kennen. Vielmehr sollen die Schüler soweit wie möglich ihren zuvor entwickelten Algorithmus wieder erkennen.

Kapitel 3

Grobziele

3.1 Grundlagen PHP

3.1.1 Begründung / Lernziel

Um ein “Gefühl” für den kausalen Zusammenhang zwischen dem Programmieren und den Computern im Allgemeinen bei den Schüler zu erzeugen, halte ich es für sinnvoll nachvollziehbare Programme vorzustellen. Ableitend davon soll der besondere Status von Scriptsprachen erwähnt werden. Damit verbunden ist das Vorgehen bei der Einbettung in HTML. Die Schüler sollen lernen einen Datentyp bestimmen zu können. Der Umgang mit Variablen wird geläufig.

3.1.2 fachliche Fundierung

Einbettung in HTML:

<html>

<body>

<?

// Das ist ein PHP-Kommentar

echo „<h1>Eine HTML U¨berschrift</h1>”;

?>

</body>

</html>

Variablen:

// Variablenbezeichner beginnen mit einem Buchstaben oder einem Unterstrich

$name = „Schmidt”;

Typisierung/Initialisierung der Variablen geschiet automatisch bei der Zuweisung.

3.2 Objekte - Modellierung und Implementierung

Es soll erreicht werden, dass der Begriff des Objektes vertraut wird. Ich will sagen, jeder Schüler kann sich etwas unter einem Objekt abstrakt vorstellen. Folgende Begriffe sind entscheidend: Klassen, Instanz, Vererbung, Attribut und Methode. Wobei deutlich wird, dass Objekte in Klassen zusammengefasst werden und Objekte immer Instanzen einer Klasse sind, Objekte Methoden usw. besitzen und genau diese Methoden an andere Objekte vererbt werden können. Noch besser gesagt, erben Objekte von anderen Objekten. Das hierarchische System wird den Schülern in diesem Zusammenhang deutlich.

Die Kapselung eines Objektes ist von großer Bedeutung und wird hier aus diesem Grund ausführlich behandelt. Ich empfehle jedes geschaffene Modell stets unmittelbar zu implementieren. Zwingend ist natürlich jedes Stückchen Code vorher zu modellieren.

“Modellierung als inhaltlicher Kern”[1] hat seine wissenschaftliche-didaktische Berechtigung. Niemals sollte auf die Implementierung verzichtet werden. Nicht

[1] Hubwieser

zuletzt deshalb, weil die Schülerinnen und Schüler selber etwas schaffen können, ist Informatik ein attraktives Fach.

3.3 Kontrollstrukturen PHP

3.3.1 Begründung

Kontrollstrukturen sind der „Lenker" eines jeden Programms. Sie werden dafür eingesetzt den Verlauf eines Programmes konditional zu steuern. Kurzum um ein komplexes Programm zu schreiben, ist das Wissen über sie unabdingbar.

3.3.2 Lernziele

Die Schüler sind in der Lage Kontrollstrukturen richtig einzusetzen. Insbesondere sind sie mit der Syntax selbiger vertraut. Als Kontrollstrukturen ist folgendes zu behandeln. Verzweigungen: IF (...) {...} ELSE {...}, SWITCH (...) {...} Schleifen: WHILE (...) {...}, DO {...}WHILE (...) und FOR (...;...;...) sind alle wichtigen Strukturen. Diese empfehle ich, losgelöst von Objekten bzw. dessen Begriffe zu vermitteln.

3.3.3 Fachliche Fundierung (Syntax)

if-bedingungen:

```
IF (Bedingung) {Anweisungen}
IF (Bedingung) {Anweisungen} ELSE {Anweisun-
gen}
IF (Bedingung) {Anweisungen} ELSEIF (Bedin-
gung) {Anweisungen}
```

Es ist nicht möglich IF ELSE Blöcke durch HTML zu trennen. Der PHP-Code muss immer vollständig sein. Folgender Code ist falsch:

```
<?
if ($name == ”schmidt”) {echo „Gu-
ten Tag Herr Schmidt”;}
?>
<font size=5>
<?
else {echo „Guten Tag Unbekannter”;}
?>
</font>
```

switch()-case Alternativen:

```
SWITCH (Ausdruck) {
CASE wert: {Anweisungen} break;
DEFAULT: {Anweisungen} break;}
```

while-Schleifen:

```
WHILE (Bedingung) {Anweisungen}
DO {Anweisungen} WHILE (Bedingungen)
```

for-Schleife:

```
FOR (Zuweisung;Bedingung;Zuweisung)
{Anweisung}
```

3.4 Projekt mit PHP

Hier sollen die Schüler in die Lage versetzt werden ein komplexes Problem zu lösen. Es ist mir wichtig, dass die Schüler nicht einzelne Sprachelemente ohne

wirklichen Sinnzusammenhang lernen. Es wird erwartet, dass die Schüler mit Hilfe der objektorientierten Modelllierung (OOM) das Problem analysieren und modellieren. Auf eine vollständige Modellierung wird Wert gelegt. Dabei empfehle ich, die Modellierung von den Schülern erarbeiten zu lassen. Es ist jedoch sinnvoll fertige Alternativen bereit zu halten, damit im Zweifelsfall genug Zeit für die Implementierung bleibt.

Hier ein Vorschlag für ein Projekt: simpler webbasierter Email-Client. Ich halte das Projektthema für sehr motivierend, denn oft haben Schüler alltäglichen Umgang mit diesem Medium. Nun sollen sie lernen, was hinter den Kulissen geschieht.

Kapitel 4

Stoffverteilungsplan

Lernabschnitt	Stunden	Inhalt
I	3	Wiederholung • Vorstellung der Verlaufsplanung, sonst. organisatorische Fragen • Wiederholung der Kenntnisse zum Internet. • Kenntisse zu HTML (Beschreibungssprachen)
II	4	PHP • Scriptsprachen • Einbettung in HTML • Datentypen • Variablen
III	10	Objekte • Analyse • Modelllierung • Implementierung
IV	6	Kontrollstrukturen • IF (...) ... ELSE ... • SWITCH (...) ... • WHILE (...) ... • DO ... WHILE () • FOR (...;...;...) ...
V	11	Projekt • webbasierter Email-client

Chapter 5

Stundenentwürfe

5.1 aus Lernabschnitt III - Objekte

5.1.1 Stundenthema

Das Thema der Doppelstunde ist Objektorientierte Analyse (OOA) und deren Implementation (OOP).

5.1.2 Dauer

90 min. (Doppelstunde)

5.1.3 Vorkenntnisse

Begriffe rund um den Objektbegriff. Die nötige PHP Syntax.

5.1.4 geplanter Unterrichtsverlauf

Zeit	Inhalt	Bemerk.	Methode
1 min	Begrüssung		
	Lehrer: "Wie können wir Autos klassifizieren?"	Motivation	
10 min	Schüler und Lehrer erarbeiten an der Tafel ein Diagramm(hieraschisch) zu Autos. Ergebniss: Diagramm mit Bezeichnung von Klasse(Autos), Objekte/Instanzen(LKW)(PKW) Vererbung wird gekennzeichnet. Weiterhin jeweils noch 2 konkrete "Autos" bzw. "LKW's" sollen benannt werden	Ergebnisse werden sofort diskutiert. Was genau ist Vererbung ?	UG,Tafel
10 min	Lehrer entwickelt an der Tafel die Implementation dieser Klasse		LV,UG,Tafel
15 min	Schüler entwickeln selbständig auf dem Papier einen Entwurf für eine Klasse „Hund" mit möglichen Attributen und Methoden	Festigung	SA
9 min	Anschreiben einer Lösung an die Tafel. z.B.: Klasse(HUNDE), Methoden(BELLEN,WEDELN) und ATTRIBUTE(FARBE,REINRASSIG)	Ergebnisse werden nicht diskutiert wenn sie nicht falsch sind. Schüler sollen selber zusammentragen.	SV
15 min	Implementieren der Klasse "Hund"	Schüler arbeiten mit ihrer eignen PHP - Dokumentation	SA
5 min	Anschreiben der Ergebnisse (Implementation)		SA
5 min	Lehrer zeigt wie ein Objekt instanziiert wird:z.B. $waldi = new HUND(...);	Schüler arbeiteten unmittelbar am Rechner mit. Synchronisieren die Eingaben des Lehrers	UG, VB

5.1.5 Lernziele

Die Schüler:

- sind in der Lage Dinge abstrakt als Objekte zu begreifen
- können Objektbestandteile bennen und kennzeichnen
- können Objektklassen als Diagramme modellieren
- können Klassen definieren
- können Methoden und Attribute in Klassen definieren.

Chapter 6

Einschätzung von PHP und OOP im Unterricht

6.1 Didaktische Reduktion und Motivation

PHP selbst ist meiner Meinung nach didaktische Reduktion. Diese Sprache besitzt keine unnötigen aber dennoch eindeutigen Formatierungen. Das weit verbreitete Konzept der Blockkennzeichnung mittels geschweifter Klammern ist auch in PHP realisiert. In der Regel sind Entwicklungsumgebungen überladen mit Funktionalität. Funktionalität, die für Unterrichtszwecke keinesfalls nötig ist und häufig zu Missverständnissen führt. Ich finde auch, dass das subjektive Empfinden selbst ein Programm geschrieben zu haben, in Entwicklungsumgebungen wie z.B. von Delphi nicht motiviert wird. Meiner Ansicht nach gleicht es eher einem Malprogramm.

Bei PHP bzw. vergleichbaren Scriptsprachen haben die Schüler die Möglichkeit, bewusst aus einer „Sache" auszuwählen und nicht auf vorfertigte Dinge zurückgreifen zu müssen. Die Programmierung nach dem Prinzip „try and error" wird nicht angeregt. In der Tat ist es so, dass PHP durch eine fehlende Entwicklungsumgebung keine aktive Hilfe[1] bietet. Die Lernenden sind damit oberflächlich

[1]Möglichkeit in vielen Entwicklungsumgebungen, z.B. Delphi, nach Methoden mit Drop-

gesehen benachteiligt. Ich halte es für sinnvoll, dass die Inhalte (Syntax) gelernt werden und das Lernen sich nicht auf die Auswahl selbst beschränkt. Die aktive Hilfe unterstützt den professionellen Programmierer bei der Schreibarbeit. Die Möglichkeit von Hilfeseiten (Manualpages) ist bei PHP selbstverständlich. Nur empfehle ich die wenigen Schlüsselwörter von PHP als Kopien an die Schüler zu verteilen.

6.2 Objektorientierung

Ich denke, Objektorientierung trägt dazu bei, dass das Wissen kognitiv gefestigt wird. Objektorientierte Vorgehensweise zwingt die Programmierer und damit die Schüler zur „sauberen" Programmierung - ein nicht zu unterschätzender Vorteil.

Bei größeren Projekten macht sich die Wiederverwendbarkeit von Objektklassen bezahlt. So sind Projekte über Jahrgangsstufen hinweg realisierbar.

Die relativ lange Erstellung gegenüber iterativen Programmen könnte als Nachteil angesehen werden. Meiner Meinung nach ist diese am Anfang notwendige Mehrarbeit so gering, dass sie mir nicht als relevant erscheint. Hinzu kommt, dass sich im späteren Verlauf der Umfang der Schreibarbeit verringert.

Down Feld auszuwählen