

# **Darstellung und Analyse eines objektorientierten Einstiegs im Anfangsunterricht der Sekundarstufe I mit Hilfe von UML und Fujaba**

Schriftliche Hausarbeit vorgelegt im Rahmen der  
Zweiten Staatsprüfung für das Lehramt für die  
Sekundarstufe I/II in Informatik

von Thorsten Reinsch

22. September 2003

I don't go looking for trouble. Trouble usually finds me.  
(Harry Potter)

# Inhaltsverzeichnis

<b>1</b>	<b>Zentrale Fragestellungen und Genese des Themas</b>	<b>1</b>
<b>2</b>	<b>Planung und Durchführung des Projektes</b>	<b>4</b>
2.1	Wieso Objektorientierung in der Sek. I? . . . . .	4
2.2	Phasenmodell . . . . .	6
2.3	Lernziele des Unterrichts . . . . .	8
2.4	Skizze der Unterrichtseinheit . . . . .	8
2.5	Lehrplanvalidität . . . . .	15
2.6	Didaktische Überlegungen . . . . .	16
2.7	Durchführung der Unterrichtseinheit . . . . .	18
2.8	Organisation . . . . .	18
<b>3</b>	<b>Evaluation des Projektes</b>	<b>20</b>
3.1	Evaluationsstandards . . . . .	20
3.2	Ziele der Evaluation . . . . .	21
3.3	Vorstellung der Evaluationsmethoden . . . . .	21
3.3.1	Vortrag und Workshop . . . . .	21
3.3.2	Klassenarbeiten . . . . .	21
3.3.3	Lerntagebuch . . . . .	22
3.3.4	Evaluationszielscheibe . . . . .	22
3.3.5	Videoanalyse . . . . .	23
3.4	Analyse und Auswertung . . . . .	23
3.4.1	Vortrag und Workshop . . . . .	23
3.4.2	Klassenarbeiten . . . . .	24
3.4.3	Lerntagebuch . . . . .	28
3.4.4	Evaluationszielscheibe . . . . .	28
3.4.5	Videoanalyse . . . . .	30

## *Inhaltsverzeichnis*

<b>4 Die Bedeutung des Projektes für zukünftige Unterrichtsvorhaben</b>	<b>32</b>
<b>Literaturverzeichnis</b>	<b>34</b>
<b>A Anhang</b>	<b>36</b>
<b>B Glossar</b>	<b>48</b>

Die in der vorgestellten Examensarbeit benutzten Bezeichnungen beziehen sich gleichermaßen auf männliche und weibliche Personen. Allein aus Gründen der besseren Lesbarkeit wurde an manchen Stellen auf die zweifache Schreibweise verzichtet.

Abkürzungen sind in der Regel direkt erläutert, sofern es sich nicht um allgemein bekannte handelt. Im Glossar sind alle Abkürzungen erklärt.

# 1 Zentrale Fragestellungen und Genese des Themas

Die Inhalte des Schulhalbjahrs 9.2 in den Gymnasien des Landes Nordrhein-Westfalen betreffen den Bereich „Methoden der Softwareentwicklung“<sup>1</sup>. Der Lehrplan verlangt, dass sich „die vermittelten Konzepte ... an der aktuellen Entwicklung des Software-Engineering orientieren“<sup>2</sup> müssen.

In der Fachwissenschaft fand in den letzten Jahren ein Paradigmenwechsel statt. Die objektorientierte Softwareentwicklung ist heute ein zentraler Bestandteil der Lehre an den Informatik-Fakultäten. So beginnen viele Universitäten heute mit der objektorientierten Programmiersprache Java.

Auch in der Wirtschaft lässt sich dieser Trend nachvollziehen. Im Dezember 2002 kaufte IBM für 2,1 Milliarden Dollar das Unternehmen Rational Software, „eine der führenden Firmen bei Werkzeugen für objektorientiertes Design sowie für Modellierungstools“<sup>3</sup>. IBM möchte mit dem Kauf seine Position in diesem Marktsegment absichern. Die anerkannte Beratungsfirma IDC prognostiziert ein Wachstum bei Werkzeugen zur Softwareentwicklung von 9 Milliarden Dollar im Jahre 2002 auf 15 Milliarden Dollar im Jahre 2006<sup>4</sup>.

Auch in der Didaktik der Informatik spielt die objektorientierte Softwareentwicklung eine große Rolle. Allerdings beschäftigen sich die meisten Forscher mit der Sekundarstufe II. In der Sekundarstufe I hingegen findet man in der Praxis meistens einen imperativen Ansatz.

Ziel dieser Arbeit ist es, einen objektorientierten Anfangsunterricht für die Jahrgangsstufe 9.2 im Rahmen der Richtlinien in NRW zu planen und zu evaluieren. Dabei greife ich auf das Phasenmodell des Projektes *life*<sup>3</sup> (Lernwerkzeuge für den Informatikunterricht: Einsetzen, Evaluieren und [Weiter-]Entwickeln) zurück. In diesem Projekt arbeitet

---

<sup>1</sup>[Lehrplan NRW (1993)], S. 43

<sup>2</sup>[Lehrplan NRW (1993)]

<sup>3</sup>[Kuri (2003)]

<sup>4</sup>ebd.

## 1 Zentrale Fragestellungen und Genese des Themas

die Forschungsgruppe Didaktik der Informatik der Universität Paderborn unter der Leitung von Prof. Dr. Johannes Magenheim mit Gymnasien der Stadt Paderborn zusammen, um eine Lernwerkstatt zum Thema „Objektorientiertes Modellieren“ aufzubauen<sup>5</sup>. Das Projekt wird durch den Universitätsverbund MultiMedia NRW (UVM) gefördert.

*life*<sup>3</sup> beschäftigt sich mit dem Anfangsunterricht in der Jahrgangsstufe 11. Für diese Arbeit müssen deshalb die vorhandenen Module abgeändert werden, um sie altersgerecht in der Jahrgangsstufe 9 einsetzen zu können. Insofern wird sich diese Arbeit lediglich mit der ersten Phase des dreistufigen *life*<sup>3</sup>-Modells befassen. Beratend stand mir für diese Aufgabe Carsten Schulte von der Universität Paderborn zur Verfügung, der im Rahmen des *life*<sup>3</sup>-Projektes promoviert. Ein Entwurf seiner Doktorarbeit stellte er mir für Detailfragen freundlicherweise zur Verfügung.

**Lehrerfunktionen:** Bis jetzt ist mir kein ausgereifter Einstieg mit einem objektorientierten Programmieransatz für die Jahrgangsstufe 9.2 bekannt<sup>6</sup>. Meine Staatsexamensarbeit versucht daher, Unterricht zu innovieren. Da die beiden Lehreraufgaben Unterrichten und Innovieren in meiner Unterrichtseinheit eng miteinander verzahnt sind, habe ich mich entschlossen, das Ergebnis zu evaluieren. Die Arbeit lässt sich daher den Lehrerfunktionen Innovieren, Unterrichten und Evaluieren zu ordnen.

**Aufbau der Arbeit:** In Kapitel 2 entwickle ich das Konzept für die Unterrichtsreihe und beschreibe kurz die Rahmenbedingungen der Durchführung. In Kapitel 3 werden ausgesuchte Unterrichtsstunden evaluiert, um den Erfolg der neuen Methoden zu erforschen. Dieses Kapitel ist in zwei Teile gegliedert. Erst werden die benutzten Evaluationsmethoden beschrieben, bevor die eigentlichen Ergebnisse ausgewertet werden. Mit der strikten Trennung sollte vermieden werden, dass man Methoden unbewusst „gewünschten“ Ergebnissen anpasst. Für die Evaluation wurden u.a. Lerntagebücher und Videoaufzeichnungen eingesetzt. Mit Hilfe der Videobänder wurden einzelne Unterrichtssequenzen analysiert. Die Auswertung dieser Analysen sind auf der dieser Arbeit beiliegenden CD-ROM in einem HTML-Dokument zusammengefasst. Im letzten Kapitel dieser Arbeit habe ich die Ergebnisse in einem Fazit zusammengefasst.

In dieser Staatsexamensarbeit ist es leider nicht möglich, alle fachlichen Facetten der Objektorientierung zu beleuchten. Ziel der Schule - sowohl in der Sekundarstufe I als

---

<sup>5</sup>vgl. <http://life.uni-paderborn.de>

<sup>6</sup>P. Hubwieser stellt einen objektorientierten Ansatz vor, der in der Sekundarstufe I fast durchgehend auf Dokumenten und Standardsoftware basiert. Für das Gymnasium fasst er in seinem Buch die Stufen 9-11 zusammen, so dass nicht klar ersichtlich wird, ob er einen OOP-Ansatz in der Sekundarstufe I befürwortet. Für die Realschule empfiehlt er in den Klassen 9,10 eine imperative Programmierung ([Hubwieser (2000)], S. 106). Jedoch benutzen er und sein Team in der Sekundarstufe I mittlerweile ebenfalls UML Notationen.

## *1 Zentrale Fragestellungen und Genese des Themas*

auch in der Sekundarstufe II - kann und sollte es nicht sein den Anspruch auf Vollständigkeit zu erheben. Statt dessen sollte der Informatikunterricht grundlegende Konzepte exemplarisch untersuchen und die Schülerinnen und Schüler in die Lage versetzen, diese Kenntnisse zu übertragen.

Auf diesem Hintergrund habe ich mich entschlossen, den Fokus dieser Arbeit nicht auf die Datenkapselung bzw. die mehrstufigen Vererbung, sondern auf Objektstrukturen<sup>7</sup> und die Interaktion von Objekten zu legen. Die Beschränkung auf diesen (wichtigen) Teilaspekt ist notwendig, um die Schülerinnen und Schüler nicht zu überfordern. Trotz dieser Einschränkung erhalten die Schülerinnen und Schüler einen wichtigen Einblick in die Softwareentwicklung, die ihnen im späteren Berufsleben hilft, als emanzipierte Benutzer an der Weiterentwicklung von Informatiksystemen zu partizipieren.

Weitere Aspekte der Objektorientierung müssen im Sinne eines Spiralcurriculums in den Jahrgangsstufen 10 bis 13 vertieft werden.

---

<sup>7</sup>Unter dem Begriff „Objektstrukturen“ verstehe ich in dieser Arbeit die Beziehungen und Abhängigkeiten zwischen verschiedenen Objekten.

# 2 Planung und Durchführung des Projektes

## 2.1 Wieso Objektorientierung in der Sek. I?

Unter den Begriff „Objektorientierung“ fallen zahlreiche verschiedene Methoden der Softwareentwicklung. Diese Methoden sind sehr komplex und überfordern Schülerinnen und Schüler - nicht nur in der Sekundarstufe I - in ihrer Gesamtheit.

Warum sollte trotzdem ein objektorientierter Zugang im Anfangsunterricht gewählt werden? Die Interessen der Schülerinnen und Schüler im Wahlpflichtbereich Informatik sind sehr ambivalent. Der eine Teil der Gruppe hat vor, Informatik in der Oberstufe fortzuführen, während der andere Teil lediglich einen zweijährigen Einstieg in die Informatik wünscht. Deshalb darf der Informatikunterricht in der Sekundarstufe I einerseits nicht zu viele Themen der Oberstufe vorwegnehmen, muss aber andererseits Schülerinnen und Schülern, die nach der Klasse 10 Informatik nicht mehr weiter belegen wollen, in die Lage versetzen, in ihrem späteren Berufsleben bei der Gestaltung neuer Informatiksysteme an ihrem Arbeitsplatz produktiv mitwirken zu können<sup>1</sup>.

Der in diesem Projekt vorgeschlagene Unterrichtsverlauf versucht, diese unterschiedlichen Motivationen durch eine didaktische Reduktion des Themas „Objektorientierung“ zu erreichen. Objektorientierung wird in dieser Arbeit als das Identifizieren von Objekten, deren Strukturen und deren Interaktion<sup>2</sup> verstanden. Diese Sichtweise ermöglicht es, Modellierungsprozesse zu erleichtern, in denen Gegenstände oder Personen der realen Welt als Objekte betrachtet werden. Die notwendige Abstraktion im Lösungsprozess wird dadurch in eine intuitive Vorgehensweise verwandelt. Diese Reduktion ermöglicht Novizen

---

<sup>1</sup>Dies bedeutet nicht, dass Schülerinnen und Schüler zu Softwareentwicklern ausgebildet werden sollen. Die Kenntnisse über grundlegende Modellierungstechniken ermöglicht, am Arbeitsplatz später im Gespräch mit Softwareentwicklern deren Sichtweise besser zu verstehen. Dadurch wird der Benutzer in die Lage versetzt, sein Wissen und seine Fähigkeiten produktiver in den Entwicklungsprozess einzubringen, was die Qualität der neuen Software (und damit den zukünftigen Arbeitsplatz des Benutzers) verbessert.

<sup>2</sup>vgl. [life<sup>3</sup> Homepage (2003)]

## 2 Planung und Durchführung des Projektes

einen einfachen Einstieg in die Thematik, ohne eine falsche Vorstellung oder ein falsches fachliches Modell zu vermitteln.

B. Oestereich stellt in diesem Zusammenhang fest: „Die Welt der objektorientierten Softwareentwicklung ist eine bilderreiche Welt: Begriffe wie Vererbung, Botschaftsaustausch, das Werkzeug-Material-Leitbild usw. belegen die Metaphorik und deuten die ontologischen Grundprinzipien an.“<sup>3</sup> Dieser Sachverhalt soll positiv im Unterricht genutzt werden.

In der unterrichtlichen Praxis bieten sich einfache Brettspiele als Aufgabentypen für den Einstieg an. Mit ihnen lassen sich bereits Objektstrukturen mit drei bis fünf Objekten realisieren. Außerdem handelt es sich bei ihnen um eine reduzierte Wirklichkeit, die mögliche Lernbarrieren umgeht. Für Anfänger ist es schwierig, aus einem realen Anwendungskontext das Wesentliche vom Unwesentlichen zu unterscheiden. Die Modellwelt des Spiels umgeht dieses Problem geschickt.

Im Gegensatz zu imperativen Ansätzen, die lediglich kleine Algorithmen implementieren, können Schülerinnen und Schüler in dieser Einheit die Probleme erkennen, die sich beim Gestaltungsprozess von Informationssystemen ergeben. Fast alle Spiele beinhalten Freiheiten bei der Auslegung der Regeln. Computer benötigen hingegen festgelegte Instruktionen. Die endgültige Festlegung der Regeln erfolgt - sowohl bei Spielen als auch bei Softwareprojekten - in einem Abstimmungsverfahren zwischen den beteiligten Personen, wobei die jeweiligen Personen unterschiedliche Interessen vertreten. Dieser Themenkomplex kann bereits in der Modellierungsphase des Anfangsunterrichts aufgegriffen werden.

Wie bereits berichtet, müssen in dieser Unterrichtsreihe Abstriche an das korrekte, ganzheitliche Modell der Fachwissenschaft gemacht werden. So können nicht alle Aspekte der Objektorientierung behandelt werden. Der klassische Einstieg über die Vererbung und Weiterentwicklung der ADT's wird erst einmal nicht thematisiert. Konkret erfolgt der Einstieg über CRC-Karten (Class, Responsibility, Collaborators) und Klassendiagramme. Diese werden in die visuelle Entwicklungsumgebung Fujaba (From UML to Java and back again) übertragen. Das Programm enthält den Debugger Mr. Dobs (Dynamic Object Browsing System), der die Objekte des Programms und deren Beziehungen grafisch darstellt.

An dieser Stelle wird bewusst auf einen textorientierten Programmieransatz verzichtet. Fujaba generiert als visuelles CASE-Tool im Hintergrund einen Java-Quelltext, der

---

<sup>3</sup>[Oestereich (1998)], S.29f

von den Schülerinnen und Schülern gestartet werden kann<sup>4</sup>. Allerdings ist der Quellcode bereits so komplex, dass dessen Betrachtung einen Anfänger überfordert. Die visuelle Darstellung hingegen ermöglicht es, allgemeinbildende Fragestellungen der Softwareentwicklung altersgerecht zu thematisieren und verständlich zu vermitteln.

Auch aus didaktischer Sicht scheint ein objektorientierter Einstieg sinnvoll. Rüdiger Baumann schreibt in seinem Standardwerk: „«Soll man objektorientiertes Denken in Informatik unterrichten?» fragen Crutzen & Hein und gelangen zu folgender Antwort: «Ja - und zwar von Anfang an und zyklisch durch alle Klassenstufen immer wieder. Es braucht dazu keine besonderen Schüler und auch keine besonderen Programmiersprachen oder Computersysteme»“<sup>5</sup>.

## 2.2 Phasenmodell

Das *life*<sup>3</sup>-Projekt hat als didaktische Grundlage ein Phasenmodell für den Unterricht entwickelt. Die drei Phasen lassen sich nach Carsten Schulte wie in Abbildung 2.1 charakterisieren.

Phase I	Phase II	Phase III
Einstieg mit einem Lösungsbeispiel.	Durch die Lehrperson unterstützte Entwicklung eines kleinen Programms.	Entwicklung eines Programms in (parallel arbeitenden) Kleingruppen.
Ziele: Brücke zum Vorwissen, grundlegende Begriffe, grundlegendes Verständnis	Ziele: Anwendung, Vertiefung, Vervollständigung: GUI	Ziele: Integration und Vertiefung, Reflexion
Methode: Objektspiel, Debugger, ... Nur Fachlogik, keine GUI	Methode: Vom Lehrer gesteuertes Projekt, Lösungsbeispiel für GUI	Methode: Projekt mit offener Aufgabenstellung, Gruppenarbeit, Plenumsphasen zur Diskussion von Alternativen
Instruktion und geleitete Erkundung	Coaching	Konstruktion

Abbildung 2.1: Phasenmodell nach Carsten Schulte

Ziel dieses Phasenmodells ist es, die Schülerinnen und Schüler in das Zentrum der Lernaktivität zu stellen. Die Lehrperson dient den Schülern als Begleiter ihres Lernpro-

<sup>4</sup>Fujaba erzeugt ein echtes JAVA Programm, bei dem (vorerst) lediglich die grafische Benutzeroberfläche fehlt.

<sup>5</sup>[Baumann (1996)], S. 281

## 2 Planung und Durchführung des Projektes

zesses. Sie stellt Hilfsmittel und Fachvokabular zur Verfügung oder hilft bei der Strukturierung auftretender Fragestellungen. In jeder Phase nehmen die Aktivitäten der Schülerinnen und Schüler zu, während die Rolle der Lehrperson abnimmt.

Das Phasenmodell wurde für die Oberstufe konzipiert. Die Prinzipien dieses Modells lassen sich jedoch auf die Sekundarstufe I übertragen. Die Nutzung des Objektspiels<sup>6</sup> kommt der kognitiven Entwicklung der Schülerinnen und Schüler in der Sekundarstufen I sogar entgegen.

Während im *life*<sup>3</sup> Projekt die Dekonstruktion des Quelltextes das Ziel des Unterrichts ist, konzentriert sich der Unterricht im Rahmen dieser Arbeit auf die visuellen Darstellungsformen von UML und Fujaba<sup>7</sup>.

Daher entwickeln die Schülerinnen und Schüler auch keine echten „Programme“, sondern lediglich Modelle. Zwar müssten die Schülerinnen und Schüler lediglich eine grafische Benutzeroberfläche ihren Modellen hinzufügen, um vollwertige Programme zu entwickeln, aber drei Gründe sprechen gegen dieses Vorgehen:

1. grafische Benutzeroberflächen sind für Novizen sehr komplex
2. Novizen bereitet es Schwierigkeiten, das Modell von den GUI-Elementen abzugrenzen
3. zur Zeit gibt es keinen GUI-Bilder, der sich einfach mit Fujaba kombinieren lässt<sup>8</sup>.

Ziel des Phasenmodells ist es nach Meinung der Autoren, „deklaratives Wissen bzw. Konzepte ... zusammen mit authentischen Verwendungsbeispielen“<sup>9</sup> einzuführen.

Die Besonderheiten der Sekundarstufe I lassen zum jetzigen Zeitpunkt eine Fokussierung auf die erste Phase dieses didaktischen Modells, dem Anfangsunterricht, sinnvoll erscheinen. In dieser Arbeit wird daher die erste Phase betrachtet.

---

<sup>6</sup>Eine Beschreibung des Objektspiels folgt auf S. 11

<sup>7</sup>Selbstverständlich wird nur ein Ausschnitt der Möglichkeiten von UML und Fujaba betrachtet.

<sup>8</sup>Die neuste Fujaba Version enthält die Klasse `fgrafik.jar`, die eine Verknüpfung mit GUI-Elementen zulässt. Allerdings denke ich, dass die Integration dieser Funktionalität in den Anfangsunterricht der Sekundarstufe I die Schüler überfordert.

<sup>9</sup>[Schulte (2003)]

## 2.3 Lernziele des Unterrichts

Die Schülerinnen und Schüler sollen

- den Unterschied zwischen Klassen und Objekten kennen und benennen können. Insbesondere sollen sie wissen, dass Objekte sich nach den in der Klasse definierten Regeln verhalten, ihr konkretes Verhalten aber von den Attributen des Objekts abhängt.
- aus einem Ausschnitt der realen Welt Objekte und Objektstrukturen benennen und daraus Klassen ableiten können.
- Klassenstrukturen in einfache UML-Klassendiagramme formalisieren können.
- CRC-Karten als Hilfsmittel zur Strukturierung einsetzen können.
- Softwareentwicklung als Teamarbeit erfahren.
- erkennen, dass zwischen einem Ausschnitt aus der Realität und einem Modell eine semantische Lücke existiert.

## 2.4 Skizze der Unterrichtseinheit

Der Anfangsunterricht konzentriert sich auf die erste Phase des didaktischen Modells des *life<sup>3</sup>*-Projekts. Auf eine zeitlich detaillierte Ausführung muss ich an dieser Stelle verzichten, einige Hinweise befinden sich aber im Anhang.

Als Thema für die erste Phase wurde das Spiel „Flaschen drehen“ ausgewählt. Spiele bieten als Unterrichtsthema in der Sekundarstufe I einige Chancen, da sie in dieser Altersgruppe die Motivation fördern. Des Weiteren haben Spiele in der Regel bereits ein eigenes Spiel-Modell, so dass die notwendige Abstraktion den Schülerinnen und Schülern erleichtert wird. Außerdem sind die Spielregeln meistens nicht bis ins letzte Detail festgelegt. Bei vielen Spielen verständigen sich die Mitspieler kurz über Unklarheiten, bevor sie mit dem eigentlichen Spiel beginnen.<sup>10</sup> Damit kann den Schülerinnen und Schülern ein erster Einblick in die Softwareentwicklung als Prozess eröffnet werden.

Grundsätzlich sollten für die Auswahl der Aufgabe folgende Aspekte berücksichtigt werden: „The problem and the solution can be quite simple, but it should have several

---

<sup>10</sup>Beim Spiel Mau-Mau gibt es z.B. unterschiedliche Varianten zur Karte 9: a) einmal ziehen b) 9 weitergeben c) keine Sanktion *ldots*

## 2 Planung und Durchführung des Projektes

classes and at least one of the classes should require multiple instantiations. You will need a problem so that the number of required objects is not greater than the number of students in your class. If this is impossible, you will need to recruit a few volunteers from outside your class for this exercise.”<sup>11</sup> Diese Kriterien werden vom Spiel „Flaschen drehen“ erfüllt. Spieler, Feld und Flasche sind die in diesem Entwurf benötigten Klassen, wobei es mehrere Spieler und Felder gibt.

Das Spiel „Flaschen drehen“ ist allen Schülerinnen und Schülern bekannt. In einem ersten Schritt wird der Kurs in Gruppen eingeteilt. Jede Gruppe erhält ein Blatt mit den Spielregeln für diese Variante und einen Beobachtungsauftrag. Die im Unterricht gespielte Variante ist eine Art von Roulette. Jeder Mitspieler legt ein Blatt vor sich, auf dem er seine Einsätze macht. Als Einsatz werden immer 1 Cent Münzen genommen. Die Spielregeln und der Beobachtungsauftrag befinden sich im Anhang. Einige Spezialfälle sind nicht explizit geregelt, z.B.: Was passiert wenn die Flasche zwischen zwei Blättern stehen bleibt? oder Wie hoch ist der Gewinn? Zu diesem Zeitpunkt des Unterrichts sollen die einzelnen Gruppen die Spielregeln nach ihrem Bedarf erweitern (Designentscheidungen treffen). Diese Situation wird später aufgegriffen, um die semantische Lücke zwischen dem fertigen Projekt und der Wirklichkeit zu reflektieren.

Im nächsten Schritt werden die Ergebnisse des Spiels an der Tafel zusammengetragen und nach zwei Gesichtspunkten sortiert. Der erste Punkt sammelt die unterschiedlichen Regelauslegungen, während unter dem zweiten Punkt die unterschiedlichen Komponenten des Spiels (gemäß dem Beobachtungsbogen) gesammelt werden. Im Unterrichtsgespräch wird geklärt, ob unterschiedliche Regelauslegungen in echten Programmierprojekten überhaupt auftreten können. Die Diskussion endet mit der Aufzählung von Personengruppen, die in der Realität solche Probleme auflösen können: Auftraggeber, Informatiker, Benutzer, ...

Der zweite Aspekt, der betrachtet wird, stellt zum ersten Mal Objekte in das Blickfeld der Schülerinnen und Schüler. Dazu sollen sie versuchen, verschiedene Objekte zu identifizieren. Im Falle dieses Spiels sind dies: die Flasche, die Felder, die Münzen und die Spieler. Die unterschiedlichen Ergebnisse der einzelnen Gruppen werden an der Tafel zusammengefasst und gesichert.

Bis zu dieser Stelle haben die Schüler noch keinen Input vom Lehrer erhalten und sich trotzdem den elementaren Begriffen der Objektorientierung genähert. Eine bekannte Schwierigkeit für Novizen ist die Unterscheidung zwischen Objekten und Klassen. Dazu führt der Lehrer CRC-Karten mit Hilfe des Objektspiels ein.

---

<sup>11</sup>[Bergin (2000)]

## 2 Planung und Durchführung des Projektes

Spieler	
- stößt Flasche an	- Flasche
- setzt Betrag auf Feld	- Feld
- besitzt Vermögen	

Flasche	
- zeigt auf ein Feld	- Feld
- drehe	
- werte Spiel aus	

Feld	
- nimmt Einsatz auf	- Feld
- gibt Einsatz zurück	- Spieler
- behalte Einsatz ein	
- kennt den setzenden Spieler	
- kennt nächstes Feld	
- kennt seine Nummer	

Abbildung 2.2: Beispiele eines CRC-Kartensets für das Spiel „Flaschen drehen“

CRC-Karten wurden von Kent Beck und Ward Cunningham als Werkzeug für die Fortbildung von Programmierern entwickelt. Sie stellten ihre Arbeit erstmals auf der ACM Konferenz OOPSLA '89 (Object-Oriented Programming, Systems, Languages and Applications) vor, die vom 1. bis 6. Oktober 1989 in New Orleans (Louisiana, USA) stattfand.

CRC-Karten sind einfache Karteikarten mit drei verschiedenen Informationen. Abbildung 2.2 zeigt ein Beispiel für CRC-Karten. In diesem Beispiel wurde das Spiel „Flaschen drehen“ modelliert<sup>12</sup>. Als erstes wird auf der Karte der Name der Klasse notiert. Des Weiteren enthält die Karte die Information, wofür die Klasse verantwortlich ist (Responsibility). Darunter versteht man, was eine Klasse „kann oder weiß“. Jedoch wird dies nicht formal auf Attribute und Methoden reduziert. In der ersten Designphase ist dies noch nicht relevant. Als letzter Punkt werden die Collaborators notiert, d.h. es werden die Klassen aufgeführt, die die Klasse der Karte benötigt um ihre Aufgaben zu erfüllen. Dies sind alle Klassen, mit dem die Ausgangsklasse im späteren Programm Botschaften austauscht.

K. Beck und W. Cunningham beschreiben in Ihrem Artikel auch schon Rollenspiele mit CRC-Karten: „We suggest driving a design toward completion with the aid of exe-

<sup>12</sup>Das Beispiel stammt aus dem *life<sup>3</sup>*-Projekt, [*life<sup>3</sup>* Homepage (2003)]

## 2 Planung und Durchführung des Projektes

cution scenarios ...“<sup>13</sup> Für den Unterricht entwickelte Joseph Bergin diese Idee zum Objektspiel<sup>14</sup> weiter. In dieser Unterrichtseinheit benutzen wir die modifizierte Variante des *life*<sup>3</sup>-Projekts, der J. Bergins Objektspiel zu Grunde liegt. Ziel der Modifikation ist es, Lernschwierigkeiten der Schülerinnen und Schüler bei der Unterscheidung Klasse versus Objekt auszuräumen. Deshalb werden neben den CRC-Karten auch Objektkarten eingeführt. Objektkarten sind wie CRC-Karten aufgebaut. Allerdings werden auf ihnen die Zustände eines konkreten Objektes notiert. Objektkarten sollten eine andere Farbe als die CRC-Karten haben, damit die Schüler auch visuell erkennen, dass es zwar Ähnlichkeiten zwischen den beiden Kartentypen gibt, sie sich aber trotzdem unterscheiden.

Im Objektspiel bestimmt der Lehrer für jedes Objekt eine Schülerin oder einen Schüler. Der Schüler erhält eine Objektkarte und einen Stift. Die Aufgabe des Schülers ist es, die Zustände seines Objektes auf seiner Karte festzuhalten. Dabei darf der Schüler jedoch nur die Kenntnisse nutzen, die auf der CRC-Karte für seine Klasse fixiert sind. Aus diesem Grund bietet es sich an, die CRC-Karten für alle Schüler sichtbar zu projizieren. An einem Whiteboard könnten die Karten z.B. mit Hilfe von Magneten befestigt werden.

Der Lehrer übernimmt im Objektspiel die Aufgabe des Benutzers und initialisiert alle Objekte. Die Schüler, die die Objekte spielen, müssen die Anfangszustände - und später alle Änderungen - auf ihrer Karte notieren. Damit die Objekte nicht über die Namen der Schüler angesprochen werden müssen, erhalten die Objekte Namen. Dies erfolgt analog zu dem Verfahren von Mr. Dobs. D.h. Felder werden f1, f2, f3, etc. und Spieler mit s1, s2, s2, etc. benannt. An dieser Stelle weist J. Bergin auf ein Problem mit den verwendeten Metaphern hin: „Since the 'objects' wear their names, it will seem like the name is an inherent part of the object, when in fact this is not the case. A reference (variable) can refer to any object of the appropriate type. It might be good to be explicit about this difference at some point, though not necessarily as the game proceeds.“<sup>15</sup> Ich stimme Bergins These zu, dass man diese Thematik zu einem späteren Zeitpunkt mit den Schülern diskutieren sollte.

Mit Hilfe der Objektkarten soll jetzt das Spiel „Flaschen drehen“ erneut gespielt werden. Durch die CRC- und Objektkarten findet das Spiel nun aber in einer formalisierten Welt statt. Das Objektspiel hilft Softwareentwicklern in Projekten ihr Modell zu überprüfen. In dieser Unterrichtsphase stehen aber didaktische Motive im Vordergrund, z.B. die Abgrenzung der Begriffe Objekt und Klasse durch die Schülerinnen und Schüler.

Das Spiel wird durch den Lehrer gestartet. Dazu gibt er einen Tennisball an ein Ob-

---

<sup>13</sup>[Beck, Cunningham (1989)]

<sup>14</sup>vgl. [Bergin (2000)]

<sup>15</sup>[Bergin (2000)]

## 2 Planung und Durchführung des Projektes

jekt. Der Ball symbolisiert den aktuellen Verarbeitungsschritt. Das aktive Objekt muss nun versuchen die ausgewählte Aufgabe zu bearbeiten. Dazu stehen ihm nur die Informationen auf der Objektkarte zur Verfügung. Allerdings hat es die Möglichkeit, die Aufgabe an einen Collaborator weiterzuleiten. Dies geschieht durch die Weitergabe des Balls. Die Aufgabe der nicht direkt am Objektspiel beteiligten Schülerinnen und Schüler ist es, darauf zu achten, dass die Objekte nur die Informationen verwenden, die als Zustand auf ihrer Objektkarte stehen. Auch müssen sie sicherstellen, dass die Objekte nur Collaborators benutzen, die sie tatsächlich kennen. Hier reicht nicht der Hinweis auf der CRC-Karte. Auf der CRC-Karte sind nur die Klassen notiert, die ein Objekt kennen kann - keine konkreten Objekte. Vermutlich wird der Lehrer zu Beginn des Objektspiels auf viele Fehler selbst hinweisen, da die Schülerinnen und Schüler auf diese Aspekte noch nicht achten. Mit zunehmender Spieldauer sollten die Schülerinnen und Schüler selbst Fehler bei der Abarbeitung entdecken.

Hat ein Objekt seine Aufgabe bearbeitet, gibt es den Ball an die aufrufende Stelle zurück. Dies ist entweder der Collaborator, der das Objekt aufgerufen hat, oder aber der Lehrer in der Rolle des Benutzers. Der Vorteil des Balls ist es, dass die Metapher auch bei nebenläufigen Prozessen korrekt bleibt. Das Starten eines neuen Prozesses wird durch das Hinzufügen eines weiteren Balls dargestellt. Parallele Prozesse sollten in der Jahrgangsstufe 9 sicherlich nicht als eigenständiges Thema besprochen werden. Es könnte jedoch sein, dass Schülerinnen und Schüler die Idee einer Parallelverarbeitung selbst im Unterricht zur Sprache bringen. Eine Runde im Objektspiel mit mehreren Bälle könnte Vor- und Nachteile von Nebenläufigkeit auf eine anschauliche Weise darstellen. Eine vertiefende Behandlung des Themas bietet sich meiner Meinung nach zu diesem Zeitpunkt nicht an.

Nach zwei oder drei Runden sollten die Schülerinnen und Schülern die Zusammenhänge verstanden haben. Deshalb werden die Grundbegriffe an der Tafel definiert. Zur Sicherung sollten die Schülerinnen und Schüler die Unterschiede und Gemeinsamkeiten der Begriffe mit eigenen Worten wiederholen und schriftlich in ihren Heften fixieren.

Die Grenzen von CRC- und Objektkarten liegen in der Visualisierung der Objektstrukturen. UML-Klassendiagramme (Unified Modeling Language) sind für diesen Zweck besser geeignet. Abbildung 2.3 zeigt das UML-Klassendiagramm in Fujaba. Jede Klasse ist durch ein Rechteck dargestellt. Der oberste Bereich im Rechteck enthält den Namen, der mittlere Bereich enthält die Attribute und der untere Bereich die Methoden der Klasse. Die Assoziationen zwischen den Klassen sind durch Verbindungslinien dargestellt, eventuelle Kardinalitäten werden am Ende der Assoziation dargestellt (z.B. 0..1)<sup>16</sup>

---

<sup>16</sup>Eine gute und empfehlenswerte Einführung in UML bietet das Buch von Bernd

## 2 Planung und Durchführung des Projektes

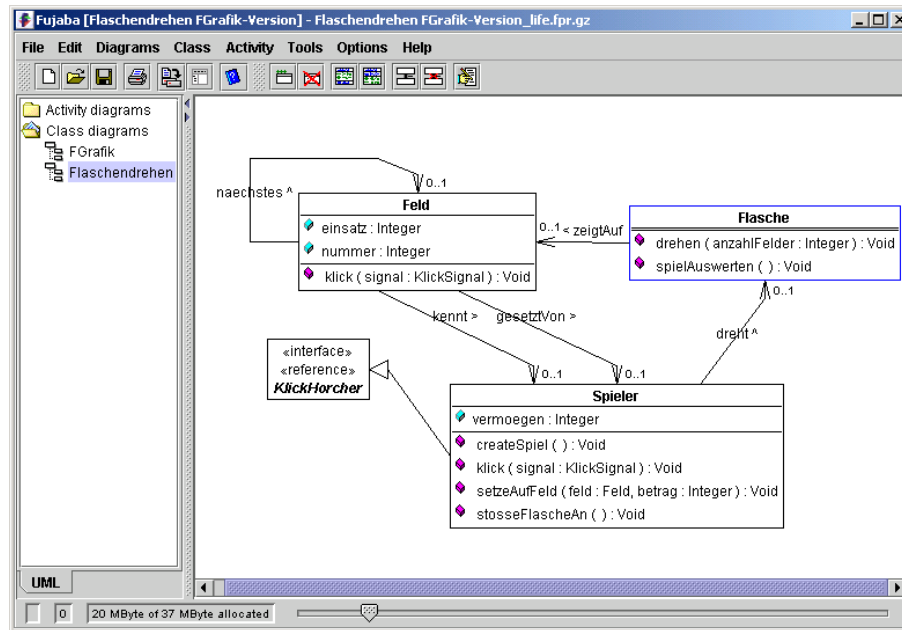


Abbildung 2.3: Klassendiagramm in Fujaba

Beim Einstieg in UML erhalten die Schülerinnen und Schüler ein ausgedrucktes Klassendiagramm. Dieses Programm stellt eine Lösung des Problems dar. An dieser Stelle kommt es zu einem didaktischen Bruch, da die Schülerinnen und Schüler nicht mit ihrem eigenen Modell weiterarbeiten können. Ich habe dies bewusst in Kauf genommen, da es mir bei der Einführung in die Software wichtig ist, dass alle Schüler über das gleiche Modell verfügen. Die Vor- und Nachteile dieses Vorgehens werden in der Evaluation nochmals besonders diskutiert.

Mit Hilfe des CRC-Kartenmodells sollen die Schülerinnen und Schüler nun die verwendeten Notationen erarbeiten. Als Hilfestellung kann im Klassenverband ein einfaches Objektdiagramm aus dem Klassendiagramm abgeleitet werden, um so z.B. die Kardinalitäten bzw. die gerichteten Assoziationen zu veranschaulichen.

Zur Vertiefung des Verständnisses bietet es sich an, das Klassendiagramm in Fujaba aufzurufen und den grafischen Debugger Mr. Dobs zu starten. Mit Hilfe von Mr. Dobs (vgl. Abbildung 2.4) können die Schülerinnen und Schüler nun interaktiv die einzelnen Objekte erzeugen und durch den Aufruf der Methoden manipulieren. In dieser Phase übernehmen die Schülerinnen und Schüler die Rolle des Lehrers während des realen Spiels. Die Arbeit mit Mr. Dobs ist für die Sicherung und Vertiefung der objektorientier-

---

Oestereich[Oestereich (1998)].

ten Kenntnisse notwendig. Zum einen macht Mr. Dobs den Unterschied zwischen Klassen und Objekten deutlich, da Klassen nicht aufgerufen werden können und nur als Bauplan für neue Objekte dienen. Zum anderen wird den Schülerinnen und Schülern deutlich, dass der Aufruf von Methoden lediglich das aktive Objekt manipuliert. Objekte besitzen also eine eigene Identität und verändern ihre Geschwister normalerweise nicht.

In dieser Unterrichtsphase muss auf Grund der gegebenen Hard- und Softwareausstattung der Schule Zeit für eine Einführung in Linux eingeplant werden (vgl. Kapitel 2.8).

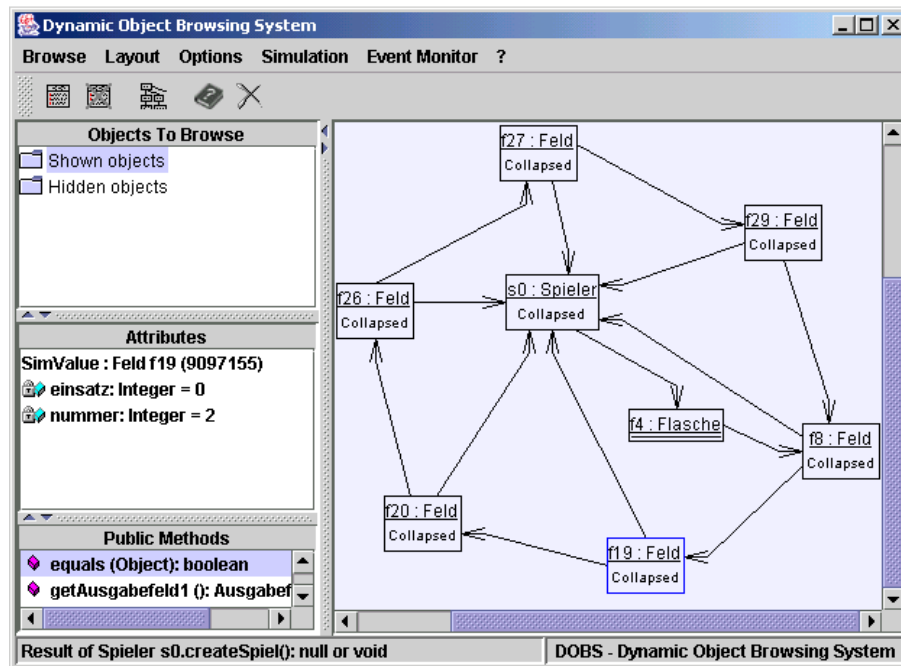


Abbildung 2.4: Der grafische Debugger Mr. Dobs

Im weiteren Unterrichtsverlauf erstellen die Schülerinnen und Schüler ein eigenes Modell. Ich habe mich dazu entschlossen, das Beispiel Schülerbibliothek zu wählen. Dieses sollen die Schüler mit Hilfe von Fujaba implementieren. Andere Beispiele wären z.B. Videoverleih, CD-Sammlung, Stundenplan online, Anmeldesystem für Tagungen<sup>17</sup> etc.

Mit dem Testen der Bibliothek endet der objektorientierte Anfangsunterricht und damit die Phase I des didaktischen Modells. Zum Abschluss reflektieren die Schülerinnen und Schüler den beschrittenen Weg von der Modellierung bis zum Programm in Fujaba.

Als nächster Schritt müssten die Schülerinnen und Schülern die Methoden implementieren. In der Praxis wechseln Softwareentwickler an dieser Stelle in den Sourcecode. Es

<sup>17</sup>nach [Seifert (2003)], S. 15

gibt verschiedene Ansätze zur visuellen Programmierung. Fujaba bietet dazu Story Pattern<sup>18</sup> an. Einfache Story Pattern könnten von Schülerinnen und Schülern selbständig analysiert, verstanden und angewandt werden. Allerdings werden in Fujaba mit Story Pattern teilweise Programmablaufpläne gezeichnet, die ein unstrukturiertes Programmieren ermöglichen. Aus diesem Grund muss aus didaktischer Sicht auf dieses Hilfsmittel verzichtet werden, da ein strukturiertes Vorgehen eines der wichtigsten Ziele des Informatikunterrichts sein muss. Die Lösung dieses Problems müsste in einer weiteren Examensarbeit oder Promotion untersucht werden<sup>19</sup>.

## 2.5 Lehrplanvalidität

Der Lehrplan für das Fach Informatik an nordrhein-westfälischen Gymnasien sieht für das zweite Halbjahr der Jahrgangsstufe 9 das Thema „Funktionsweisen von Software“ vor. „Die in diesem Halbjahr vorgesehenen Inhalt betreffen den Bereich „Methoden der Softwareentwicklung.“<sup>20</sup>. „Die Wirkungsweise von Anwenderprogrammen wird den Schülerinnen und Schülern verständlicher, wenn sie die zugrundeliegenden Konzepte exemplarisch anhand selbstverfasster Programme erproben können . . . Die vermittelten Konzepte müssen sich an der aktuellen Entwicklung des Software-Engineering orientieren. Hierbei gewinnt eine funktionale und objektorientierte Konzeption zunehmend an Bedeutung.“<sup>21</sup>

Aus diesen Zitaten wird deutlich, dass das Konzept der Unterrichtseinheit im Sinne des Lehrplans entwickelt wurde. Die CASE-Software Fujaba wurde von Prof. Dr. W. Schäfer in der Fachgruppe Softwaretechnik an der Universität Paderborn entwickelt. Mit Hilfe der Software können die Schülerinnen und Schüler einen exemplarischen Einblick in die aktuelle Forschung des Softwareengineering gewinnen. Dass dieses Thema ebenfalls eine große wirtschaftliche Bedeutung hat, wurde bereits auf Seite 1 erläutert.

Neben den objektorientierten Sichtweisen ermöglicht das Unterrichtskonzept auch die Betrachtung des funktionalen Paradigmas. Wie der Artikel von I. Diethelm, L. Geiger und A. Zündorf<sup>22</sup> zeigt, kann Fujaba Story Pattern implementieren. Methoden können durch die Angabe einer Start-Objektstruktur und einer End-Objektstruktur implementiert werden. Die interne Codierung des Quelltextes generiert Fujaba automatisch.<sup>23</sup>

---

<sup>18</sup>Story Pattern sind von der AG Schäfer der Universität Paderborn entworfene Diagramme. Diese Diagramme sind nicht in UML enthalten - bauen aber auf dem Standard auf.

<sup>19</sup>Insbesondere sollte geprüft werden, ob Fujaba nicht um Struktogramme erweitert werden kann.

<sup>20</sup>[Lehrplan NRW (1993)], S. 43

<sup>21</sup>[Lehrplan NRW (1993)], S. 37

<sup>22</sup>[Diethelm, Geiger, Zündorf (2002)]

<sup>23</sup>Leider werden bei Story Pattern Programmablaufpläne benutzt. Dies spricht gegen den Einsatz in der

Da die Software auf der visuellen Modellierungssprache UML basiert und der grafische Debugger Mr. Dobs ein interaktives Testen der entworfenen Modelle ermöglicht, erfüllt die Einheit ein weiteres wichtiges Kriterium der Richtlinien: „Für Schülerinnen und Schüler der Sekundarstufe I ist es vorteilhaft, weil altersangemessen, wenn die Visualisierung von Algorithmen . . . und ein interaktives Arbeiten ermöglicht werden.“<sup>24</sup>

Die Struktur der skizzierten Unterrichtseinheit stimmt also mit dem Lehrplan und den Richtlinien des Fachs Informatik in der Sekundarstufe I überein.

## 2.6 Didaktische Überlegungen

Wie das Kapitel 2.5 gezeigt hat, erfüllt die vorgestellte Unterrichtseinheit die formalen Kriterien der Richtlinien. In den nächsten Abschnitten soll gezeigt werden, dass die Unterrichtsreihe aus didaktischer Sicht ebenfalls sinnvoll ist.

Peter Hubwieser von der TU München hat ein Konzept für ein Pflichtfach Informatik an bayrischen Gymnasien erstellt: „Der Informatik-Pflichtunterricht beginnt in der 6. Jahrgangsstufe, worauf ein zweijähriges Wahl- oder Wahlpflichtangebot folgt. Ab der 9. Jahrgangsstufe bis zur 11. Jahrgangsstufe wird die Informatik als Pflichtfach fortgesetzt. Das Fach ist in allen genannten Jahrgangsstufen mindestens zweistündig.“<sup>25</sup>

Dieser Vorschlag soll ab dem Schuljahr 2003/2004 in Bayern flächendeckend umgesetzt werden. Bereits jetzt laufen zahlreiche Versuchsklassen, die sowohl intern als auch extern evaluiert werden<sup>26</sup>.

In seinen Unterrichtsbeispielen benutzt P. Hubwieser von der Jahrgangsstufe 6 an UML-Notationen zur Visualisierung. Er benutzt in seinem Buch zwar eine vom aktuellen UML-Standard abweichende Notation<sup>27</sup>, aber Matthias Spohrer, Mitarbeiter von P. Hubwieser an der TU München, benutzt in seiner neusten Publikation zum Informatikunterricht in der Sekundarstufe I ebenfalls UML-Diagramme gemäß dem aktuellen Standard<sup>28</sup>. Dieser Artikel zeigt darüber hinaus, dass Modellieren mit Hilfe von UML-CASE-Tools in der Mittelstufe aktueller Forschungsgegenstand ist<sup>29</sup>.

Betrachten wir nun die eingesetzten Methoden aus didaktischer Sicht. K. Beck und

---

Schule. Das Thema wird auf Seite 24 ausführlicher behandelt.

<sup>24</sup>[Lehrplan NRW (1993)], S. 52

<sup>25</sup>[Hubwieser (2000)], S. 97f

<sup>26</sup>vgl. [Hubwieser, Humbert und Schubert (2001)]

<sup>27</sup>Der aktuelle UML-Standard ist jünger als Hubwiesers Buch.

<sup>28</sup>[Spohrer (2002)]

<sup>29</sup>Allerdings beschäftigt sich der Artikel über endliche Automaten und Zustandsdiagrammen. Der nordrhein-westfälische Lehrplan sieht dies allerdings zu diesem Zeitpunkt des Unterrichts nicht vor.

W. Cunningham schreiben zum Einsatz von CRC-Karten: „Using cards in group settings we found that even weaker programmers, without a deep understanding of objects, could contribute to objects designs. We speculate that because the designs are so much more concrete, and the logical relationship, between objects explicit, it is easier to understand, evaluate, and modify a design.“<sup>30</sup> Diese Aussage lässt vermuten, dass die Methoden von CRC-Karten und dem Objektspiel erfolgreich in der Sekundarstufe I eingesetzt werden können.

Es stellt sich natürlich die Frage, wieso man mit dem echten Spiel beginnt, anstatt direkt die CRC-Karten und das Objektspiel zu nutzen. Zwei Gründe sprechen meiner Meinung nach für die Beibehaltung des echten Spiels. Zum einen fördert dies die Motivation, da der Unterricht die sonst übliche rein kognitive Ebene verlässt. Zum anderen erarbeiten sich die Schüler das Basiswissen zum Prozessablauf. K. Beck und W. Cunningham halten dies für unabdingbar: „We know of one case where finished cards were delivered to a client as (partial) design documentation. Although the team that produced the cards was suite happy with the design, the recipient was unable to make sense of the cards out of the context.“<sup>31</sup> Die Einführung in den Anwendungskontext fördert also das Verständnis des Modells.

UML und Fujaba ermöglichen einen gleitenden Übergang von der stark handlungsorientierten Einstiegsphase mit den CRC-Karten und dem Objektspiel hin zur Modellierung mit UML am Rechner (und damit zum Generieren von Sourcecode). Die Schülerinnen und Schüler nehmen den ganzen Prozess als schrittweise Formalisierung wahr. Sie erleben Softwareentwicklung als evolutionären Prozess. Bereits der Lehrplan schlägt diese Strategie vor: „An seiner Stelle [klassische Phasenmodell, Anm. d. Autors] wird für den Wahlpflichtbereich eine Strategie vorgeschlagen, die sich an einer Entwicklungsmethode orientiert, die evolutionäres Entwickeln genannt wird. Unter evolutionärem Entwickeln versteht man die sukzessive Weiterentwicklung eines Minimalansatzes (Prototyps) unter ständigem Dialog mit der Zielgruppe. Diese Vorgehensweise ist nicht nur altersangemessen, sondern sie ist fachwissenschaftlich begründet und wird bei der kommerziellen Softwareentwicklung angewendet.“<sup>32</sup>

Die theoretischen Vorüberlegungen zu dieser Unterrichtseinheit lassen also den Schluss zu, dass diese Methode für den Einsatz in der Sekundarstufe I geeignet ist. Die bisherigen Unterrichtsversuche in der Oberstufe<sup>33</sup> haben die Möglichkeit der schulischen Umsetzung

---

<sup>30</sup>[Beck, Cunningham (1989)]

<sup>31</sup>[Beck, Cunningham (1989)]

<sup>32</sup>[Lehrplan NRW (1993)], S. 53f

<sup>33</sup>vgl. [*life*<sup>3</sup> Homepage (2003)]

bewiesen. Inwieweit sich das Konzept im unterrichtlichen Alltag bewährt, wird in Kapitel 3 evaluiert.

## 2.7 Durchführung der Unterrichtseinheit

Die im Kapitel 2.4 dargestellte Einheit wurde von mir mit einem Differenzierungskurs 9 Informatik am Ernst-Moritz-Arndt-Gymnasium in Bonn zwischen dem 6. März und 16. Mai 2003 durchgeführt. Durch die Osterferien wurde die Einheit in zwei Blöcke geteilt<sup>34</sup>. Wichtige Stunden wurden auf Video aufgezeichnet, so dass eine spätere Evaluation der Einheit möglich wurde. Eine genaue Beschreibung der Evaluation befindet sich im Kapitel 3.

## 2.8 Organisation

Die Software Fujaba steht unter der GPL-Lizenz und ist daher frei verfügbar<sup>35</sup>. Die Software ist selbst in Java geschrieben und kann daher auf verschiedenen Betriebssystemen eingesetzt werden. Da ein Schüler zu Hause über einen Macintosh verfügte, war dies ein wichtiger Aspekt bei der Auswahl der Software.

Alle Schülerinnen und Schüler erhielten eine CD-ROM mit der Software, so dass sie zu Hause ihre Kenntnisse aus dem Unterricht vertiefen konnten. Aufgrund der GPL konnte die Software frei verteilt werden.

Allerdings gab es ein Problem mit der Software. Da Java sehr viele Ressourcen benötigt, lief das Programm auf den Computern des Ernst-Moritz-Arndt-Gymnasium (Pentium Celeron, 500 MHz, 64 MB RAM) nicht. Das größte Problem war der kleine Hauptspeicher. Als Lösung wurden zwei Application-Server<sup>36</sup> eingerichtet. Als Betriebssystem wurde Linux ausgewählt. Auf diesen Servern wurde Fujaba ausgeführt, so dass lediglich die Bildausgabe über einen X-Server auf die Clients gespiegelt wurde.

Zur besseren Lastverteilung der 12 Schüler-Clients wurden zwei Server eingesetzt. Mit Hilfe von NIS und NFS wurde sichergestellt, dass die Accounts nur einmal eingerichtet werden mussten und den Schülern auf beiden Servern die gleiche Arbeitsumgebung zur Verfügung stand.

---

<sup>34</sup>Durch verschiedene Austauschprogramme mussten in der Zeit weitere Stunden ausfallen.

<sup>35</sup><http://www.fujaba.de>

<sup>36</sup>Pentium III mit 700 MHz/512MB RAM bzw. 900 MHz/256MB RAM

## *2 Planung und Durchführung des Projektes*

Auf den Clients wurde die X-Server-Client-Software X-Win-32 installiert. Wir nutzten eine kostenlose Demo-Version, deren Funktionalität auf zwei Stunden beschränkt ist. Für den schulischen Gebrauch reichte dies völlig aus. Zum Start von Fujaba mussten die Schüler den X-Server für Windows starten und sich anschließend mit SSH auf dem Linux-Server anmelden. Dort konnten sie Fujaba starten. Eine kurze Einführung in Linux für die Schülerinnen und Schüler war notwendig, da sie keine Erfahrung mit einem UNIX-Betriebssystem hatten.

# 3 Evaluation des Projektes

Die Rahmenbedingungen einer Staatsexamensarbeit lassen keine ausführliche, wissenschaftliche Evaluation zu. Jedoch ist es möglich, und dieses Ziel wird in diesem Kapitel verfolgt, eine praxisorientierte Evaluation durchzuführen.

Im ersten Abschnitt versuche ich, an ausgewählten Kriterien des amerikanischen *Joint Committee on Standards for Educational Evaluation* zu zeigen, dass trotz der Praxisorientierung viele Standards einer wissenschaftlichen Evaluation eingehalten wurden.

Im zweiten Abschnitt stelle ich die Evaluationsmethoden vor, bevor im letzten Abschnitt die Evaluationsergebnisse vorgestellt und analysiert werden.

## 3.1 Evaluationsstandards

Bereits seit 1981 arbeitet das amerikanische *Joint Committee on Standards for Educational Evaluation*<sup>1</sup> an einem Standardkatalog für Evaluationen. In die Standards flossen Erfahrungen vieler Evaluationen ein, und seit ihrer Veröffentlichung haben sich tausende von Evaluationen an diesen Kriterien orientiert. Aber nicht nur in Nordamerika, sondern auch in Deutschland, in Österreich und in der Schweiz werden diese Evaluationsstandards immer wieder eingesetzt.

Normalerweise wird eine Evaluation von einem Auftraggeber initiiert und von einem Evaluator in einem vorgegebenen Kostenrahmen verwirklicht. Diese Faktoren treffen auf diese Arbeit nicht zu, so dass eine korrekte, wissenschaftliche Evaluation nach den Standards nicht möglich ist.

Im folgenden möchte ich darstellen, dass ich in dieser Arbeit trotzdem versuche, möglichst viele Standards einzuhalten. Die Standards lassen sich in vier Kategorien einteilen<sup>2</sup>:

- Nützlichkeitsstandards
- Durchführbarkeitsstandards

---

<sup>1</sup>Diese Evaluationsstandards werden vom Studienseminar Bonn als Standardliteratur angegeben.

<sup>2</sup>vgl. [Sanders (Hrsg.) (2000)]

- Korrektheitsstandards
- Genauigkeitsstandards

Eine Erläuterung der in dieser Arbeit angewendeten Evaluationsstandards befindet sich im Anhang auf Seite 36. Insbesondere die Standards zur Wahrung der Persönlichkeitsrechte der Schülerinnen und Schüler wurden wegen der Videoaufnahmen besonders beachtet.

## 3.2 Ziele der Evaluation

Ziel der Evaluation ist es zu überprüfen, ob ein objektorientierter Einstieg im Anfangsunterricht der Sekundarstufe I realisierbar ist. Drei Fragestellungen stehen dabei im besonderen Interesse der Evaluation:

- Lassen sich durch die Unterrichtseinheit grundlegende, fachliche Sichtweisen der Objektorientierung den Schülerinnen und Schülern der Sekundarstufe I vermitteln?
- Ermöglicht der beschriebene Ansatz ein stark handlungsorientiertes Arbeiten der Schülerinnen und Schüler?
- Welche Stärken besitzt der Entwurf und an welchen Stellen muss der Ansatz verbessert werden?

## 3.3 Vorstellung der Evaluationsmethoden

### 3.3.1 Vortrag und Workshop

Am 7. April 2003 hatte ich auf dem „Informatiktag NRW“ die Möglichkeit, meine Unterrichtsreihe in einem Vortrag und einem Workshop vorzustellen. Der Informatiktag fand an der Universität Siegen statt und wurde von der Gesellschaft für Informatik, Fachgruppe InformatiklehrerInnen NRW, veranstaltet. An meinem Vortrag nahmen ca. 20 Lehrerinnen und Lehrer aus ganz Nordrhein-Westfalen teil, darunter zahlreiche Fachleiter.

### 3.3.2 Klassenarbeiten

Da diese Examensarbeit im Differenzierungsbereich statt fand, mussten die Klassenarbeiten an den von der Stufenleitung vorgegebenen Terminen geschrieben werden. Eine

Verschiebung war nicht möglich, da im zweiten Halbjahr der Jahrgangsstufe 9 in unserer Schule mehrere Austauschprogramme mit Partnerschulen im Ausland stattfinden.

Deshalb musste ich die Klassenarbeiten am 27. März 2003 und 26. Juni 2003 schreiben. In der ersten Klassenarbeit konnte das Wissen des ersten Abschnitts geprüft werden. Allerdings bezogen sich ca. 50 Prozent der Arbeit auf den vorherigen Themenblock.

Vor der zweiten Klassenarbeit wurde ein neues Thema<sup>3</sup> angesprochen, so dass sich auch diese Arbeit über verschiedene Wissensgebiete erstreckte. Dies hatte allerdings den Vorteil, dass sich mit der zweiten Arbeit eine Nachhaltigkeit prüfen ließ, da das neue Thema bereits vier Wochen lang behandelt wurde.

Um die fehlende Datenlücke am Ende der Unterrichtsreihe schließen zu können, habe ich eine schriftliche Übung schreiben lassen. Zwar konnte damit nur ein kleiner Ausschnitt des Stoffs abgefragt werden, aber zur qualitativen Abrundung der Daten reichten die gewonnenen Informationen aus.

#### 3.3.3 Lerntagebuch

Im Vorfeld der Unterrichtseinheit wurde mit den Schülern vereinbart, ein Lerntagebuch einzuführen. Die Schüler erhielten am Ende jeder Stunde ein Formblatt (vgl. S. 46), in dem sie über ihren Lernfortschritt Buch führen sollten. Für das Ausfüllen der Blätter erhielten die Schüler einen festen Zeitrahmen im Unterricht. Die Blätter wurden zu jeder Stunde vom Lehrer verteilt. Sie mussten von den Schülerinnen und Schülern direkt in einen Schnellhefter geheftet werden, der ihnen vom Lehrer zur Verfügung gestellt wurde.

Zwischen dem Lehrer und den Schülern wurde vereinbart, dass die Lerntagebücher nicht in die Benotung einfließen. Sie dienten in erster Linie der Selbstevaluation der Schüler. Die Schüler erklärten sich bereit, dem Lehrer nach der Einheit das Lerntagebuch als Kopie zur Auswertung der Unterrichtseinheit zur Verfügung zu stellen.

#### 3.3.4 Evaluationszielscheibe

In der letzten Stunde der Einheit wurde eine kurze Evaluation der Einheit durch die Schülerinnen und Schüler vorgenommen. Dazu wurde die Evaluationszielscheibe von Christoph Burkhard und Gerhard Eikenbusch<sup>4</sup> benutzt und leicht modifiziert. Jeder Schüler erhielt eine Zielscheibe, auf der er markieren musste, wie ihm die Unterrichtsreihe gefallen hat. Eine Markierung im Zentrum bedeutet eine hohe Zustimmung, eine Markierung

---

<sup>3</sup>Hypertext Markup Language

<sup>4</sup>[Burkhard, Eikenbusch (2000)], S. 159

außerhalb der Zielscheibe eine Ablehnung. Nachdem alle Schülerinnen und Schüler ihre Zielscheibe ausgefüllt hatten, wurden die Ergebnisse auf eine große Evaluationszielscheibe übertragen. Dazu wurde ein Plakat mit der Zielscheibe auf dem Whiteboard befestigt. Über das Ergebnis dieser Evaluation wurde ein Unterrichtsgespräch geführt, um so ein differenzierteres Bild der Schüleraussagen zu erhalten.

#### 3.3.5 Videoanalyse

Die komplette Einheit des Anfangsunterricht wurde auf Video aufgezeichnet. Wie auf Seite 36 dokumentiert, wurden die Persönlichkeitsrechte der Schüler gewahrt.

Für die Aufzeichnungen stand ein DV-Camcorder mit 60 Minuten Bändern zur Verfügung. In den Doppelstunden mussten die Bänder deshalb gewechselt werden. Zu Beginn jeder Stunde wurde die Kamera auf einem Stativ mit einer festen Position aufgebaut. In der Regel wurde die Perspektive nicht verändert.

Insgesamt wurden 17 Bänder aufgezeichnet. Die Bänder wurden digitalisiert und auf Festplatten (ca. 150 GB) gespeichert.

Das gesamte Videomaterial wurde gesichtet, um interessante Schlüsselstellen des Unterrichts zu identifizieren. Diese Unterrichtsstellen wurden zusammengeschnitten und kommentiert. Dieser Arbeit liegt eine CD-ROM mit einem Hypertextdokument bei, die diese Sequenzen enthält.

### 3.4 Analyse und Auswertung

In diesem Abschnitt stelle ich die Ergebnisse der Evaluation vor. Dabei orientiere ich mich an der Gliederung des vorherigen Kapitels, in dem ich die verwendeten Methoden beschrieben habe. An Stellen, bei denen eine inhaltliche Verknüpfung der Evaluationsergebnisse sinnvoll erschien, habe ich die Grenzen zwischen den Abschnitten aufgeweicht.

#### 3.4.1 Vortrag und Workshop

Auf dem „Informatiktag NRW“ habe ich das Konzept der Unterrichtsreihe vorgestellt und im Workshop einige Schlüsselstellen die Teilnehmer simulieren lassen. Die Diskussion mit den erfahrenen Kolleginnen und Kollegen war sehr fruchtbar. Insgesamt stieß das Thema auf ein sehr großes Interesse. Obwohl der Lehrplan objektorientierte Vorgehensweisen in der Sekundarstufe I vorsieht (vgl. Abschnitt 2.5), gibt es sehr wenige konkrete Erfahrungen damit. Das Interesse an diesem neuen Zugang zeigt deutlich den innovativen

Charakter dieser Arbeit. Die stark handlungsorientierte Arbeitsweise mit vielen Gruppenarbeitsphasen wurde von den Informatiklehrern begrüßt. Ebenfalls fand die Arbeit mit einem grafischen Entwicklungstool die Zustimmung der meisten Kollegen.

Von einigen Teilnehmern wurde der didaktische Bruch beim Wechsel von den CRC-Karten zu den Klassendiagrammen angemerkt, den ich bereits in der Beschreibung der Reihe vorgestellt habe (vgl. Seite 13). Es gab eine kontroverse Diskussion über die Vor- und Nachteile dieses Vorgehens. In der Zukunft muss untersucht werden, welches Verfahren sich in der konkreten Unterrichtssituation besser eignet. Die Evaluationsdokumente zeigen, dass es meinen Schülern keine Probleme bereitet hat, diesen Wechsel nachzuvollziehen. Allerdings müssen für eine endgültige Aussage quantitative Untersuchungen durchgeführt werden.

Ebenfalls diskutiert wurde, wie die Implementierung der Methoden altersgerecht eingeführt werden könnte. Thematisch würde dies zwar erst nach meiner Unterrichtsreihe beginnen, trotzdem wurde dieses Thema diskutiert. Mehrheitlich waren die Kollegen der Meinung, dass man dies im Idealfall ebenfalls grafisch machen sollte. Die von Fujaba zur Verfügung gestellten Möglichkeiten bieten zwar gute Ansätze, aber die Gefahr des „Spaghetti-Codes“ sind zu groß. Insbesondere L. Humbert<sup>5</sup> verwies auf den Anspruch des Informatikunterrichts, strukturierte Vorgehensweisen zu fördern und regte die Implementierung von Struktogrammen an. Diese Forderung unterstütze ich mit dieser Arbeit.

Ein Kollege thematisierte ein sprachliches Problem an. Mit dem Begriff „Klasse“ verknüpfen die Schülerinnen und Schüler implizit konkrete Vorstellungen in der Schule. Diese Vorstellungen passen in der Regel nicht zum abstrakten Begriff der Informatik. Die Fachdidaktik müsste untersuchen, inwieweit durch die Verwendung des Begriffs „Klasse“ Lernbarrieren bei Schülerinnen und Schülern in der Sekundarstufe I entstehen. Außerdem müsste geklärt werden, welchen anderen Begriff man benutzen könnte, falls es durch die falschen Assoziationen zu Lernschwierigkeiten kommt. Der Begriff des Bauplans, der in einigen didaktischen Publikationen genannt wird, bietet sich eventuell an. Allerdings zeigt das folgende Kapitel, dass Schülerinnen und Schüler diesen Begriff nicht intuitiv übernehmen.

#### 3.4.2 Klassenarbeiten

Da die Wörter Klasse und Objekt zentrale Fachbegriffe der Unterrichtsreihe sind, wurden sie auf unterschiedliche Weise in allen drei Übungen abgefragt. Den Fokus der Evaluation

---

<sup>5</sup>Fachleiter am Studienseminar Hamm

### 3 Evaluation des Projektes

richte ich daher auf diese Begriffe.

Tabelle 3.1 zeigt das Ergebnis einer qualitativen Analyse der drei schriftlichen Überprüfungen. Alle Antworten wurden im Evaluationsprozess erneut ausgewertet und in drei Kategorien eingeteilt. In der ersten Kategorie waren die Schülerinnen und Schüler in der Lage, die Begriffe Objekt und Klasse exakt zu beschreiben und zu definieren<sup>6</sup>, z.B.: *Objekte werden in der Informatik konkret auftretende Personen oder Gegenstände bezeichnet, die immer, obwohl Objekte manchmal das gleiche Aussehen haben, einzigartig sind. In Klassen werden nur gleichartige Objekte zusammengefasst, damit sie besser geordnet werden können. Man sagt auch Klassen sind „Baupläne“ aus denen Objekte erzeugt werden.*<sup>7</sup> (Schüler E.)

	1. Klassenarbeit	schriftliche Übung	2. Klassenarbeit
exakte Beschreibung	6	4	3
gutes Verständnis	8	9	10
fehlerhafte Vorstellung	4	3	4

Tabelle 3.1: Qualität der Antworten zu den Begriffen Objekt und Klasse in den schriftlichen Überprüfungen

In der Kategorie „gutes Verständnis“ wurden Antworten gezählt, die das Verständnis zeigten, jedoch noch nicht exakt genug formuliert waren. Dazu gehörten zum Beispiel die folgenden Aussagen: *In einer Klasse sind Objekte gegliedert, die das gleiche Wissen, Können und die gleichen Aufgaben haben.* (Schüler A.) *Klasse: fasst Objekte zu einer Gruppe zusammen; Objekt: ein konkret auftretender Gegenstand bzw. Person* (Schüler Je.) *Ein Objekt gehört zu einer Klasse. In einem Objekt wird beschrieben, welche Aufgaben ein Objekt übernimmt und welche anderen Objekte es kennt.* (Schüler F.) *Objekte sind einmalige, individuelle Gegenstände (Dinge, Sachen) die ihrer Ähnlichkeit eingeteilt.* (*sic!*) (Schülerin J.)

Antworten, aus denen die grundlegenden Ideen von Objekten und Klassen nicht hervorgingen, wurden in der letzten Kategorie gesammelt.

Die Analyse zeigt, dass die meisten Schülerinnen und Schüler in der Lage waren, die grundlegenden Ideen der Objektorientierung zu reproduzieren. Die Abnahme der exakten

<sup>6</sup>Als Maßstab wurden die im Unterricht eingeführten Definitionen herangeführt. Im Rahmen einer didaktischen Reduktion weichen einige Definitionen von fachlichen Publikationen ab, ohne jedoch eine falsche Vorstellung bei den Schülerinnen und Schülern zu erzeugen. So wurde z.B. der Fokus beim Modellieren auf reale Objekten gelegt - „künstliche“, technische Objekte wurden aus didaktischen Gründen ausgeblendet.

<sup>7</sup>Alle Zitate von Schülerinnen und Schülern sind als kursives Schriftbild abgedruckt. Rechtschreib- oder Ausdrucksfehler wurden aus dem Original übernommen!

### 3 Evaluation des Projektes

Beschreibungen lässt sich durch den Zeitpunkt der Arbeiten erklären. Die zweite Klassenarbeit fand vier Wochen nach der Unterrichtsreihe statt und betraf inhaltlich schon die neue Thematik. Insofern gehe ich davon aus, dass das Auswendiglernen in dieser Arbeit nicht in dem selben Maße wirkte wie in der ersten Klassenarbeit. Um so positiver ist zu bewerten, dass die Grundvorstellungen der Objektorientierung über die eigentliche Unterrichtseinheit hinaus bei den Schülern vorhanden war. Dass einige Schüler nicht zu erreichen waren, hängt vermutlich auch mit der mangelnden Motivation dieser Schüler zusammen (vgl. Kapitel 3.4.4).

Interessant ist die Beobachtung, dass die Schülerinnen und Schüler ihre Vorstellung von den Begriffen im Laufe der Unterrichtseinheit mit den neuen Methoden, insbesondere UML, verknüpften. Zum einen nutzen die Schüler UML-Notationen um Objekte bzw. Klassen darzustellen oder die Begriffe an konkreten Beispielen zu erläutern. Zum anderen scheint die grafische Notation die Schülerinnen und Schüler dazu zu verleiten, in Strukturen zu denken. Häufig werden in Antworten die Beziehungen zwischen Klassen und Objekten erwähnt. *Objekte werden in Klassen unterteilt. Jedes Objekt muss einer Klasse angehören. Es können unendlich viele Objekte geben. Jedes Objekt hat eine eigene Identität.* (Schüler A.)

Äußerungen von Schüler lassen mich vermuten, dass die Benutzung von Fujaba und Mr. Dobs sehr zum Verständnis von OOP beitragen. Ein typisches Zitat ist: *Klassen bestehen aus Objekten. Sie treten im späteren Programm **nicht** mehr zum Vorschein, sie geben an, was ein Objekt kann, weiß und kennt. Objekte sind immer nur **einer** Klasse zugeteilt.* (Schüler S.) Das Modellieren von Klassen in Fujaba und das anschließende Testen der Objekte in Mr. Dobs scheint sich positiv auf den Lerneffekt auszuwirken.

Zum Abschluss dieses Themas möchte ich ein Zitat aus der zweiten Klassenarbeit heranziehen: *Zu einer Klasse werden gleichartige Objekte zusammengefasst. Klassen sind Baupläne, aus denen Objekte erzeugt werden. Ein Objekt ist ein konkreter Gegenstand oder eine Person. Jedes Objekt ist einzigartig.* (Schüler E.) An dieser Stelle benutzt der Schüler den Begriff Bauplan, der im vorherigen Kapitel bereits angesprochen wurde. Im Unterricht wurde dieser Begriff als Metapher eingeführt. Die Auswertung des schriftlichen Materials ergab aber, dass nur wenige Schüler diesen Begriff übernommen haben. Stattdessen orientierten sich die Schüler eher an den abgewandten UML Diagrammen. Ich vermute, dass ein Grund dafür ist, dass der Bauplan lediglich als Metapher benutzt wurde. Sollte man in der Sekundarstufe I den Begriff Bauplan wirklich einführen, so müsste den Schülerinnen und Schüler die Möglichkeit gegeben werden, den Begriff praktisch anzuwenden. Immer weniger Schüler haben Erfahrungen im Modellbau, so dass das

### 3 Evaluation des Projektes

Anwenden von Bauplänen nicht unbedingt in den Köpfen verankert ist. Im nächsten Unterrichtsversuch sollte den Schüler ein kleines Lego-Modell mit einem Bauplan vorgelegt werden. Dies könnte die Einführungsstunde zum Thema Klassen und Objekte abrunden.

Eine Hoffnung des objektorientierten Ansatzes ist es, die Schülerinnen und Schüler stärker für Probleme der Softwareentwicklung zu sensibilisieren, als dies die bisherige imperative Vorgehensweise erreicht. Diese wurde im Unterricht thematisiert und ebenfalls überprüft. In der zweiten Klassenarbeit wurden die Schüler daher nach Problemen gefragt, die in der Modellierungsphase auftreten können. Hier exemplarische Antworten: *Beim Modellieren tritt die Schwierigkeit auf, dass man auf vielen Wegen zu einem Ziel gelangen kann.* (Schüler M.) *Für viele Probleme gibt es verschiedene Lösungen. Man muss genau wissen, was das Programm nachher können und wissen soll, damit man zwischen den Lösungen die richtige wählen kann. Fehler im Modell können nachher nur schwer entfernt werden, evtl. muss alles komplett neu programmiert werden.* (Schüler S.) *Es gibt ein Problem und mehrere Lösungen. Jedoch hat jede Lösung ihre Vor- und Nachteile.* (Schüler K.) *Das Problem beim Modellieren ist, dass es viele Möglichkeiten gibt ein Programm umzusetzen, und jeder Mensch, der Teil des Projektes ist, eine andere Vorstellung hat.* (Schüler B.)

Die Äußerungen zeigen, dass die Schülerinnen und Schüler einen wichtigen Aspekt der Softwareentwicklung gelernt haben, der es ihnen später ermöglicht, als Benutzer bei Softwareprojekten besser zu partizipieren. Selbstverständlich muss der Informatikunterricht dieses Bild weiter ausdifferenzieren. Dies zeigt meiner Meinung nach aber die Stärke des objektorientierten Ansatzes. Man muss bedenken, dass diese Auswertung am Ende des Anfangsunterrichts stattfand und sie mit den imperativen Ansätzen vergleichen, die dieses Ziel in der Regel erst sehr viel später erreichen.

Positiv kann man das Verwenden von UML-Diagrammen bewerten. Die Schülerinnen und Schüler hatten keine Probleme mit der Bearbeitung dieser Aufgaben. UML Diagramme können vielfältig eingesetzt werden. Schüler können Diagramme selbst erstellen, vorhandene Diagramme ergänzen oder auf Fehler untersuchen. Auch für die Lehrperson sind solche Aufgaben sehr angenehm. Mit Hilfe der vorhandenen Tools können Diagramme einfach erstellt werden. Auch die Korrektur von UML Aufgaben in Klassenarbeiten ist sehr angenehm. Durch geeignete Aufgabenstellungen können mit Hilfe der Diagramme auch komplexere Sachzusammenhänge und Verständnisfragen sehr einfach überprüft werden.

### 3.4.3 Lerntagebuch

Die Auswertung der Lerntagebücher ergab, dass die Schülerinnen und Schüler mit solchen Evaluationsmaßnahmen nicht vertraut sind. Die Schülerinnen und Schüler sind sehr zurückhaltend mit ihren Einträgen gewesen. Trotzdem lassen sie interessante Rückschlüsse zu.

Die erste Analyse ergibt, dass viele Schüler die Gruppenarbeit positiv bewertet haben. Auf die Frage „Was hat mir Spaß gemacht?“ haben viele Schüler als Antwort „Gruppenarbeit“ gegeben (z.B. die Schüler Ja., Ei., Eu., M., An., F., S., Je., Jo., Lu.).

Aus den Fragebögen kann man entnehmen, dass das Ziel, die Schüler mit den Gruppenarbeitsphasen zu selbstständigem Lernen zu aktivieren, erreicht und von den Schülern positiv bewertet wurde. Folgende Zitate aus dem Bereich „Spaß“ finden sich in den Lernbüchern wieder, die diese These belegen: *Mir hat Spaß gemacht, verschiedene Teile des Spiels zu klassifizieren und zu unterteilen.* (Schüler Ar.), *selbstständiges Überlegen, was Klassen, Objekte, etc. sein können* (Schüler K.), *Diskussion* (Schüler P.), *Das Diskutieren mit meinem Nachbarn* (Schüler An.), *in der Gruppe [...] sich was auszudenken* (Schüler N.), *Die gut funktionierende Gruppenarbeit* (Schüler Ei.) oder *CRC-Karten selbst erstellen* (Schüler Eu.)

Zwei Schüler fallen mit ihren Aussagen aus den bisherigen Kommentaren heraus. Der Schüler Ni. fällt durch eine durchgehend negative Einstellung zum Unterricht auf. Häufig sind Kommentare wie *nichts, ödel, ödel* oder *so tun als würd ich aufpassen* (sic!) in seinem Lerntagebuch zu finden. Bei einem zweiten Schüler (B.) setzt eine ähnliche Haltung nach der zweiten Stunde ein. Von dieser Stunde an wird die Frage „Was hat mir keinen Spaß gemacht?“ beispielsweise wie folgt beantwortet: *Diskutieren über Unterschiede und Gemeinsamkeiten, Das Spiel spielen, alles.* Leider sind die Antworten nicht ausführlich genug, um Aussagen über die Ursachen dieser Abneigung gegenüber dem Unterricht festzustellen. Allerdings bestätigt dies die These der Demotivation einiger Schüler aus dem vorherigen Kapitel.

### 3.4.4 Evaluationszielscheibe

Das Ergebnis der Evaluationszielscheibe (Abbildung 3.1) hat mich überrascht. Für den ersten Durchgang einer Unterrichtsreihe, der ein neues Konzept zu Grunde lag, wurde von den Schülerinnen und Schülern eine meiner Meinung nach sehr gute Bewertung abgegeben. Mit einer so guten Bewertung hatte ich im Vorfeld der Evaluation nicht gerechnet.

An der Zielscheibe kann erkannt werden, dass den Schülerinnen und Schülern in der

### 3 Evaluation des Projektes

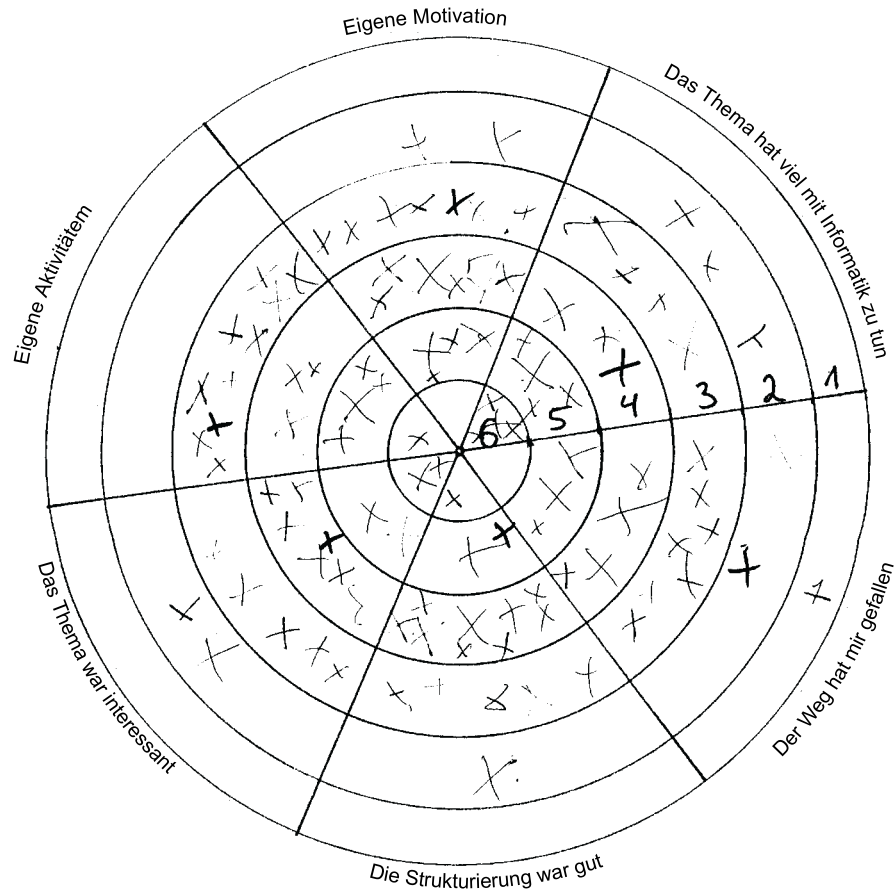


Abbildung 3.1: Evaluationszielscheibe nach C. Burkhard und G. Eikenbusch

Einheit sehr gut die Grundprinzipien der Softwareentwicklung vermittelt werden konnten. Die Schüler haben den hohen Zusammenhang zwischen der (Fach-)Informatik und dem Unterricht erkannt. Auch die gute Bewertung der Strukturierung zeigt meines Erachtens, dass die Schüler in der Reflexionsphase die Bedeutung der einzelnen Phasen im Softwareentwicklungsprozess erkannt haben. Andernfalls müsste dieser Punkt in der Bewertung wesentlich schlechter abschneiden.

Die Angaben zur Motivation entspricht dem Erwartungshorizont einer Klasse 9. Die sehr wenig motivierten Schüler werden sich vermutlich auch in den Lernerfolgskontrollen nicht besonders angestrengt haben, so dass sich damit auch der nicht vorhandene Lernfortschritt in der dritten Kategorie erklären lässt.

Überraschend ist, dass die eigenen Aktivitäten durch mehr als die Hälfte der Schülerinnen und Schüler nur im Mittelfeld bewertet wurden. Die Videobänder zeigen, dass im Unterricht sehr viele Gruppenarbeitsphasen bzw. Arbeiten am Computer stattfanden.

Auch steht diese Aussage im Widerspruch zu den Lerntagebüchern.

Ich vermute, dass die Schüler die eigenen Aktivitäten mit der Arbeit am Computer gleichsetzten. Im Verlauf des Unterrichts haben die Schüler öfters gefragt, wann sie endlich wieder am Computer arbeiten könnten.

Hier stellt sich meiner Meinung nach eine zentrale Frage für die Fachdidaktik der Sekundarstufe I: Die Arbeit mit dem Computer macht sicherlich einen großen Reiz des Fachs aus. Die Erwartungshaltung der Schüler, fast ausschließlich mit dem Rechner zu arbeiten, halte ich aber für kontraproduktiv. Ein Großteil der Arbeitszeit eines Softwareentwicklers findet bekanntlich nicht am Rechner statt. Den Fehlvorstellungen, dass Informatik nur vor dem Bildschirm stattfindet, die sicherlich durch Freunde oder ältere Geschwister geprägt werden<sup>8</sup>, sollten im Informatikunterricht der Mittelstufe gezielt entgegengewirkt werden. Die relativ hohe Abbrecherquote in den Informatik-Studiengängen an den Universitäten lässt sich sicherlich auch durch diese Fehlvorstellung erklären. Deshalb sollte trotz der schlechten Bewertung an diesem Teil des Konzeptes festgehalten werden.

Sehr positiv im Gespräch wurde von den Schülerinnen und Schülern die Arbeit in einer heterogenen Softwareumgebung hervorgehoben. Die Benutzung von X-Windows, um auf einem entfernten Linux-Server seine Programme laufen zu lassen, war für die Schüler faszinierend und spannend, obwohl die Software relativ langsam lief. Trotz einiger Reibungsverluste durch den Einsatz eines neuen Betriebssystems sollten meiner Meinung nach im Informatikunterricht verstärkt solche Techniken eingesetzt werden. Der Vorwurf einer Produktschulung kann umgangen und die Schülerinnen und Schüler auf das Arbeiten in einer vernetzten Gesellschaft vorbereitet werden.

#### 3.4.5 Videoanalyse

Die Auswertung des umfangreichen Videomaterials stellte sich, insbesondere bei den Gruppenarbeitsphasen, als schwierig heraus. Die Qualität der Tonaufnahmen sinkt rapide, wenn mehrere Schüler reden - was bei Diskussionen in der Gruppe erwünscht war. Im Rahmen dieser schriftlichen Arbeit möchte ich nur kurz auf zwei Phasen des Unterrichts eingehen. Eine ausführliche Analyse inklusive Videoausschnitten ist auf der beiliegenden CD-ROM enthalten.

---

<sup>8</sup>Diese Beobachtung spiegelt sich auch in den Aussagen der PISA-E Studie wieder. Deutsche Schüler zeigen zwar ein großes Interesse an Computern, schätzen ihre Fähigkeiten aber sehr gering ein. Die Forscher begründen dies mit dem Nutzungsverhalten deutscher Jugendlicher. 58,4% der Jugendlichen nutzen Computer für Spiele mindestens einmal die Woche, während nur 12,4% mindestens einmal wöchentlich Lern- oder Arbeitssoftware einsetzt. vgl. [PISA (2003)] S. 195-209, insb. S. 198 und S. 200

### 3 Evaluation des Projektes

Die Präsentationsphasen im Anschluss an die Gruppenarbeitsphasen zeigen einen deutlichen Lernfortschritt der Schülerinnen und Schüler. Insbesondere Probleme der Softwareentwicklung werden von den Schülern sehr früh im Unterricht angesprochen und diskutiert. Dabei erreichten die Schüler bereits nach wenigen Stunden grundlegende Fragestellungen der Informatik. Selbstverständlich erlangte dieses Verständnis nur eine begrenzte Breite und Tiefe. Jedoch zeigen die Videoaufnahmen, dass die gewählten Methoden und die Diskussionen der Schüler untereinander es ermöglichen, derartige Themen bereits im Anfangsunterricht der Sekundarstufe I erfolgreich anzusprechen.

Von dem Objektspiel liegen zwei Videosequenzen aus unterschiedlichen Stunden vor. Beide Sequenzen zeigen, dass die Schülerinnen und Schüler das Modell zügig verstehen. Fehler von Spielern werden sehr schnell von den Mitschülern erkannt und korrigiert. Der Lehrer kann sich daher nach kurzer Zeit aus seiner aktiven Rolle zurückziehen. Am ersten Tag des Objektspiels waren die Schülerinnen und Schüler sehr diszipliniert. Am zweiten Tag kamen einige Schüler auf die Idee den Ball fester zu werfen. Sofern der Lehrer dies frühzeitig unterbindet, lässt sich das Spiel ohne Komplikationen durchführen.

Die Videos zeigen, dass sich das Objektspiel effektiv und motivierend in der Sekundarstufe I einsetzen lässt.

## 4 Die Bedeutung des Projektes für zukünftige Unterrichtsvorhaben

Die Erfahrungen der Durchführung und die Auswertung der Evaluation lassen ein sehr positives Fazit der Unterrichtsreihe zu. Zusammenfassend können die drei zentralen Fragestellungen der Evaluation (vgl. Seite 21) wie folgt beantwortet werden:

1. Die Reihe vermittelte den Schülerinnen und Schülern grundlegende Konzepte der objektorientierten Softwareentwicklung. Sie waren am Ende der Einheit in der Lage, die Begriffe Klasse und Objekt zu unterscheiden und aus kleinen realen Problemen Objekte und Klassen zu identifizieren, Beziehungen zu erkennen und entsprechende UML Diagramme zu erstellen.
2. Die gewählten Methoden erlaubten den Schülerinnen und Schülern, sehr selbstständig zu lernen. Dies wurde insbesondere in den Auswertungen der Lerntagebücher deutlich. Die langen Gruppenarbeitsphasen zu Beginn der Unterrichtsreihe wurden von vielen Schülern zum selbstentdeckenden Lernen genutzt. Gleichzeitig kamen von den Schülern Fragen, die grundlegende Probleme der Modellierung und Softwareentwicklung aufwarfen.
3. Die erhoffte starke Schülerzentrierung und die bessere Wahrnehmung für Probleme der Modellierung wurden in dieser Unterrichtsreihe umgesetzt. Einige Arbeitsblätter sollten überarbeitet und grafisch ansprechender gestaltet werden. Der Versuch, statt dem Begriff „Klasse“ den Begriff „Bauplan“ stärker zu nutzen, sollte im nächsten Durchlauf erprobt werden. Das größte Problem des Gesamtkonzeptes bildet der Übergang von der Implementierung der Methoden zu der Erstellung einer grafischen Benutzeroberfläche.<sup>1</sup> Das von Fujaba angebotene Konzept des Story Pattern's

---

<sup>1</sup>Dieser Schritt gehört thematisch erst an den Anschluss der Einheit meiner Examensarbeit. Trotzdem sei an dieser Stelle auf die Problematik hingewiesen. Eine weitere Arbeit zu diesem Thema wird vermutlich die Schwierigkeiten lösen können, was aus zeitlichen Gründen an dieser Stelle nicht geleistet werden konnte.

#### 4 Die Bedeutung des Projektes für zukünftige Unterrichtsvorhaben

ist sehr schön, hat aber die nicht zu akzeptierende Schwachstelle der unstrukturierten Sprünge. Es wäre wünschenswert, wenn die Entwickler von Fujaba stattdessen Struktogramme implementieren würden. Zur Koppelung des Modells mit einer GUI steht unter Fujaba bereits eine Bibliothek zur Verfügung, die aber im Rahmen dieser Examensarbeit nicht erprobt werden konnte. Da die Erstellung eines „echten“ Programms, d.h. inklusive einer Benutzeroberfläche, einen hohen Motivationswert für die Schülerinnen und Schüler hat, sollte die Reihe um diesen Punkt erweitert werden.

Gemessen an den Lernzielen der Einheit lässt sich positiv feststellen: Die Schülerinnen und Schüler konnten am Ende der Unterrichtseinheit,

- die Begriffe Klasse und Objekt unterscheiden und anwenden.
- Objekte und Objektstrukturen an einem Beispiel erkennen und entsprechende
- CRC-Karten erstellen.
- UML-Klassendiagramme zeichnen und mit Hilfe von Fujaba in lauffähige Programme umsetzen.
- produktiv in der Gruppe arbeiten.

Insgesamt zeigt diese Arbeit, dass objektorientierte Techniken den Unterricht in der Sekundarstufe I beleben. Nicht nur in der Theorie, sondern auch in der Praxis lassen sich frühzeitig Sichtweisen vermitteln, die sich im imperativen Ansatz erst sehr viel später (zu spät?) verwirklichen lassen. Die kleineren Schwächen, die sich in einzelnen Phasen noch befinden, können in den nächsten Erprobungen leicht ausgebessert werden.

Der erfolgreiche Abschluss dieser Arbeit sollte viele Informatiklehrerinnen und -lehrer motivieren, einen objektorientierten Ansatz in der Sekundarstufe I zu erproben. Die Materialien dieser Arbeit werden im Internet zur Verfügung gestellt.

# Literaturverzeichnis

- [Baumann (1996)] Baumann, Rüdiger: Didaktik der Informatik. Stuttgart, München, Düsseldorf, u.a.: 1996.
- [Beck, Cunningham (1989)] Beck, Kent;Cunningham, Ward: A Laboratory For Teaching Object-Oriented Thinkinig. From: OOPSLA '89 Conference Proceedings. New Orleans, Louisiana. <http://c2.com/doc/oopsla89/papers.html> 26. Januar 2003 (also published: SIGPLAN Notices, Volume 24, Number 10, October 1989)
- [Bergin (2000)] Bergin, Joseph: The Object Game. An excercise for studying objects: <http://csis.pace.edu/~bergin/patterns/objectgame.html> 25. Januar 2003
- [Burkhard, Eikenbusch (2000)] Burkhard, Christoph; Eikenbusch, Gerhard: Praxishandbuch Evaluation in der Schule. Berlin: 2000
- [Diethelm, Geiger, Zündorf (2002)] Diethelm, Ira; Geiger, Leif; Zündorf, Albert: UML im Unterricht: Systematische objektorientierte Problemlösungen mit Hilfe von Szenarien am Beispiel der Türme von Hanoi. In: Schubert, S.; Magenheimer, J.; Hubwieser, P.; u.a.: Forschungsbeiträge zur Didaktik der Informatik - Theorie, Praxis, Evaluation. Tagungsband des 1. Workshops der GI-Fachgruppe „Didaktik der Informatik“ (DDI'02). Schwerpunkt: Modellierung in der informatischen Bildung. Bonn: 2002, S. 33-42
- [Hubwieser (2000)] Hubwieser, Peter: Didaktik der Informatik. Grundlagen, Konzepte, Beispiele. Berlin, Heidelberg, New York, u.a.: 2000
- [Hubwieser, Humbert und Schubert (2001)] Hubwieser, Peter; Humbert, Ludger; Schubert, Sigrid: Evaluation von Informatikunterricht. In: Keil-Slawik, R.; Magenheimer, J. (Hrsg.): Informatikunterricht und Medienbildung. INFOS 2001. 9. GI-Fachtagung Informatik und Schule. 17.-20. September 2001 in Paderborn. Bonn: 2001, S. 213-215

## Literaturverzeichnis

- [Lehrplan NRW (1993)] Kultusminister des Landes Nordrhein-Westfalen: Richtlinien und Lehrpläne. Informatik. Düsseldorf, Frechen: 1993
- [*life*<sup>3</sup> Homepage (2003)] Homepage des *life*<sup>3</sup>-Projekts: <http://life.uni-paderborn.de> 14. Januar 2003
- [Kuri (2003)] Kuri, Jürgen: IBM kauft Rational Software. c't magazin für computer technik, Hannover (2003) 1, S. 40
- [Oestereich (1998)] Oestereich, Bernd: Objektorientierte Softwareentwicklung. Analyse und Design mit der Unified modeling language. München, Wien, Oldenbourg: 1998
- [PISA (2003)] Deutsches PISA-Konsortium (Hrsg.): PISA 2000. Ein differenzierter Blick auf die Länder der Bundesrepublik Deutschland. Opladen: 2003
- [Sanders (Hrsg.) (2000)] Sanders, James R. (Hrsg.): Handbuch der Evaluationsstandards. Die Standards des „Joint Committee on Standards for Educational Evaluation. Opladen: 2000
- [Schulte (2003)] Schulte, Carsten: Auszüge aus der Dissertation (in Arbeit; bis jetzt unveröffentlicht) 2003
- [Seifert (2003)] Seifert, Monika: Informatik in der Sekundarstufe II. Vom Gesamtkonzept zum Curriculum - Planung von Kurssequenzen. In: LOG IN. Informatische Bildung und Computer in der Schule, Berlin (2003) Nr. 124, S. 10-16
- [Spohrer (2002)] Spohrer, Matthias: Zustandsorientierte Modellierung. Grundgedanken zur Umsetzung dieses Themas im Informatikunterricht der Mittelstufe. In: Schubert, S.; Magenheimer, J.; Hubwieser, P.; u.a.: Forschungsbeiträge zur Didaktik der Informatik - Theorie, Praxis, Evaluation. Tagungsband des 1. Workshops der GI-Fachgruppe „Didaktik der Informatik“ (DDI'02). Schwerpunkt: Modellierung in der informatischen Bildung. Bonn: 2002, S. 23-32

# A Anhang

## **Erläuterung der Evaluationsstandards**

Wie im Kapitel 3.1 erwähnt, hat das amerikanische Joint Committee on Standards for Educational Evaluation in seiner Veröffentlichung [Sanders (Hrsg.) (2000)] Standards für Evaluationen festgelegt. In den folgenden Abschnitten wird gezeigt, dass sich meine Evaluation - im Rahmen der organisatorischen Möglichkeiten - sich an die definierten Standards gehalten hat.

*N1 Ermittlung der Beteiligten und Betroffenen:* Durchgeführt wird die Evaluation von Thorsten Reinsch. Betroffen sind die Schülerinnen und Schüler des Informatik Differenzierungskurs, teilweise deren Erziehungsberechtigten. Eine externe Evaluation ließ sich aus schulorganisatorischen Gründen nicht verwirklichen. Das Studienseminar Bonn und die Schulleitung des Ernst-Moritz-Arndt Gymnasium Bonn waren im Genehmigungsverfahren von der Evaluation betroffen.

*N4 Feststellung von Werten:* Die Unterrichtseinheit beruht auf der Annahme, dass ein objektorientierter Anfangsunterricht in der Sekundarstufe I mit visuellen Werkzeugen wie UML und Fujaba einen altersgerechten und motivierenden Einstieg in die Softwareentwicklung liefert. Während der Evaluation ist darauf zu achten, dass die Ergebnisse aus Sicht dieser Annahme nicht verfälscht werden.

*N5 Klarheit des Berichts:* Dieses Kapitel versucht die Ziele, Verfahren und Befunde der Evaluation transparent zu beschreiben.

*D1 Praktische Verfahren:* Im Rahmen der Evaluation werden im wesentlichen Methoden eingesetzt, die sich einfach in den normalen Unterrichtsablauf integrieren lassen. Aus diesem Grund wird z.B. auf eine zweite Videokamera verzichtet, da kein Personal zum Auf- und Abbau zur Verfügung steht. Des Weiteren werden schultypische Dokumente (z.B. Klassenarbeiten) zur Evaluation herangezogen.

*K3 Schutz individueller Menschenrechte:* Videoaufzeichnungen während des Unterrichts beeinträchtigen die Rechte der Schülerinnen und Schüler sowie deren Erziehungsberechtigten. Aus diesem Grund wurde vor der Evaluation das Einverständnis von Schülern und Eltern eingeholt. Die Planung wurde so rechtzeitig begonnen, dass alle Eltern

auf dem Elternsprechtag zu den Halbjahreszeugnissen die Möglichkeit hatten sich ausführlich über die geplanten Videoaufzeichnungen zu informieren. Außerdem wurde das Projekt dem Studienseminar und der Schulleitung zur Genehmigung vorgelegt.

*K5 Vollständige und faire Einschätzung:* Die Evaluation versucht Stärken und Schwächen der Unterrichtseinheit aufzuspüren. In der Beurteilung wird versucht eine objektive Sicht einzunehmen, die es ermöglicht den Unterricht konstruktiv weiterzuentwickeln.

*K6 Offenlegung der Ergebnisse:* Diese Examensarbeit wird nach ihrem Abschluss im Internet veröffentlicht. Interessenten können beim Autor ein Exemplar bestellen. Bei der Offenlegung wird auf eine Anonymisierung der Daten geachtet, so dass die Schutzinteressen der beteiligten Schülerinnen und Schüler gewahrt werden. Eine Veröffentlichung von Videosequenzen geschieht nur mit einer weiteren Einverständniserklärung der betroffenen Schülerinnen und Schülern sowie derer Eltern.

*G1 Programmdokumentation:* Im Rahmen der schriftlichen Hausarbeit und ihren Anlagen wird versucht die Evaluation klar und genau zu beschreiben.

*G3 Beschreibung von Zielen und Vorgehen:* Das nächste Kapitel versucht die angewendeten Methoden ausführlich zu dokumentieren. Damit soll den Lesern die Möglichkeit gegeben werden, die Ergebnisse der Evaluation besser einzuschätzen.

*G4 Verlässliche Informationsquellen:* Die im Abschnitt 3.3 erläuterten Dokumente sollen eine verlässliche Evaluation ermöglichen. So dienen die Videoaufnahmen dazu Unterrichtssequenzen im Nachhinein zu analysieren, um so Details zu betrachten, die im ersten Durchgang übersehen wurden.

*G10 Begründete Schlussfolgerungen:* Die im Kapitel 4 gezogenen Schlussfolgerungen werden durch die Ergebnisse der Evaluation begründet.

Weitere Evaluationsstandards werden bei der Beschreibung der angewendeten Methoden berücksichtigt.

# Arbeitsblätter

Im folgenden befinden sich ausgewählte Arbeitsblätter der Unterrichtsreihe.

## *Arbeitsblatt 1*

### **Arbeitsauftrag**

Bestimmt eine(n) Protokollführer/in aus eurer Gruppe. Dokumentiert den Spielablauf genau. Spielt mehrere Runden. Verteilt die Münzen ggf. neu.

Überlegt Euch folgende Fragen und beantwortet sie schriftlich:

- Welche Gegenstände werden für das Spiel benötigt?
- Welche Personen sind an dem Spiel beteiligt?
- Sind die Spielregeln präzise genug formuliert?

Übertragt die Ergebnisse aus eurem Protokoll für die Präsentation auf die Karteikarten (Stichworte reichen aus).

Bitte gebt euer Protokoll am Ende der Stunde ab!

### **Spielregeln „Flaschen drehen“**

Zu Beginn des Spiels erhält jeder Spieler 10 Münzen. Die Spieler stellen sich im Kreis um die Flasche auf und legen vor sich ein Spielfeld auf den Boden.

In jeder Runde setzen die Spieler einen Betrag auf ein Spielfeld. Das Spielfeld auf das die Flasche zeigt gewinnt. Anschließend wird eine neue Runde gespielt.

## Arbeitsblatt 2

### Hintergrundinformationen

In der letzten Stunde habt Ihr bereits die beteiligten Gegenstände und Objekte unseres Spiels identifiziert. In der Informatik heißen konkret auftretende Gegenstände und Personen **Objekte**. Jedes Objekt hat eine eigene Identität, selbst dann, wenn zwei Objekte gleich aussehen. Ihr könnt Euch dies an den Münzen klar machen: Alle Münzen sehen identisch aus. Trotzdem ist jede Münze einmalig.

Gleichartige Objekte werden zu **Klassen** zusammengefasst. Dabei wird jedem Objekt genau eine Klasse zugewiesen. Natürlich können mehrere Objekte zur selben Klasse gehören, allerdings kann ein Objekt nicht gleichzeitig zwei Klassen zugewiesen werden.

Bevor mit dem Programmieren einer Software begonnen werden kann, müssen die Klassen und Objekte gefunden werden, die zur Problemlösung benötigt werden. Dieser Prozess heißt **Modellierung**. Die Modellierung ist eine der wichtigsten Phasen bei der Softwareentwicklung, da Fehler im Modell später sehr schwierig zu korrigieren sind. Bei einem gravierenden Fehler muss ggf. alles neu programmiert werden.

Eine Schwierigkeit beim Modellieren ist, dass es keine eindeutigen Lösungen gibt. Häufig können mehrere Modelle das Problem lösen, wobei jedes Modell Vor- und Nachteile hat. Modelle die die gegebene Aufgabenstellung jedoch nicht lösen sind falsch. Die Wahl des endgültigen Modells hängt von den unterschiedlichen Interessen der beteiligten Entscheidungsträger ab.

Das Finden von Klassen und Objekten kann durch CRC-Karten erleichtert werden. CRC steht für **Class-Responsibility-Collaboration** (Klasse, Aufgabe/Verantwortlichkeit, Mitarbeiter/Kollaborateur).

Eine CRC-Karte hat folgenden Aufbau:

Name der Klasse	
(Responsibility) - Welche Aufgaben hat die Klasse?  - Was kann die Klasse?  - Was weiß die Klasse?	(Collaborators) - Welche anderen Gegenstände oder Personen sind dieser Klasse bekannt, die bei der Erledigung der Aufgaben mitarbeiten können?

## *A Anhang*

Jede CRC-Karte steht für eine Klasse. Von jeder Klasse die im Modell vorkommt muss mindestens ein Objekt existieren. Häufig existieren mehrere Objekte pro Klasse. In einem konkreten Spiel existieren nur Objekte. Die Klassen dienen lediglich zur Strukturierung, wenn man während einer Modellierungsphase versucht das Modell zu analysieren.

Objekte wissen nur Dinge bzw. können nur Aktionen ausführen, die im Feld „Responsibility“ aufgeführt sind. Auch können sie nur Objekte kennen, die von einer Klasse sind die im Bereich „Collaborators“ steht.

### *Arbeitsblatt 3*

#### **Aufgaben**

Eine Gruppe von Studentinnen und Studenten hat ein Programm „Flaschen drehen“ geschrieben. Bei der Modellierung hat die Gruppe CRC-Karten erstellt. Diese Karten stehen euch nun zur Verfügung.

1. Vergleicht die CRC-Karten mit dem Spiel das Ihr gespielt habt. Welche Gemeinsamkeiten und welche Unterschiede gibt es?
2. Aus welchen unterschiedlichen Bereichen besteht eine CRC-Karte? Erläutert die Funktionen der einzelnen Bereiche.
3. Versucht am Beispiel des CRC-Karten-Modells die Begriffe Objekt und Klasse zu erklären.
4. Was versteht man unter dem Begriff Modellieren? Warum ist die Phase der Modellierung in der Softwareentwicklung wichtig?

## Arbeitsblatt 4

### Wiederholung

Die erste Phase der Softwareentwicklung heißt \_\_\_\_\_. In dieser Phase versucht man die \_\_\_\_\_ zu finden, die zur Problemlösung beitragen. Gleichartige \_\_\_\_\_ werden zu \_\_\_\_\_ zusammengefasst.

CRC-Karten helfen in dieser Phase. Dabei steht CRC für: \_\_\_\_\_ - \_\_\_\_\_ - \_\_\_\_\_. Eine CRC-Karte ist hier abgebildet. Wofür stehen die einzelnen Bereiche der Karte? Schreibe die Bedeutung in die CRC-Karte hinein.


Jedes Objekt ist \_\_\_\_\_ einer Klasse zugeordnet, jedoch können einer Klasse \_\_\_\_\_ Objekte zugeordnet sein. Obwohl Objekte gleich aussehen können, ist jedes Objekt \_\_\_\_\_. Man sagt auch, dass jedes Objekt seine eigene \_\_\_\_\_ besitzt.

## Arbeitsblatt 5

### Arbeitsauftrag

Versucht für das CRC-Karten-Modell der Studenten Objektkarten anzulegen. Objektkarten (orange) sind genauso aufgebaut wie CRC-Karten. Allerdings werden auf einer Objektkarte nur die Dinge aufgeschrieben, die ein Objekt tatsächlich kennt. Hier ein Beispiel für eine Objektkarte zur Klasse „Spieler“:

$s_1$	
8 Cent	$f_1$ (Flasche) $f_4$ (Feld 3)

Das Objekt steht hier für den ersten Spieler und heißt  $s_1$ . Es kennt die Flasche  $f_1$  und hat sein Geld auf das Feld 3 mit dem Namen  $f_4$  gelegt. Die Aktionen „stößt Flasche an“ und „setze Betrag auf Feld“ werden nicht auf die Objektkarte geschrieben, da es sich um ausführbare Handlungen handelt.

Legt Objektkarten für alle Klassen (Spieler, Flasche und Feld) an. Bereitet die Präsentation eurer Objektkarten an der Tafel vor.

## *Arbeitsblatt 6*

### **Aufgabe**

Für die Schülerbibliothek am Ernst-Moritz-Arndt-Gymnasium soll ein Programm geschrieben werden, mit dem die Ausleihe der Bücher optimiert werden kann.

Entwerft in Eurer Gruppe mit CRC-Karten ein Modell der Software.

# Lerntagebuch von

**Informatik Differenzierungskurs 6. März - 10. April 2003**

Ich schreibe das Lerntagebuch in erster Linie für mich selbst! Es wird nicht benotet, es wird nicht angestrichen und nicht verbessert. Was ich denke, kann nur ich beschreiben. Es ist immer richtig, weil es einfach das ist, was ich denke.

Ich kann mit diesem Lerntagebuch aber etwas über mich in Erfahrung bringen, ich kann erkennen, wie Lernen bei mir am besten klappt. Wenn ich es regelmäßig benutze, kann ich immer wieder zurückblättern, kann ich Dinge nachlesen, ich kann sehen, ob sich etwas verändert hat, ob ich besser, schlechter geworden bin, ob ich anders lerne als vorher - ich kann schauen, was ich lerne, wie ich lerne, und was und wie ich lieber lernen will.

Ich kann auch mit meinem Lehrer über mein Lerntagebuch sprechen. Ich brauche keine Angst zu haben, dass ich blöd da stehe. Denn vielleicht kann auch mein Lehrer etwas von mir lernen, was mir leicht und was mir schwer fällt, was mich noch interessiert.

Ich kann aber auch mit meinen Klassenkameraden und -kameradinnen über mein Lerntagebuch sprechen. Vielleicht ist es interessant, wie andere lernen, was anderen leicht und schwer fällt. Vielleicht können wir dadurch auch mehr voneinander lernen.



## Zeitlicher Ablauf der Unterrichtseinheit

Abschrift aus dem Kursheft:

Datum	Kurzbeschreibung
06.03. (D)	Spiel „Flaschen drehen“; Grundlagen der OO
11.03. (E)	CRC-Karten
13.03. (D)	Objektkarten; Objektspiel; UML Klassen. und Objektdiagramme
20.03. (D)	UML Klassen- und Objektdiagramme (Beispiele)
25.03. (E)	Objektdiagramme; Wiederholung
27.03. (D)	Klassenarbeit
01.04. (E)	Rückgabe und Besprechung der Klassenarbeit
03.04. (D)	Einführung in LINUX, X-Win 32 und Fujaba
10.04. (D)	Mr. Dobs
29.04. (E)	Von der CRC-Karte zum Klassendiagramm - Schritt 1
06.05. (E)	Wiederholung Mr. Dobs
08.05. (D)	Übungen - Assoziationen
13.05. (E)	Von der CRC-Karte zum Klassendiagramm - Schritt 2
15.05. (D)	Übung - Attribute und Methoden
20.05. (E)	schriftliche Übung

## B Glossar

**ACM** [Association for Computing Machinery] „The First Society in Computing“ - amerikanisches Pendant zur Gesellschaft für Informatik. Weltweit eine der wichtigsten Vereinigung von Informatikern.

**ADT** [Abstrakter Datentyp] Programmierkonzept (Vorgänger der Objektorientierung), indem die Daten und Zugriffsmethoden, z.B. in Bibliotheken), gekapselt werden.

**CASE-Tools** [Computer Aided Software Engineering] Softwareprogramme die das Entwickeln von Software nach ingenieurmäßigen Methoden unterstützen.

**CRC-Karten** [Class-Responsibility-Collaboration] CRC-Karten wurden von K. Beck und W. Cunningham erfunden. Es handelt sich dabei um Karteikarten die in drei Bereiche eingeteilt sind. Der erste Bereich enthält den Namen der Klasse (class), der zweite Bereich die Verantwortlichkeiten der Klasse (Responsibility) während der dritte Bereich die Klassen enthält, mit der die Klasse interagiert(Collaboration). Ein Beispiel ist auf Seite 10 abgedruckt.

**Mr. Dobs** [Dynamic Object Browsing System] Visueller Debugger der es ermöglicht Objektstrukturen grafisch anzuzeigen und interaktiv zu manipulieren. Bestandteil von Fujaba.

**Fujaba** [From UML to Java and back again] Ein von der Fachgruppe „Softwaretechnik“ an der Universität Paderborn unter der Leitung von Prof. Dr. W. Schäfer entwickeltes CASE-Tool, welches aus UML Diagramme Java-Code erzeugt bzw. aus Java-Code UML Diagramme erstellt. Die Software ist in Java geschrieben und steht unter der GPL.

**GPL** [General Public License] Die in der Open Source Community am meisten benutzte Lizenz. Sie ermöglicht Programme frei zu benutzen und weiter zu entwickeln. Ausführliche Informationen gibt es im Internet unter <http://www.gnu.org/licenses/licenses.html>

**GUI** [Graphical User Interface] Grafische Benutzeroberfläche

**life** [*life*<sup>3</sup>: Lernwerkzeuge für den Informatikunterricht: Einsetzen, Evaluieren und (Weiter) Entwickeln] Projekt der Fachgruppe „Didaktik der Informatik“ der Universität Paderborn unter der Leitung von Prof. Dr. J. Magenheim. Das Projekt stellt den theoretischen Rahmen für diese Examensarbeit zur Verfügung.

**NFS** [Network File System] Dieser Dienst ermöglicht es UNIX-Servern ihre Dateisysteme gemeinsam zu nutzen. s. NIS

**NIS** [Network Information System] Ein von der Firma Sun entwickelter Dienst, der es verschiedenen Servern in einem Netzwerk ermöglicht, die Benutzeraccounts bzw. andere Ressourcen gemeinsam zu verwalten. Ursprünglich wurde das System unter dem Namen yp (yellow pages) bekannt. Da die Firma British Telecom ein Copyright auf den Namen yp besitzt, musste Sun den Namen geändert.

**UML** [Unified Modeling Language] Standardisierte grafische Modellierungssprache. Geht insbesondere auf die Arbeiten von Grady Booch, James Rumbaugh und Ivar Jacobson („die drei Amigos“) zurück. Mittlerweile gibt es zahlreiche Programme die aus UML-Diagrammen Quellcode für verschiedene Programmiersprachen generieren können.

Besonderen Dank möchte ich meinem Vater Paul Reinsch aussprechen, der mir das Equipment für die Videoaufzeichnungen zur Verfügung gestellt und mir die zeitaufwendige Arbeit des Kopieren der Videobänder auf die Festplatte abgenommen hat.

Der Firma Hansen&Gieraths, Bonn, und Herrn Prof. Dr. J. Magenheim von der Universität Paderborn möchte ich danken, die mir die beiden Linux-Server für das Projekt zur Verfügung gestellt haben.

Zum Schluss möchte ich bei der National Aeronautics & Space Administration (NASA) für ihren Hinweis bedanken, wie man unter  $\LaTeX$  eine Tilde darstellt.

Ich versichere, dass ich die schriftliche Hausarbeit einschließlich evtl. beigefügter Zeichnungen, Kartenskizzen und Darstellungen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, habe ich in jedem Falle unter genauer Angabe der Quelle deutlich als Entlehnung kenntlich gemacht.

Bonn, den 22. September 2003

*Ceterum censeo carthaginem esse delendam.*