

Erprobung der

Portfolioarbeit

zur Förderung des Lösens von Problemen durch Algorithmen
unter Verwendung erweiterter objektorientierter Konzepte

Ein Unterrichtsvorhaben im
Basiskurs Informatik
der Humboldt-Oberschule

Schriftliche Prüfungsarbeit für das Amt des Studienrats

vorgelegt von
Lars-Christian Pelz
Studienreferendar

1. Schulpraktisches Seminar
im Bezirk Reinickendorf (S)
(Berlin, 19.9.2008)

Hauptseminarleiter: Herr Kraft
Fachseminarleiterin: Frau Thalemann

Inhalt

1) Einleitung.....	3
2) Planungsvoraussetzungen.....	4
2.1) Lerngruppenanalyse.....	4
2.2) Lernausgangslage.....	6
3) Konzeption der Unterrichtseinheit.....	9
3.1) Förderung der Problemlösefähigkeit.....	9
3.2) Portfolioarbeit und Problemlösen im Informatikunterricht.....	12
3.3) Messbarkeit des Lernzuwachses.....	14
3.4) Erkenntnisinteresse.....	16
3.5) Lehrer- und Schülerrolle.....	17
3.6) Technische Voraussetzungen.....	18
4) Praktischer Teil.....	19
4.1) Sachanalyse.....	19
4.2) Planung der Sequenz.....	21
4.3) Synopse.....	23
4.4) Durchführung der Sequenz.....	25
4.4.1) Beratung und Interviews.....	25
4.4.2) Portfolios und Präsentationen.....	27
4.4.3) Bewertung.....	34
4.5) Analyse und Alternativen.....	38
4.6) Überprüfung der Hypothesen.....	41
4.6.1) Binnendifferenzierung.....	41
4.6.2) Betreuung.....	42
4.6.3) Lernzuwachs und Organisationsaufwand.....	43
5) Gesamtreflexion und Ausblick.....	44
6) Literaturverzeichnis.....	47
7) Anhang.....	48

1) Einleitung

„Computer lösen Probleme, die wir ohne sie nicht hätten.“

IT-Spruchwort (unbekannter Verfasser)

In der IT-Branche gilt die Fähigkeit, Probleme schnell und effizient zu lösen, als eines der höchsten Güter. Leider führt das obige Zitat in die Irre, denn es sind nicht die Computer, die informationstechnische Probleme lösen, sondern Menschen. Computer führen die von Menschen erdachten Lösungen aus, doch nur der Mensch verfügt über das kognitive Potential, Probleme zu strukturieren und eine Lösung zu planen. Trotzdem ist problemorientierter Informatikunterricht aus vielen Gründen noch nicht alltäglich.

Die in dieser Arbeit vorgestellte Unterrichtssequenz soll die angesprochene Problemlösefähigkeit bei meinen Schülern¹ trainieren. Da die kognitiven Strategien zum Lösen von Problemen von Schüler zu Schüler unterschiedlich sind, möchte ich ihnen die Wahl geeigneter Problemstellungen in gewissem Rahmen freistellen. Um die Anstrengungen der Schüler beobachten und analysieren zu können, werden sie dazu angehalten, ihre Ideen und Fortschritte in einem Portfolio zu dokumentieren, welches sie mithilfe einer Online-Plattform erstellen und pflegen werden.

Da die Leistungsniveaus der einzelnen Schüler stark unterschiedlich sind, interessiert mich vor allem, ob mit der Portfolioarbeit auch eine starke Binnendifferenzierung zu erreichen ist. Darüber hinaus ist es wichtig, festzustellen, ob und inwieweit eine solche Sequenz mit offenem Unterricht in der Sekundarstufe II einsetzbar und durchführbar ist. Ich werde daher untersuchen, ob es mir gelingt, die Schüler innerhalb ihres jeweiligen Themas optimal zu betreuen. Interessant ist auch, ob und inwiefern alle Schüler in der Phase des offenen Unterrichts den gewünschten Lernzuwachs durch selbstorganisiertes Lernen an einem zu Beginn der Reihe nicht feststehenden Thema erreichen.

Eine weitere Schwierigkeit des Informatikunterrichts besteht in der Notwendigkeit, theoretisch behandelte Planungsmethoden praktisch anzuwenden. Generell dominiert das Implementieren von Lösungen das Unterrichtsgeschehen. Die Durchführung eines kompletten Planungsprozesses für ein wenige Zeilen umfassendes Programm wirkt allerdings trivial und ist daher nicht motivierend. In dieser Unterrichtssequenz sollen meine Schüler deshalb Gelegenheit erhalten, die gelernte Theorie in größerem Maßstab anzuwenden und dabei informationstechnische Probleme lösen, indem sie umfangreiche objektorientierte Programme verfassen.

¹ Ich verwende in diesem Dokument zugunsten des Leseflusses nur die männliche Form bei der Bezeichnung von Personengruppen (z.B. Lehrer, Schüler).

2) Planungsvoraussetzungen

2.1) Lerngruppenanalyse

Am Basiskurs Informatik nehmen acht Schüler und zwei Schülerinnen teil. Ich führe in diesem Kurs seit Beginn meines Referendariats eigenverantwortlichen Unterricht durch. Anfangs bestand der Kurs aus neun Schülern und einer Schülerin. Ein Schüler wurde zum Halbjahr aufgrund seines geringen Alters (er begann die elfte Klasse an der Humboldt-Oberschule mit zwölf Jahren) in eine andere Schule versetzt, die außerdem näher an seinem Wohnort lag. Ein anderer Schüler verließ den Kurs, da er seine Neigungen in anderen Fächern sah. Liane trat dem Kurs im Dezember 2007 bei, nachdem sie von einem Frankreich-Austausch zurückkehrte. Philipp begann den Kurs ebenfalls nach einem Frankreich-Austausch im Februar 2008.

Das Arbeitsklima im Kurs ist sehr angenehm; es herrscht Kooperation zwischen dem Lehrer und den Schülern im Rahmen der zu bearbeitenden Themen. Bei fast allen Schülern ist das Interesse am Fach groß genug, um auch über den Unterricht hinaus Beachtung zu finden. Es herrscht also eine intrinsische Motivation vor, sich mit dem Unterrichtsstoff zu beschäftigen. Einige Schüler bringen Ideen und Probleme mit in den Unterricht, die sie im Umgang mit dem eigenen Rechner erleben. Sie diskutieren spontan und konstruktiv sowohl über im Unterricht entstandene Fragen, als auch vom Unterrichtsthema abweichende Themen.

In der Leistungsverteilung der einzelnen Schüler steckt das eigentliche unterrichtstechnische Problem, welches auch zur Wahl des Themas dieser Arbeit geführt hat: In der objektorientierten Programmierung sind drei Schüler bereits weit über die Anforderungen des Kurses hinaus kompetent, während zum Beginn der darzustellenden Sequenz zwei andere keinerlei Erfahrungen in dieser Hinsicht haben.

Stellvertretend für die Gruppe der weit fortgeschrittenen Schüler kann Torsten bereits eine beachtliche Sammlung von eigenen Programmen vorweisen und kennt sich mit Vererbung, Klassenhierarchien und dem abstrakten Datentyp *Liste* aus. Am Einsatz des Letzteren in seinen Programmen kann ich erkennen, dass er die Begrifflichkeiten der Objektreferenz und der Objektdatentypen vollständig durchdrungen hat und in Implementierungen eigener Problemlösungen einsetzen kann.

Philipp und Liane hingegen haben Teile des ersten Halbjahrs als Austauschschüler in Frankreich verbracht und kennen sich somit weder mit der imperativen Programmierung in Java noch mit den Grundbegriffen der Objektorientierung aus. Philipp hat schon imperativ in PHP programmiert, steht jedoch auch dort am Anfang. Nachholen ließ sich der Stoff aufgrund des

Umfangs offenbar nicht.

Als vierter exemplarischer Schüler sei hier Peter erwähnt, der für das „Mittelfeld“ der Lerngruppe steht. Er hat – wie die anderen auch – das erste Halbjahr über gut mitgearbeitet und kann somit die Planungsinstrumente *Spezifikation* (vereinfachtes UML-Klassendiagramm ohne Beziehungen) und *Struktogramm* für die Planung eigener Problemlösungen einsetzen. Darüber hinaus beherrscht er die grundlegenden Java-Sprachkonstrukte und kann somit seine geplanten Lösungen innerhalb eines vom Lehrer geschaffenen Rahmens in Java implementieren.

Am Vergleich dieser drei Schülergruppen lässt sich der wichtigste Grund für die Wahl meines Themas ableiten: die Leistungsniveaus der Schüler unterscheiden sich sehr stark voneinander. Einerseits versuche ich daher, mit meiner Unterrichtsreihe eine möglichst starke Binnendifferenzierung zu erreichen, um die einzelnen Schüler ihren Voraussetzungen entsprechend optimal zu fordern und zu fördern. Andererseits möchte ich sie befähigen, aus eigenem Antrieb heraus Probleme zu strukturieren, Lösungsansätze für Teilprobleme zu entwickeln und Lösungen eigenständig zu planen und umzusetzen. Die Portfolioarbeit soll hierbei als Mittel der Strukturierung und Reflexion des eigenen Lern- und Arbeitsprozesses dienen. Nach der Darstellung der Fähigkeiten einzelner Schüler möchte ich nun auf das Sozialverhalten der Lerngruppe eingehen.

Disziplinverstöße kommen im Kurs in der Regel selten vor. Die Gruppe ist ausgeglichen und eher ruhig. Ich kann keine offen ausgetragenen oder versteckten Antipathien wahrnehmen.

Auf Zusammenarbeit bzw. gegenseitige Hilfe lassen sich die Schüler trotzdem nur ungern ein. Das liegt nach meiner Einschätzung zum einen an der mangelnden Geduld, jemandem etwas zu erklären, was man selbst mit wenigen Tastendrücken fertig stellen könnte. Da heißt es dann: „Ich tipp' dir das mal kurz,“ oder „lass mich 'mal.“ Da der Lernfortschritt desjenigen, der die Hilfe angefordert hat, in diesem Fall vernachlässigbar gering ist, habe ich in der Vergangenheit solche Hilfsangebote mit dem Hinweis unterdrückt, jeder Schüler möge nur seine Tastatur benutzen. Zum anderen hat die Beschäftigung mit dem eigenen Rechner für die Schüler einen viel höheren Stellenwert als die Hilfestellung bei anderen Schülern.

Aus diesem Grund habe ich die Bildung von Arbeitsgruppen zugelassen (s. Abschnitt 4.2), die gemeinsam eine Problemstellung bearbeiten. Im Kontext einer gemeinsamen Bearbeitung kann ein Programm nur dann vollständig fertig gestellt werden, wenn alle Gruppenmitglieder sich abstimmen und unterschiedliche Teile des Programms bearbeiten. Natürlich gilt es auszu-

schließen, dass ein Mitglied die gesamte Implementierung allein ausführt. Da alle Schüler ein eigenes Portfolio erstellen müssen, sollte ich durch die Betrachtung der Portfolios den Anteil der jeweiligen Schüler am gemeinsamen Produkt erkennen können.

2.2) Lernausgangslage

In diesem Abschnitt stelle ich die Lernvoraussetzungen dar, welche die Schüler in die durchzuführende Unterrichtssequenz einbringen. Es handelt sich hierbei um die Kompetenzen, die sie in den vorhergehenden Unterrichtssequenzen erwerben konnten und erworben haben.

Die folgenden grundlegenden Fähigkeiten im Umgang mit einem Rechnersystem brachten die Schüler bereits zu Beginn des Basiskurses mit:

- Bedienung der Eingabegeräte Tastatur und Maus
- Benutzung verschiedener grafischer Benutzerschnittstellen bzw. -oberflächen (im Unterricht *Gnome*, daheim evtl. andere Linux-Oberflächen oder Microsoft Windows)
- Umgang mit Standardsoftware: Internet- und Dateisystem-Browser, Textverarbeitung, Grafikprogramme

Die zur Erstellung und Pflege von Inhalten auf Webseiten nötigen Kenntnisse der Seitenbeschreibungssprache HTML erwarben die Schüler im sog. HTML-Kurs, ein epochal durchgeführter Zusatzunterricht in der achten Klasse der Humboldt-Schule. Diese Kompetenz ist von entscheidender Bedeutung, um den Schülern die Erstellung des Portfolios mithilfe des Internet zu ermöglichen. Ich gehe darauf in Abschnitt 4.2 näher ein.

Im meinem Basiskurs erwarben die Schüler die neue Fähigkeit, Gegenstände der realen Welt als abstrakte informationstechnische Darstellung in einem Computersystem zu modellieren. Diese Fähigkeit geht einher mit dem Verständnis der Trennung von Information und Repräsentation, was wiederum eine Schlüsselkompetenz in der Informatik darstellt (vgl. Schubert/Schwill 2004 S. 107). Für die Schüler ist es essentiell, begriffen zu haben, dass Objekte der realen Welt durch Rechenmodelle dargestellt werden können. Das bedeutet nicht, dass das Modell hundertprozentig exakt sein muss, sondern es können – je nach der Komplexität der Anforderungen – Einschränkungen vorgenommen werden, die das aus dem Modell entstehende Abbild des realen Gegenstands prägen.

Eine andere Schlüsselkompetenz der Informatik ist die Fähigkeit, Abläufe in einzelne Operationen zu zerlegen und diese in der richtigen Reihenfolge auszuschreiben (vgl. Schubert/Schwill 2004 S. 51). Hierbei handelt es sich um die Beschreibung von Abläufen

durch ein mathematisches Modell, den Algorithmus. Die Schüler haben verinnerlicht, dass beliebige Sequenzen von atomaren Schritten (auch „Befehle“ oder „Anweisungen“ genannt), zur Veränderung der Information in einem Modell herangezogen werden können. Selbstverständlich gehen sie damit noch intuitiver um, als es die im vorangegangenen Satz gegebene Beschreibung ausdrücken kann. Sie orientieren sich noch vielfach an der Repräsentation (z.B. die Veränderung einer Grafik auf dem Bildschirm durch ein Programm, das sie geschrieben und ausgeführt haben), um Rückschlüsse auf die im Modell aktuell enthaltene Information zu ziehen. Dennoch ist ihnen klar, dass sie, um eine Veränderung der Darstellung zu erreichen, die Information in ihrem Modell durch einen Algorithmus verändern müssen.

Neben diesen wichtigsten Grundkenntnissen stehen den Schülern eine Vielzahl von weiteren Kompetenzen zur Verfügung, die es ihnen ermöglichen, die Modellierung von Gegenständen und Abläufen mit computergestützten Modellen vorzunehmen:

- Aufzeichnen der theoretischen Modelle mit vereinfachten UML-Klassendiagrammen (Gegenstände, Objekte) und Nassi-Shneiderman-Diagrammen (auch *Struktogramme* genannt, Darstellung von Abläufen)
- Implementieren der Modelle als Programm in der Programmiersprache Java

Die Schüler sind darüber hinaus vertraut mit dem Entwurfsvorgang eines Programms, der im folgenden Schema dargestellt ist:

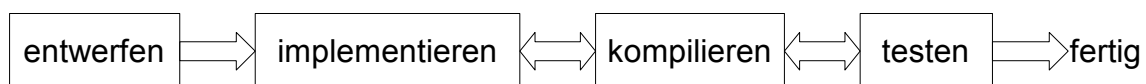


Abbildung 1: Schematischer Ablauf des Kurzentwurfs in der Softwareentwicklung

Bisher jedoch galt das Hauptaugenmerk der Implementierung. Die Schüler sollten die grundlegenden Strukturen der Programmiersprache Java kennen lernen und im Rahmen vorstrukturierter Aufgaben einsetzen. Ich möchte sie deshalb in dieser Unterrichtseinheit zum Lösen von komplexen Problemen anleiten, damit sie lernen, die bereits erworbenen Kompetenzen zu kombinieren und koordiniert einzusetzen, um umfangreichere Programme als bisher zu entwerfen.

Um einzuschätzen, in welchem Maße die Schüler bereits in der Lage sind, umgangssprachlich dargestellte Probleme mithilfe der Planungsmethoden *Struktogramm* und *Klassendiagramm* zu strukturieren und ggf. zu lösen, stellte ich der Unterrichtssequenz einen Test voran. Dieser Test (s. Anhang S. 49) enthielt sowohl eine Aufgabe, bei der aus einem gegebenen Problem ein Klassendiagramm zu erstellen war, als auch eine Aufgabe, die eine Implementierung einer

gegebenen Spezifikation (UML-Klassendiagramm mit einer Klasse) in Java erforderte.

Die Auswertung des Tests spiegelte meine Einschätzung der Lerngruppe wider: sechs der zehn Schüler lösten die Implementierungsaufgabe korrekt, zwei mit Einschränkungen und zwei mit groben Fehlern. Nur zwei Schüler (Raphael und Torsten) vermochten die Strukturierungsaufgabe in meinem Sinne zu lösen (Ableitung dreier Klassen aus dem Text, Zuordnung von Attributen und Methoden), ein Schüler löste die Aufgabe nur teilweise, vier weitere entnahmen dem Text zumindest Namen von Objekten, die sie für geeignete Klassenkandidaten hielten und zwei Schüler bearbeiteten die Aufgabe nicht. Dies zeigte, dass die Fähigkeit der Schüler meiner Lerngruppe zur Zerlegung der Probleme in ihre wesentlichen Bestandteile und somit der Planung einer Lösung der Förderung bedurfte.

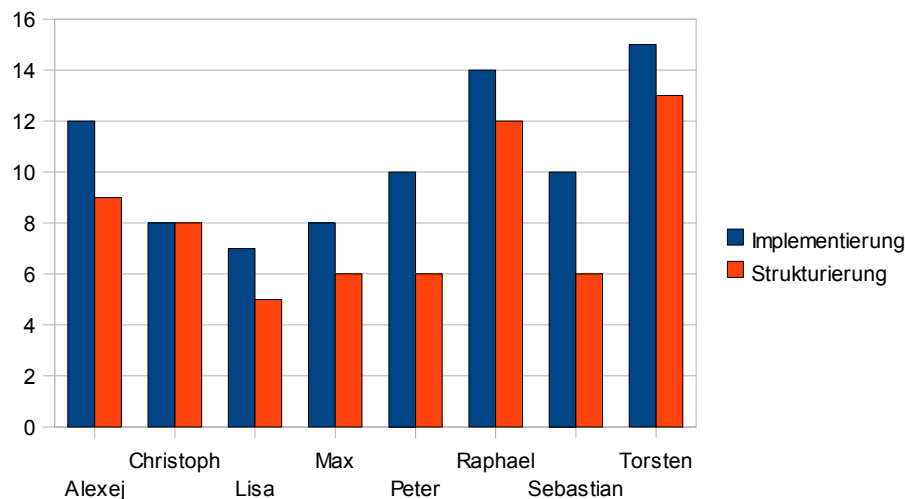


Abbildung 2: Ergebnisse des Tests zur Erhebung des Lernstandes

Im obigen Diagramm sind die Ergebnisse des Tests grafisch dargestellt. Die Ergebnisse von Philipp und Liane ließ ich außen vor, da beide wie erwähnt keine Vorkenntnisse in der Programmiersprache Java hatten und auch die bereits behandelten Planungsmethoden nicht konnten.

3) Konzeption der Unterrichtseinheit

3.1) Förderung der Problemlösefähigkeit

Humbert sieht in einem Problem „eine nicht routinemäßig lösbare Aufgabe“ (vgl. Humbert 2005, S. 36), eben eine solche, zu welcher zwar Ausgangs- und gewünschte Zielsituation bekannt sind, aber der Lösungsweg erarbeitet werden muss. Zur Lösungsfindung existieren – je nach Schwierigkeitsgrad und Anforderungen – verschiedene Methoden: heuristische, strukturierende, hierarchisierende, etc. (vgl. Schubert/Schwill 2004, S. 50 und Humbert 2005, S. 37) Grundsätzlich sollte jedem Lösungsversuch eine intensive Planung vorausgehen.

Das Lösen von Problemen durch Algorithmen hat einen hohen Stellenwert in der Informationstechnologie. Offensichtlich herrschen dabei umgangssprachlich gegebene und damit schwach spezifizierte Probleme vor, beispielsweise:

- „Erstellen Sie eine grafische Oberfläche, damit unsere Kunden Online-Banking nutzen können.“
- „Unser Rennspiel braucht ein besseres Fahrzeug-Handling. Arbeiten Sie mit den Spieltestern zusammen, um es zu verbessern.“

Um den Anforderungen an Studium und Beruf gewachsen zu sein, müssen die Schüler Strategien entwickeln, um die hinter solchen Problemen stehenden informationstechnischen Strukturen zu erkennen und somit Lösungsansätze entwickeln zu können. Diese Strategien können für jeden Schüler unterschiedlich sein, denn jeder hat für sich genommen anderes Wissen erlangt und andere Erfahrungen gemacht. In der Informatik ist diese Vielfalt auch durchaus gewollt, wenn die erarbeiteten Lösungen der Problemstellung gerecht werden.

In der Schule sollte – anders als im Berufsleben – die Möglichkeit bestehen, sich auszuprobieren und sich eine Problemlösestrategie anzueignen, die man selbst als wirksam und als der eigenen Denkweise angemessen empfindet. Der Informatikunterricht sollte hierfür mögliche Vorgehensweisen vorstellen und die nötigen Hilfsmittel anbieten. Aus der Individualität des Problemlöseprozesses leite ich die Notwendigkeit zur individuellen Betreuung und damit zur Binnendifferenzierung im Unterricht ab (vgl. Humbert 2005, S. 75 f.).

Anhaltspunkt für alle Problemlösestrategien, deren Produkt eine Implementierung ist, sollte im Informatikunterricht jedoch der Prozess der Softwareentwicklung sein, welcher in der nebenstehenden Grafik verdeutlicht wird.

- Problemanalyse: Was ist gefordert? Welche Aufgabe soll erfüllt bzw. gelöst werden? Welche Hilfsmittel, Geräte oder Software-Werkzeuge werden benötigt?
- Wissenserwerb (Recherche): Welche Prinzipien und Technologien können verwendet werden? Existiert bereits eine Lösung für das Problem?

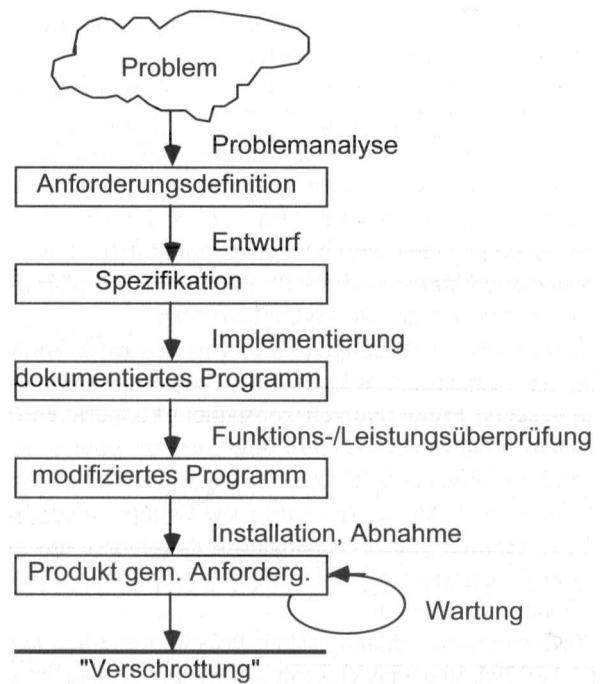


Abbildung 3: Software Life Cycle (aus Schwill/Schubert 2004 S. 302)

- Strukturanalyse: Welche Gliederung hat das Problem? Wie hängen die einzelnen Teile voneinander ab? In welcher Reihenfolge müssen sie bearbeitet werden?
- Implementierung der einzelnen Teile, Zusammenführung
- Test, in Abhängigkeit vom Ergebnis Rücksprung zu jeweiligem Schritt in der Planung

Schwill und Schubert (vgl. Schwill/Schubert 2004 S. 35) nennen die für die jeweiligen Schritte im informationstechnischen Problemlösungsprozess gebräuchlichen Methoden. Am bedeutendsten sind hierbei die *Modellierung von Prozessen (Abläufen)* und die *strukturierte Zerlegung* eines Problems in Teilprobleme. Zu letzterer gehören die in der Informatik zentralen Prinzipien *Modularisierung, Hierarchisierung* und *Orthogonalisierung* („Fundamentale Idee der Informatik“, vgl. Schwill/Schubert 2004 S. 86).

Über die Bedeutung im Informatikunterricht hinaus hat das Lösen von Problemen auch einen hohen Stellenwert in anderen Fächern. Im Wesentlichen sind dies die mathematisch-naturwissenschaftlichen Fächer, aber auch in den sprach- und geisteswissenschaftlichen Fächern können zu lösende Probleme als Unterrichtsinhalt aufgegriffen werden. Deshalb ist die Förderung der Problemlösefähigkeit nicht nur ein Selbstzweck des Informatikunterrichts, sondern sie dient der Allgemeinbildung der Schüler. Dass im Informatikunterricht auf die Problemlösefähigkeit besonderen Wert gelegt wird, liegt an der anderen Sicht auf die Repräsentation von Wissen einerseits und das Problemlösen als einen davon abhängigen Inferenzmechanismus

andererseits (vgl. Schwill/Schubert S. 53 f.).

Folgerichtig ist das Problemlösen im Informatikunterricht im Berliner Rahmenplan der Sekundarstufe II verankert. Hier ist im Themenfeld „L2 – Grundlagen der Programmierung“ das „selbstständige Lösen von Problemen in Einzel- oder Gruppenarbeit am Computer“ als obligatorischer Kompetenzerwerb festgelegt (vgl. Berliner RLP Informatik Sek II, S. VIII). Da der Schwerpunkt dieser Unterrichtseinheit auf dem Themenfeld „L2“ liegt, wird selbstverständlich auch der Erwerb aller zusätzlich dort aufgeführten Kompetenzen gefördert.

Des Weiteren sind die im Abitur zu erreichenden Standards für die Informatik-Leistungs- und Grundkurs ebenfalls richtungsweisend für diese Unterrichtseinheit. Ich trainiere meine Schüler darin, informationstechnische Modelle zu erstellen und zu nutzen (vgl. Berliner RLP Informatik Sek. II, S. 14) und Information in Form von Daten darzustellen und zu verarbeiten (vgl. Berliner RLP Informatik Sek II, S. 15). Hier findet sich auch erneut die Maßgabe, dass die Schüler informationstechnische Probleme rechnergestützt lösen (vgl. Berliner RLP Informatik Sek. II, S. 16). Die Standards im Rahmenlehrplan für die Sekundarstufe II orientieren sich direkt an den *Einheitlichen Prüfungsanforderungen in der Abiturprüfung* (EPA) Informatik (Beschluss der Kultusministerkonferenz vom 01.12.1989 i.d.F. vom 05.02.2004, vgl. Berliner RLP Informatik Sek. II, S. 33).

Die Schüler meiner Lerngruppe haben sich in meinem Unterricht bisher die Grundlagen zur Lösung von Problemen durch Algorithmen angeeignet. Es wurden einerseits mehrere Strukturierungsmodelle behandelt (vereinfachte Klassendiagramme in UML und Struktogramme), andererseits verfügen die Schüler über grundsätzliche Kenntnisse im Umgang mit der Programmiersprache Java. Beides zusammengenommen sollte die Schüler befähigen, für beliebige informationstechnische Problemstellungen algorithmische Lösungen zu erstellen.

Im Unterricht hat sich jedoch auch gezeigt, dass die Modellierung eines Problems und dessen Lösung sowie die Umsetzung der Modellierung als Programm nicht trivial sind. Die Modellierung der Problemlösung als Klassengerüst in der objektorientierten Programmierung ist dabei der schwierigste Schritt im Lösungsprozess. Ein guter Entwurf des Klassengerüsts entscheidet meist schon über Erfolg oder Misserfolg der Implementierung. Ist die Klassenstruktur zu umfangreich, erschwert dies die Fertigstellung des Programms in der verfügbaren Zeit. Ist sie zu flach, verhindert dies die Umsetzung komplexer Abläufe oder lässt den Quelltext unübersichtlich werden.

Dennoch können gute Entwürfe nur mit viel Erfahrung erstellt werden, da es (bislang) keine

Algorithmen gibt, die aus einer umgangssprachlich formulierten Problemstellung eine Klassenstruktur erstellen. Ich möchte deshalb meine Schüler in dieser Unterrichtssequenz das erste Mal zu einem vollständig eigenen Entwurf anleiten. Dabei spielt genau die Förderung der Problemlösefähigkeit eine zentrale Rolle.

3.2) Portfolioarbeit und Problemlösen im Informatikunterricht

Ein Portfolio im allgemeinen Sinne ist eine Sammlung von Dokumenten, die nach bestimmten Kriterien ausgesucht wurden. Eine solche Sammlung hat die Aufgabe, besondere Fähigkeiten oder Eigenschaften des Erstellers hervorzuheben.

Portfolios findet man häufig in der Wirtschaft. Dort geben etwa Aktienportfolios und Produktportfolios Auskunft über die Fähigkeiten und Ausstattung von Banken und Firmen. In der Bildung werden Portfolios seit den 80er Jahren in den USA an Highschools und Universitäten eingesetzt, um den Studenten Möglichkeiten der Selbstkritik zu geben und gleichzeitig eine Arbeitsmappe zu haben, die Grundlage für die abschließende Bewertung sein kann. Ein Portfolio erfüllt damit sowohl die Funktion eines Bewertungsinstruments als auch die eines Lehr-Lern-Instruments (vgl. Kolb 2007 S. 25, Häcker 2006).

In Deutschland werden Portfolios häufig in der Grundschule eingesetzt, um den Lernfortschritt der Schüler zu dokumentieren und zu bewerten und die Schüler zur Reflexion ihres Lernprozesses anzuregen (vgl. Kolb 2007, S. 21 f.). Deshalb existiert kaum Literatur über die Verwendung von Portfolios in der Sekundarstufe II, weshalb die folgenden Ausführungen hauptsächlich Erklärungen und Erfahrungsberichte aus Primarstufe und Sekundarstufe I referenzieren. Humbert geht in seinem Werk nur kurz auf den Einsatz eines Portfolios als Beurteilungsinstrument ein (vgl. Humbert 2005, S. 146). In den anderen mir vorliegenden Informatik-Didaktiken erfährt die Portfolioarbeit keine Erwähnung. Ich entnehme daher den Werken Häcker 2006 und Kolb 2007 die meiner Meinung nach wichtigen Informationen und baue darauf meine Unterrichtsplanung auf.

Man unterscheidet mehrere Arten von Portfolios im Hinblick auf deren Verwendung in der Schulbildung: z.B. Bewerbungsportfolio, Entwicklungsportfolio, Assessmentportfolio, etc. (vgl. Häcker 2006). Alle genannten Arten können zwei großen Kategorien zugeordnet werden: den produktorientierten und prozessorientierten Portfolios (vgl. Kolb 2007, S. 22).

Ein produktorientiertes Portfolio ist eine Sammlung von Schülerprodukten, die als Ganzes betrachtet den aktuellen Lern- und Fähigkeitsstand eines Schülers dokumentieren sollen. Schülerprodukte können jegliche im Unterricht gefertigten Arbeiten sein. Die Auswahl der zu inte-

grierenden Arbeiten nimmt der Schüler unter bestimmten Qualitätskriterien vor.

Ein prozessorientiertes Portfolio dagegen soll die einzelnen Schritte zur Erstellung eines Produktes dokumentieren. Hierbei tritt das eigentliche Produkt in den Hintergrund. Die zur Erstellung des Produktes unternommenen Anstrengungen werden hingegen viel detaillierter dargestellt. Dies kann in Form von Übungskalendern, verschiedenen Editionen eines Textes, Skizzen, Schemata, Mindmaps, Notizen u.ä. geschehen (vgl. Häcker 2006). Im Vergleich zum produktorientierten Portfolio erfordert das prozessorientierte Portfolio ein höheres Maß an Strukturierung, um den Zusammenhang zwischen den einzelnen Inhalten herzustellen. Während ein Produkt-Portfolio aus gleichberechtigt nebeneinander stehenden Exponaten bestehen kann (vgl. „Ausstellung“), erfordert somit die Darstellung eines Prozesses durch einzelne Arbeitsergebnisse nach meinem Verständnis der Quellen (vgl. Kolb 2007, S. 117) zumindest eine chronologische Ordnung innerhalb des jeweiligen Prozesses. Da ich in dieser Unterrichtseinheit Wert auf Problemlösungen und die damit verbundenen Prozesse lege, werde ich meine Schüler dazu anhalten, ein prozessorientiertes Portfolio zu erstellen.

Im Folgenden erläutere ich, wie Schüler dazu angeleitet werden sollten, ihre Produkte in einem prozessorientierten Portfolio zu organisieren und selbst Kriterien für deren Bewertung zu finden.

Portfolioarbeit wird in der Literatur durchgehend mit der Diversität des Lernprozesses in Verbindung gebracht (vgl. Kolb 2007 S. 27, Häcker 2005). Das schließt auch die Offenheit bezüglich der in ein Portfolio aufgenommenen Inhalte ein, was wiederum das Problem der Vergleichbarkeit der entstandenen Ergebnisse aufwirft. Ohne diese Vergleichbarkeit kann aber keine Selbstkritik entstehen. Schließlich ist ein Produkt, welches nicht mit anderen verglichen werden kann, das beste in der durch es selbst geschaffenen Kategorie und muss somit auch nicht mehr verbessert werden. Doch gerade diese ständige kritische Betrachtung der eigenen Leistungen soll erreicht werden. Deshalb werden Kriterien benötigt, welche an alle Portfolios eine einheitliche Messlatte anlegen.

Da eine Sammlung eigener Arbeitsergebnisse etwas sehr Persönliches ist, sollten die Schüler unbedingt in die Aufstellung der Qualitätskriterien mit einbezogen werden. Wenn sie gemeinsam die wichtigen Normen festlegen, können sie sich später besser mit diesen identifizieren und sie leichter akzeptieren. Aus den in Häcker gefundenen Ausführungen zum Thema Bewertung von Portfolios (vgl. Häcker 2006) leite ich folgende wichtige Grundkriterien ab, auf deren Einhaltung die Lehrperson achten sollte:

- Die Normen dürfen nicht zu hoch oder zu niedrig angesetzt werden, sondern müssen die Schüler entsprechend ihres Lernstandes fordern.
- Der Normenkatalog sollte nicht so umfangreich werden, dass er unübersichtlich wird.
- Die Schülerarbeiten sollten ein breites Spektrum an Inhalt abdecken, damit keine einseitige Darstellung entsteht. Dies ist auch wichtig, um deutlich erkennen zu können, welchen Lernzuwachs der Schüler in den jeweils im Unterricht behandelten Bereichen erlangt hat.
- Es darf keine Diskriminierungen geben (z.B. pauschale Abwertung bestimmter Schülergruppen aufgrund der Differenzen im Leistungsstand).

Die Organisation verschiedener Materialien in einem Verbund ist – gerade für Anfänger – so kompliziert, dass eine Anleitung zur Strukturierung des Inhalts zwingend erforderlich ist. Hier eignet sich die gemeinsame Erarbeitung möglicher, aber dennoch verbindlicher Anordnungen, um Willkür zu verhindern. Dabei halte ich folgende Punkte für beachtenswert:

- Mögliche Sortierkriterien sind die chronologische Reihenfolge und die inhaltliche Ähnlichkeit von Arbeitsergebnissen (oder eine Kombination aus beiden).
- Die einzelnen Inhaltsbereiche müssen klar voneinander getrennt sein (z.B. durch Deckblätter oder in einer Fächermappe).
- Ein Inhaltsverzeichnis ist äußerst hilfreich.

Unabhängig davon, ob prozess- oder produktorientierte Portfolios erstellt werden sollen, ist eine kontinuierliche Betreuung der Erstellung notwendig (vgl. Kolb 2007 S. 145 ff.). Hier geht es vor allem darum, die Schüler in ihrer Selbstkritik zu unterstützen. Innerhalb der Betreuung soll es mehrfache persönliche Beratungsgespräche geben. Diese Gespräche dienen dem Aufbau einer Beziehung zwischen Lehrer und Schüler, die über die konventionelle Rollenverteilung hinausgeht. Mehr dazu findet sich in Abschnitt 3.5.

3.3) Messbarkeit des Lernzuwachses

Die zentrale in diesem Unterrichtsvorhaben zu fördernde Kompetenz ist die bereits erläuterte Problemlösefähigkeit. Im Folgenden werde ich ausführen, mit welchen Mitteln diese Kernkompetenz im Informatikunterricht untersucht und der in der Unterrichtsreihe erreichte Lernfortschritt gemessen werden kann. Zunächst jedoch betrachte ich kurz die fachlichen Kompetenzen.

Auf der Ebene der fachlichen Kompetenzen ist ein Lernfortschritt genau dann erreicht, wenn die Schüler Strukturen und/oder Befehle in ihr Programm einbauen, die ich in meinem bisherigen Unterricht nicht behandelt habe. Um zu überprüfen, ob ein Lernzuwachs eingetreten ist, werde ich die fertigen Programme betrachten und nach entsprechenden Neuerungen untersuchen. Ich erwarte selbstverständlich die Verwendung *erweiterter* objektorientierter Konzepte. Worum es sich hierbei handelt, definiere ich in der Sachanalyse (Abschnitt 4.1). Des weiteren erwarte ich u.a. die Verwendung Java-spezifischer Programmierschnittstellen (APIs) und bisher nicht behandelte Kontrollstrukturen (z.B. *switch/case* zur Fallunterscheidung oder *try/catch* zur Fehlerbehandlung).

Die Messung der Problemlösefähigkeit gestaltet sich weitaus schwieriger. Wie bereits erwähnt ist die Lösung eines Problems durch einen Algorithmus ein Prozess, der sich durch eine schrittweise Annäherung an eine Lösung auszeichnet. Die Anzahl der Schritte und die daraus resultierende Zeit bis zur Fertigstellung der Lösung ist nicht intuitiv aus der Problemstellung erkennbar. Einen Lernzuwachs erreichen die Schüler, indem sie trotzdem lernen, den Umfang eines gegebenen Problems einzuschätzen und die richtigen Entwurfsmethoden auswählen. Zwei Beispiele sollen die erwartete Schätzleistung deutlich machen:

- „kleines“ algorithmisches Problem (auch „Programmieraufgabe“ im Informatikunterricht): „Schreiben Sie ein Programm, welches eine ungeordnete Menge von Zahlen aufsteigend sortiert.“
- „großes“ Problem (Softwareentwurf): „Schreiben Sie ein Programm, welches es Ihnen und ihren Mitschülern ermöglicht, ihre Hausaufgaben im Internet einzutragen und zu verwalten.“

Im ersten Fall können einige Schritte des vollständigen Softwareentwurfszyklus entfallen, im zweiten Fall kann nicht ohne umfangreiche Planung und Anforderungserhebung mit der Implementierung begonnen werden.

Das Problem der Messbarkeit der Problemlösefähigkeit als Ganzes besteht in ihrer Abhängigkeit von vielen unterschiedlichen Faktoren. Zwei wesentliche dieser Faktoren sind die Formulierung und das Verständnis der Problemstellung. Darüber hinaus spielt das persönliche Analyseverhalten von bekannten und unbekanntem Problemstellungen eine Rolle. Außerdem bedingt die individuelle Einstellung zum zielgerichteten Arbeiten natürlich den Fortschritt der einzelnen Schülerinnen.

Es ist also schwer, die Problemlösefähigkeit an sich durch eine objektive Größe darzustellen.

Sie ist vielmehr das Zusammenspiel verschiedener Prozesse, die von Schülerin zu Schülerin unterschiedlich sind. Genau hier sollten mir die Portfolios der Schüler Aufschluss über ihre persönlichen Strategien zur Problemlösung geben. Bei der Betrachtung der Portfolios untersuche ich daher, ob sich in den Ausführungen die in der Informatik vorkommenden Arbeitsphasen der Softwareentwicklung widerspiegeln. Da dies ein längerfristiger Prozess ist, erachte ich es als sinnvoll, den Lernzuwachs im Bereich der Planung und Implementierung an den Ergebnissen der Portfolioarbeit abzulesen und kurzfristige Prozesse (Algorithmisierung von kleinen Problemen, s.o.) mit einer Klausur zu testen. Eine detaillierte Ausführung zur Bewertung der Portfolios und zur Klausur findet sich im Kapitel 4.4.3.

3.4) Erkenntnisinteresse

In diesem Abschnitt stelle ich dar, welche Erkenntnisse ich aus der Durchführung der Unterrichtseinheit zu gewinnen versuche. Um zu überprüfen, ob die Schüler die von mir intendierten Ziele erreicht haben, stelle ich drei Hypothesen auf, deren Überprüfung im Abschnitt 4.6 Aufschluss über den Erfolg des Unterrichtsvorhabens liefern soll.

Hypothese 1 (Binnendifferenzierung): *Durch den Einsatz der Portfolioarbeit an sich kann im Informatikunterricht eine ausreichend starke und individuelle Förderung der Schüler erreicht werden.*

Mein größtes Interesse gilt der Frage, ob die Portfolioarbeit von sich aus eine Binnendifferenzierung ermöglicht. Damit hätte ich eine adäquate Methode gefunden, um der im Informatikunterricht allgegenwärtigen starken Leistungsdiskrepanz Herr zu werden.

Hypothese 2 (Betreuung): *Durch Monitoring und das Studium der Schülerarbeiten und -portfolios kann ich die Schüler über den gesamten Zeitraum der Unterrichtseinheit individuell betreuen.*

Die zweite Hypothese befasst sich mit der Realisierbarkeit der Betreuung. In der bisherigen Unterrichtsarbeit stellte sich heraus, dass in Phasen, in denen die Schüler selbstständig am Rechner arbeiteten, auch die Lehrperson stark gefordert wurde, wenn es Rückfragen oder technische Probleme gab. Dennoch wurde immer am gleichen Thema bzw. mit gleichen Operationsobjekten gearbeitet, währenddessen in der hier beschriebenen Einheit die Schüler das Thema frei wählen dürfen. Durch die starke Diversität erwarte ich einen großen Aufwand in der Einsichtnahme in die jeweiligen Schülerarbeiten, da ich die Änderungen und Erweiterungen in den Programmen der Schüler eventuell nicht verfolgen kann, sondern bei Fragen immer erst den Quelltext lesen muss. Erfolg auf dieser Ebene ist jedoch die Voraussetzung für

eine optimale Betreuung.

Hypothese 3 (Lernzuwachs und Organisationsaufwand): *Der zur Durchführung der Portfolioarbeit nötige Organisationsaufwand ist durch den Umfang des erreichten Lernzuwachses gerechtfertigt.*

Ich postuliere hier, dass der sowohl auf Schüler- als auch auf Lehrerseite erhebliche Organisationsaufwand der Erstellung und Betreuung der Portfolios einem ausreichend großen Lernzuwachs im Bereich der Problemlösefähigkeit gegenüber steht. Die Beantwortung dieser Frage kann entscheidend für eine Wiederverwendung der Portfolioarbeit in meinem Unterricht sein. Sollte sich herausstellen, dass zwar die Organisation und Durchführung der Portfolioarbeit gelingt, die Schüler aber dennoch nur das bereits Gelernte in ihren Programmen verwenden, lassen sich daraus wichtige Schlüsse zur Freistellung der Themenwahl und der Zielsetzung der Portfolios ziehen.

3.5) Lehrer- und Schülerrolle

In der Unterrichtsreihe soll eine starke Binnendifferenzierung durch die individuellen Programmierthemen erreicht werden. Individuelle Themen ziehen aber auch eine themenspezifische Betreuung mit sich. Dies hat auch eine Veränderung der Lehrer-/Schülerbeziehung zur Folge, die vom klassischen Produzenten-/Rezipienten-Modell des Frontalunterrichts oder allgemein des anweisenden Unterrichts abweicht.

Die Schüler sollen zu selbstständigem Lernen angeregt werden. Im Unterrichtsgeschehen müssen ihnen also größtmögliche Freiräume geschaffen werden, in denen sie ihre Ideen in die Praxis umsetzen können. Dazu gehört vor allem die Schaffung von genügend Lernzeit. Als Lernzeit wird in diesem Zusammenhang die Zeit bezeichnet, welche die Schüler mit der Arbeit an ihren Produkten und Portfolios verbringen. Am meisten Lernzeit haben die Schüler also, wenn sie selbstständig und zielorientiert arbeiten.

Eine selbstständige Arbeit ist nur dann möglich, wenn die Lehrperson ihre Steuerungsfunktion so selten wie möglich wahrnimmt. Gänzlich ungesteuertes Lernen führt allerdings zu Willkür und kann unter Umständen darin enden, dass die erwarteten Lernzuwächse nicht erreicht werden. Deshalb sehe ich meine Aufgabe darin, die Schüler in ihrem Lern- und Arbeitsprozess zu beraten. Diese Beraterfunktion übe ich in zweierlei Hinsicht aus.

Zunächst fällt mir die Herstellung einer lernförderlichen Umgebung zu. Neben einem angenehmen und respektvollen Umgang in der Lerngruppe gehört dazu die Bereitstellung der tech-

nischen Gegebenheiten, auf die ich in Abschnitt 3.6 näher eingehe.

Darüber hinaus kontrolliere ich den Ablauf der individuellen Lernzuwächse. Einerseits verschaffe ich mir während der Unterrichtszeit Einblicke in die Arbeiten der Schüler, andererseits sollen die Blogeinträge der Schüler mir helfen, die jeweiligen Prozesse nachzuvollziehen. Ich stehe den Schülern mit dieser Form der Beratung auch für die Beantwortung von Fragen zu spezifischen Problemen zur Verfügung. In solchen Fällen gebe ich natürlich nur Hinweise auf eventuelle Lösungsansätze und keine vollständigen Lösungen, da mir die selbstständige Erarbeitung der Lösung durch die Schüler wichtiger ist als die Fertigstellung des Produktes.

Das gegen Ende des Abschnitts 3.2 angesprochene Beratungsgespräch hat auf die Rollenverteilung im Unterricht weitere, tiefgreifende Folgen. Es dient dem Aufbau einer persönlichen Beziehung zwischen Lehrer und Schüler. Dies erfordert einerseits vom Lehrer die Würdigung und den Respekt vor den Leistungen des Schülers. Der Lehrer muss sich von seinen eigenen Vorstellungen lösen und versuchen, sich in die Sichtweise des Schülers hinein zu versetzen, um die Beratung auf der Begriffsebene des Schülers durchführen zu können. Es nützt nichts, wenn Entwicklungsvorschläge gemacht werden, die nicht zum aktuellen Stand des Schülers passen.

Der Schüler sollte sich entsprechend auf die Beratungsfunktion des Lehrers einlassen. Hierzu gehört das aktive Einfordern von Hilfsangeboten. In diesem Zusammenhang trainieren die Schüler das konstruktive Gespräch über ihre Arbeiten. Sie lernen, ein Problem, das sie nicht allein lösen können, möglichst eng einzugrenzen und präzise zu beschreiben, um schnell eine Antwort vom Lehrer zu bekommen (zur Notwendigkeit präziser Problembeschreibungen vgl. Schwill/Schubert 2004, S. 108). Hier zeigt sich auch, ob die Schüler Vertrauen in die vom Lehrer gegebenen Informationen haben oder ob sie sich eher auf andere Informationsquellen verlassen: Internet, Programmierbeispiele oder Unterrichtsmaterialien.

3.6) Technische Voraussetzungen

Die Erstellung des Portfolios erfolgt mithilfe meiner Internetseite. Dazu ist eine Software erforderlich, welche die Erstellung von Webseiten und die Einbindung von digitalen Medien für die Schüler erleichtert. Ich habe mich hier für das *Drupal Content Management System* (www.drupal.org) entschieden, da ich damit schon gute Erfahrungen im Hinblick auf Stabilität und Benutzerkomfort gemacht habe. Das System ist auf meinem privaten Internetbereich installiert und damit öffentlich zugänglich. Der Unterschied zu einem Wiki besteht im Umfang der zur Verfügung stehenden Module. Die Schüler haben nicht nur die Möglichkeit, Seiten zu

erstellen und zu verwalten, sondern verfügen zusätzlich über persönliche Blogs und Bildergalerien. Die Bedeutung der Blogs spreche ich auf Seite 22 (Abschnitt 4.2) an.

Des Weiteren müssen die Schüler Zugriff auf die zur Programmierung notwendigen Werkzeuge erhalten. Dabei handelt es sich einerseits um Editoren (*BlueJ*, *Eclipse*), welche das Erstellen und Testen von Programmen erleichtern, andererseits um Werkzeuge zur Kommunikation und zum Datenaustausch zwischen Mitgliedern einer Arbeitsgruppe. Zu letzterem Zweck habe ich ein Konto bei einem Internetdienst angefordert, der *Subversion*-Umgebungen kostenfrei zur Nutzung anbietet.

Subversion (SVN) ist ein Nachfolger der Kollaborationssoftware CVS (*Concurrent Versioning System*) und ermöglicht wie diese eine gemeinsame und vor allem ort- und zeitunabhängige Bearbeitung desselben Projekts durch mehrere Teammitglieder. Für meine Schüler ist die zur Benutzung des SVN-Kontos benötigte Software leicht verfügbar, da sie quelloffen (von eng. *open source*) entwickelt wird und daher im Internet kostenfrei verfügbar ist. Das SVN-Konto soll das Übertragen der Programmdateien zwischen dem Heimarbeitsplatz der Schüler und der Schule erleichtern und den Aufwand des Abgleichs verschiedener Bearbeitungszustände der Projektdateien minimieren.

4) Praktischer Teil

4.1) Sachanalyse

In dieser Unterrichtseinheit sollen die Schüler Programme erstellen, die *erweiterte objektorientierte* Konzepte beinhalten. Hierbei gehe ich davon aus, dass der Klassenbegriff die Grundlage aller Begrifflichkeiten der objektorientierten Programmierung ist. Eine Klasse wird durch ihren Namen, ihre Attribute und ihre Methoden definiert. Zur Laufzeit des Programms können beliebig viele Instanzen einer Klasse erstellt werden. Ihre Anzahl ist nur durch den verfügbaren Speicher begrenzt. Mit den Instanzen einer Klasse kann dann gearbeitet werden, d.h. sie dienen der Datenspeicherung und -manipulation im Sinne der Klassendefinition. Diese Konzepte stellen die Basis der objektorientierten Programmierung dar.

Erweiterte objektorientierte Konzepte sind nach meiner Auffassung die folgenden:

- **Schnittstellen:** Klassen können die Implementierung einer oder mehrerer Schnittstellen sein. Auf diese Weise wird die Definition generischer Methoden ermöglicht.
- **Vererbung:** Klassen können von anderen Klassen abgeleitet werden, um Klassenhierarchien zu erzeugen. Dies dient der Abstraktion und dem Verbergen von Implementie-

rungen (eng. *implementation hiding*). Beides wird benötigt, um ein Programm modular aufbauen zu können und damit Übersichtlichkeit, Stabilität und Austauschbarkeit einzelner Programmteile zu ermöglichen.

- **Abstrakte Datentypen:** Eigentlich ein Konzept aus der funktionalen Programmierung, so finden Abstrakte Datentypen auch in der objektorientierten Programmierung Verwendung. Im Allgemeinen handelt es sich hierbei um immer wiederkehrende Entwurfsmuster von Datenstrukturen, die eine definierte Menge von Operationen zulassen und denen strikte Gültigkeitsgrenzen (eng. *constraints*) gesetzt sind.

Bei den Abstrakten Datentypen (ADT) sei hier beispielhaft der ADT *Menge* dargestellt. Dieser erlangt als einfachster und am häufigsten verwendeter ADT besondere Bedeutung für die durchzuführende Sequenz.

Der ADT *Menge* ist eine ungeordnete Menge von Objekten. Er stellt in der einfachsten Variante nur zwei Operationen zur Verfügung: *einfügen*, *entfernen* und *ausgeben*. Mit dieser geringen Funktionalität lassen sich zusammen mit generischen Klassenmethoden schon Strukturen implementieren, die eine iterative Verarbeitung von Objekten gleichen Typs ermöglichen.

Des weiteren stelle ich den Schülern zur Wahl mit einem von mir implementierten Grafiksystem namens *Zeichenfläche* zu arbeiten. Dieses System abstrahiert die zur Ausgabe von Grafik in der Programmiersprache Java notwendigen Abläufe und ermöglicht so den Schülern einen einfachen Zugang zur Erstellung von grafischen Darstellungen und Animationen auf dem Bildschirm. Unter anderem stellt die *Zeichenfläche* auch die Methoden eines ADT *Menge* zur Verfügung: Objekte, die einer speziellen Grafikklassse angehören, können eingefügt und wieder entfernt werden. Alle in die *Zeichenfläche* eingefügten Objekte werden in jeder Zeichenphase animiert (falls Animationsalgorithmen implementiert wurden) und gezeichnet. Die Schüler haben im Verlauf des bisherigen Unterrichts bereits mit der *Zeichenfläche* gearbeitet und sind damit vertraut.

Des weiteren plane ich, in der Sequenz die objektorientierte Programmierumgebung *Greenfoot* zur Verfügung zu stellen. Dies ist ein auf die Java-Entwicklungsumgebung *BlueJ* aufbauendes System, welches ähnlich der *Zeichenfläche* mit einfachen Mitteln eine grafische Ausgabe ermöglicht. Es können hier Java-Klassen definiert werden, denen jeweils ein kleines Symbol in einer virtuellen Welt zugeordnet wird. Die Erleichterung, die *Greenfoot* hier bietet, ist durch die Verfügbarkeit von Methoden zur Erkundung der Welt gegeben. Durch Erweiterung an den richtigen Stellen kann der Benutzer die Bewohner der *Greenfoot*-Welt mit spezifischem Verhalten ausstatten, welches er als Algorithmen in Java implementiert. *Greenfoot* ist für die Zielgruppe der Java-Anfänger konzipiert. Da ich zwei Schüler in meiner Lerngruppe

habe, die bisher noch keine Erfahrungen mit Java haben, biete ich den Einsatz von *Greenfoot* zusammen mit deutschem Lernmaterial vom Berliner Bildungsserver (Bebis 2008) an. Das Lernmaterial bietet eine Einführung in Greenfoot und gibt Anregungen zur selbstständigen Erkundung des Systems.

4.2) Planung der Sequenz

In diesem Abschnitt stelle ich die grundsätzliche Planung der Unterrichtssequenz dar und begründe meine Planungsentscheidungen.

Grundsätzlich möchte ich in dieser Einheit die Schüler dazu anleiten, sich selbstständig die bereits beschriebenen erweiterten Konzepte der objektorientierten Programmierung anzueignen und den Prozess der Aneignung in einem Portfolio zu dokumentieren. Die Motivation der Aneignung geschieht durch die Wahl einer zu lösenden Problemstellung. Um das gestellte Problem zu lösen, müssen sich die Schüler die in der Sachanalyse (Abschnitt 4.1) definierten Programmierkonzepte erarbeiten und erlangen somit in beiden Bereichen einen Lernzuwachs. Für die Erstellung des Portfolios sollen die Schüler – wie in Abschnitt 3.6 bereits erwähnt – ein Content-Management-System verwenden.

Zunächst ist festzuhalten, dass in dieser Einheit die gemeine dreiteilige Stundenstruktur aufgrund des offenen Unterrichtscharakters nur rudimentär vorhanden sein wird. Es ist mir wichtig, die Stunden trotzdem nicht „einfach so“ beginnen zu lassen. Die kurzen Einleitungs- und Schlussphasen werde ich einhalten und für die Abfrage individueller Zielsetzungen oder Schwierigkeiten nutzen, die es in der jeweiligen Stunde zu bearbeiten gilt.

Die grobe Einteilung der Sequenz gestaltet sich wie folgt:

- Einführung in die Portfolioarbeit und Themenfindung
- Arbeitsphase
- Meilensteine und Zwischenpräsentation der Portfolios
- Arbeitsphase
- Abschlusspräsentation der Produkte und Portfolios
- Klausur

Die Einleitung hat die Funktion die Schüler auf die Portfolioarbeit einzustimmen. Sie besteht neben einer kurzen Erläuterung des Portfoliobegriffs aus der Besprechung der möglichen Themen.

Um die Schüler bei der Bearbeitung ihrer Programme zu betreuen und die Möglichkeit für

Rückfragen einzuräumen, setze ich einen großen Teil der Unterrichtssequenz für die Arbeit mit den Rechnern ein. In diesen Arbeitsphasen schaue ich den Schülern beim Recherchieren und Programmieren über die Schulter und versuche, individuell auftretende Schwierigkeiten bei der Bedienung der Technik, der Suche nach Informationen oder der konkreten Implementierung einer Problemlösung zu beseitigen. Dabei greife ich keinesfalls in die Bearbeitung von Schülerprojekten ein, sondern gebe möglichst eindeutige Hinweise auf mögliche Lösungen bzw. Lösungsansätze.

Zur Stabilisierung der Arbeit an den Portfolios wird zur Halbzeit der Sequenz eine Bestandsaufnahme durchgeführt, indem die Schüler ihre bis zu diesem Zeitpunkt erstellten Portfolios kurz präsentieren. Diese Phase soll einerseits mir einen detaillierten Einblick in die hinter den Portfolios stehenden Ideen geben, falls diese aufgrund der laufenden Bearbeitung noch nicht explizit erkennbar sind. Andererseits soll sie auch den anderen Lernenden eventuelle Anregungen geben, was noch in ihren Portfolios fehlt bzw. welche Inhalte zur Bereicherung und Vertiefung führen könnten.

Die Präsentationen der fertigen Portfolios bilden den Abschluss der Unterrichtssequenz. Die Schüler geben hier einen kurzen Überblick über das bearbeitete Thema und beschreiben dann wichtige Details der Dokumentation ihres Lernfortschritts. Sie erhalten somit die Möglichkeit, die Aufmerksamkeit ihrer Mitschüler auf die Informationen zu richten, die ihnen besonders wichtig sind bzw. die sie für besonders aussagekräftig bezüglich ihres erarbeiteten Lernfortschritts halten. Die jeweilige Präsentation soll nicht länger als zehn Minuten dauern. Schüler, die in einer Gruppe gemeinsam ein Thema bearbeiteten, stellen dieses Thema auch gemeinsam vor. Ihre Portfolios sollen selbstverständlich individuell vorgestellt werden, da jeder einen anderen Beitrag zur Gruppenarbeit geleistet hat.

Danach erfolgt die Bewertung der Portfolios durch die Mitschüler. Hierzu erstelle ich einen Bewertungsbogen auf der Grundlage der im Unterricht von den Schülern zu erarbeitenden Gütekriterien für die Portfolios (siehe Abschnitt 4.4.3).

Während der gesamten Unterrichtseinheit erhalten die Schüler die Aufgabe, ihre Fortschritte in Form von Blogeinträgen festzuhalten. Hierzu steht ihnen die Blog-Komponente der bereits erwähnten Internetplattform zur Verfügung. Ich erwarte mehrmals wöchentlich eine kurze formlose Notiz über die aktuelle Arbeit der Schüler. Dies soll mir einerseits ein weiteres Instrument zur Verfolgung der Fortschritte meiner Schüler zur Verfügung stellen, andererseits haben die Schüler bei kontinuierlicher Nutzung des Blogs bereits eine chronologische Aufstellung ihrer Tätigkeiten. Der Blog hat also im Rückblick auf das Geleistete eine starke Strukturierungsfunktion.

4.3) Synopse

Std. Nr.	Inhalt	Kompetenzzuwachs	Methode
1	Einstufung	Personale K.: Selbsteinschätzung	Fragebogen, Einzelarbeit
2	Einführung: Was ist ein Portfolio?	Fach-K.: Kennenlernen verschiedener Portfolio-Varianten, Sinn und Zweck	Lehrervortrag, Partnerarbeit, Unetrachtsgespräch
3	Gruppenfindung, Themenstellung und erste Zielsetzung	Meth.-K.: Abschätzen der eigenen Fähigkeiten, Zerlegen eines großen Problems in Teilprobleme, Machbarkeitsanalyse	Gruppenarbeit, Partnerarbeit, Einzelarbeit
4	Arbeit an den Programmen u. Portfolios	Meth.-K.: Problemlösen, Dokumentation Fach.-K.: Aneignung des zur Durchführung der Implementierung nötigen Wissens; Benutzung der Implementierungs- und Dokumentationswerkzeuge Soziale K.: Abstimmung mit Teammitgliedern Personale K.: Meilensteine setzen, Machbarkeitsanalyse	Offener Unterricht
5	Arbeit an den Programmen u. Portfolios	s. Std. 4	Offener Unterricht
6	Erarbeitung von Bewertungskriterien für Portfolios	Personale K.: Selbsteinschätzung, Bewertung eigener Leistungen Meth.-K.: Mindmap	Gruppenarbeit, Unterrichtsgespräch, Mindmap
7	Arbeit an den Programmen u. Portfolios: Überprüfung der Portfolios anhand der Bewertungskriterien	s. Std. 4	Offener Unterricht
8	Beratungsgespräche	Meth.-K.: Selbstdarstellung, Planung von Meilensteinen Personale K.: Selbsteinschätzung	Einzelinterviews
9	Beratungsgespräche	s. Std. 9	Einzelinterviews
10	Arbeit an den Programmen u. Portfolios	s. Std. 4	Offener Unterricht

Std. Nr.	Inhalt	Kompetenzzuwachs	Methode
11	Zwischenpräsentation	Meth.-K.: Bewertung von Arbeiten anderer Schüler nach best. Kriterien	
12	Arbeit an den Programmen u. Portfolios	s. Std. 4	Offener Unterricht
13	Arbeit an den Programmen u. Portfolios	s. Std. 4	Offener Unterricht
14	kurze Beratungsgespräche	s. Std. 9	Einzelinterviews
15	Präsentationen I	Meth.-K.: Präsentieren, Selbstdarstellung, Bewertung von Arbeiten anderer Schüler nach best. Kriterien Soziale K.: Üben konstruktiver Kritik	Schülervortrag
16	Präsentationen II	s. Std. 15	Schülervortrag
17	Präsentationen III	s. Std. 15	Schülervortrag
18	Klausur		

4.4) Durchführung der Sequenz

Zum Anfang dieses Abschnitts möchte ich einen kurzen Überblick über den Ablauf der Sequenz geben und dann drei Aspekte deutlicher untersuchen. Darüber hinaus macht die genaue Darstellung einzelner Stunden wenig Sinn, da viele Stunden wie zu Anfang dieses Kapitels erwähnt durch die grobe Strukturiertheit des offenen Unterrichts gekennzeichnet sind. Die meisten Stunden enthalten einen hohen Anteil an Einzel- bzw. Gruppenarbeit. Sicherungsphasen fehlen ebenso wie Wiederholungs- bzw. Vertiefungsphasen, da die individuellen Themenstellungen das Besprechen von Arbeitsergebnissen im Plenum überflüssig machen.

Der zeitliche Ablauf der Sequenz wurde wie geplant eingehalten. Portfolios und Programme der Schüler waren zur Abschlusspräsentation in einem vorzeigbaren Zustand. Durch die kurz vorher durchgeführte Umstellung der Rechneranlage auf ein neues Betriebssystem wurde die Unterrichtseinheit von kleineren technischen Problemen begleitet (der Editor *BlueJ* nahm beispielsweise keine geschweiften Klammern mehr entgegen, die aber für das Schreiben von Quelltext in Java essentiell sind). Das war zwar stellenweise hinderlich, betraf aufgrund der verschiedenen Themen jedoch meist nur einzelne Schüler.

Seitens der Schulverwaltung erfuhr ich großes Entgegenkommen bei der Verschiebung des Klausurtermins für den Basiskurs, der zunächst in der zweiten Woche nach Beginn meiner Sequenz angesetzt war. Ich durfte mir einen beliebigen Termin aussuchen. Das Fehlen der Schüler im anderen Unterricht wurde ausnahmslos entschuldigt.

4.4.1) Beratung und Interviews

Im Laufe der Unterrichtsreihe führte ich in mehreren Stunden neben dem üblichen Monitoring Beratungsgespräche und Einzelinterviews durch, deren Sinn und Ergebnis ich in diesem Abschnitt erläutere.

Kontinuierliche Beratung ist laut Fachliteratur (siehe Abschnitte 3.2 und 3.5, vgl. Kolb 2007 S. 145 ff.) ein integraler Bestandteil der Methode Portfolioarbeit. Die Beratung soll einerseits dem Lehrer einen Einblick in die individuellen Fortschritte der Lernenden ermöglichen, andererseits können die Schüler auf diesem Wege Rückmeldungen zu ihrer Arbeit erhalten. Als Berater hat der Lehrer also die Aufgabe, den Schülern beim Überprüfen ihres Fortschritts zu helfen und richtungsweisend Anregungen zu geben, die zum Erreichen der Ziele wichtig sein können. Mein Ziel in den Beratungsstunden war es demnach, den aktuellen Fortschritt der Schüler einzuschätzen und bei Schwierigkeiten entsprechende Materialien oder Hilfsmittel zur Verfügung zu stellen. Aufgrund der Themenvielfalt war schwer vorauszusagen, welcher

Art diese Hilfsmittel sein könnten.

In möchte nun detailliert auf die Einzelinterviews eingehen, da sich hier eine Entwicklung darstellen lässt, die auf das zentrale Problem der Unterrichtseinheit hinweist.

Zur Vorbereitung auf die Interviews habe ich mir einen Fragenkatalog erstellt (s. Anhang S. 51). Ich habe die Fragen zwar nicht als obligatorisch angesehen, dennoch brauchte ich eine Ideensammlung, die es mir erlaubte, die in meinen Augen wichtigsten Aspekte der Schülermeinung im Interview zu erfahren. Die Dauer eines Interviews habe ich auf maximal fünf Minuten beschränkt. Des weiteren habe ich vor den Interviews das – zu diesem Zeitpunkt in den wenigsten Fällen vorhandene – Portfolio des zu interviewenden Schülers durchgesehen.

Grundsätzlich beantworteten die Schüler die Frage nach dem Fortschritt ihres Portfolios ehrlich – es war wenn überhaupt nur in Ansätzen vorhanden. Dem entgegen hatte die Programmierarbeit gute Fortschritte gemacht. Es stellt sich also die Frage nach dem Grund dieser Konzentrierung auf das Produkt.

Auf die Frage hin, welchen Sinn das Portfolio in ihren Augen hätte, antworteten die meisten Schüler damit, dass es dazu diene, ihren persönlichen Lernweg und -fortschritt zu dokumentieren und ihrem Lehrer darzustellen, was sie sich erarbeitet hätten. Bei drei Schülern, deren Portfolio nicht einmal ansatzweise vorhanden war, mutmaßte ich, sie hätten also noch nichts gelernt. Christophs Antwort darauf soll hier exemplarisch für die anderen stehen: „Natürlich habe ich schon etwas getan! Ich habe es schon hinbekommen, ein Fenster mit Swing aufzumachen² und einen JButton darin eingefügt, aber das ist doch erst der Anfang. In mein Portfolio wollte ich dann das fertige Fenster aufnehmen.“

An dieser Äußerung wird deutlich, wie schwer es den Schülern fiel, die eigene Leistung nachzuvollziehen und zu hinterfragen. Natürlich ist es ein enormer Fortschritt, das „Fenster“ korrekt programmiert und die damit verbundenen Konzepte verstanden zu haben. Dennoch erscheint dies dem Schüler zu trivial um der Aufnahme im Portfolio würdig zu sein.

Klar erkennbar ist an Christophs Aussage auch die Schwierigkeit, den eignen Lernprozess bewusst wahrzunehmen und als ähnlich wichtig wie das fertige Produkt anzuerkennen. Das fertige „Fenster“ ist schließlich das Ergebnis dieses Lernprozesses, den es aufzuzeichnen galt.

Ähnlich äußerte sich Peter, der in einer Gruppe mit zwei anderen Schülern an einem Kampfspiel arbeitete (s. nächsten Abschnitt). Als ich ihn fragte, was er zu seinem Projekt beiträgt, antwortete er: „... Wir haben also erstmal die verschiedenen Bewegungsabläufe und so Sachen

2 Bedeutet: „auf dem Bildschirm anzuzeigen“, Anm. d. Autors

gemacht und ich hab' das alles dokumentiert, wobei Alexej das Größte bisher programmiert hat. [...] Wir haben immer zusammen über die Methoden geredet und er hat das gleich eingetippt und ich hab's dokumentiert. [...] haben wir noch nicht so genau geplant, wie wir das Ganze aufteilen [...].“

Hier zeigt sich die Strategie des Planens: Die Methoden (der einzelnen Klassen) müssen vereinbart und die Verteilung der Aufgaben muss vorgenommen werden. Interessanterweise dauert die Planung in dieser Gruppe offenbar sehr lange, wenn man bedenkt, dass die Beratungsgespräche etwa zur Halbzeit der Sequenz stattfanden. Dennoch ist Peter deutlich klar, in welche Richtung die Arbeit geht und was als nächstes zu tun ist. Ich riet, einige Detailentscheidungen zu vertagen und sich auf die Aufgabenverteilung und -bearbeitung zu konzentrieren, damit die Implementierung voran schreiten konnte.

Meine Frage nach dem Stand der Ausarbeitung des Portfolios beantwortete Peter ausweichend, denn bisher hatten die Schüler seiner Gruppe nur zwei Blogeinträge erstellt (Erwartet hatte ich eigentlich eine kontinuierliche Rückmeldung im Blog mit etwa drei Einträgen pro Woche). Nach dem vermeintlichen Sinn des Portfolios gefragt äußerte sich Peter wie folgt: „Also erstmal ist es ja ganz gut um nochmal reinzukommen, weil ich mir das gerade nochmal angeschaut habe [...] weil ich war ja letzte Woche nicht da [...] und na ja, ich denk' mal, das ist eigentlich ganz gut nochmal nachzuvollziehen, wenn man das Projekt nachher fertiggestellt hat.“

Peter sieht den alleinigen Sinn des Portfolios in seiner Funktion als Dokumentationsinstrument. Einen Nutzen für seine persönliche Entwicklung sah er nicht. Dies macht deutlich, dass das Portfolio hier noch nicht als Reflexionsinstrument verstanden wurde, wie es bei allen Schülern der Lerngruppe der Fall war.

Ich nutzte die persönlichen Gespräche deshalb hauptsächlich, um nochmal die Selbstkritik im Zusammenhang mit dem Portfolio zu erklären. Dies war essentiell, um schließlich die erwarteten Ausprägungen der Portfolios zu erhalten. Anderenfalls hätten die Schüler nur Dokumentationen ihrer Programme erstellt.

4.4.2) Portfolios und Präsentationen

In diesem Abschnitt möchte ich kurz auf die von den Schülern in der Reihe erstellten Arbeiten eingehen. Ich zeige daran auf, welche Leistung die Schüler gezeigt und welchen Lernzuwachs im Sinne der Messbarkeit (Abschnitt 3.3) sie erreichten. Es folgt zunächst eine tabellarische Übersicht aller bearbeiteten Themen und der darin verwendeten objektorientierten Konzepte.

Ich werde danach die Schülerarbeiten beschreiben, damit die Erklärungen zu den Portfolios klarer werden, da sie sich stark auf die erarbeiteten Produkte beziehen.

Name(n)	Thema	angewandte Konzepte der OOP
Max, Alexej, Peter	„MAP Brothers“, ein Action-Kampfspiel für zwei Spieler	Erstellung eigener Klassenstruktur inkl. Attribute, Methoden;
Lisa, Christoph	Währungsumrechner	Grafische Oberfläche mit Swing: Einbindung und Konfiguration von Eingabeelementen, Ereignisbehandlung bei Benutzerinteraktion
Raphael, Philipp	Graphenzeichner für polynomiale Funktionen	Grafische Oberfläche mit Swing: Einbindung und Konfiguration von Eingabeelementen, Ereignisbehandlung bei Benutzerinteraktion, Anzeige des Funktionsgraphs
Liane	OOP mit Greenfoot	Grundlagen der OOP: Klassen, Methoden, Attribute und einfache Algorithmen in Java
Torsten	Erweiterung eines Online-Rollenspiels	Vektoren (ADT <i>Menge</i>), Server-Client-Kommunikation, Sprite-Grafik
Sebastian	Content-Management-System in PHP und MySQL	wenig berücksichtigt

Im Allgemeinen betrachtet haben alle Schüler im Rahmen ihrer Möglichkeiten gute Beiträge geleistet. Keiner der Schüler hat sich verweigert oder die Charakteristik des offenen Unterrichts ausgenutzt, um die Zeit mit Computerspielen oder dem Surfen im Internet zu verbringen. Es wurde selbstständig und zielstrebig gearbeitet.

Diese Beobachtungen sind hauptsächlich durch den Umfang und den Detailreichtum der von den Schülern angefertigten Arbeiten begründet. Die meisten Programme waren zum Zeitpunkt der Präsentation mindestens in einem Zustand, in dem die grundlegenden Funktionen bereits

implementiert waren.

Am weitesten entwickelt war der Graphenzeichner von Raphael und Philipp. Das präsentierte Programm unterstützte die Eingabe eines Polynoms von theoretisch beliebigem Grad durch Angabe der jeweiligen Koeffizienten. Im nächsten Schritt konnten wiederum beliebig viele Polynome in einem kartesischen Koordinatensystem eingezeichnet und skaliert werden.

Raphael und Philipp arbeiteten streng nach objektorientierten Prinzipien. In-

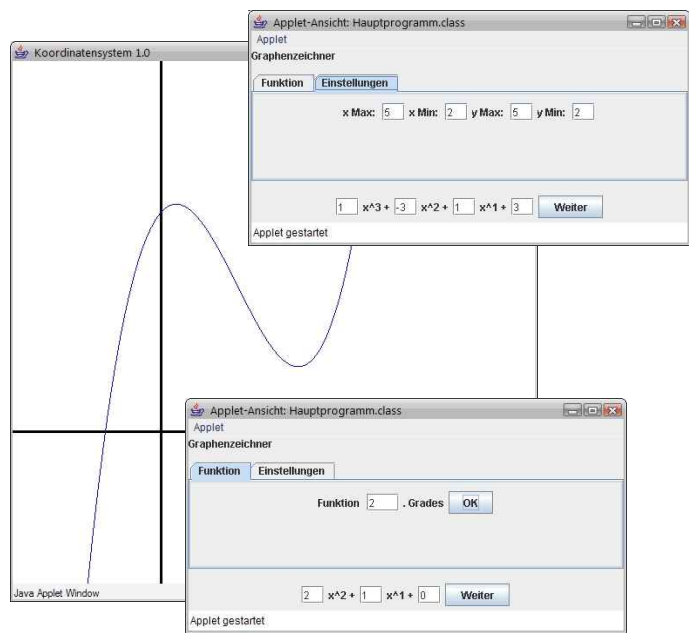


Abbildung 4: Bildschirmausschnitt der Ausgabe des Graphenzeichners

teressant ist in ihrer Klassenstruktur die Trennung zwischen dem Graphen selbst und dem Koordinatensystem. Für beide Konzepte wurde eine eigene Klasse angelegt. Dies ermöglichte es den Schülern, die Arbeiten an beiden Klassen gleichzeitig und unabhängig voneinander durchzuführen. Da das Koordinatensystem weniger aufwändig zu programmieren war als die Darstellung eines Funktionsgraphen, konnte schnell eine Binnendifferenzierung erreicht werden: Philipp, der durch sein Auslandshalbjahr wenig Erfahrung in der Java-Programmierung hatte, wurde von Raphael in die Grundlagen von Java eingeführt und implementierte die Klasse für das Koordinatensystem (benutzte Strukturen: einfache Schleifen, Zeichnen von Geraden in der *Zeichenfläche*), Raphael dagegen erstellte die Graphen-Klasse (benutzte Strukturen: Felder, Vererbung, ADT *Menge*, Verwaltung mehrerer Instanzen der Klasse „Graph“).

Max, Alexej und Peter arbeiteten nach ähnlichen Prinzipien wie Raphael und Philipp. Sie erstellten ein Kampfspiel für zwei Spieler. Zur Zeit der Präsentation konnten sie zwei Avatare (dargestellt durch ausgefüllte Kreise) unabhängig voneinander mit der Tastatur steuern. Darüber hinaus konnten die Avatare auf mehreren Plattformen laufen, die durch farbige Balken dargestellt wurden.

Auch hier stellt sich die Klassenstruktur als intelligent strukturiert dar. Die einzelnen Konzepte (Figur, Balken, Text, etc.) sind jeweils als Klasse modelliert und konnten somit unabhängig voneinander implementiert werden. Die drei Schüler investierten viel Zeit und Mühe in die Konzeption des Programms (s. Interview mit Peter, S. 26 f.), wodurch ihnen zum Schluss die

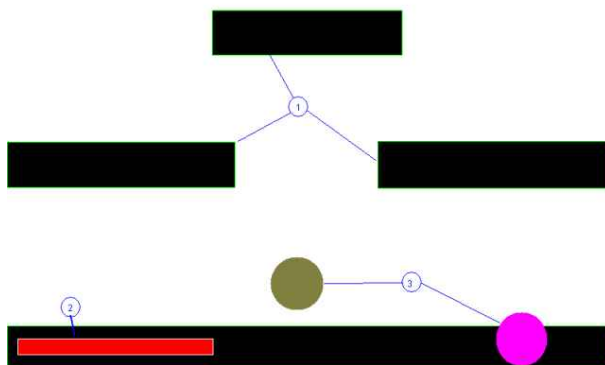


Abbildung 5: Bildschirmausschnitt des Spiels "MAP Brothers": (1) Plattformen, (2) Trefferanzeige, (3) Figuren

Alexej informierte sich über die grundlegende Simulation von physikalischen Vorgängen wie der Gravitation und dem elastischen Stoß. Die Implementierung der Spielphysik und von Kollisionsprüfungen lag in seinem Hauptinteresse. Max kümmerte sich um den generellen Spielablauf, die Punktezahl und -ausgabe und die Verarbeitung der Tastatureingaben.

Torsten hatte sich ebenfalls zum Thema gesetzt, ein Spiel zu bearbeiten. Es handelte sich hierbei um ein browserbasiertes Online-Rollenspiel. Der Spielclient wird als Java-Applet im Browser des Spielers ausgeführt und verbindet sich zum ebenfalls in Java geschriebenen Server. Durch diese Struktur wird eine Benutzerinteraktion sofort an alle am Spiel teilnehmenden Benutzer weitergeleitet, d.h. man sieht die Bewegungen der anderen Spieler sofort und könnte mit ihnen kommunizieren, handeln, kämpfen etc.

An genau dieser Interaktionskomponente fehlte es dem Spiel bisher noch. Im Rahmen der Portfolio-Sequenz nahm Torsten sich vor, das Spiel um eine Kampfschnittstelle zu ergänzen, in der Spieler rundenbasiert ihre strategischen Fähigkeiten messen können. Torsten erweiterte entsprechend die Klassenstruktur des Spielsystems um die benötigten Klassen (*Gegenstand*, *Waffe* etc.). Er verwendete dabei die Vererbung, um die verschiedenen *Gegenstände* in einer generischen Datenstruktur speichern zu können (*Vektor*, besitzt ähnliche Eigenschaften wie der in Abschnitt 4.1 beschriebene ADT *Menge*).

Torsten brachte stets umfangreiche Erfahrungen in der Programmierung mit Java in den Unterricht ein und stellte durch die Bearbeitung seines Themas dar, über welche Fähigkeiten er darüber hinaus verfügt. Sein Lernzuwachs beschränkt sich allerdings auf die Erstellung der entsprechenden Dokumentation, was in seinem Fall auch sehr wichtig ist, da er das Dokumentieren seiner eigenen Programme für überflüssig hält (s.u.).

Zeit zur Implementierung fehlte. Darüber hinaus hatten sie sich im Anforderungsniveau überschätzt. Sie hatten zunächst geglaubt, ihr Programm nur mit im Unterricht gelernten Fähigkeiten umsetzen zu können, mussten aber schnell feststellen, dass sie sich zusätzliche Informationen aus dem Internet beschaffen mussten. Peter beschäftigte sich z.B. mit zusätzlichen Funktionen beim Zeichnen von Grafikprimitiven zur Darstellung

Sebastian erstellte ein Content-Management-System in der Skriptsprache PHP. Er hatte damit das am weitesten gesteckte Thema, denn die Erstellung eines solchen Systems ist extrem aufwändig und kann mitunter ganze Online-Gemeinschaften beschäftigen (siehe www.drupal.org). Die Voraussetzung, objektorientierte Konzepte zu nutzen, hat Sebastian jedoch nicht umgesetzt. Obwohl er die von der Sprache PHP zur Verfügung gestellten objektorientierten Sprachkonzepte verwendet hat, so war sein Programm dennoch nicht objektorientiert: Von jeder Klasse existierte zur Laufzeit maximal eine Instanz. Entsprechend unpassend sah seine Klassenstruktur aus.

Richtungsweisungen meinerseits halfen nichts, um wenigstens Teile in Sebastians Programm objektorientiert zu gestalten. Er war sehr sicher in allem, was das Zusammenspiel von PHP und SQL anging und die Schnittstellendefinitionen (API) kannte er auswendig. Nur hat er bisher stets imperativ (bzw. sequentiell) programmiert. Ein Umdenken in der ihm so vertrauten Umgebung war sehr schwer für ihn. Ich sah auch ein, dass für das Thema, so wie er es angelegt hatte, eine imperative Lösung viel nahe liegender war als eine objektorientierte. Eventuell wäre es besser gewesen, ihm von vornherein die Verwendung von PHP/SQL zu verweigern und ihn zu bitten, ein Java-Programm zu erstellen, obwohl er daran weniger interessiert war. Denn so hat er den von mir angestrebten Lernzuwachs in der objektorientierten Programmierung leider nicht erreicht.

Im Gegensatz zu den anderen Schülern hatte Liane kein Thema für sich gefunden, das sie bearbeiten konnte. Das war auch nicht anders zu erwarten, denn durch ihre Abwesenheit im Auslandshalbjahr hatte sie keine Erfahrung in der Programmierung mit Java und konnte deshalb auch nicht einschätzen, wie komplex die Umsetzung eines Programmierthemas sein konnte (vgl. meine Anforderungen an die Einschätzung der Problemkomplexität im Abschnitt 3.3). Ich habe ihr deshalb eine einleitende Arbeit vorgegeben. Diese bestand darin, sich in die in Abschnitt 4.1 beschriebene *Greenfoot*-Umgebung einzuarbeiten. In den Beratungsgesprächen habe ich Liane dann wie beschrieben dazu angeleitet, sich ein Problem auszudenken, welches sie mit der *Greenfoot*-Umgebung lösen kann. Sie nahm sich vor, eine Spinne durch ein beliebiges Labyrinth aus Hindernissen manövrieren zu lassen. Den Algorithmus hat sie dann in ihrer Präsentation vorgestellt. Das fertige Programm in einer offensichtlich von ihr stammenden Implementierung lässt mich darauf schließen, dass sie sich in *Greenfoot* zurecht findet und einfache Probleme selbstständig damit lösen kann.

Mit dem Währungsumrechner von Christoph und Lisa beschäftige ich mich hier nicht mehr, da dieser stark ähnliche Konzepte aufweist wie die Eingabekomponente des Graphenzeich-

ners von Raphael und Philipp. Die Ähnlichkeit entsteht in der Verwendung der Swing-Schnittstelle von Java.

Die Informationen, aus denen ich die Beschreibungen zusammengestellt habe, entnahm ich sowohl den Programmen selbst als auch den Gesprächen mit den Schülern und den Portfolios. Ich werde nun näher auf die Strukturen der Schülerportfolios eingehen um zu zeigen, dass auch hier meine Anforderungen von den Schülern größtenteils erfüllt wurden.

Die Portfolios wurden ausnahmslos auf meiner Webseite erstellt; keines wurde schriftlich nachgereicht o.ä. Alle Portfolios waren zum jeweiligen Präsentationstermin vorhanden und wurden vorgestellt bzw. bewertet.

Da die Schüler die Qualitätskriterien, nach denen ihre Portfolios bewertet würden, selbst erarbeitet hatten (siehe nächsten Abschnitt) war ihnen klar, auf welche Punkte sie bei der Erstellung ihrer Portfolios Wert legen mussten. Die meisten Schüler hatten also ein Inhaltsverzeichnis, sowie mehrere, untereinander verknüpfte und chronologisch angeordnete Seiten und eingebettete Bilder (Konzeptzeichnungen, Bildschirmfotos, Klassendiagramme etc.). Inhaltlich bewegten sich die Portfolios zwischen der gewünschten Dokumentation der angeeigneten Fähigkeiten bzw. des erarbeiteten Wissens und der Beschreibung der Funktionen des geschriebenen Programms.

Torstens Portfolio beispielsweise stand hinter meinen Erwartungen an ihn zurück. Er schrieb sein Portfolio als Handbuch und interne Dokumentation seines Programms. So fehlten bei ihm die Darstellungen von Problemen und dazugehörigen Lösungen genauso wie die Aufzeichnungen dessen, was er sich während der Bearbeitung seines Themas angeeignet hat.

Natürlich bin ich mir im Klaren darüber, dass er durch seine ausgezeichneten Fähigkeiten und Erfahrungen nicht mehr oft in der Java-Schnittstellendokumentation nach Funktionen nachschlagen muss. Doch gerade deshalb definiert und löst er Probleme umso öfter und schneller. Seine Ablehnung der ausführlichen Dokumentation ist meines Erachtens bedingt durch seinen großen Vorsprung als „Einzelkämpfer“. Torsten musste sich bisher nie mit anderen Programmierern oder Kunden abstimmen, weil er immer genau die Problemstellungen löste, die er sich selbst stellte. Dies tat er immer allein. Daraus erwächst eine gewisse Überheblichkeit gegenüber dem Aufzeichnen der Arbeitsschritte: „Wer braucht das schon?“, oder: „Ist doch nur für mich“, waren die Argumente, die ich hörte, wenn ich ihn auf die Notwendigkeit der Kooperation in der Informatik aufmerksam machte. Kennzeichnend ist hier auch, dass er sich von vornherein keiner Gruppe anschließen wollte.

Christophs Portfolio hat vollständig die Anforderungen erfüllt, die ich für angemessen in Bezug auf ein prozessorientiertes Portfolio halte. Bei ihm spiegelte sich deutlich wider, in welcher Reihenfolge er sich die Strukturen der Swing-API erarbeitet hat. Am Anfang steht ein in der Gruppe erarbeitetes Muster einer Oberfläche, welche die vom Währungsumrechner benötigten Eingaben ermöglicht. Dann schreibt er die Webseiten auf, die er besucht hat, um die Grundlagen der Erstellung von grafischen Oberflächen mit Bedienelementen zu lernen. Als nächstes folgen erste Versuche, d.h. kleine Programme, in denen er die verschiedenen Bedienelemente miteinander kombiniert und mit den Layoutmanagern experimentiert. Zum Schluss beschreibt er die Erstellung der eigentlichen Benutzeroberfläche.

Hieran lässt sich zweierlei darstellen: Einerseits ist deutlich, wie die Aufgaben innerhalb der Arbeitsgruppe zum Thema Währungsumrechner verteilt waren. Aus Christophs Portfolio geht klar hervor, dass er alle wichtigen Informationen zum Thema GUI gesammelt und verarbeitet hat. Ein Blick in Lisas Portfolio zeigt, dass sie sich stattdessen um Währungskurse und die entsprechende Umrechnung gekümmert hat. Beide hatten also klar abgegrenzte Aufgabenfelder und mussten sich in deren Überlappungsbereichen entsprechend verständigen, damit das Programm am Ende funktionierte.

Andererseits kann an beiden Portfolios wieder der Bezug zur Binnendifferenzierung dargestellt werden. Während sich der in der Programmierung schon erfahrene Christoph mit einem programmieraufwändigen Teilaspekt des Themas beschäftigte, suchte sich Lisa einen eher mathematisch-theoretischen Teil aus. Auf den ersten Blick mag die eigentliche Umrechnung von einem Währungskurs in einen anderen zwar trivial erscheinen, dennoch musste auch die Speicherung und der Abruf der Kurse berücksichtigt werden. Hier konnte auch Lisa ihr Wissen und Können optimal einbringen. Sie konnte auf ihrer Kompetenzebene arbeiten und sich somit in einer ihrem Lerntempo gerechten Weise weiterentwickeln. Sowohl Lisa als auch Christoph haben dies exzellent in ihren Portfolios dokumentiert.

Interessant ist noch die Betrachtung von Lianes Portfolio, da ich hier noch einen wichtigen Aspekt der Dokumentation eines Prozesses aufzeigen möchte. Es geht um die Konservierung der einzelnen Arbeitsschritte. Liane hat im Laufe der Bearbeitung ihres Themas immer an ein und derselben Datei gearbeitet. Sobald ein Programm fertig war, hat sie den durch sie erstellten Teil wieder gelöscht und durch das Ergebnis des nächsten Arbeitsschritts ersetzt. Da sie – symptomatisch für die gesamte Reihe – ihr Portfolio nicht kontinuierlich erweiterte, sobald sie einen neuen Arbeitsschritt anging, fehlte es ihr an Material, das sie ins Portfolio hätte aufnehmen können. Leider ist mir das zu spät aufgefallen, um verloren gegangenen Quelltext wieder

herzustellen. So konzentrierte sich Liane stärker auf die Präsentation ihres Programms und hatte ein dementsprechend stärker produktorientiertes Portfolio.

4.4.3) Bewertung

Die Kriterien zur Bewertung eines Portfolios wurden von den Schülern in der sechsten Stunde der Unterrichtssequenz erarbeitet. Die von den Schülern zusammengestellten Punkte hielt ich in der folgenden Mindmap fest, um sie nach Kategorien geordnet darzustellen.

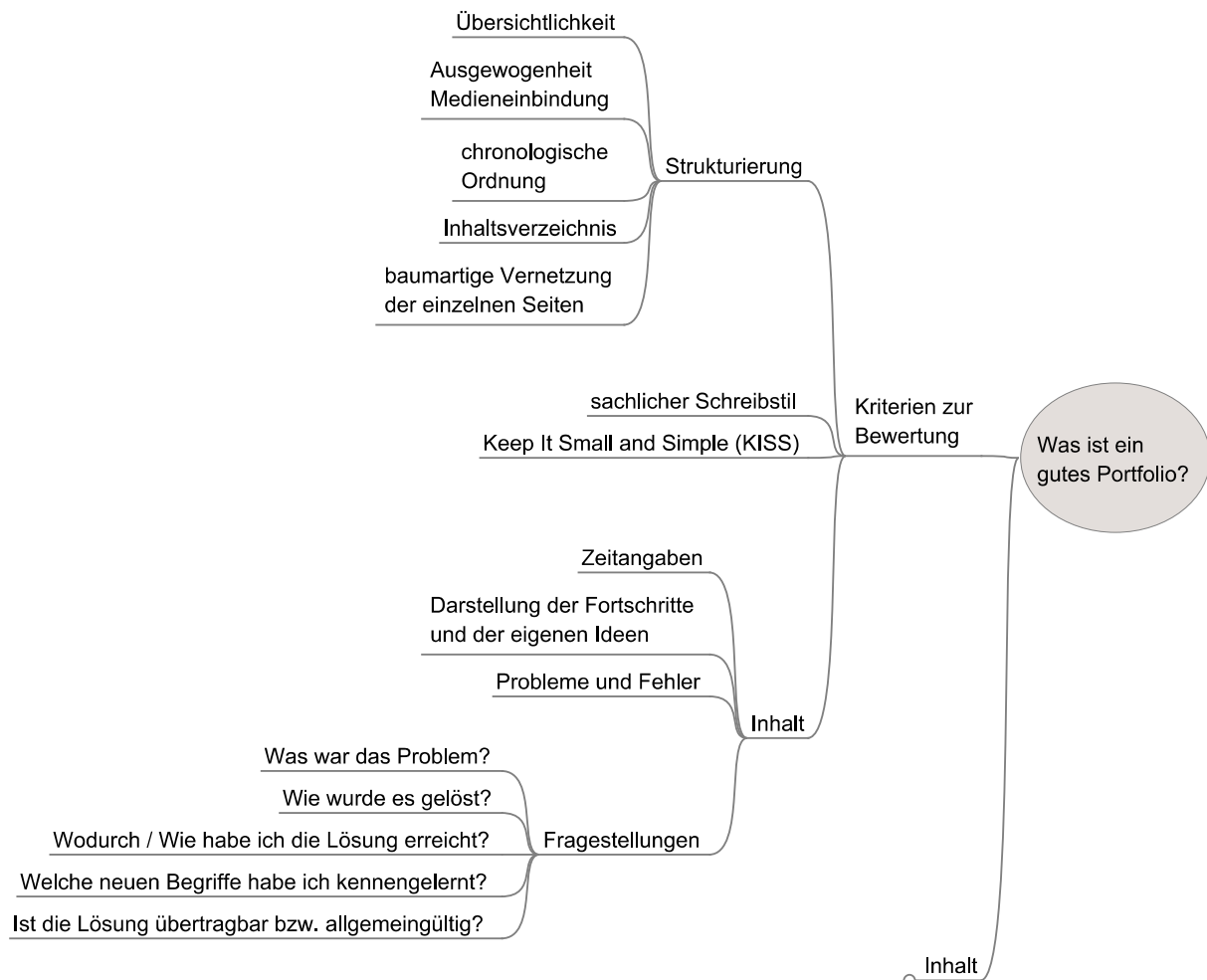


Abbildung 6: Mindmap mit Bewertungskriterien für Portfolios

Das wichtigste Merkmal des Portfolios stellen die fünf Fragestellungen dar, die ich gemeinsam mit den Schülern erarbeitete. Werden diese Fragen beantwortet, so decken die Antworten genau die Bereiche ab, die ich in dieser Unterrichtssequenz für interessant halte: Wie lösen die Schüler Probleme und welche neuen Konzepte eignen sie sich dabei an?

Durch die Hervorhebung dieser zentralen Fragestellungen sind die Ausführungen hierzu auch einfach im Portfolio wiederzufinden, im Kontrast zu abstrakten Kriterien wie „Übersichtlichkeit“ oder „Ausgewogene Darstellung“. Dies erleichtert die Bewertung der Darstellungen

durch die Mitschüler.

Unter dem Punkt „Inhalt“ (rechts unten) in der Mindmap haben die Schüler beispielhafte Inhaltskategorien zusammengetragen, um einen Überblick zu haben, welche Inhalte sich zur Integration eignen. Es wurden hier Ideen von einfachem Text über Bildschirmfotos bis hin zu Videobeiträgen genannt.

Aus den in der Mindmap dargestellten Kriterien erarbeitete ich einen Fragebogen (s. Anhang S. 48), der zur Bewertung der Portfolios herangezogen werden sollte. Jeder Schüler bewertete in einer Präsentationsstunde bis zu zwei Portfolios. So verfügten schließlich alle Schüler über mehrere Bewertungen ihrer Arbeit durch die Mitschüler. Ich bewertete alle Portfolios nach denselben Kriterien und gab meine Bewertung ebenfalls an den jeweiligen Portfolio-Autor weiter.

Es stellte sich heraus, dass der Bewertungsbogen die zu erfüllenden Normen gut abdeckte und an einigen Stellen die Schwächen der Portfolios bloßstellen konnte. Ich habe alle Ergebnisse der Bewertungen für vier exemplarische Schüler in der folgenden Grafik dargestellt:

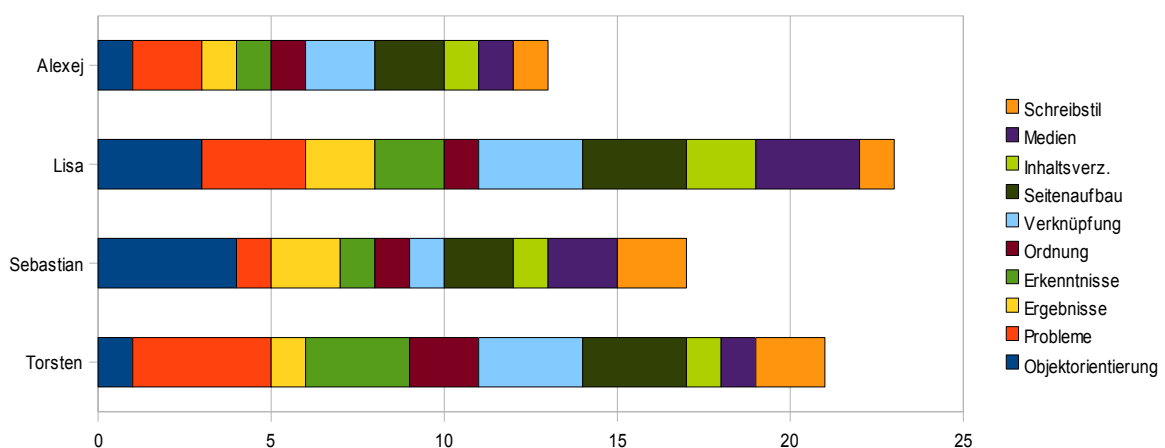


Abbildung 7: Bewertungen der Portfolios von ausgewählten Schülern, **kurze Balken** stellen ein **besseres Ergebnis** dar, Fragebogen s. Anhang

Klar lässt sich hier erkennen, was ich schon im Abschnitt 4.4.2 angesprochen habe: Sebastian hat ein gutes Portfolio erstellt; er verfügt auch über ausgezeichnete Fähigkeiten was die Programmierung von PHP und MySQL angeht. Dennoch hat er – auch nach mehreren Aufforderungen und Anregungen meinerseits – kaum mit Konzepten der Objektorientierung gearbeitet.

Torstens Portfolio hingegen war sehr produktlastig. Es las sich wie ein Handbuch zu seinem Programm und enthielt nur geringe Anteile, die seinen Lernprozess darstellten. Lisa hingegen hat – entsprechend ihrer Lernausgangslage – wenig Praktisches zu ihrem Projekt beigetragen und dennoch viel über Arithmetik und Währungen gelernt, wenn man nach ihrem Portfolio

geht. Sie scheint auch allgemein einen guten Überblick darüber zu haben, wie sie lernt. Alexejs Portfolio schließlich vereinigte alle wichtigen Punkte in sich: Prozessorientierung, Dokumentation der Konzepte, die er sich aneignete und sehr gute Aufbereitung der Informationen. Die Übereinstimmung der Bewertungen der Schüler mit meinem allgemeinen Eindruck bestätigt die Validität des Fragebogens.

Um eine adäquate Bewertung des Leistungsstandes am Ende der Unterrichtsreihe vornehmen zu können, schloss ich die Reihe mit einer Klausur ab. Ein Vergleich mit den anfangs durchgeführten „Einstufungstests“ (s. Anhang S. 49) sollte Klarheit darüber bringen, ob und welcher Lernzuwachs bei den Schülern eingetreten ist.

Um den diversen bearbeiteten Themen Rechnung zu tragen, entwickelte ich eine Klausur auf der Grundlage erwarteter Gemeinsamkeiten in der Bearbeitung der Projekte. Die Schüler bekamen die Möglichkeit, aus fünf unterschiedlichen Aufgaben diejenige auszuwählen, die ihnen mit ihrem Wissen am besten lösbar erschien.

So nahm ich beispielsweise an, dass Alexej, Peter und Max dieselben Grundkenntnisse über die „Zeichenfläche“ haben müssten und deshalb in der Lage wären, ein auf dieses Programm bezogenes Problem zu lösen. Ich griff dabei ein Problem auf, welches während der Projektarbeit oft diskutiert wurde (Was passiert, wenn mehrere Tasten gleichzeitig gedrückt werden?).

Liane erhielt eine speziell für sie konzipierte Aufgabe, die sich um die Programmierung mithilfe des Greenfoot-Systems drehte. Diese Aufgabe hatte aufgrund meiner Einschätzung von Lianes Lernstand einen viel geringeren Schwierigkeitsgrad in der Problematik als die anderen Aufgaben. Dennoch war sie für alle anderen Schüler unlösbar, da nur Liane über das spezielle Wissen über Greenfoot verfügte, welches für die Lösung zwingend notwendig war. Damit glich ich die Ungerechtigkeit gegenüber dem Schwierigkeitsgrad der anderen Aufgaben aus.

Um den Schülern über die Präsentationen der Portfolios hinaus noch einmal die Gelegenheit zur Darstellung des Erarbeiteten zu geben, wies ich sie an, ein in ihrem Projekt bearbeitetes und gelöstes Teilproblem in einem kurzen Aufsatz darzustellen. Bei der Bewertung dieses Aufsatzes legte ich Wert auf eine dem Problemlöseprozess entsprechende Struktur: Ich erwartete sowohl die knappen Beschreibungen der Problemstellung, des Lösungswegs und der fertigen Lösung.

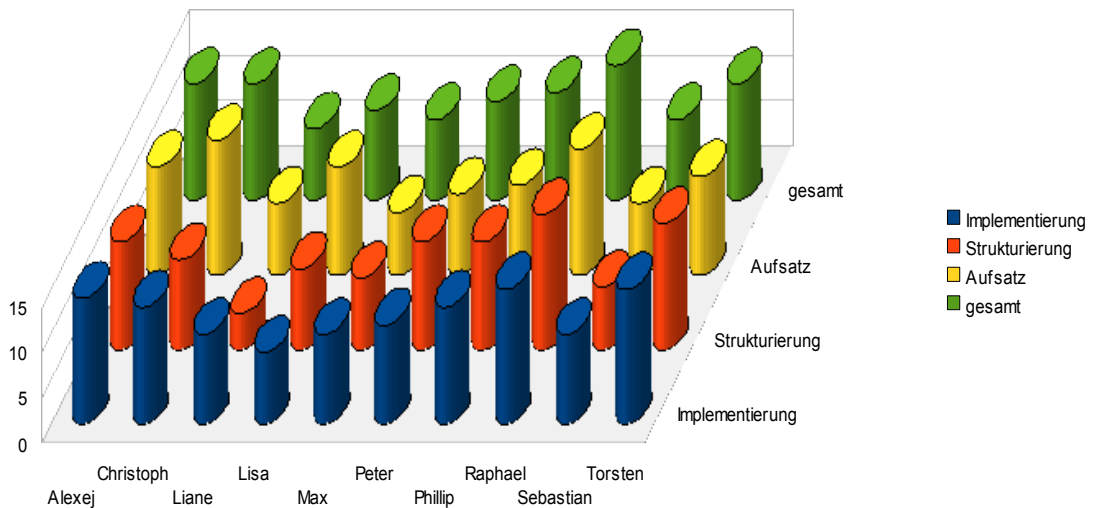


Abbildung 8: Klausurergebnis (Noten in Punktwerten der Oberstufe), Durchschnitt 11,2 Punkte

Die Klausur fiel gut aus (s. oben stehendes Diagramm). Sechs von zehn Schülern konnten die Strukturierungsaufgaben der Klausur gut bis sehr gut bewältigen. Dies zeigt den Lernzuwachs im Bereich der Problemlösefähigkeit deutlich im Vergleich zum Eingangstest. Die Implementierungsaufgaben mit insgesamt höherem Schwierigkeitsgrad als im Eingangstest wurden von acht von zehn Schülern gut bis sehr gut gelöst. Der insgesamt niedriger liegende Durchschnitt der Klausur begründet sich darin, dass ein Großteil der Schüler Probleme hatte, präzise zu formulieren. So konnten fünf von zehn Schüler den Aufsatz nur mit ausreichend oder schlechter abschließen, da die von mir geforderten Strukturierungskriterien nicht erfüllt waren. Darüber hinaus konnte die Klausur auch nicht die Lernzuwächse von Raphael und Torsten in Bezug auf die Implementierungsfähigkeit adäquat erfassen, da das Ergebnis beider Schüler bereits im Eingangstest nah an der maximalen Punktzahl lag.

Die Auswertung der Klausur bestätigte darüber hinaus einige Erwartungen, die sich mir schon während der Durchführung der Sequenz erschlossen. So konnte Sebastian, der in seinem Projekt nur wenig objektorientierte Strukturen verwendete, die allgemein gefasste Strukturierungsaufgabe (s. Anhang S. 50) nicht befriedigend lösen. Aufgrund seiner hohen Programmierkenntnis hatte er im Unterricht meistens „aus dem Kopf“ gearbeitet und sich deshalb nicht ausreichend mit den Planungsmethoden beschäftigt. Im Portfolio fiel das nicht auf, da er trotz allem seine Probleme und die entsprechenden Lösungen gut darstellen konnte.

Liane hatte für ihre Verhältnisse in der kurzen Zeit gute Fortschritte gemacht. Sie konnte die Implementierungsaufgabe für die Greenfoot-Umgebung gut bewältigen, versagte aber ebenfalls bei der Strukturierungsaufgabe.

4.5) Analyse und Alternativen

Das wohl interessanteste beobachtete Phänomen in dieser Unterrichtssequenz gibt Aufschluss über das Verhältnis der Schüler zum eigenen Lernen. Die Portfolioarbeit zielte auf die bewusste Analyse und Dokumentation des eigenen Lernprozesses. Für die Schüler in meinem Basiskurs war dies offenbar sehr ungewohnt. Für sie zählte nur das fertige Produkt ihrer Arbeiten, wie ich schon in der Darstellung der Beratungsgespräche feststellte. Ich führe dieses Verhalten auf den meiner Meinung nach generell geringen Anteil an Reflexionen des eigenen Lernens im allgemeinen Schulunterricht zurück. Es ist allgemein bekannt, dass hauptsächlich Produkte und erreichte Leistungen bewertet werden, eher als die Wege, die dorthin geführt haben. Dies könnte nach meinem Empfinden der Grund dafür sein, dass meine Schüler die einzelnen Stationen ihres Lernens als trivial oder unwichtig empfanden und sich größtenteils darauf konzentrierten, ihre Produkte möglichst detailliert zu beschreiben.

Diese starke Produktorientierung ist offensichtlich kontraproduktiv für die Erstellung von prozessorientierten Portfolios. Mein ständiges Bestreben, die Schüler wenigstens zum Aufzeichnen ihrer Lernschritte zu bewegen, erzeugte zusätzlich eher Antipathien gegenüber dieser Arbeit: Das war für mich am interessantesten, für meine Schüler jedoch eine zusätzliche Belastung, die ihnen Zeit stahl, die sie für die Bearbeitung ihrer Programme hätten einsetzen können.

Um mir die verinnerlichte Produktorientierung der Schüler zunutze zu machen, hätte ich stärker auf das prozessorientierte Portfolio als eigentliches bzw. als wichtigeres Produkt der Unterrichtseinheit hinweisen können. Dies hätte jedoch wiederum einen Motivationseinbruch verursachen können, denn für die Schüler war es wichtig, an ihren Programmen zu arbeiten und damit voranzukommen. Deshalb war es auch wichtig, am Schluss der Sequenz beides zu präsentieren: Portfolio und Programm.

Ein weiterer Diskussionspunkt ist der Einsatz des Content-Management-Systems zur Erstellung und Pflege der Portfolios. Einerseits bot das System jenen Schülern, die bereits gut mit HTML umgehen konnten, eine ausgezeichnete Basis zur kreativen Gestaltung ihrer Portfolios. Den weniger mit HTML vertrauten Schülern gelang die Einarbeitung ebenfalls schnell genug, um am Ende der Sequenz ein fertiges Portfolio zu haben. Dennoch wurde mehrmals die Frage gestellt, ob nicht ein handschriftliches Portfolio ausreichen würde, da dies effizienter wäre.

Dennoch denke ich, dass im Hinblick auf eine hochschul- und berufsorientierte Laufbahn in der Informatik der Einsatz eines digitalen Dokumentationssystems gerechtfertigt ist und geübt

werden sollte. Die Schüler erhalten so auch ein Gefühl für die Effizienz eines solchen Systems und können später entscheiden, ob und wann sie selbst ein solches System einsetzen.

Interessanterweise machten die Schüler von der Möglichkeit, die Portfolios von ihrem Heimarbeitsplatz aus zu bearbeiten, wenig Gebrauch. Auch hier griffen sie häufiger auf die Kollaborationsplattform SVN zu als auf die Webseite, mit der sie die Portfolios erstellten. Dies geht aus dem Vergleich der Aktivitätsaufzeichnungen der beiden Plattformen hervor. Deutlich erkenne ich daran, dass die Schüler hauptsächlich im Unterricht auf die jeweiligen Plattformen zugegriffen, außerhalb des Unterrichts jedoch häufiger auf den SVN-Bereich. Erst zum Schluss erhöht sich die Aktivität auf der Portfolio-Webseite stark. Ich begründe dieses Verhalten mit dem Fehlen von obligatorischen „Hausaufgaben“. Ganz offenbar habe ich hier die Fähigkeiten meiner Schüler zur Selbstorganisation unterschätzt und demnach hätte ich – genau wie die Schüler für ihre Programme – Meilensteine für den Fertigstellungsgrad der Portfolios festlegen müssen. Davon habe ich während der Einheit aus dem Grund abgesehen, um den Schülern am Ende der Reihe bewusst machen zu können, welche Folgen ihr Verhalten in dieser Hinsicht hatte: Sie mussten in der letzten Phase große Anstrengungen unternehmen, um ihre Portfolios noch rechtzeitig fertig zu stellen und hatten teilweise zu wenig Material im Verlauf ihrer Arbeit gesammelt.

Problematisch an der Unterrichtssequenz war auch die zu kurze Zeitspanne, in welcher die Portfolioarbeit durchgeführt wurde. Daher verringerte sich die Auswahl der Themen, welche die Schüler in ihrem Portfolio behandeln konnten. Mit der Festlegung der zu implementierenden Funktionen bzw. der Arbeitsteilung innerhalb der Gruppen änderte sich die persönliche Aufgabenstellung nur geringfügig. Da die Schüler immer innerhalb der selbst gesteckten Grenzen auf die Fertigstellung ihres Programms oder Programmteils hin arbeiteten, empfanden sie auch die Darstellung ihrer Arbeit in den Portfolios als einseitig, da eben nur die Erarbeitung eines einzelnen großen Themas stattfand.

Ein Beispiel hierzu ist Christophs Portfolio. Es handelte thematisch von der Erstellung einer grafischen Benutzeroberfläche mithilfe der Java-Schnittstelle *Swing*, denn die Einarbeitung in *Swing* und die Erstellung der Oberfläche waren seine Aufgaben innerhalb der Gruppe. Sein Portfolio entsprach auch genau meinen Vorgaben: es stellte die Schritte dar, die er vom ersten „leeren Fenster“ zu einer kompletten Oberfläche mit Schaltflächen, Eingabefeldern usw. gegangen war.

Demgegenüber bietet die Arbeit mit Portfolios viel mehr Möglichkeiten. So sollte nicht nur ein Thema zur Auswahl stehen, sondern die Schüler sollten zwischen verschiedenen Themen

auswählen können. Dies schafft neben einem größeren Anreiz auch die Möglichkeit zum Ausgleich, falls das eine oder andere Thema einen Schüler weniger begeistern konnte bzw. er oder sie bei der Bearbeitung eine geringere Leistung gezeigt hat als sonst. Dies war in der vorliegenden Unterrichtssequenz nicht möglich.

Verschiedene Themenbereiche können jedoch meiner Ansicht nach nur angeboten werden, wenn die Methode Portfolioarbeit sequenzübergreifend eingesetzt wird. Generell bezieht sich auch die Fachliteratur meist auf die Anwendung der Methode über den Zeitraum von mindestens einem Halbjahr bis hin zu mehreren Schuljahren (vgl. z.B. Kolb 2007, S. 122 f.). Die Verlängerung des Einsatzes hätte nicht nur einen positiven Effekt auf die Diversität der Themen, sondern würde auch das Verhältnis der für die Betreuung der Portfolios aufgewendeten Zeit zur eigentlichen Unterrichtszeit verbessern.

Zuletzt möchte ich den Einsatz einer Klausur als Bewertungsinstrument einer Unterrichtsreihe mit individueller Themenstellung generell in Frage stellen. Auf der einen Seite konnte ich mit der Klausur überwiegend nur die Fähigkeit der Algorithmisierung überprüfen. Dies begründet sich in der Vorgabe der Verteilung der Anforderungsniveaus in einer Klausur. Der größte Teil der Klausuraufgaben muss den Anforderungsbereichen I und II entsprechen. Eine zu lösende Problemstellung entsteht nach Schubert und Schwill (vgl. Schubert/Schwill 2004, S. 104 f.) jedoch auch immer durch das Fehlen einiger zur Lösung notwendiger Informationen, die erst recherchiert oder aus den gegebenen Informationen abgeleitet werden müssen. Dies entspricht jedoch der Definition des Anforderungsbereichs III, der nur einen vergleichbar kleinen Teil in einer Klausur stellen darf.

Dazu kam noch, wie bereits erwähnt, dass die Leistungsanforderung der einzelnen Aufgaben ungerecht verteilt war, da die im Unterricht eingesetzte Binnendifferenzierung auch gerechterweise in der Klausur fortgeführt werden musste. An dieser Stelle entstehen grundsätzlich (auch bei meinen Schülern) Diskussionen, ob eine solche Mischung leichter und schwerer Aufgaben für schwache und starke Schüler statthaft oder in sich gerecht ist.

Ich führe im Folgenden zwei Alternativen an um diese vermeintliche Ungerechtigkeit der Leistungsbewertung zu umgehen bzw. aufzuheben.

Zunächst hätte ich die Programme und dazugehörigen Präsentationen als schriftliche Leistung bewerten können. Hierin investierten die Schüler die meiste Energie und den größten Aufwand, also wäre anzunehmen, dass dies auch die eigentliche Leistung sei, deren Bewertung sie am ehesten erwarten und akzeptieren würden. Die gefühlte Ungerechtigkeit der Binnendif-

ferenzierung wäre zwar nicht verhindert worden, doch man hätte sie besser erklären können: Ein einfaches Programm von Liane beispielsweise kann, sollte es den Ansprüchen an die Entwicklung der Fähigkeiten und der Dokumentation genügen, ebenso mit sehr gut bewertet werden wie ein viel umfangreicheres Programm von Torsten. Die Unterschiede in der Leistungsbewertung könnten dann anhand der festgelegten Kriterien für Entwicklung und Dokumentation begründet werden. Dem entgegen steht allerdings die Fokussierung meiner Unterrichtsreihe auf den Lernprozess. Hätte ich die Programme und Präsentationen von vornherein als Bewertungsgrundlage festgelegt, so hätte dies die Motivation zur Unterhaltung der Portfolios noch weiter gemindert.

Die bessere Alternative wäre es nach meiner Auffassung, das Portfolio selbst als schriftliche Prüfungsleistung des Halbjahres einzusetzen. Das Schulrecht erlaubt die Erbringung der schriftlichen Teilleistung in der Oberstufe auch durch andere Prüfungsformen anstatt einer Klausur. Ich habe diese Möglichkeit während der Reihe nicht in Betracht gezogen, da ich aufgrund der späten Fertigstellung der Portfolios durch die Schüler die Gefahr sah, dass die guten Leistungen einiger Schüler durch ein fehlendes Portfolio nicht entsprechend bewertet würden. Schließlich war auch die Frage nach der allgemeinen Problemlösefähigkeit der Schüler zu beantworten, die sich meiner Meinung nach nur durch die Stellung eines „neuen“ Problems effektiv bewerten ließe.

4.6) Überprüfung der Hypothesen

4.6.1) Binnendifferenzierung

Das wichtigste Merkmal der Binnendifferenzierung ist die Abstufung des Schwierigkeitsgrades nach individuellem Leistungsstand der Schüler. Die Abstufung wurde im Groben durch die individuelle Themenwahl der Implementierungen realisiert. Im Detail haben die Schüler sich während der Bearbeitung der Themen selbstständig Problemstellungen erarbeitet, die sie lösen konnten. Dabei knüpften sie an bereits Bekanntes an und erarbeiteten sich neue Begriffe durch Recherche (z.B. in Online-Materialien oder Java-Referenzbüchern).

Wie in der Beschreibung der Schülerarbeiten bereits dargestellt, suchten sich die Schüler Themen, für die sie von sich aus Ideen zur Bearbeitung entwickeln konnten. Daraus entnehme ich, dass sie selbst eine gesunde Einschätzung ihrer Fähigkeiten vornehmen konnten. Leider lässt sich daran nicht mehr ablesen, ob ich von mir aus eine innere Abstufung der Leistungsanforderungen vornehmen kann, die zu den Schülern passt. Dennoch wird auch durch die bereits beschriebene exklusive Verteilung der Aufgaben innerhalb der Arbeitsgruppen klar, dass

in dieser Sequenz stark binnendifferenziert gearbeitet wurde.

Als Beispiel weise ich an dieser Stelle noch einmal auf die Schüler Max, Alexej und Peter hin (s. Seite 29). Alexej, der im Eingangstest schon gut abschnitt, suchte sich die Beschäftigung mit den komplexen Zusammenhängen der Physiksimulation im Kampfspiel aus. Max hingegen, der weniger gut implementieren konnte, konzentrierte sich auf sequentielle Abläufe des Spielgeschehens und musste sich somit auch nicht mit der Grafikausgabe beschäftigen, die Peter bearbeitete. Eine ähnliche Situation herrschte in den anderen Arbeitsgruppen bzw. bei den einzeln arbeitenden Schülern.

Damit sehe ich es als belegt an, dass durch die Verwendung der Methode Portfolioarbeit im Informatikunterricht eine Binnendifferenzierung realisiert werden kann.

4.6.2) Betreuung

Eine gute individuelle Betreuung kam nach meinem Befinden in meinem Unterricht nicht zustande. Dies hatte zwei wesentliche Gründe.

Einerseits habe ich die Produktivität meiner Schüler unterschätzt. Nahe dem Ende der Unterrichtseinheit umfassten alle von den Schülern erarbeiteten Programme zusammengenommen mehr als 1000 Zeilen Quelltext. Das sind mehr als 30 DIN-A4-Seiten, die gleichzeitig an verschiedenen Stellen von zehn Personen verändert werden. Hier den Überblick zu behalten hieße, jeden Tag einen Taschenroman auf geänderte Textstellen zu überfliegen. Darüber hinaus ist Quelltext ein nicht-linearer Text, in dem geringste Änderungen weitreichende Folgen für den Rest des Programms haben können.

Hierzu ein Beispiel: Raphael arbeitete für seinen Graphenzeichner mehrere Tagen an der Transformation des virtuellen Koordinatensystems in den Koordinatenraum des Bildschirms um den darin gezeichneten Graphen in beliebiger Skalierung anzeigen zu können. Das Programm gab zwar eine Darstellung aus, doch es war auf den ersten Blick zu erkennen, dass die ausgegebenen Graphen nicht mit den eingegebenen Funktionen übereinstimmten. Wir arbeiteten gemeinsam in einer Freistunde am Programm – im Unterricht selbst hatte ich nicht genügend Zeit – und fanden nach einer halben Stunde heraus, dass an einer Stelle im Programm die Transformation doppelt vorgenommen und deshalb das Ergebnis verfälscht wurde.

Da bei Raphael wie bei den anderen Schülern eine umgangssprachliche Dokumentation des Quelltextes fehlte, konnte der Fehler nur durch schrittweises Durchgehen des Quelltextes gefunden werden: eine zeitaufwändige Methode, aber dennoch fand hier im Gespräch mit dem

Schüler echte individuelle Betreuung statt. Im Unterricht selbst habe ich mir für solche Gespräche maximal fünf Minuten pro Fall gestattet, um die anderen Schüler nicht zu benachteiligen. Von dieser schon knappen Zeit musste ich noch einen Teil für die Behandlung technischer Probleme aufwenden (z.B. überfüllte Benutzerverzeichnisse, Installation und Konfiguration der Anwendungsumgebung etc.). Für elf Lerngruppenmitglieder war dieses Verfahren also nur teilweise ausreichend, für größere Lerngruppen sehe ich diese Art von Betreuung unter den dargestellten Voraussetzungen als gänzlich unpraktikabel an.

4.6.3) Lernzuwachs und Organisationsaufwand

Den Lernzuwachs beurteile ich danach, inwieweit die Schüler in der Lage sind, Probleme mithilfe dem Problem gerechter Implementierungen zu lösen und dabei erweiterte

Nach der Messung des Lernzuwachses zu urteilen, wurde die angestrebte Aneignung und Verwendung erweiterter objektorientierter Konzepte wie z.B. abstrakter Datentypen, Aggregationen oder Vererbung bei einigen Schülern nicht vollständig erreicht. Dies sehe ich durch mehrere Gründe verursacht.

Zunächst stellt sich die Wahlfreiheit der Themen als problematisch dar. Ich konnte nicht sicherstellen, dass allein die „Überschrift“ des Themas als Garant diente, dass sich dahinter auch wirklich Problemstellungen verbargen, die sich unter Verwendung objektorientierter Konzepte lösen ließen. Auch die begleitende Betreuung war aufgrund der in Abschnitt 4.6.2 erwähnten Probleme nicht ausreichend, um entsprechend auf die angestrebte Verwendung der Konzepte hinzuarbeiten, jedenfalls nicht bei allen Schülern. Im schlimmsten Fall ergibt sich hier eine enorme Ungerechtigkeit: Schüler, die Beratungsangebote annehmen oder einfordern und entsprechend geleitet werden, haben viel bessere Chancen auf einen guten Abschluss, als Schüler, die lange Zeit nur für sich bleiben und still arbeiten. Sollten sich die stillen Schüler zwar exzellent mit ihrem Thema auseinandersetzen, aber an der übergeordneten Zielsetzung vorbei arbeiten, so entsteht eine Winner/Loser-Situation, in der die Leistung des Schülers nichts mehr zählt. Solch eine Situation ist auf jeden Fall zu vermeiden. In der vorliegenden Sequenz war Sebastian zeitweilig in der Gefahr, vollends an der Objektorientierung vorbei zu arbeiten und somit die Klausuranforderungen nicht zu schaffen.

Des Weiteren war die Arbeitsbelastung der Schüler durch die gleichzeitige Bearbeitung von Portfolio und Programm sehr hoch. Sie mussten viele verschiedene Werkzeuge handhaben (z.B. Browser, CMS, SVN, BlueJ, Greenfoot, Eclipse, ZIP, Zeichenprogramm etc.), um ein ansehnliches Ergebnis zu erhalten. Die Knappheit der zur Verfügung stehenden Zeit ver-

schlimmerte diesen Zustand zusätzlich, da für einige Schüler z.B. die Handhabung der Webseite schwieriger war als angenommen.

Durch die Auswertung der Klausur zeigt sich außerdem, dass die Schüler zwar kompetenter geworden sind, was den Entwurf von Algorithmen zu gegebenen Aufgabenstellungen angeht. Dennoch bereitet die Umsetzung der Lösung mit objektorientierter Modellierung einigen immer noch Schwierigkeiten. Ich weiß, dass es in jeder Lerngruppe eine statistische Verteilung der Leistungsfähigkeit um einen bestimmten Punkt herum gibt. Dennoch ging ich davon aus, dass nach dieser Unterrichtssequenz zumindest die Modellierung einfacher Strukturen bei allen Schülern generell verfügbar sein sollte. Offenbar fehlt hier doch der ständige Umgang mit Modellierungsszenarien eher als das Umsetzen der Modelle einer Programmiersprache.

5) Gesamtreflexion und Ausblick

Insgesamt bin ich mit dem Ablauf der Reihe zufrieden. Wichtig war mir, die Portfolioarbeit als Möglichkeit zur Durchführung von binnendifferenziertem Informatikunterricht im Hinblick auf das selbstorganisierte Lösen von Problemen zu evaluieren. Die Chancen sowie die Nachteile der Portfolioarbeit wurden dabei deutlich sichtbar. Trotz aller Nachteile haben die Schüler größtenteils den von mir angestrebten Lernzuwachs erreicht und nebenbei viel über die eigene Einstellung zu ihren Leistungen und ihren Lernmethoden gelernt.

Beeindruckt hat mich die Leistungsbereitschaft und Kreativität meiner Schüler. Ich wusste bereits, dass Schülern immer die Möglichkeit zur kreativen Ausgestaltung ihrer Arbeiten gegeben werden sollte. Doch zu sehen, welche Vielfalt entstehen kann, wenn man ihnen in einem weiten Rahmen freie Hand lässt, bestärkt mich darin, weiterhin den Schülern Freiräume für eigene Entscheidungen und persönliche Entfaltung einzuräumen.

Ich sehe aber auch, dass die Schüler viel öfter und längerfristiger zur Reflexion des eigenen Lernens angeregt und angeleitet werden müssen. Wie auch Kolb bemerkt (vg. Kolb 2007, S. 42), ist der Aufbau einer „Kultur des Nachdenkes über das eigene Lernen“ ein komplexer und langwieriger Prozess. Deshalb halte ich es für wichtig, diesen Prozess immer öfter nicht implizit bei den Schülern als gegeben vorauszusetzen, sondern ihn zum Thema des Unterrichts zu machen. Neue, allgemein verfügbare Informationsquellen erfordern selbstverständlich auch neue Methoden der Wissensaneignung (etwa kollaborativen Wissenserwerb auf Web-2.0-Plattformen) und damit auch bewusst gewordene und ständig erweiterte Strukturen des eigenen Lernens.

Deutlich wurde mir auch, dass Formen offenen Unterrichts im Vorfeld viel mehr Vorbereitung benötigen als die Durchführung einer Sequenz, deren Thema und zu erstellende Schülerarbeiten schon vorher festgelegt sind. Man muss den Schülern eine ihren Interessen entsprechende Auswahl an Materialien schon zu Beginn der Reihe zur Verfügung stellen. Diese Materialien müssen natürlich erstellt und möglichst auf ihre Tauglichkeit bzw. Funktionalität überprüft worden sein. Nicht zuletzt muss man während der Reihe auf alle Eventualitäten vorbereitet sein und bei unvorhergesehen auftretenden Problemen schnell handeln können. Somit sehe ich beispielsweise Schwierigkeiten an Schulen, an denen der Rechnerbetrieb von einer externen Firma betreut wird und der Lehrer selbst keine Softwareinstallationen durchführen kann oder keinen Zugriff auf die Administration hat.

Bei der Planung der Reihe rechnete ich damit, dass es Fälle geben könnte, in denen Schüler vorgeben, schwer überprüfbare Aktivitäten durchgeführt zu haben (z.B. „Ich habe schon jede Menge Internetseiten zum Thema gelesen, bin auch schon beim Programmieren, aber da ist noch nichts fertig.“) und war erstaunt, dass die Schüler mit großem Enthusiasmus auch kleine Fortschritte innerhalb der persönlichen Betreuung zeigen wollten. Umso bedauerlicher ist ihre Abneigung gegen das Aufschreiben der eigenen Leistungen, da diese dennoch vorlagen. Hier auf muss ich in Zukunft nicht nur im Informatikunterricht besonderen Wert legen, da die Fähigkeit zur Reflexion des eigenen Lern- und Arbeitsprozesses von größter Bedeutung für den zukünftigen Weg der Schüler in Hochschul- und Berufswelt sein wird. Darüber hinaus würde ich in kommenden Sequenzen mit Projektarbeit die Bildung von Arbeitsgruppen erzwingen, da die großen Vorteile der Gruppenarbeit im Hinblick auf das Üben von Kommunikation und Kooperation in dieser Sequenz deutlich waren.

Der Einsatz der Portfolioarbeit ist nach wie vor eine interessante Option für stark leistungsdifferente Lerngruppen. Natürlich existieren noch viele Bereiche, in denen ich diese Methode für meinen Unterricht verfeinern kann, wie z.B. die Themenauswahl, der zeitliche Rahmen, die Bewertung der Leistungen und so weiter.

Wichtig ist es – wie ich bereits in der Analyse erwähnte – die Portfolioarbeit über einen längeren Zeitraum als nur einige Wochen zu betreiben. Des weiteren sollten den Schülern konsequente Meilensteine gesetzt werden, an denen eine Überprüfung ihrer Arbeiten erfolgt. Durch die Verlängerung der Einsatzzeit wird außerdem die Balance zwischen den Arbeitsergebnissen der Schüler an sich (Aufgabenlösungen, Programmierideen etc.) und dem Portfolio wiederhergestellt, so dass das Portfolio eher als Begleitinstrument denn als vordergründige Bewertungsgrundlage gesehen werden kann. Wichtig ist es dabei, die Funktion des Portfolios als in-

dividuell gestaltete Entwicklungsmappe gegenüber dem Schülerhefter abzugrenzen, um das Gefühl der Doppelbelastung bei den Schülern zu vermeiden. Eventuell sind noch weitreichendere Einschnitte in den herkömmlichen Unterricht zu betrachten:

- Kann die Portfolioarbeit als ständige Hausaufgabe gelten?
- Wie muss der Unterricht als Ganzes strukturiert sein, damit vielfältige Schülerarbeiten entstehen, die ins Portfolio integriert werden können?
- Reicht die in dieser Arbeit vorgestellte Verknüpfung von Portfolioarbeit und Offenem Unterricht aus, um beispielsweise den Erwerb von Grundkenntnissen sicherzustellen?

In diesem Sinne schließe ich diese Ausführung und freue mich darauf, in Zukunft andere Methoden im Informatikunterricht auf ihre Eignung zu überprüfen und somit einen modernen, schülerorientierten Informatikunterricht mit aktuellen Inhalten und Strukturen durchzuführen.

6) Literaturverzeichnis

Didaktik:

- Baumann, Rüdiger: *Didaktik der Informatik*, Klett 1990
- Hubwieser, Peter: *Didaktik der Informatik*, Springer 2004
- Humbert, Ludger: *Didaktik der Informatik*, Teubner 2005
- Meyer, Hilbert: *Unterrichtsmethoden, I: Theorieband*, Cornelsen 1994
- Schwill, Andreas u. Schubert, Sigrid: *Didaktik der Informatik*, Spektrum 2004

Methode „Portfolioarbeit“:

- Häcker, Thomas: *Das Handbuch Portfolioarbeit: Konzepte - Anregungen - Erfahrungen aus Schule und Lehrerbildung*, Kallmeyer 2006
- Kolb, Annika: *Portfolioarbeit*, Gunter Narr Verlag 2007

Weitere Literatur:

- Senatsverwaltung für Bildung, Wissenschaft und Forschung in Berlin: *Rahmenlehrplan Informatik für die gymnasiale Oberstufe*, 2006
- Bebis 2008: Berliner Bildungsserver: *Java mit BlueJ*
URL: http://www.bebis.de/themen/faecher/informatik/programmiersprachen/Java_mit_BlueJ/Greenfoot/, erstellt am 18.3.2007, zuletzt geprüft am 11.9.2008.

Verwendete Software (frei und/oder Open Source):

- Bluej: www.bluej.com
- Greenfoot: www.greenfoot.org
- Subversion (SVN): subversion.tigris.org
- Drupal: www.drupal.org
- Eclipse, PHP, MySQL

7) Anhang

Fragebogen zur Bewertung der Portfolios

Basiskurs Informatik – Portfolio-Bewertungsbogen

Name des Portfolio-Autors/der Portfolio-Autorin: _____

Name des Bewerter: _____

Problemstellung (Thema): _____

Bewertungsteil	1	2	3	4
----------------	---	---	---	---

A) Form

- | | | | | |
|--|--------------------------|--------------------------|--------------------------|--------------------------|
| 1) <i>Schreibstil</i> (1 = sachlich, neutral; 4 = Slang) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2) <i>Wurden verschiedene Medien sinnvoll eingesetzt</i> (etwa zur Verdeutlichung komplizierter Ideen)?
(1 = sinnvoll, 4 = nur Text oder zuviel Überflüssiges) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3) <i>Inhaltsverzeichnis</i> (1 = übersichtlich, 4 = nicht vorhanden) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4) <i>Ist der Seitenaufbau übersichtlich und durch verschiedene Schriftstile gegliedert?</i> (1 = übersichtlich, 4 = kreuz und quer) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5) <i>Sind die einzelnen Seiten sinnvoll miteinander verknüpft</i> (z.B. mit Verweisen auf die nächste und die vorhergehende Seite)?
(1 = Navigationsleiste o.ä., 4 = unsinnige oder keine Links) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 6) <i>Sind die Seiten nach einem klar erkennbaren Merkmal angeordnet</i> (z.B. zeitliche Abfolge, Thema usw.)?
(1 = deutlich sichtbar, 4 = unsortiert) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

B) Inhalt

- | | | | | |
|---|--------------------------|--------------------------|--------------------------|--------------------------|
| 7) <i>Sind die neuen Erkenntnisse des Autors/der Autorin dargestellt?</i>
(1 = deutlich sichtbar, 4 = nicht erkennbar bzw. vorhanden) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 8) <i>Wurden Arbeitsergebnisse integriert</i> (Skizzen, Diagramme, etc.)?
(1 = mehr als fünf Bilder, Codeschnipsel etc., 4 = keine) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9) <i>Hat der Autor/die Autorin die von ihm/ihr bearbeiteten Probleme und deren Lösungen dargestellt?</i> (1 = deutlich sichtbar, 4 = nicht erkennbar) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 10) <i>Wie gut lassen sich die objektorientierten Prinzipien erkennen, nach denen der Autor/die Autorin gearbeitet hat?</i> (1 = deutlich, 4 = nicht vorh.) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

C) **Summe:** _____ (Wertung: 10 = sehr gut, 40 = mangelhaft)

Test zur Erhebung der Strukturierungs- und Implementierungsfähigkeit

Aufgabe 1)

Der Betreiber einer Bibliothekenkette möchte digital kodierte Kundenkarten ausgeben, um die Diebstähle zu reduzieren. Deshalb müssen sich alle Kunden beim Betreten der Bibliothek anmelden und beim Verlassen wieder abmelden. Kunden können Bücher ausleihen oder abgeben, wenn sie angemeldet sind. Die Kunden müssen bei Beantragung der Nutzungserlaubnis ihren Namen und ihren Geburtstag angeben. Die in der Bibliothek vorhandenen Bücher sind im alten Katalog mit Titel, Autor und ISBN-Nummer verzeichnet.

Aufgabe: Entwickeln Sie ein Klassendiagramm für ein Programm, welches die Verwaltung der Kunden und Bücher übernehmen kann! Geben Sie für jede Klasse, die sie anlegen, die Spezifikation an!

[Erwartetes Diagramm:]

Kunde	Buch	Bibliothek
Attribute - name : String - geburtstag : int - nummer : int	Attribute - titel : String - autor : String - isbn : int	Attribute - buecher : Buch[] - kunden : Kunde[]
Methoden + nameSetzen (String) + adresseSetzen (String) + anmelden () + abmelden () + nummerHolen () : int ...	Methoden + ausleihen () + abgeben () + istAusgeliehen () : boolean + titelSetzen (String) + titelHolen () : String ...	Methoden + kundeRegistrieren (Kunde) + kundeAnmelden (Kunde) + kundeAbmelden (Kunde) + buchAusleihen (Buch) + buchAbgeben (Buch) ...

Aufgabe 2)

Die folgenden Spezifikationen beschreiben Klassen aus dem Programm für ein Fahrzeug-Navigationssystem. Implementieren Sie die Klassen in Java! Die Namen der Methoden geben die jeweilige Funktion an.

Fahrzeug	Strasse	Position
Attribute - geschwindigkeit : float - farbe : Color - typ : String - aktPos : Position	Attribute - name : String - laenge : float	Attribute - laenge : int - hoehe : int - breite : int
Methoden + geschwindErhoehen (float) + geschwindVerringern (float) + strasseSetzen (Strasse) + positionSetzen (Position)	Methoden + nameHolen () : String + laengeHolen () : float	Methoden + setzePosition (int,int,int) + hoeheHolen () : int + laengeHolen () : int + breiteHolen () : int

Ausschnitte aus der Klausur

Datum: 13. Juni 2008

Informatik Basiskurs

Klausur im Zweiten Halbjahr 2007/08

Name: _____

Teil A/4: Objektorientiertes Programmieren mit der **Zeichenfläche**

Aufgabe 1:

- a) Benennen Sie drei Methoden der Klasse Zeichenfläche und beschreiben Sie kurz deren Funktion! ___/3P
- b) Erläutern Sie die Anforderungen an eine Klasse, die in die Zeichenfläche eingefügt und gezeichnet werden soll! ___/3P

Aufgabe 2:

Implementieren Sie eine Klasse Ball, die in die Zeichenfläche eingefügt werden kann und als roter Kreis dargestellt wird. ___/7P

Aufgabe 3:

Die Zeichenfläche besitzt eine Methode `istTasteGedrueckt(int tastenCode)`. Schreiben Sie einen kurzen Programmteil in Java, der abfragt, ob verschiedene Tasten gerade gedrückt werden.

Konstanten für Tastencodes der Cursortasten:
`VK_DOWN`, `VK_UP`, `VK_LEFT`, `VK_RIGHT`

___/7P

Aufgabe 4:

Die aktuelle Version der Zeichenfläche kann keine Kollisionsprüfung zwischen den einzelnen Zeichenobjekten ausführen. Schlagen Sie eine Erweiterung der Klasse vor, so dass die Kollisionsprüfung dort ausgeführt werden kann.

Sie können den Vorschlag sowohl umgangssprachlich ausführen, als auch Skizzen oder Spezifikationen angeben. Sie müssen nichts implementieren!

___/4P

Teil B: Lernfortschritt

Aufgabe:

Rufen Sie sich Ihr Projekt in Erinnerung, das Sie in den letzten sechs Wochen bearbeitet haben. Entscheiden Sie sich nun, welches das größte, wichtigste oder interessanteste Problem war, das Sie während des Projektes gelöst haben. Schreiben Sie eine kurze Besinnung (Aufsatz) zu dem ausgesuchten Problem aus ihrem Projekt. Dies soll mir einen Eindruck darüber geben, ob Sie in der Lage sind, über Ihre Arbeitsprozesse strukturiert nachzudenken.

Ablauf:

- a) Machen Sie sich Stichpunkte zum Problem: Ausgangslage, Fragestellung, Ideen, Lösungsansätze, passende Lösung, etc. Die Stichpunkte ziehe ich (bei knapper Zeit) auch in die Bewertung mit ein.
- b) Formulieren Sie einen kurzen Aufsatz (keine Stichpunkte) zum Ablauf der Problemlösung und zeigen Sie darin auf, wie Sie an das Problem herangegangen sind, welche Schritte Sie gewählt haben und welche Idee schlussendlich zur Lösung geführt hat. Der Aufsatz soll mind. 100 Wörter umfassen. Achten Sie bitte auf Folgendes: klare Strukturierung, zeitliche Abfolge und präzise Formulierungen! ___/12P

Fragen zur Unterstützung der Interviews

Fragen zum Projekt

- Was tragen Sie zu Ihrem Projekt bei? Wie sind die Aufgaben innerhalb der Projektgruppe verteilt?
- Welches Ziel verfolgten Sie beim Erstellen des Klassendiagramms?
- Beschreiben Sie ein aktuelles Problem, an dem Sie arbeiten? Gibt es eine Stelle, an der Sie bisher nicht weitergekommen sind?

Fragen zum Portfolio

- Welches Thema bearbeiten Sie? Welchen Titel soll Ihr Portfolio haben?
- Welche Struktur bzw. Einteilung soll ihr Portfolio haben?
- Welche anderen Medien außer Text wollen Sie in Ihr Portfolio integrieren?
- Beschreiben Sie aus ihrer Sicht den Sinn des Portfolios!
- Welche Vor- und Nachteile sehen Sie?

Erklärung

Hiermit versichere ich, dass ich die vorliegende Prüfungsarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Berlin, den 19. September 2008

Lars-Christian Pelz