

Humboldt-Universität zu Berlin

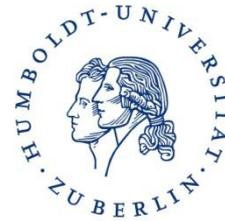
Mathematisch-Naturwissenschaftliche Fakultät II

Institut für Informatik

Seminar: Schulpraktische Studien - Nachbereitung

Dozent: Dr. rer. nat. Nguyen-Thanh Le

Semester: WS 2014 / 2015



Praktikumsbericht

zum Unterrichtspraktikum im Fach Informatik am Humboldt-Gymnasium
in Berlin-Tegel

07.11.2014

Kevin Rimek

E – Mail: randco@gmx.net

Master of Education: 3. Fachsemester

Kernfach: Informatik

Zweifach: Geschichte

Inhaltsverzeichnis

1. Überblicksdarstellung über die am Humboldt-Gymnasium angebotenen Informatikkurse und deren Inhalte	1
1.1 Informatiknahe Unterrichtsfächer in der Sekundarstufe I	1
1.2 Schulinternes Gesamtcurriculum und Profilierungskurs Informatik	2
1.3 Informatik in der Sekundarstufe II	3
2. Stundenentwurf zum Thema: „Realisierung eines Dateikopierers in C#“	5
2.1 Erläuterungen zum Thema der Unterrichtsreihe	5
2.1.1 Grobziele der Unterrichtsreihe	6
2.1.2 Begründung der Themenwahl für die Unterrichtsreihe	7
2.1.3 Struktur der Unterrichtsreihe	9
2.2 Aufgabenstellung der Stunde und Aufgabenanalyse	10
2.2.1 Ausformulierung der Aufgabenstellung	10
2.2.2 Mit der Aufgabenstellung verbundene Lernziele und Aufgabenlösung	11
2.2.3 Bestimmung der Arbeitsschritte	15
2.3 Bedingungsanalyse	16
2.3.1 Richtlinien- und Fachvorgaben	16
2.3.2 Lernvoraussetzungen der Schülerinnen und Schüler	19
2.3.3 Lehrvoraussetzungen	26
2.3.4 Institutionelle Rahmenvorgaben, materielle Lernvoraussetzungen und Raumausstattung	27
2.4 Lernziele	27
2.4.1 Durch die Unterrichtsstunde berührte Grobziele der Reihe	28
2.4.2 Aus den Grobzielen abgeleitete Feinziele der Unterrichtsstunde	29
2.5 Didaktische Strukturierung	32
2.6 Tabellarischer Stundenverlaufsplan	57
2.7 Vorüberlegungen zur Unterrichtsauswertung	61
3. Reflexion der Unterrichtsstunde	63
3.1 Besonderheiten beim realen Stundenverlauf gegenüber dem geplanten Verlauf	63
3.2. Fazit zur Stunde – Was habe ich gelernt?	67
4. Hospitationsprotokolle	70
4.1 Hospitationsprotokoll I	70
4.2 Hospitationsprotokoll II	75
5. Reflexion über das Praktikum – Was habe ich gelernt?	80

6. Stundenübersicht	82
6.1 Übersicht über die Hospitationsstunden.....	82
6.2 Außerschulische Aktivitäten	83
6.3 Übersicht über die selbst gehaltenen Stunden	83
Anhang 1:.....	84
Anhang 2:.....	86
Anhang 3:.....	89
Anhang 4:.....	90
Anhang 5:.....	92
Anhang 6:.....	94
Anhang 7:.....	95
Literaturverzeichnis	96

1. Überblicksdarstellung über die am Humboldt-Gymnasium angebotenen Informatikkurse und deren Inhalte

In diesem Abschnitt soll ein Überblick über die schulinternen Rahmenbedingungen und Richtlinien gegeben werden, unter denen der Informatikunterricht am Humboldt-Gymnasium durchgeführt wird. Hiermit wird jedoch kein Vorgriff auf die Bedingungsfeldanalyse des Unterrichtsentwurfs intendiert, weshalb an dieser Stelle auf die Beschreibung von Räumlichkeiten sowie der schulspezifischen, zeitlichen Vorgaben verzichtet wird. Vielmehr konzentriert sich der Abschnitt auf die Erläuterung der verschiedenen Angebotsstrukturen, welche die Schule den Schülerinnen und Schülern (im Folgenden abgekürzt mit SuS) im Fachbereich Informatik in verschiedenen Altersstufen offeriert, wobei insbesondere auf die durch die Schule festgelegten Inhalte der jeweiligen Kurse sowie die Intentionen und (institutionellen) Zwangslagen, welche die Genese jener Kurse und Inhalte prägten, eingegangen werden soll. Hierdurch soll es erleichtert werden, die später in der Bedingungsanalyse betrachteten Faktoren jener Felder in ihrer Sinnhaftigkeit zu erfassen. Weiterhin werden die Bedeutungen bestimmter Begriffe mit Blick auf ihren schulspezifischen Anwendungskontext definiert.

1.1 Informatiknahe Unterrichtsfächer in der Sekundarstufe I

Am Humboldt-Gymnasium werden SuS in den Klassenstufen 5 bis 12 unterrichtet.

In der siebten Klasse wird den SuS die Teilnahme am Unterrichtsfach „Memo / ITG“ (Medien und Methoden / Informationstechnische Grundbildung) angeboten.¹ Hierbei orientiert sich das Fachkollegium an den Vorgaben des Berliner Rahmenlehrplans für ITG² sowie dem ebenfalls daran orientierten Fachbrief Nr. 1 ITG der Senatsverwaltung für Bildung, Jugend und Wissenschaft,³ ohne die in jenen Dokumenten bezüglich der Inhalte und Kompetenzen formulierten Vorgaben vollständig umzusetzen. Das Fach behandelt schwerpunktmäßig die thematischen Module „Aufbau und Wirkungsweise von Informatiksystemen“ (ITG1)⁴ sowie „Nutzung von Standardsoftware“ (ITG2),⁵ wohingegen aus den Feldern „Informationsbearbeitung“ (ITG3)⁶ sowie „Leben mit vernetzten Systemen“ (ITG4)⁷ nur ausgewählte Inhalte betrachtet werden.⁸ Stattdessen werden die im Rahmenlehrplan vorgegebenen Themen um das Feld „Medien und Methoden“

¹ Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_gesamt.pdf [01.10.2014], S. 155-158.

² Vgl. https://www.berlin.de/imperia/md/content/sen-bildung/schulorganisation/lehrplaene/sek1_itg_informatik.pdf?start&ts=1406800774&file=sek1_itg_informatik.pdf [01.10.2014].

³ Vgl. http://bildungserver.berlin-brandenburg.de/fileadmin/bbb/unterricht/rahmenlehrplaene_und_curriculare_materialien/fachbriefe_berlin/itg/fachbrief_itg_01.pdf [01.10.2014].

⁴ Vgl. https://www.berlin.de/imperia/md/content/sen-bildung/schulorganisation/lehrplaene/sek1_itg_informatik.pdf?start&ts=1406800774&file=sek1_itg_informatik.pdf [01.10.2014], S. 18 f.

⁵ Vgl. ebd. [01.10.2014], S. 20 f.

⁶ Vgl. ebd. [01.10.2014], S. 22 f.

⁷ Vgl. ebd. [01.10.2014], S. 23 f.

⁸ Vgl. <http://www.humboldtschule-berlin.de/images/pdf/memo/memo.pdf> [01.10.2014].

ergänzt, worunter die Schule ein Methodentraining versteht, „das die Schülerinnen und Schüler mit Lerntechniken, Möglichkeiten der Texterschließung und Formen der Präsentation vertraut macht.“⁹

Als weiteres Additum erhalten die SuS der achten, neunten und zehnten Klasse die Möglichkeit, das Wahlpflichtfach „Technik und Naturwissenschaft“ (TuN) zu belegen.¹⁰ Neben Themen der Physik, Biologie und Chemie werden die SuS der achten Jahrgangsstufe in die logischen Schaltungen eingeführt, wobei die Thematik das Kennenlernen des Aufbaus eines Stromkreises, der UND-, der ODER- sowie der Wechselschaltung und das Arbeiten mit Funktionstabellen umfasst.¹¹ In der neunten Klasse wird der ITG-Anteil an TuN ausgeweitet, wobei neben dem Aufbau des Digitalrechners und dem Publizieren im Internet die Datenverwaltung und -verarbeitung vordergründig thematisiert wird.¹²

1.2 Schulinternes Gesamtcurriculum und Profilierungskurs Informatik

Betrachtet man die einzelnen Fachcurricula des Humboldt-Gymnasiums, so stellt man schnell fest, dass die Entwicklung des Schulcurriculums in Abhängigkeit vom jeweiligen Fach ein unterschiedliches Niveau erreicht hat. Während in manchen Fächern (bspw. in Chinesisch) noch kein Curriculum existiert oder dieses nicht zwischen den Jahrgangsstufen differenziert (bspw. in Ethik), wurden in anderen Fächern (bspw. in Erdkunde oder Geschichte) Teilcurricula konzipiert, welche mehrdimensional zwischen den Jahrgangsstufen sowie den Leistungen der SuS (in Form von Regel- und Schnellläuferklassen) differenzieren. Weiterhin ist auffällig, dass die wenigsten Fachcurricula die Sekundarstufe II vollständig umfassen, oftmals endet die Konzeption nach der 10. Klasse, der Einführungsphase oder der 11. Klasse.¹³

Ein ähnlich differenziertes Bild ergibt sich für die Curricula der Informatik und der Informatiknahen Fächer: So weist das Curriculum für ITG / Memo detailliert aus, welche Themen zu welchem Zeitpunkt und in welcher Zeitspanne zu bearbeiten sind, Überlegungen zu Kompetenzen, Standards oder Lernzielen fehlen jedoch gänzlich.¹⁴ In gleicher Weise werden im Curriculum des Fachs Technik und Natur Themen für Unterrichtsreihen sowie die jeweils dafür vorgesehene Anzahl an Stunden festgelegt, eine Formulierung von Kompetenzen, Standards oder Zielen für die jeweilige Reihe findet jedoch auch hier nicht statt. Es werden lediglich Grobziele für das Fach formuliert.¹⁵

Anders verhält es sich beim Curriculum für Informatik als Wahlpflichtkurs in der Jahrgangsstufe 10. Das Curriculum orientiert sich dabei nicht etwa am Rahmenlehrplan Informatik der Sekundarstufe I, obwohl dieser den Rahmen für den Informatikunterricht als Wahlfach in der Sekundarstufe I setzt, sondern an der Einführungsphase des Rahmenlehrplans der Sekundarstufe II.¹⁶ Die Wahl der Einführungsphase als Grundlage für das Curriculum ist insofern sinnvoll, als dass der Besuch dieses Wahlpflicht- bzw. Profilierungskurses vom Humboldt-Gymnasium als Voraussetzung für die Möglichkeit, Informatik in der

⁹ <http://www.humboldtschule-berlin.de/index.php/unterricht/faecher-m-z/memo-itg> [01.10.2014].

¹⁰ Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_gesamt.pdf [01.10.2014], S. 211-218.

¹¹ Vgl. <http://www.humboldtschule-berlin.de/images/pdf/tun/Rahmenplantun8.pdf> [01.10.2014].

¹² Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_gesamt.pdf [01.10.2014], S. 215 f.

¹³ Vgl. ebd. [01.10.2014].

¹⁴ Vgl. ebd. [01.10.2014], S. 157 f.

¹⁵ Vgl. ebd. [01.10.2014], S. 213-217.

¹⁶ Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_informatik.pdf [01.10.2014].

Sekundarstufe II als Leistungskursfach zu belegen, festgesetzt wurde.¹⁷ Diese Kondition leitet sich wiederum aus dem Rahmenlehrplan der Sekundarstufe II ab, welcher bestimmte Eingangsvoraussetzungen für den erfolgreichen Kompetenzerwerb der SuS im Informatikunterricht der Qualifikationsphase benennt. Mit diesen Eingangsvoraussetzungen beschreibt der Rahmenlehrplan bestimmte inhalts- und prozessbezogene Kompetenzen, wobei über die Formulierung von Standards für die einzelnen Kompetenzen transparent gemacht wird, in welchem Umfang bzw. in welchem Anforderungsbereich / auf welcher Stufe die SuS die jeweilige Kompetenz bereits erworben haben sollen.¹⁸ Dies ist notwendig, da der Rahmenlehrplan in der Qualifikationsphase von den gleichen Kompetenzen ausgeht, mit dem Unterschied, dass die abschlussorientierten Standards umfangreicher und ausdifferenzierter sind, sodass die SuS am Ende der Qualifikationsphase die Kompetenzen auf einem höheren Anforderungsniveau bzw. einer höheren Stufe beherrschen sollen.¹⁹ Die Leistung des schulinternen Curriculums für den Profilierungskurs besteht dabei in der Zuordnung der Eingangsvoraussetzungen zu den Inhalten der Einführungsphase: So übernahmen die FachlehrerInnen wortwörtlich die vom Rahmenlehrplan vorgegebenen Themen, wobei die übergeordneten Themenbereiche im Curriculum jeweils als Thema bzw. Schwerpunkt einer eigenen Unterrichtsreihe dargestellt wurden und die im Rahmenlehrplan formulierten Unterpunkte jeweils als geplante Inhalte der Reihe dienten. Für jede Reihe bzw. jeden Schwerpunkt wurden anschließend zentrale Standards (inklusive der jeweils zugehörigen Kompetenzen) benannt, welche durch die Unterrichtsreihe erworben werden sollten.²⁰ Desweiteren stellt das Humboldt-Gymnasium eine exemplarische und daher unvollständige Liste von Unterrichtsthemen für den Informatikunterricht in der 10. Klasse (dort als erstes Unterrichtsjahr bezeichnet) sowie in der Qualifikationsphase (zweites und drittes Unterrichtsjahr) zur Verfügung, an dem sich die FachlehrerInnen und ReferendarInnen orientieren können.²¹

1.3 Informatik in der Sekundarstufe II

Für die Qualifikationsphase stellt dieses Dokument sogar die einzige, von der Schule ausgehende Richtlinie dar, da ein schulinternes Curriculum für die Sekundarstufe II im Fachbereich Informatik noch nicht erarbeitet wurde. Die Fachlehrer haben somit nur die Möglichkeit, sich an den Vorgaben höherer Institutionen wie dem Rahmenlehrplan für die Sekundarstufe II oder den „Einheitlichen Prüfungsanforderungen“ (EPA) für Informatik der Kultusministerkonferenz zu orientieren.²² Als formale, schulspezifische Besonderheit bezüglich des Unterrichts in der Sekundarstufe II sei noch erwähnt, dass die Bezeichnung der Klassenstufe nicht dem eigentlichen Jahrgang entspricht, sondern, orientiert an den erworbenen Kompetenzen und behandelten Themen, auf die Bezeichnung im alten, 13 Jahrgangsstufen umfassenden System abbildet. Die SuS, welche das 11. Schuljahr unterrichtet werden und sich so im ersten Jahr der Qualifikationsphase befinden, werden daher als 12. Klasse bezeichnet, da im 13 Jahrgangsstufen

¹⁷ Vgl. https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1406800775&file=sek2_informatik.pdf [01.10.2014], S. V-VII f. sowie <http://www.humboldtschule-berlin.de/index.php/unterricht/faecher/informatik/102-informatik-varianten-der-belegung> [01.10.2014].

¹⁸ Vgl. https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1406800775&file=sek2_informatik.pdf [01.10.2014], S. 12 f.

¹⁹ Vgl. ebd. [01.10.2014], S. 13-17.

²⁰ Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_informatik.pdf [01.10.2014].

²¹ Vgl. www.humboldtschule-berlin.de/index.php/unterricht/faecher/informatik/103-informatik-unterricht [01.10.2014].

²² Vgl. http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf [01.10.2014].

umfassenden System die Qualifikationsphase in der 12. Jahrgangsstufe begann. Für den Bericht soll diese schulspezifische Bezeichnung nachfolgend nicht übernommen werden, sodass die Angabe der Jahrgangsstufe einer Klasse auch der tatsächlichen Anzahl der Unterrichtsjahre entspricht.

2. Stundenentwurf zum Thema: „Realisierung eines Dateikopierers in C#“

Name: Kevin Rimek

Fach: Informatik

Seminarleiter: Dr. Nguyen-Thinh Le

Thema: „Realisierung eines Dateikopierers in C#“

Datum: 16.09.2014

Zeit: 10:45-12:00

Raum: 209

Schule: Humboldt-Gymnasium

Mentor: Lars P.

Klassenstufe: 11 (Leistungskurs; 12 SuS: 10 Jungen, 2 Mädchen)

Fachlehrer: Lars P.

Thema der Unterrichtsreihe: „Entwicklung von Desktopanwendungen in C# – Einführung in die moderne Oberflächenprogrammierung mit WPF und XAML“

Thema der Unterrichtsstunde: „Realisierung eines Dateikopierers in C#“

2.1 Erläuterungen zum Thema der Unterrichtsreihe

Die vier Unterrichtsstunden umfassende Reihe zum Thema „Entwicklung von Desktopanwendungen in C# – Einführung in die moderne Oberflächenprogrammierung mit WPF und XAML“ wurde vollständig von mir geplant und durchgeführt, wobei mein Fachlehrer mir sowohl in der Wahl des Themas als auch in der didaktischen Umsetzung freie Hand ließ. Die Unterrichtsreihe intendiert ein hohes Maß an horizontalem und vertikalem Transfer.²³ Der vertikale Transfer wird durch die Wahl des Themas und der Ziele sowie durch die sich daraus ergebende, nachfolgend zu erläuternde Struktur der Reihe bedingt. Hierbei lernen die SuS in jeder Stunde neue Sprachelemente von C# kennen und setzen diese nach und nach zu immer komplexer werdenden Gebilden zusammen. Der horizontale Transfer ergibt sich aus der thematischen Einbettung der Reihe in das vom Fachlehrer konzipierte Kurshalbjahr. Vor meiner Reihe wurde mit den SuS das Thema „Betriebssysteme“ behandelt, weshalb ich an diversen Stellen meiner eigenen Reihe die Strukturen von C# sowie die der Sprache zugrundeliegenden Technologien in Relation zum Betriebssystem gesetzt habe, indem ich mit den SuS erarbeitet habe, wie C# das Betriebssystem über das .Net-Framework anspricht. So ist es auch kein Zufall, dass die Unterrichtsstunde, welche in diesem Entwurf vorgestellt werden soll, die Realisierung eines Dateikopierers zum Gegenstand hat. Weiterhin ergibt sich der horizontale Transfer aus dem Umstand, dass viele Sprachelemente in C# ihrem Pendant in Java in hohem Maße ähneln. Da Java im Profilierungskurs für den Themenschwerpunkt „Einführung in das informatische Modellieren und Umsetzung mit einer Programmiersprache“ verwendet wurde,²⁴ stellt die Auseinandersetzung mit C# für SuS somit eine Wiederholung der Sprachelemente in einem neuen Bezug / einer neuen Programmiersprache dar. Dies ist insofern vorteilhaft, als dass der Fachlehrer nach meiner Unterrichtsreihe damit begonnen hat, die Skriptsprache PHP einzuführen. Dabei griff er die Erarbeitung von C# auf, indem er mit den SuS Gemeinsamkeiten und Unterschiede zwischen der Skriptsprache PHP und den objektorientierten Programmiersprachen Java und C# identifizierte.

²³ Zur Definition des horizontalen und vertikalen Transfers: vgl. Meyer 2012, S. 202 f.

²⁴ Vgl. <http://www.humboldtschule-berlin.de/index.php/unterricht/faecher/informatik/103-informatik-unterricht> [02.10.2014].

Wie dieses Beispiel zeigt, werden durch den starken Transfercharakter der Reihe die Überlegungen zur Stundenkonzeption in Teilen durch die Themen und Lernziele der vorangegangenen Unterrichtsstunden der Reihe sowie der benachbarten Reihen geprägt. Aus diesem Grund soll dem Stundenentwurf nachfolgend eine Nennung der groben Lernziele sowie eine kurze Begründung der Themenwahl vorangestellt werden, wobei kurz bedeutet, dass hiermit keine ausführliche, didaktische Analyse formuliert werden soll. Diese wird in der didaktischen Strukturierung geleistet, wobei die hier formulierten Erläuterungen als Vorarbeit für diese auf der Ebene der Reihe verstanden werden sollen.

2.1.1 Grobziele der Unterrichtsreihe

Kognitive Lernziele:

Zur Hierarchisierung der kognitiven Lernziele beziehe ich mich auf die kognitive Lernzieltaxonomie der Arbeitsgruppe Bloom.²⁵ Der Name der Stufe steht dabei jeweils in Klammern am Ende des jeweiligen Lernziels, dem diese Stufe zugeordnet wurde.

Die SuS sollen...

- lernen, die verschiedenen Strukturen, in die der Programmcode unterteilt ist, zu benennen und ihre Bedeutung für den Code zu erläutern. (Verständnis)
- lernen, die verschiedenen Dateien und Dateitypen in einer Projektmappe zu benennen und ihre Funktion im später daraus kompilierten Programm zu erläutern. (Verständnis)
- lernen, zu erläutern, wie die Trennung des Programmcodes in Oberflächen- und logischen Code strukturiert ist und welche Wege es gibt, über die Oberfläche mit der Logik bzw. über die Logik mit der Oberfläche zu kommunizieren. (Verständnis)
- lernen, darzustellen, wie das Betriebssystem mithilfe von Befehlen aus Frameworkbibliotheken angesprochen werden kann. (Verständnis)
- lernen, die Struktur und die Funktion unbekannter Elemente einer Programmiersprache zu analysieren, indem sie sich selbständig in vorgegebenen Dokumentationen über diese informieren und ihre Erkenntnisse durch exemplarisches Implementieren der Elemente in verschiedenen Kontexten verifizieren. (Analyse)
- lernen, wie sie sich selbständig weiterführendes Wissen über eine Programmiersprache aneignen, indem sie geeignete Medien finden und verwenden, welche ein Anknüpfen an ihr bisheriges Wissen zulassen. (Beurteilung)
- lernen, bereits bekannte Sprachelemente in einfachen oder bekannten Kontexten zu implementieren. (Übung)
- lernen, Lösungen für komplexere Probleme zu realisieren, indem sie die sich aus dem Problem ergebenden Anforderungen analysieren, aus neuen und bekannten Sprachelementen Algorithmen zur Umsetzung jener Anforderungen formulieren und diese Algorithmen programmatisch umsetzen. (Synthese)
- lernen, eigenständig die Güte ihres Programms zu beurteilen, indem sie unter Generierung verschiedener Anwendungsfälle testen, ob die funktionalen Anforderungen an das Programm auch durch die konkrete Implementierung geleistet werden können. (Begründung)
- lernen, Sprachelemente in einem kurzen, zeitlichen Rahmen und reduziert auf die wesentlichen Eigenschaften, d. h. solche Eigenschaften, welche den Aufbau oder die Funktionsweise des

²⁵ Bloom (zit. nach Sitte 2001, S. 474 f.).

Elementes beschreiben, gegenüber ihren MitschülerInnen zu vermitteln, indem sie jene wesentlichen Eigenschaften aus ihrem bisher gesammelten Wissen über das Element selektieren und diese anschließend in einer strukturgebenden Darstellungsreihenfolge organisieren. (Analyse)

Affektive Lernziele:

Um die affektiven Lernziele bezüglich ihres Grades der Internalisierung zu hierarchisieren, beziehe ich mich auf Krathwohls Internalisierungsstufen der affektiven Taxonomie.²⁶ Der Name der Stufe steht dabei jeweils in Klammern am Ende des jeweiligen Lernziels, dem diese Stufe zugeordnet wurde.

Die SuS sollen...

- lernen, in ähnlich beschaffenen bzw. ähnlich aussehenden Codeabschnitten auf geringfügige Unterschiede zu achten und dabei zu prüfen, ob bzw. wie sich die Unterschiede auf die Funktionalität des jeweiligen Abschnitts auswirken, indem sie die Funktionalitäten der Codeabschnitte vergleichend gegenüberstellen und auch kleinere Abweichungen als mögliche Folgen der Unterschiede wahrnehmen. (Aufmerksamwerden / Beachten)
- erkennen, dass die Zusammenarbeit mit anderen Schülern einen Mehrwert für ihren eigenen Wissensstand generiert. (Organisation)
- lernen, Ergebnisse / Erzeugnisse und Positionen / Haltungen ihrer Mitschülerinnen und Mitschüler kritisch zu reflektieren und ihren eigenen Ergebnissen / Erzeugnissen bzw. Positionen / Haltungen als mögliche Alternative unvoreingenommen, d. h. anhand objektiver Bewertungskriterien, gegenüberzustellen. (Charakterisierung durch einen Wert oder eine Wertstruktur)
- die Bereitschaft entwickeln, neue Impulse aufzunehmen und in ihren bisher erarbeiteten Wissenskontext zu integrieren bzw. diesen Wissenskontext neu zu organisieren. (Aufmerksamwerden / Beachten)
- die Bereitschaft entwickeln, sich aus eigenem Antrieb heraus weiter mit einer Programmiersprache zu beschäftigen, indem sie neue Sprachelemente finden, diese in ihrer Struktur und Funktion beschreiben sowie versuchen, Handlungsfolgen auf diese anzuwenden, welche für ähnliche Elemente erarbeitet wurden. (Reagieren)
- lernen, ein Interesse daran zu entwickeln, dass ihre Mitschülerinnen und Mitschüler die Mindestanforderungen erfüllen. (Werten)
- die Bereitschaft entwickeln, Interesse an den Beiträgen aller Mitschülerinnen und Mitschüler zu zeigen und diese zu Beiträgen zu ermutigen. (Werten)
- lernen, bewusst Verantwortung für den Lernfortschritt ihrer Mitschülerinnen und Mitschüler zu übernehmen. (Werten)

2.1.2 Begründung der Themenwahl für die Unterrichtsreihe

Mit der Durchführung der Unterrichtsreihe zum Thema „Betriebssysteme“ wurde ein Gegenstand behandelt, welcher vom Rahmenlehrplan im Themenbereich „Grundlagen informatischen Arbeitens“ der Einführungsphase verortet wird.²⁷ Ebenso stellt die Einführung in die Skriptsprache PHP ein Thema dar, das in seiner Konzeption mit dem Bereich „Algorithmen und Softwareentwicklung“ der Einführungsphase

²⁶ Vgl. Meyer (zit. nach http://www.zfsl-recklinghausen.nrw.de/Seminar_GyGe/Seminarprogramm/Lernzielkonzept/Anhang/Anhang_5/ [02.10.2014]).

²⁷ Vgl. https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1406800775&file=sek2_informatik.pdf, S. V f.

größere Schnittstellen aufweist als mit dem Themenfeld der „Softwareentwicklung“ der Qualifikationsphase und daher ebenso in die Einführungsphase fällt. Die separate Behandlung der Themenfelder „Grundlagen informatischen Arbeitens“ sowie „Algorithmen und Softwareentwicklung“ vernachlässigt dabei den sowohl im Rahmenlehrplan²⁸ als auch im schulinternen Curriculum²⁹ formulierten Vorschlag, diese Themenfelder integrativ zu behandeln. Mit der Konzeption meiner Unterrichtsreihe intendierte ich, diesen Vorschlag umzusetzen. Hieraus ergibt sich zum Teil die Zukunfts- und Gegenwartsbedeutung des Themas. Dabei wurde bereits auf den horizontalen Transfer als wesentliches Element für die Gegenwartsbedeutung verwiesen. Die Auseinandersetzung mit der Sprache C# verlangt von den SuS in einem hohen Maße die Reaktivierung ihrer Kenntnisse über Java bzw. über die bereits erarbeiteten Strukturen und Elemente objektorientierter Programmiersprachen. Gleichzeitig bietet sich für die SuS die Gelegenheit, Java und C# gegenüberzustellen und aus der Erarbeitung von Gemeinsamkeiten und Unterschieden Hypothesen darüber zu formulieren, welche Bedingungen eine Sprache erfüllen muss, um als objektorientiert zu gelten. Ebenso stellt die Unterrichtseinheit sicher, dass die SuS das vorangegangene Thema „Betriebssysteme“ multiperspektivisch betrachten, indem sie mithilfe von C# und dem .Net Framework die Entwicklersicht auf das Betriebssystem ergänzt. Mit der Integration von WPF und XAML wurde intendiert, die Auseinandersetzung mit objektorientierten Programmiersprachen an die Lebenswelt der SuS anzupassen, in der nicht Konsolenfenster, sondern GUI- und ModernUI-Oberflächen als Interaktionsschnittstellen vorherrschend sind. In der Befähigung zur programmatischen Implementierung moderner Oberflächen liegt ebenso eine Zukunftsbedeutung für die SuS, zumal das in C# genutzte Konzept der Trennung von Design- und logischem Code ebenso in anderen Sprachen und auf anderen Plattformen, wie bspw. bei Java in Verbindung mit Android, Anwendung findet. Da die Android-Entwicklung den SuS als Unterrichtsthema der 12. Jahrgangsstufe am Humboldt-Gymnasium begegnet,³⁰ wird hiermit ein grundlegendes Verständnis für das später notwendig anzuwendende Konzept vorbereitet. Ebenso ist die Wahrscheinlichkeit groß, dass die SuS, wenn sie sich nach dem Besuch des Gymnasiums in einem schulischen oder beruflichen Umfeld weiterhin mit Informatik beschäftigen, der Anforderung begegnen, moderne Oberflächen zu programmieren. Als weitere Zukunftsbedeutung wurde bereits angeführt, dass nicht nur die objektorientierten Sprachen untereinander, sondern in der nachfolgenden Unterrichtseinheit mit der Skriptsprache PHP verglichen werden, wobei insbesondere die Unterschiede zwischen der Skriptsprache und den objektorientierten Sprachen herausgearbeitet werden sollen.³¹ Andererseits gewährleistet die Auseinandersetzung mit C#, dass diejenigen Sprachelemente, die sich als Gemeinsamkeit von C# und PHP identifizieren lassen, in ihrer Bedeutung während der Auseinandersetzung mit PHP nicht noch einmal wiederholt werden müssen.

Nachfolgend soll ein kurzer Überblick darüber gegeben werden, welche konkreten Themen in den einzelnen Stunden der Reihe behandelt wurden.

²⁸ Vgl. ebd. [02.10.2014], S. VI.

²⁹ Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_informatik.pdf [02.10.2014], S. 4.

³⁰ Vgl. <http://www.humboldtschule-berlin.de/index.php/unterricht/faecher/informatik/103-informatik-unterricht> [02.10.2014].

³¹ Vgl. <http://icebear.pelz-familie.de/?q=node/585> [02.10.2014].

2.1.3 Struktur der Unterrichtsreihe

1. Stunde der Reihe	Montag, 08.09.2014	14:00-15:15
Thema: Einführung in die moderne Oberflächenprogrammierung mit XAML		
Geplante Inhalte: <ul style="list-style-type: none">• Überblick über wichtige Bibliothekskategorien des .Net-Framework• Einführung in die IDE Visual Studio 2012 Express• Einführung in die Windows Presentation Foundation• Überblick über die Dateien in der Projektmappe einer WPF-Anwendung• Erarbeitung von grundlegenden UI-Elementen in XAML• Erarbeitung des Property-Konzept in XAML		

2. Stunde der Reihe	Donnerstag, 11.09.2014	12:30-13:45
Thema: Einführung in die Logikprogrammierung für mit XAML erstellte Oberflächen über C#		
Geplante Inhalte: <ul style="list-style-type: none">• Erarbeitung grundlegender Sprachelemente und -strukturen (Namensräume, Klassen, Methoden Datentypen)• Erweiterung des Property-Konzepts auf C#• Kennenlernen des Konzepts der Trennung von logischem und Oberflächencode und der damit verbundenen Sprachen (C# und XAML)• Erarbeitung der Kommunikationswege zwischen Elementen in C# und XAML• Kennenlernen der Parent-Children-Kommunikation in XAML und C#		

3. Stunde der Reihe	Montag, 15.09.2014	14:00-15:15
Thema: Bedingungsabfragen und Kontrollstrukturen		
Geplante Inhalte: <ul style="list-style-type: none">• Erarbeitung der logischen Operatoren in C#• Erarbeitung der Vergleichsoperatoren in C# sowie der vergleichbaren Datentypen• Zusammensetzen grundlegender logischer Operationen bzw. Vergleichsoperationen zu komplexeren Ausdrücken und Abfragen• Beginn der Erarbeitung der if-else- sowie der switch-case-Verzweigung• Beginn der Erarbeitung der for- und der while-Schleife		

4. Stunde der Reihe	Dienstag, 16.09.2014	10:45-12:00
Thema: Realisierung eines Dateikopierers in C#		
Geplante Inhalte: <ul style="list-style-type: none">• Finalisierung der Erarbeitung der if-else- sowie der switch-case-Verzweigung• Finalisierung der Erarbeitung der for- und der while-Schleife• Kennenlernen spezifischer C#-Sprachelemente• Erarbeitung der Methoden der File-Klasse• Kennenlernen des methodischen Dreischritts der Anforderungsanalyse, Planung und Implementierung und Anwenden des Dreischritts im Rahmen eines Softwareprojekts		

2.2 Aufgabenstellung der Stunde und Aufgabenanalyse

In diesem Abschnitt soll die konkrete Aufgabenstellung der Unterrichtsstunde ausformuliert und analysiert werden, wobei ich mich an der Aufgabenanalyse nach Meyer orientiere.³² Um Dopplungen zu vermeiden, werden die aufgabenbezogene Lernstandsanalyse sowie die Klärung der Fragen nach der Passung der Aufgabenstellung und der inhaltlichen Verknüpfung im Rahmen der Bedingungsfeldanalyse sowie der didaktischen Strukturierung behandelt.

2.2.1 Ausformulierung der Aufgabenstellung

„In der heutigen Stunde werden wir eine WPF-Anwendung in C# realisieren, welche es dem Nutzer ermöglicht, eine von ihm gewählte Datei beliebig oft an einen von ihm gewählten Speicherort zu kopieren. Um diese Aufgabe zu lösen, benötigen wir neben den Kenntnissen, die wir uns in den letzten Stunden über Betriebssysteme und die Programmiersprache C# angeeignet haben, weiterführendes Wissen über die Kontrollstrukturen in C# sowie über bestimmte Klassen, die uns einen Zugriff auf das Betriebssystem erlauben.

Aus diesem Grund sollt ihr euch zunächst wieder in den Dreiergruppen zusammenfinden, die wir zu Beginn der vorangegangenen Stunde gebildet haben. Jedes Gruppenmitglied ist dabei Experte für mindestens eine Kontrollstruktur in C#. Als Experte seid ihr dafür verantwortlich, dass die anderen Gruppenmitglieder eure Kontrollstruktur(en) verstehen und diese anwenden können. Um dies sicherzustellen, sollen die Experten ihre Kontrollstruktur innerhalb von 5 Minuten gegenüber ihren Gruppenmitgliedern erläutern, wobei sie insbesondere auf den Aufbau und die Funktionsweise der jeweiligen Struktur eingehen sollen. Dabei können die Experten auch auf die Arbeitsblätter aus der letzten Stunde eingehen, indem sie mit den anderen Gruppenmitgliedern die Verständnis- und Kontrollaufgaben bearbeiten, um sich somit zu vergewissern, dass diese die Struktur auch wirklich verstanden haben.

Anschließend betrachten wir die Funktionen einiger ausgewählter Klassen, Methoden und Eigenschaften, die wir für die Implementierung des Dateikopierers benötigen. Dabei werden wir insbesondere auf die Methoden der Klasse File eingehen, welche die Verzeichnisstruktur des Betriebssystems manipulieren. Eure Aufgabe besteht darin, sicherzustellen, dass ihr die Bedeutung und Funktionsweise der behandelten Sprachelemente verstehen und wiedergeben könnt. Da für die Methoden der Klasse File noch keine Beschreibung der jeweiligen Funktion vorliegt, müsst ihr euch diese eigenständig erarbeiten, indem ihr diese in der MSDN-Dokumentation sucht und sie mit eigenen Worten in euren Notizen festhaltet.

Danach haben wir alle notwendigen Kenntnisse erworben, um mit der Realisierung des Projekts zu beginnen. Hierfür werden uns konkrete Vorgaben bereitgestellt, die die Struktur und die Funktionalität des Programms beschreiben und uns somit erläutern, wie wir das Programm realisieren sollen. Dabei ist das Implementieren des Programms immer der letzte und nicht der erste Schritt. Vorher müssen wir analysieren, welche konkreten Schritte wir unternehmen müssen, um die einzelnen Anforderungen zu erfüllen. Anschließend müssen wir planen, in welcher Reihenfolge diese Schritte abgearbeitet werden müssen, wobei wir auf Abhängigkeiten achten müssen, die zwischen den Schritten bestehen. Erst dann können wir mit der Implementierung beginnen.“

³² Vgl. Meyer 2012, S. 184-191.

2.2.2 Mit der Aufgabenstellung verbundene Lernziele und Aufgabenlösung

Da die Lernziele der Unterrichtsstunde nachfolgend noch näher spezifiziert werden, soll in diesem Abschnitt nur auf diejenigen Lernziele eingegangen werden, welche bereits als Grobziele der Unterrichtsreihe enumeriert wurden und auf denen der primäre Fokus der ausformulierten Aufgabenstellung und ihrer Lösung liegt.

Im Mittelpunkt der Unterrichtsstunde steht die Realisierung der Dateikopierer-Anwendung als Produkt einer handlungsorientierten Projektarbeit.³³ Gudjons unterteilt den Verlauf eines Projekts in vier Schritte:

1. eine für den Erwerb von Erfahrungen geeignete, problemhaltige Sachlage auswählen
2. gemeinsam einen Plan zu Lösung entwickeln
3. sich mit dem Problem handlungsorientiert auseinandersetzen
4. die erarbeitete Problemlösung an der Wirklichkeit überprüfen.³⁴

Die Schritte 2-4 finden sich auch in Deweys methodischem Dreischritt zur Problemlösung, bestehend aus der Hypothesenbildung, dem Entwurf einer Lösung und dem Ausprobieren dieser,³⁵ welcher ebenso Eingang in die Unterrichtsstunde findet, wobei der Anteil der direkten Instruktion innerhalb der Stunde das für das Konzept des handlungsorientierten Projektunterrichts übliche Maß deutlich übersteigt,³⁶ da die SuS in den meisten Phasen des Projekts vom Lehrer angeleitet werden und erst im letzten Abschnitt des Unterrichts den von Dewey formulierten Dreischritt eigenständig durchführen. Die zunächst starke Anleitung durch den Lehrer erscheint jedoch als legitim, wenn man sich das primäre kognitive Lernziel für die Unterrichtsstunde vor Augen führt, welches als Grobziel der Unterrichtsreihe wie folgt formuliert wurde: „Die SuS sollen lernen, Lösungen für komplexere Probleme zu realisieren, indem sie die sich aus dem Problem ergebenden Anforderungen analysieren, aus neuen und bekannten Sprachelementen Algorithmen zur Umsetzung jener Anforderungen formulieren und diese Algorithmen programmatisch umsetzen.“³⁷ Der sich hieraus ergebende Dreischritt der Anforderungsanalyse, der Planung der Anforderungsumsetzung und der Implementierung soll dabei als Methodik zur Problemlösung eingeführt und eingeübt werden. Die SuS sollen diesen Dreischritt also nicht nur abarbeiten, um somit das gewünschte Produkt zu realisieren (Handeln nach Vorgabe),³⁸ sie sollen gleichzeitig ihr Vorgehen reflektieren und dabei den Dreischritt als Methodik der Problemlösung inkorporieren, d. h. sie sollen lernen, die Funktionsweise der Methode zu verstehen und ohne weitere Belehrung in unterschiedlichen, passenden bzw. zieladäquaten Fachbezügen mit dieser zu arbeiten (Handeln nach Einsicht).³⁹ Um dieses Ziel zu erreichen, wird der Dreischritt mehrmals innerhalb der Stunde durchlaufen, indem das Programm in drei Abschnitte aufgeteilt wird,⁴⁰ wobei durch die Variation der Sozialformen und der zur Verfügung stehenden Orientierungshilfen von den SuS bei jedem Durchgang ein unterschiedlich hohes Maß an Eigenständigkeit abverlangt wird:

Der erste Durchlauf wird als Plenumsunterricht inszeniert, wobei über das Handlungsmuster des fragend-entwickelnden Unterrichtsgesprächs⁴¹ die SuS zusammen mit der Lehrperson, welche das Gespräch

³³ Vgl. Humbert 2006, S.39-47.

³⁴ Gudjons (zit. nach Humbert 2006, S. 42).

³⁵ Dewey (zit. nach Humbert 2006, S. 41).

³⁶ Vgl. Meyer 2012, S. 57-74.

³⁷ Vgl. 2.1.1 Grobziele der Unterrichtsreihe.

³⁸ Dies entspricht der zweiten Stufe in Meyers pragmatischem Kompetenzstufenmodell. Vgl. Meyer 2012, S. 157.

³⁹ Dies entspricht der dritten Stufe in Meyers pragmatischem Kompetenzstufenmodell. Vgl. ebd., S. 157.

⁴⁰ Vgl. Anhang 2.

⁴¹ Vgl. Meyer 1994, S. 288.

moderiert, eine Art Musterlösung erarbeiten. Die SuS, welche sich am Unterrichtsgespräch beteiligen, sind also ganz oder zumindest teilweise in der Lage, die auf dem Arbeitsblatt formulierten oder sich während des Unterrichtsgesprächs ergebenden Anweisungen und Regeln zur Umsetzung des Dreischritts zu verstehen und diese schrittweise auszuführen, d. h. nach Vorgabe zu handeln. Aus dem Unterrichtsgespräch als gezeigter Leistung ist für den Lehrer jedoch nur auf der Ebene der Performanz, d. h. auf der Oberfläche, erkennbar, inwieweit der Dreischritt als Methode von den SuS verstanden wurde und infolgedessen angewendet werden kann, der tatsächliche Fortschritt der Aneignung auf der Ebene der Kompetenz bleibt jedoch verborgen, weshalb durch die Lehrkraft nicht zuverlässig sichergestellt werden kann, dass die gezeigte Leistung der erworbenen Kompetenzstufe entspricht, vor allem in den Fällen nicht, in denen ein Schüler oder eine Schülerin keinen Beitrag leistet.⁴² Hierin liegt auch das inkrementelle Vorgehen bei der Realisierung des Dreischritts begründet: durch die Unterteilung des Programms in kleinere Abschnitte soll dabei die Komplexität des exemplarischen Gegenstands, an dem die Methode durchgeführt wird, reduziert und der Fokus auf die Arbeit mit der Methode gelegt werden. Weiterhin soll über das wiederholte Anwenden des Dreischritts das Verständnis der SuS über die Funktionsweise dieser Methode gefestigt werden.

Die Anwendung des Dreischritts auf den zweiten Anforderungsabschnitt soll den SuS die Möglichkeit geben, für sich selbst zu überprüfen, inwieweit sie die Methode bereits inkorporiert haben. Aus diesem Grund wird die Eigenständigkeit der SuS durch den Wechsel der Sozialform vom Plenumsunterricht zur Partnerarbeit erhöht. Die Notwendigkeit der Diskussion über die konkrete Ausgestaltung des Ergebnisses für den jeweiligen Schritt mit dem Partner gewährleistet dabei, dass sich alle SuS mit jedem der drei Schritte aktiv auseinandersetzen müssen und sich nicht, wie es im Unterrichtsgespräch der Fall ist, hinter ihren Mitschülerinnen und Mitschülern als passive Rezipienten verstecken können. Durch die Partnerarbeit müssen die SuS also eigene Ideen zur Anwendung der Methode entwickeln, gleichzeitig haben sie jedoch die Möglichkeit, die durch ihren Partner generierten, externen Impulse aufzunehmen, dessen Lösungsvorschläge unvoreingenommen und anhand von objektiven Kriterien zu beurteilen bzw. den eigenen Ideen gegenüberzustellen, woraus sich für die SuS die Möglichkeiten ergeben, alternative Lösungsvorschläge in reflektierter Form für sich zu übernehmen sowie aus den Impulsen des Partners heraus Fehler in der eigenen Verwendung der Methode zu identifizieren. Die Partnerarbeit trägt also zum Erreichen der folgenden, für die Unterrichtseinheit formulierten, affektiven Grobziele bei:

- Die SuS sollen lernen, Ergebnisse / Erzeugnisse und Positionen / Haltungen ihrer Mitschülerinnen und Mitschüler kritisch zu reflektieren und ihren eigenen Ergebnissen / Erzeugnissen bzw. Positionen / Haltungen als mögliche Alternative unvoreingenommen, d. h. anhand objektiver Bewertungskriterien, gegenüberzustellen.
- Die SuS sollen die Bereitschaft entwickeln, neue Impulse aufzunehmen und in ihren bisher erarbeiteten Wissenskontext zu integrieren bzw. diesen Wissenskontext neu zu organisieren.
- Die SuS sollen erkennen, dass die Zusammenarbeit mit anderen Schülern einen Mehrwert für ihren eigenen Wissensstand generiert.

Die Anforderungsbeschreibung des zweiten Abschnitts (und damit der Gegenstand, an dem die Methode durchgeführt werden soll) ist bis auf wenige Einzelheiten nahezu identisch mit den im ersten Teil formulierten Anforderungen, weshalb die durch die Anwendung der Methode auf den ersten Abschnitt entstandene Musterlösung in weiten Teilen ebenso eine Musterlösung für den zweiten Abschnitt darstellt.

⁴² Vgl. Meyer 2012, S. 147.

Die SuS sollen somit die Gelegenheit erhalten, ihr eigenständig bzw. mit dem Partner erarbeitetes Ergebnis der Durchführung des Dreischritts in seiner Güte zu evaluieren, indem sie es der Musterlösung des ersten Teils gegenüberstellen. Da die Anforderungen jedoch nur nahezu, aber nicht vollständig identisch sind, müssen die SuS gerade den Unterschieden besondere Aufmerksamkeit zukommen lassen und dabei insbesondere reflektieren, welche Veränderungen die Unterschiede im Ergebnis des Dreischritts bewirken, damit die Veränderungen im Vergleich mit dem Ergebnis des ersten Dreischritts nicht fälschlicherweise als Fehler identifiziert werden. Die aufmerksame Auseinandersetzung mit den Unterschieden eines Codefragments und seiner nur geringfügig veränderten Variation wurde ebenfalls als Grobziel der affektiven Lernziel dimension für die Reihe formuliert.⁴³ Insbesondere bietet die Musterlösung den SuS eine Orientierungshilfe für die Frage, wie detailliert die Anforderungsanalyse und die Planung ausgestaltet werden sollen.

Aus der großen Ähnlichkeit der Gegenstände, d. h. der Anforderungen des ersten und zweiten Abschnitts, sowie der unmittelbar aufeinanderfolgenden Anwendung des Dreischritts auf diese ergibt sich die Gefahr, dass die SuS die Durchführung des Dreischritts nicht eigenständig neu konstruieren, sondern ihr Vorgehen (zumindest in weiten Teilen) ein unreflektiertes Nachahmen der gerade im Plenum erarbeiteten Lösung darstellt,⁴⁴ in der der Dreischritt nicht als Methodik zur Lösung einer bestimmten Problemklasse wahrgenommen wird. Um zu gewährleisten, dass die Ergebnisse der SuS keine bloße Nachahmung darstellen, wird für die dritte Wiederholung des Dreischritts ein Gegenstand gewählt, welcher sich sehr viel deutlicher von den ersten beiden Anforderungsbeschreibungen unterscheidet. In diesem Durchlauf können die SuS also nicht einfach die vorangegangene Lösung memorieren und mit geringfügigen Änderungen unreflektiert wiedergeben, sondern sie müssen verstanden haben, was die einzelnen Schritte der Methode beinhalten bzw. erfordern, d. h. sie müssen wissen, wie die Methode funktioniert und wie sie auf den neuen Gegenstand abgebildet werden kann.

Auch wenn das Projekt und das damit verbundene kognitive Groblernziel, nach dem die SuS lernen sollen, Lösungen für komplexere Probleme zu realisieren, indem sie die sich aus dem Problem ergebenden Anforderungen analysieren, aus neuen und bekannten Sprachelementen Algorithmen zur Umsetzung jener Anforderungen formulieren und diese Algorithmen programmatisch umsetzen, im Mittelpunkt der Aufgabenstellung stehen, muss zunächst sichergestellt werden, dass die SuS die notwendigen Vorkenntnisse besitzen, um das Projekt durchzuführen. Aus diesem Grund werden vor dem Beginn des Projekts solche Sprachelemente eingeführt oder wiederholt, die für den als Anforderungen formulierten Gegenstand von Bedeutung sind. Mit dem Austausch der Experten über die Kontrollstrukturen wird an die Aufgabenstellung aus der letzten Stunde angeknüpft. Hierbei wurden zunächst vier Dreiergruppen in der Form gebildet, dass jede Dreiergruppe in sich ein hohes Maß an Leistungsheterogenität aufwies. Die Mitglieder der vier Dreiergruppen wurden anschließend leistungsdifferenziert auf drei Expertengruppen aufgeteilt: Das jeweils leistungsschwächste Gruppenmitglied wurde der Expertengruppe zugeordnet, welche die if-else-Verzweigung erarbeitete.⁴⁵ Das in seiner Leistung jeweils mittelstarke Gruppenmitglied bearbeitete das Aufgabenblatt zur switch-case-Verzweigung,⁴⁶ wobei bereits rudimentäre Kenntnisse über die if-else-

⁴³ „Die SuS sollen lernen, in ähnlich beschaffenen bzw. ähnlich aussehenden Codeabschnitten auf geringfügige Unterschiede zu achten und dabei zu prüfen, ob bzw. wie sich die Unterschiede auf die Funktionalität des jeweiligen Abschnitts auswirken, indem sie die Funktionalitäten der Codeabschnitte vergleichend gegenüberstellen und auch kleinere Abweichungen als mögliche Folgen der Unterschiede wahrnehmen.“ (2.1.1 Grobziele der Unterrichtsreihe).

⁴⁴ Dies entspricht der zweiten Stufe in Meyers pragmatischem Kompetenzstufenmodell. Vgl. Meyer 2012, S. 157.

⁴⁵ Vgl. Anhang 4.

⁴⁶ Vgl. Anhang 5.

Verzweigung vorausgesetzt wurden. Die jeweils leistungsstärksten Mitglieder einer Gruppen sollten sowohl die for- als auch die while-Schleife erarbeiteten.⁴⁷ Neben dem Umstand, dass die Arbeitsaufträge für die ersten beiden Gruppen kleinschrittiger und angeleiteter formuliert waren, als dies bei der leistungsstarken Gruppe der Fall war, unterschieden sie sich weiterhin in der methodischen Herangehensweise: Während die if-else- sowie die switch-case-Gruppe den Aufbau und die Funktionsweise ihrer jeweiligen Struktur aus einem vorgegebenen Lehrbuchtext entnehmen und ihre Ergebnisse anschließend durch geschlossene Verständnis- und Übungsaufgaben sichern konnte, sollte sich die leistungsstarke Gruppe den Aufbau und die Funktionsweise der beiden Schleifen durch forschend-entdeckendes Lernen erarbeiten, indem sie vorgegebene Schleifenimplementierungen im Visual Studio schrittweise ausführten und infolgedessen Hypothesen über den Aufbau und die Funktionsweise der beiden Schleifen formulierten.

In der Stunde, für die dieser Unterrichtsentwurf konzipiert wird, sollen die SuS als Experten für eine der beiden Verzweigungen oder die beiden Schleifen in die in der vorangegangenen Stunde gebildeten, leistungsheterogenen Gruppen zurückkehren und ihren anderen Gruppenmitgliedern die von ihnen mit unterschiedlichem Aufwand und Anforderungsgrad erarbeiteten Strukturen in einem zeitlichen Rahmen von jeweils fünf Minuten in der Form erklären, dass die beiden anderen Gruppenmitglieder nach diesen fünf Minuten den Aufbau und die Funktionsweise der jeweiligen Struktur verstanden haben und diese anwenden können. Hierdurch wird wiederum das Erreichen des für die Unterrichtsreihe formulierten, kognitiven Lernziels gefördert, nach dem die SuS lernen sollen, „Sprachelemente in einem kurzen, zeitlichen Rahmen und reduziert auf die wesentlichen Eigenschaften, d. h. solche Eigenschaften, welche den Aufbau oder Funktionsweise des Elementes beschreiben, gegenüber ihren Mitschülern zu vermitteln, indem sie jene wesentlichen Eigenschaften aus ihrem bisher gesammelten Wissen über das Element selektieren und diese anschließend in einer strukturgebenden Darstellungsreihenfolge organisieren.“⁴⁸ Neben diesem kognitiven Lernziel fördert das gegenseitige Erklären der Strukturen ein weiteres, affektives Lernziel, nach dem die SuS lernen sollen, „bewusst Verantwortung für den Lernfortschritt ihrer Mitschülerinnen und Mitschüler zu übernehmen“,⁴⁹ wobei die SuS in der jeweils von ihnen erarbeiteten Struktur über Expertenwissen verfügen, welches in der Dreiergruppe nur einmal vorhanden ist. Aus diesem Monopolwissen erwächst die alleinige Verantwortung für die zieladäquate Vermittlung der jeweiligen Struktur gegenüber den Mitschülerinnen und Mitschülern.

Nach der gegenseitigen Erläuterung der Kontrollstrukturen sollen die SuS mithilfe des Arbeitsblatts „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“ solche Sprachelemente, welche für die Anwendung des Dreischritts auf den Gegenstand des Dateikopierer-Projekts benötigt werden, wiederholen bzw. neu erarbeiten, wobei der Fokus auf die Funktionalität der Elemente gelegt wird.⁵⁰ Das Arbeitsblatt stellt den SuS eine kleine Dokumentation der benötigten Elemente zur Verfügung, bei der mehrere Sprachelemente bereits durch Anmerkungen des Lehrers erläutert wurden. Für jene Elemente müssen die SuS lediglich sicherstellen, dass sie die vom Lehrer formulierten Erläuterungen auch verstanden haben und infolgedessen die Funktionen bzw. die möglichen Auswirkungen benennen können, welche das Element auf den mit diesem in Verbindung stehenden Codeabschnitt haben kann. Für die File-Klasse werden den SuS lediglich die zu betrachtenden Methoden vorgegeben, weshalb eine Dokumentation für diese durch die SuS noch vorgenommen werden muss. Hierbei sollen sie sich an der Dokumentation des Microsoft

⁴⁷ Vgl. Anhang 6.

⁴⁸ 2.1.1 Grobziele der Unterrichtsreihe.

⁴⁹ Ebd.

⁵⁰ Vgl. Anhang 1.

Developer Networks (MSDN) orientieren, wobei sie die dort aufgeführten Inhalte nicht einfach transkribieren, sondern aufgabenorientiert in eigenen (und daher möglicherweise für sie besser verständlichen) Formulierungen zu ihrer Dokumentation hinzufügen sollen. Die Aufgabe trägt also zum Erreichen des oben formulierten, kognitiven Grobziels der Unterrichtsreihe bei, nach dem die SuS lernen sollen, „die Struktur und die Funktion unbekannter Elemente einer Programmiersprache zu analysieren, indem sie sich selbständig in vorgegebenen Dokumentationen über diese informieren und ihre Erkenntnisse durch exemplarisches Implementieren der Elemente in verschiedenen Kontexten verifizieren.“⁵¹ Vor Beginn der Erarbeitung der Dokumentation für die File-Klasse werden die SuS vom Lehrer in leistungsheterogene Paare aufgeteilt, wobei die Partner bis zum Ende der Stunde zusammenbleiben und somit auch die Realisierung des Dateikopierers zusammen durchführen. Dieser Umstand wird gegenüber den SuS vor der Paarbildung mit der Intention kommuniziert, dass die SuS reflektieren, dass der Partner, mit dem sie die das Projekt vorbereitende Aufgabe lösen, auch der Partner ist, mit dem sie später das Projekt durchführen müssen und somit ein gewisses Eigeninteresse daran entwickeln, dass der Partner einen bestimmten Wissensstand über die später benötigten Sprachelemente erreicht. Dies wurde in den affektiven Grobzielen allgemeiner wie folgt formuliert: „Die SuS sollen lernen, ein Interesse daran zu entwickeln, dass ihre Mitschülerinnen und Mitschüler die Mindestanforderungen erfüllen.“⁵²

2.2.3 Bestimmung der Arbeitsschritte

Schritt 1: Die SuS erarbeiten sich in Dreiergruppen den Aufbau und die Funktionsweise der if-else- und der switch-case-Verzweigung sowie der while- und der for-Schleife, wobei sie die hierfür notwendigen Informationen aus den fünfminütigen Kurzreferaten entnehmen, welche für jede Kontrollstruktur jeweils von demjenigen Gruppenmitglied gehalten wird, welches sich diese Struktur bereits in der vorangegangenen Stunde erarbeitet hat und daher Experte für diese ist.

Schritt 2: Die SuS lesen die auf dem Arbeitsblatt „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“ gegebene Dokumentation und stellen dabei sicher, dass sie die vom Lehrer formulierten Erläuterungen zu den dort beschriebenen Sprachelementen inhaltlich nachvollziehen und infolgedessen die Funktionen bzw. die möglichen Auswirkungen benennen können, welche das Element auf den mit diesem in Verbindung stehenden Codeabschnitt haben kann.

Schritt 3: Die SuS bilden Paare und vervollständigen in Partnerarbeit die auf dem Arbeitsblatt „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“ gegebene Dokumentation, indem sie die Funktionen der vorgegebenen Methoden der Klasse File in der Dokumentation ergänzen, wobei sie die hierfür notwendigen Informationen zunächst durch die eigenständige Arbeit mit der MSDN-Dokumentation finden.

Schritt 4: Die SuS erarbeiten mit dem Lehrer im Plenum gemeinsam die Methodik des im Arbeitsauftrag des Arbeitsblatts „Projekt Dateikopierer“ anzuwendenden Dreischritts der Anforderungsanalyse, Planung der algorithmischen Umsetzung und Implementierung, indem sie diese Schrittfolge im Rahmen eines fragend-entwickelnden Unterrichtsgesprächs exemplarisch am Gegenstand der Anforderung 1 des Arbeitsblatts durchführen.

Schritt 5: Die SuS führen in den zuvor gebildeten Paaren den Dreischritt der Anforderungsanalyse, Planung der algorithmischen Umsetzung und Implementierung selbständig am Gegenstand der Anforderung 2 des

⁵¹ 2.1.1 Grobziele der Unterrichtsreihe.

⁵² Ebd.

Arbeitsblatts „Projekt Dateikopierer“ durch und vergleichen ihr hierdurch entstandenes Produkt mit dem im Plenum vorgestellten Ergebnis eines Schülerpaars sowie dem Ergebnis der Schrittfolgenanwendung für Anforderung 1.

Schritt 6: Die SuS führen in den zuvor gebildeten Paaren den Dreischritt der Anforderungsanalyse, Planung der algorithmischen Umsetzung und Implementierung selbständig am Gegenstand der Anforderung 3 des Arbeitsblatts „Projekt Dateikopierer“ durch und vergleichen ihr hierdurch entstandenes Produkt mit dem im Plenum vorgestellten Ergebnis eines Schülerpaars.

2.3 Bedingungsanalyse

2.3.1 Richtlinien- und Fachvorgaben

Im Rahmen der Betrachtung des Informatikangebots des Humboldt-Gymnasiums wurde bereits festgestellt, dass für die Sekundarstufe II kein schulinternes, fachbezogenes Curriculum existiert, weshalb für diese Jahrgangsstufen nur eine Orientierung an den Vorgaben der der Schule übergeordneten Institutionen wie dem Berliner Rahmenlehrplan für die Sekundarstufe II im Fach Informatik möglich ist.⁵³ Weiterhin wurde bereits beschrieben, dass der Rahmenlehrplan für zwei unterschiedliche Zeitpunkte fachbezogene Standards definiert. Dabei stellten die Eingangsvoraussetzungen Standards bzw. fachliche Anforderungen dar, welche die SuS bereits zu Beginn der Qualifikationsphase bewältigt haben sollten. Die abschlussorientierten Standards stellten dagegen fachliche Anforderungen dar, die die SuS am Ende der Qualifikationsphase erreicht haben sollten. Die fachbezogenen Standards werden bestimmten, fachbezogenen Kompetenzen zugeordnet, wobei die fachbezogenen Kompetenzen am Anfang und Ende der Qualifikationsphase zwar gleich sind, jedoch alternieren die mit der jeweiligen Kompetenz konnotierten fachlichen Anforderungen zwischen diesen beiden Zeitpunkten.⁵⁴ Der Rahmenlehrplan unterscheidet zwischen sechs fachbezogenen Kompetenzen: „Informatisches Modellieren“, „Mit Informationen umgehen“, „Informatiksysteme verstehen“, „Problemlösen“, „Kommunizieren und Kooperieren“ sowie „Wechselwirkungen zwischen Informatiksystemen, Mensch und Gesellschaft beurteilen“.⁵⁵

Aus der oben ausformulierten Aufgabenstellung und -lösung wird bereits ersichtlich, dass in der Stunde überwiegend fachliche Anforderungen der Kompetenzen „Problemlösen“ sowie „Kommunizieren und Kooperieren“ aktiviert werden. Unter der Problemlösekompetenz versteht der Rahmenlehrplan das Erfassen von Problemen sowie die selbständige und sachangemessene Nutzung von Informationssystemen zu deren Lösung. In der Problemlösung kompetente SuS sind in der Lage, informatische Strategien in den verschiedenen Phasen des Problemlöseprozesses zielorientiert einzusetzen. Weiterhin erkennen und reflektieren sie die Grenzen des Problemlösens mit Informatiksystemen.⁵⁶ Unter „Kommunizieren und Kooperieren“ definiert der Rahmenlehrplan die kompetente Organisation und Koordinierung von Teamarbeit: „Die Schülerinnen und Schüler erkennen, dass Teamarbeit bei der Erstellung von Informatiksystemen zwingend erforderlich ist. Sie sind in der Lage, ihre Arbeit in Projektgruppen zunehmend selbständig zu organisieren und zu koordinieren. Sie verwenden dabei die Fachsprache angemessen, dokumentieren und präsentieren Arbeitsergebnisse. Die Schülerinnen und Schüler nutzen Informatiksysteme

⁵³ Vgl. 1.3 Informatik in der Sekundarstufe II.

⁵⁴ Vgl. 1.2 Schulinternes Gesamtcurriculum und Profilierungskurs Informatik.

⁵⁵ https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1406800775&file=sek2_informatik.pdf [05.10.2014], S. 10 f.

⁵⁶ Vgl. ebd. [05.10.2014], S. 11.

zur Kooperation und reflektieren die Kommunikationsprozesse.⁵⁷ Für beide Kompetenzen werden im Rahmenlehrplan⁵⁸ Eingangsvoraussetzungen und abschlussorientierte Standards benannt, wobei bei den abschlussorientierten Standards in beiden Fällen teilweise zwischen Grund- und Leistungskurs differenziert wird.⁵⁹ Im schulinternen Fachcurriculum des Profilierungskurses Informatik werden die im Rahmenlehrplan formulierten Eingangsvoraussetzungen für die Sekundarstufe II als abschlussorientierte Standards bzw. als Dispositionen⁶⁰ für die 10. Klasse verwendet und mit in dieser Jahrgangsstufe zu behandelnden Inhalten verknüpft. Hierbei muss jedoch hervorgehoben werden, dass manche Eingangsvoraussetzungen in die Qualifikationsphase nicht explizit vom schulinternen Curriculum berücksichtigt wurden, weshalb insbesondere bei diesen Standards nicht davon ausgegangen werden kann, dass sie von den SuS zum Eintritt in die Qualifikationsphase erreicht wurden: Für die Kompetenz der Problemlösung wurden bis auf den Standard „Die Schülerinnen und Schüler wählen zur Lösung eines Problems geeignete Standardsoftware (Textverarbeitung, Tabellenkalkulation, Erfassen und Verwaltung von Daten, Bildbearbeitung) aus“ alle im Rahmenlehrplan formulierten Eingangsvoraussetzungen als im Rahmen des Themenbereichs „Einführung in das informatische Modellieren und Umsetzung mit einer Programmiersprache“ zu erwerbende Standards gekennzeichnet, wobei diese Unterrichtseinheit inhaltlich den im Rahmenlehrplan für die Einführungsphase formulierten Themenbereich „Algorithmen und Softwareentwicklung“ umfasst⁶¹ und konkret durch eine Einführung in die objektorientierte Java-Programmierung mit den Programmierumgebungen Greenfoot und BlueJ umgesetzt wird.⁶² Im Gegensatz dazu finden die im Rahmenlehrplan für die Kompetenz „Kommunizieren und Kooperieren“ formulierten Eingangsvoraussetzungen im schulinternen Curriculum kaum Beachtung. Lediglich die Nutzung von Rechnernetzwerken zur Kommunikation wird im Rahmen des Themenfelds „Computernetze“ explizit als zu erwerbender Standard formuliert.⁶³ Aus der unvollständigen Integration der Eingangsvoraussetzungen in das schulinterne Curriculum erwächst die Notwendigkeit, bei der nachfolgenden Analyse der Lernvoraussetzungen der SuS genauer darauf zu achten, inwieweit der Erwerb jener Standards zum Beginn der Qualifikationsphase tatsächlich vorangeschritten ist. Dies ist umso wichtiger, als dass bereits im Rahmen der Begründung der Themenwahl für die Unterrichtsreihe festgestellt wurde, dass mit den SuS in der Qualifikationsphase zum Teil Themen aus der Einführungsphase bearbeitet werden, wobei exemplarisch die Auseinandersetzungen mit Betriebssystemen und mit PHP angeführt wurden.⁶⁴ Hieraus folgt, dass für die Genese der für die Unterrichtsstunde zu formulierenden Dispositionen nicht nur auf die abschlussorientierten Standards eingegangen werden darf, sondern auch die Eignung der Eingangsvoraussetzungen dafür geprüft werden muss, welche insbesondere dann vorliegt, wenn die entsprechende Eingangsvoraussetzung in der Einführungsphase nicht oder nur marginal als zu erwerbender Standard berücksichtigt wurde.

⁵⁷ Ebd., S. 11.

⁵⁸ Sofern nicht explizit anders gekennzeichnet, ist mit der Bezeichnung „Rahmenlehrplan“ im Folgenden der Berliner Rahmenlehrplan der Sekundarstufe II im Fach Informatik gemeint.

⁵⁹ Auf eine Nennung der einzelnen Standards wird an dieser Stelle verzichtet.

⁶⁰ Vgl. http://www.member.uni-oldenburg.de/hilbert.meyer/download/7.4.Trainingsbogen_zur_Lernzielanalyse.pdf [05.10.2014], S. 3.

⁶¹ Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_informatik.pdf [05.10.2014], S. 3.

⁶² Vgl. <http://www.humboldtschule-berlin.de/index.php/unterricht/faecher/informatik/103-informatik-unterricht> [05.10.2014].

⁶³ Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_informatik.pdf [05.10.2014], S. 4 f.

⁶⁴ Vgl. 2.1.2 Begründung der Themenwahl für die Unterrichtsreihe.

Da sich der Rahmenlehrplan bei der Formulierung der fachlichen Kompetenzen auf die „Einheitlichen Prüfungsanforderungen in der Abiturprüfung“ (EPA) stützt,⁶⁵ sollen die dort formulierten Kompetenzen und abschlussorientierten Standards bei der Bedingungsanalyse ebenfalls berücksichtigt werden. Die abschlussorientierten Standards werden in der EPA in drei Anforderungsbereichen kategorisiert:

Unter dem Anforderungsbereich I fasst die EPA

- die Wiedergabe von bekannten Sachverhalten aus einem abgegrenzten Gebiet im gelernten Zusammenhang,
- die Beschreibung und Darstellung bekannter Verfahren, Methoden und Prinzipien der Informatik sowie
- die Beschreibung und Verwendung gelernter und geübter Arbeitstechniken und Verfahrensweisen in einem begrenzten Gebiet und in einem wiederholenden Zusammenhang

zusammen.⁶⁶

Der Anforderungsbereich II wird durch

- die selbständige Verwendung (Auswählen, Anordnen, Verarbeiten und Darstellen) bekannter Sachverhalte zur Bearbeitung neuer Frage- oder Problemstellungen unter vorgegebenen Gesichtspunkten in einem durch Übung bekannten Zusammenhang,
- die selbständige Übertragung des Gelernten auf vergleichbare neue Situationen, wobei es entweder um veränderte Fragestellungen oder um veränderte Sachzusammenhänge oder um abgewandelte Verfahrensweisen gehen kann, sowie
- die Anwendung bekannter Verfahren, Methoden und Prinzipien der Informatik zur Lösung eines neuen Problems aus einem bekannten Problemkreis

beschrieben.⁶⁷

Der Anforderungsbereich III umfasst

- die planmäßige Verarbeitung komplexer Gegebenheiten mit dem Ziel, zu selbständigen Gestaltungen bzw. Deutungen, Folgerungen, Begründungen, Wertungen zu gelangen, sowie
- die bewusste und selbständige Auswahl und Anpassung geeigneter gelernter Methoden und Verfahren in neuartigen Situationen. Dabei werden aus gelernten Denkmethoden bzw. Lösungsverfahren die zur Bewältigung der Aufgabe geeigneten selbständig ausgewählt und einer neuen Problemstellung angepasst.⁶⁸

Da die meisten abschlussorientierten Standards in den EPA lediglich am Anforderungsbereich orientiert, jedoch nicht kompetenzorientiert zugeordnet werden, soll sich im Folgenden auf das

⁶⁵ Vgl. https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1406800775&file=sek2_informatik.pdf [06.10.2014], S. 9 f.

⁶⁶ Vgl. http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01_EPA_Informatik.pdf [06.10.2014], S. 10.

⁶⁷ Vgl. ebd., S. 10 f.

⁶⁸ Vgl. ebd., S. 11.

Kompetenzstufenmodell von Fothe bezogen werden, welches sowohl eine kompetenz- als auch eine anforderungsorientierte Zuordnung der abschlussorientierten Standards leistet.

Der Kompetenz „Problemlösen“ ordnet Fothe folgende Standards zu:

- einfaches Erweitern einer vorgegebenen Problemlösung in geübtem Zusammenhang (Anforderungsbereich I)
- planvolles Einsetzen bekannter Informatiksysteme zur Lösung einer neuen Problemstellung aus einem bekannten Bereich (Anforderungsbereich II)
- Nutzen vorhandener Programmbibliotheken für die eigene Problemlösung (Anforderungsbereich II)
- Dokumentieren einer Problemlösung mit angemessenen Darstellungsmitteln (Anforderungsbereich II)
- Durchführen einer komplexen Problemanalyse (Anforderungsbereich III)
- Zerlegen eines gegebenen anspruchsvollen Problems in geeignete Teilprobleme (Anforderungsbereich III)
- Beurteilen der eigenen Modellierung und Problemlösung im Anwendungskontext (Anforderungsbereich III).⁶⁹

Für die in den EPA formulierte Kompetenz „Algorithmen“ ergibt sich bei Fothe folgende Zuordnung:

- Wiedergeben von Begriffsdefinitionen, Regeln, Zusammenhängen, bekannten Verfahren, einfachen Algorithmen, einfachen Modellierungen und Strukturen in einer im Unterricht behandelten Darstellungsform (Anforderungsbereich I)
- Verwenden einfacher Modellierungen und bekannter einfacher Algorithmen (Anforderungsbereich I)
- Analysieren eines gegebenen Algorithmus (Anforderungsbereich II)
- Übertragen von Aufwandsbetrachtungen auf einen vergleichbaren aber nicht bekannten Algorithmus (Anforderungsbereich II)
- Begründen von bestimmten Eigenschaften (z. B. Terminierung, Zeit- und Speicheraufwand) eines gegebenen Algorithmus durch nicht formale Überlegungen (Anforderungsbereich II)
- Entwickeln eines Verfahrens bzw. Algorithmus zur Lösung eines neuen Problems (Anforderungsbereich III).⁷⁰

2.3.2 Lernvoraussetzungen der Schülerinnen und Schüler

Für die thematische Verortung einer Aufgabenstellung, welche die Realisierung eines Softwareprojekts zum Gegenstand hat, bietet der Rahmenlehrplan für die Qualifikationsphase nur wenige Alternativen. Als dritten Themenbereich gibt der Rahmenlehrplan die „Softwareentwicklung“ vor, wobei die SuS nicht nur lernen sollen, durch die Auseinandersetzung mit Algorithmen und Datenstrukturen sowie den Konzepten der deklarativen und der objektorientierten Programmierung Software zu implementieren, sie sollen darüber hinaus durch die Auseinandersetzung mit der objektorientierten Modellierung sowie dem Softwarelebenszyklus ebenfalls lernen, ein Softwareprojekt zu planen und die Güte der implementierten

⁶⁹ Vgl. https://ddi.informatik.uni-erlangen.de/EN/service/events/FachgruppeDDI2008/vortragsfolien/Fothe_giws08.pdf [06.10.2014], S. 17.

⁷⁰ Vgl. ebd. [06.10.2014], S. 14.

Lösungen zu evaluieren.⁷¹ Dieser Themenbereich soll im Leistungskurs entweder im ersten oder im zweiten Kurshalbjahr der Qualifikationsphase behandelt werden.⁷²

Vergleicht man die im Rahmenlehrplan formulierten Inhalte und Kompetenzen des Themenbereichs „Softwareentwicklung“ mit den obigen Ausführungen zur Aufgabenstellung der Stunde,⁷³ so wird ersichtlich, dass die Aufgabenstellung in ihren Lernzielen sowie in den für ihre Lösung von den SuS zu erbringenden Anforderungen zum Teil hinter dem Rahmenlehrplan zurückbleibt. Dieser Umstand wird dadurch bedingt, dass bei der Konzeption der Aufgabenstellung nicht nur auf die Vorgaben des Rahmenlehrplans eingegangen, sondern insbesondere intendiert wurde, durch die Berücksichtigung der Vorkenntnisse und Kompetenzen der SuS eine Passung der Aufgabenstellung für die Klasse sicherzustellen.

Die Analyse der Lernvoraussetzungen ergibt insbesondere für den Aspekt der Vorkenntnisse bzw. Vorerfahrungen der SuS sowie den Aspekt der Lernkompetenzen ein sehr heterogenes Klassenbild. Um diese Heterogenität zu skizzieren, sollen im Folgenden einige für die Aufgabenstellung sowie den im Rahmenlehrplan formulierten Themenbereich „Softwareentwicklung“ wichtige inhalts- und prozessbezogene Kompetenzen definiert werden, wobei die Stufung einer Kompetenz jeweils entweder auf Meyers pragmatisches Kompetenzstufenmodell,⁷⁴ auf Blooms Lernzieltaxonomie⁷⁵ oder auf Fothes Zuordnung der Standards der EPA rekurriert.⁷⁶ Aus Komplexitätsgründen sollen für die Kompetenzen jedoch nur diejenigen Stufen angegeben werden, welche bei der Analyse in der Klasse auch tatsächlich identifiziert werden konnten.

Aus der Betrachtung des schulinternen Curriculums in Verbindung mit der exemplarischen Themenübersicht für das Fach Informatik folgt, dass im Profilierungskurs grundlegende Elemente der Programmiersprache Java behandelt wurden, wobei in diesem Zusammenhang die folgenden Inhalte Gegenstand des Unterrichts waren:

- Algorithmen im Kleinen: Sequenz, Auswahl, Wiederholung, Variablenkonzept, einfache Datentypen, Parameterkonzept
- Grundlagen der objektorientierten Modellierung und Programmierung (Klasse, Attribut, Methode, Exemplar)
- Verwendung vorhandener Klassen
- Modellierung und Implementierung eigener Klassen
- Exemplarische Einblicke in Klassenbeziehungen und Objektinteraktion (Aggregation, Assoziation, Vererbung).⁷⁷

Wie aus dem tabellarischen Überblick über die in der Unterrichtsreihe behandelten Themen ersichtlich wird,⁷⁸ waren nur solche Elemente von C# Gegenstand des Unterrichts, welche den SuS bereits aus Java bzw.

⁷¹ Vgl. https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1406800775&file=sek2_informatik.pdf [06.10.2014], S. 21.

⁷² Vgl. ebd. [06.10.2014], S. 28.

⁷³ Vgl. 2.2 Aufgabenstellung der Stunde und Aufgabenanalyse.

⁷⁴ Vgl. Meyer 2012, S. 157.

⁷⁵ Bloom (zit. nach Sitte 2001, S. 474 f.).

⁷⁶ Vgl. https://ddi.informatik.uni-erlangen.de/EN/service/events/FachgruppeDDI2008/vortragsfolien/Fothe_giws08.pdf [07.10.2014].

⁷⁷ Vgl. http://www.humboldtschule-berlin.de/images/pdf/curricula/schulinternes_curriculum_informatik.pdf [07.10.2014], S. 3.

der Klassenstufe 10 bekannt waren. Dennoch zeigten die SuS bei der Erarbeitung jener Sprachelemente in C# ein sehr heterogenes Vorwissen über diese. So waren für die inhaltsbezogene Kompetenz „Kenntnisse über die grundlegenden Elemente objektorientierter Programmiersprachen“ zu Beginn der Reihe die folgenden, am Modell von Bloom orientierten Stufen innerhalb der Klasse beobachtbar:

1. Die SuS können die Sprachelemente sowie ihre Funktion benennen. (Kenntnisse)
2. Die SuS können die Sprachelemente in ihrer Struktur sowie ihrer Funktionsweise erläutern und wissen, in welchen Kontexten sie angewendet werden können. (Verstehen)
3. Die SuS wählen für einfache Programmieraufgaben selbständig die erforderlichen Sprachelemente aus und implementieren diese semantisch und syntaktisch korrekt. (Anwendung)
4. Die SuS sind in der Lage, ähnliche Sprachelemente zwischen den Programmiersprachen zu identifizieren und diese bezüglich ihrer Struktur und Funktionsweise vergleichend gegenüberzustellen. (Analyse)

Hierbei ergab sich eine Zweiteilung der Klasse, wobei der leistungsschwächere Teil der Klasse oftmals nur die Kompetenzstufe 1 und nur bei einfacheren Sprachelementen auch die Stufe 2 erreichte, während die leistungsstärkere Hälfte oftmals die Ähnlichkeit zwischen den Sprachelementen in C# und Java erkannte und sich daher in der Regel auf der vierten, in seltenen Fällen auf der dritten Kompetenzstufe befand. Hierbei muss jedoch erwähnt werden, dass das oftmalige Herstellen von Relationen zwischen Java und C# unter anderem dem Umstand geschuldet war, dass sich zwei der leistungsstärkeren Schüler das Programmieren in C# bereits autodidaktisch beigebracht hatten. Mit Blick auf die leistungsschwächere Hälfte wurden die Arbeitsaufträge zur Erarbeitung der Sprachelemente in C# in der Form gestaltet, dass die SuS sich das jeweilige Element zunächst in seiner Struktur und Funktionsweise erarbeiten und dieses anschließend in diversen Kontexten bzw. Szenarien anwenden mussten. Hierdurch sollte sichergestellt werden, dass die SuS das jeweilige Sprachelement nach der erneuten Erarbeitung in C# auch anwenden können und somit für das jeweilige Element mindestens die dritte Kompetenzstufe erreichen.

Für die inhaltsbezogene Kompetenz der „Kenntnisse über den Aufbau und die Funktionsweise von Betriebssystemen“ ergibt sich ein homogeneres Bild. Die Inkorporation der in der Reihe „Betriebssysteme“ behandelten Inhalte Aufgaben des Betriebssystems, Anmeldung und Authentifikation, Dateisystem sowie Zugriffsrechte erfolgte in der Klasse auf drei Stufen:

1. Die SuS können die Aufgaben des Betriebssystems erläutern, sie können die Struktur des Dateisystems sowie die Zugriffsrechte erläutern und wissen, wie sie sich beim System anmelden / authentifizieren können. (Verstehen)
2. Die SuS können das Betriebssystem über das Terminal / die Eingabeaufforderung mithilfe bekannter Befehle oder über geeignete Oberflächen manipulieren. (Anwendung)
3. Die SuS können das Terminal / die Eingabeaufforderung nutzen, um sich selbständig Befehle zur Manipulation des Betriebssystems anzueignen. Weiterhin sind sie in der Lage, die aus Nutzersicht kennengelernten Bestandteile den jeweiligen Bibliotheken des Betriebssystems zuzuordnen. (Analyse)

Zum Ende der Reihe über Betriebssysteme erreichten alle SuS mindestens die zweite Kompetenzstufe. Die dritte Kompetenzstufe wurde hingegen nur von der leistungsstärkeren Hälfte der Klasse erreicht.

⁷⁸ Vgl. 2.1.3 Struktur der Unterrichtsreihe.

Für die prozessbezogene Kompetenz des Problemlösens gibt das schulinterne Curriculum für den Profilierungskurs Informatik die folgenden, mit dem Abschluss der Unterrichtseinheit „Einführung in das informatische Modellieren und Umsetzung mit einer Programmiersprache“ zu erreichenden Standards vor:

Die SuS...

- beschreiben algorithmische Abläufe umgangssprachlich und grafisch,
- modellieren einfache Abläufe mit Algorithmen (Sequenz, Auswahl, Wiederholung),
- setzen Algorithmen in Programme um.

Für das Problemlösen gestaltet sich eine zuverlässige Einschätzung über das Kompetenzniveau der SuS als schwierig, da ich während meines Praktikums nur wenige damit verbundene Standards als Performanz der SuS überhaupt beobachten konnte. Dies war dem Umstand geschuldet, dass das Produkt / das Ergebnis der zuvor durchgeführten Problemlöseaufgaben oftmals nur in der Implementierung bestand, wohingegen die Gedanken der SuS zur Anforderungsanalyse und Planung in der Tiefenstruktur verborgen blieben.

Von den Standards, welche Fothe der Kompetenz „Problemlösen“ zuordnet, konnte ich folgende in der Performanz der SuS identifizieren

- einfaches Erweitern einer vorgegebenen Problemlösung in geübtem Zusammenhang (Anforderungsbereich I)
- planvolles Einsetzen bekannter Informatiksysteme zur Lösung einer neuen Problemstellung aus einem bekannten Bereich (Anforderungsbereich II)
- Nutzen vorhandener Programmbibliotheken für die eigene Problemlösung (Anforderungsbereich II)

Für den Kompetenzbereich „Algorithmen“ wurden folgende Standards ersichtlich:

- Wiedergeben von Begriffsdefinitionen, Regeln, Zusammenhängen, bekannten Verfahren, einfachen Algorithmen, einfachen Modellierungen und Strukturen in einer im Unterricht behandelten Darstellungsform (Anforderungsbereich I)
- Analysieren eines gegebenen Algorithmus (Anforderungsbereich II)
- Entwickeln eines Verfahrens bzw. Algorithmus zur Lösung eines neuen Problems (Anforderungsbereich III).

Die SuS sind also in der Lage, für eine gegebene Problemstellung eine eigenständig erarbeitete Lösung zu entwickeln. Da als Produkt bzw. Erzeugnis jedoch lediglich die Implementierung vorliegt, kann nicht evaluiert werden, in welchem Umfang und in welcher Qualität die anderen Bereiche des Softwarelebenszyklus Eingang in die Problemlösung (und damit auch in die Problemlösefähigkeit der SuS) gefunden haben bzw. ob in den Phasen der Anforderungsanalyse und der Planung im Unterricht erlernte Techniken angewendet wurden oder diese lediglich auf informellem Handeln beruhen.

Dass für die einzelnen Phasen der Softwareentwicklung keine Zwischenprodukte verlangt werden, hängt mit den Bemühungen des Fachlehrers zusammen, Problemlöse- und Programmieraufgaben möglichst oft im Kontext von Lernaufgaben einzusetzen, wobei ein möglichst hohes Maß an selbstreguliertem Lernen intendiert wird: Die SuS sollen die vom Lehrer gestellten Arbeitsaufträge bearbeiten, in welchen Sozialformen und Handlungsmustern diese umgesetzt werden sollen, wird jedoch oftmals nicht explizit durch den Arbeitsauftrag festgelegt und bleibt dann der Schülerin / dem Schüler überlassen. Dabei beziehen

sich die als Lernaufgaben formulierten Arbeitsaufträge auf in der Klasse ritualisierte Handlungsmuster des forschend-entdeckenden Lernens, denen die SuS jedoch mit unterschiedlichen Lernkompetenzen begegnen:

Für die Sachkompetenz⁷⁹ unterscheide ich, orientiert an Meyers pragmatischem Kompetenzstufenmodell, folgende Stufen:

1. Die SuS erwerben sachlich-fachliche Kenntnisse aus einfachen, inhaltlich auf den zu erwerbenden Gegenstand reduzierten und vom Lehrer vorgegebenen Darstellungstexten. (unreflektiertes Nachvollziehen)
2. Die SuS erwerben sachlich-fachliche Kenntnisse, indem sie vom Lehrer vorgegebene oder selbst gefundene Sachtexte auf den zu erwerbenden Gegenstand reduzieren sowie eigenständig durchgeführte, vom Lehrer vorgeschriebene Handlungen in ihrer Bedeutung für den zu erwerbenden Gegenstand vom Lehrer angeleitet reflektieren. (Handeln nach Vorgabe)
3. Die SuS erwerben sachlich-fachliche Kenntnisse, indem sie eigenständig durchgeführte, vom Lehrer vorgeschriebene Handlungen in ihrer Bedeutung für den zu erwerbenden Gegenstand ohne Hilfe reflektieren. (Handeln nach Einsicht)
4. Die SuS erwerben sachlich-fachliche Kenntnisse, indem sie den zu erwerbenden Gegenstand mit fachbezogenen oder fachfremden Kenntnissen und Einsichten eigenständig und ohne Vorgabe der Methoden in Beziehung setzen. (selbständige Prozesssteuerung).

Hierbei konzentriert sich die Klasse wieder im mittleren Kompetenzfeld, wobei die leistungsschwächeren SuS eher dazu tendieren, ihren Wissenserwerb stark an den Vorgaben des Lehrers auszurichten (Handeln nach Vorgabe), während sich die leistungsstärkeren Schüler eher trauen, die Vorgaben zu verlassen, indem sie die vorgegebenen Handlungsaufträge modifizieren, Schrittfolgen einfügen oder auslassen (Handeln nach Einsicht). Die Beobachtung der Ausprägung der Sachkompetenz der SuS bestätigt somit eine Aussage Meyers, nach der die klare Strukturierung des Unterrichts für leistungsschwächere SuS einen ganz wesentlichen Schlüssel zum Erfolg darstelle,⁸⁰ wobei er (insbesondere für leistungsschwächere SuS) eine Überlegenheit der direkten Instruktion beim fachlichen Lernen festmacht.⁸¹ Der Fall, dass die SuS auf der Kompetenzstufe 1 stehen bleiben, tritt in der Regel nur in den Fällen auf, in denen sie den Unterrichtsgegenstand nicht richtig verstanden haben oder sich durch diesen überfordert fühlen. Kompetenzstufe 4 wird regelmäßig von den beiden Schülern erreicht, welche sich auch eigenständig C# beigebracht haben. Sie verfügten sowohl in meiner Unterrichtseinheit als auch in der vorhergehenden bzw. nachfolgenden Einheit über Betriebssysteme bzw. PHP über ein großes fachliches Vorwissen.

Der Gegensatz zwischen dem eher auf Sicherheit bedachten, stark am Lehrer orientierten Kenntniserwerb der leistungsschwächeren und dem eher experimentierfreudigen, vom Lehrer teilweise emanzipierten Ansatz der leistungsstärkeren SuS zeigt sich auch in der Methodenkompetenz: Die SuS sind es gewohnt, mehrere Lernaufgaben in einem Block, das heißt nacheinander in Einzel-, Partner- oder Gruppenarbeit zu bearbeiten, ohne dass diese Arbeit durch Zwischensicherungen im Plenum unterbrochen wird. Dabei werden die Lernaufgaben eines Blocks meist in der Art konzipiert, dass sie sukzessive mindestens die ersten drei Stufen der Bloomschen Lernzieltaxonomie abdecken, wobei Analyseaufgaben ebenso häufig vertreten sind. Bei den leistungsschwächeren SuS konnte ich beobachten, dass sie einen Großteil der Zeit, die ihnen für den jeweiligen Block zur Verfügung steht, für diejenigen Aufgaben aufwenden, welche sich auf den ersten beiden

⁷⁹ Vgl. Meyer 2012, S. 154.

⁸⁰ Vgl. ebd., S. 123.

⁸¹ Vgl. ebd., S. 58.

Stufen der Kenntnis und des Verstehens bewegen. Dabei machen sie sich detailliert Notizen über den zu inkorporierenden Lerngegenstand, was in manchen Fällen jedoch auch dazu führt, dass beinahe alle Informationen, welche die SuS aus den Aufgaben und den damit verbundenen Medien gewinnen können, von diesen auch festgehalten werden, sodass sie nur im geringen Maße Informationen in ihrer Bedeutung für den Gegenstand und dessen Verständnis gewichten. Bei den leistungsstärkeren SuS liegt der Fokus eher auf den Anwendungs- und Analyseaufgaben. Sie sehen die Kenntnis- und Verständnisaufgaben als notwendige Vorarbeiten an, über die sie sich die notwendigen Tools, d. h. die notwendigen Kenntnisse und Fähigkeiten aneignen können. Sie betrachten also (im Gegensatz zu den leistungsschwächeren Schülern) den in den Lernaufgaben behandelten Gegenstand nicht als für sich allein stehendes Lernobjekt, sondern immer im Kontext der Handlungsmöglichkeiten, welche den SuS durch die Auseinandersetzung mit diesem eröffnet werden. Daher stellen die Kenntnis- und Verständnisaufgaben für die leistungsstärkeren SuS in erster Linie hinführende Zwischenschritte dar, weshalb die hierbei gewonnenen Informationen seltener und in einem sehr viel geringeren Umfang festgehalten werden als bei den leistungsschwächeren SuS. Bei der Lernkompetenz der SuS ist besonders auf die Gründe zu achten, aus denen sich die Fokussierung auf die Kenntnis- und Verständnisaufgaben ergibt: Bei den leistungsschwächeren SuS ergibt sich der Umstand, dass sie häufig die Anwendungs- und Analyseaufgaben nicht vollständig bearbeiten, als Folge des methodischen Vorgehens, wonach das detaillierte Aufzeichnen beim Lösen der Aufgaben der Stufen 1 und 2 oftmals so viel Zeit in Anspruch nimmt, dass die SuS die nachfolgenden Aufgaben in der verbleibenden Zeit nicht adäquat bearbeiten können. Bei den beiden Schülerinnen machte ich jedoch weiterhin die alarmierende Beobachtung, dass sich die Fokussierung auf die Kenntnis- und Verständnisaufgaben ebenso als Folge einer Vermeidungsstrategie ergibt, welche genetisch durch eine geringe Selbstkompetenz in Verbindung mit einer geringen Handlungs- und Selbstwirksamkeit bedingt ist. Die Schülerinnen zeigen dabei ein Verhalten, welches durch Jerusalem empirisch belegt wurde: „Schüler mit geringen Fähigkeitsüberzeugungen zum Beispiel bevorzugen leichtere Aufgaben, strengen sich bei schwierigen Problemen weniger an und geben eher auf als Schüler mit hohen Kompetenzüberzeugungen.“⁸² Der hohe zeitliche Aufwand, welchen die Schülerinnen in die Aufgaben der Stufen 1 und 2 investieren, ist also Teil einer Lernstrategie, wonach sie versuchen, die aus ihrer Sicht erfolgversprechenden Aufgaben so gut und genau wie möglich zu lösen, während sie nicht bereit sind, auf diejenigen Aufgaben, für die sie von vornherein schlechte Ergebnisse erwarten, ein hohes Maß an Lernressourcen aufzuwenden. Bedenkt man einerseits, wie stark sich die leistungsschwächeren SuS sich am Lehrer und den Kenntnis- und Verständnisaufgaben orientieren, und wie die leistungsstärkeren SuS andererseits ebendiese Aufgaben vernachlässigen (indem sie sich beispielsweise keine Notizen machen) und in den Aufgaben der höheren Stufen teilweise autark vom Lehrer agieren, so wird die von Jerusalem formulierte Differenz zwischen dem Selbstkonzept und der objektiv wahrnehmbaren Leistung der SuS deutlich: „Ein geringes Selbstkonzept etwa trägt angesichts neuer Aufgaben zu einer Unterschätzung eigener Leistungsfähigkeiten und auch der eigenen Leistungsergebnisse bei, während ein hohes Selbstkonzept eher mit einer leichten Überschätzung einhergeht.“⁸³ Die schulbezogene Selbstwirksamkeitserwartung der SuS sowie ihr Selbstkonzept sind dabei nicht nur dafür entscheidend, wie die SuS arbeiten, sondern auch mit wem. Da bei der Bearbeitung von Arbeitsaufträgen oftmals keine Sozialform vom Fachlehrer vorgegeben wird, arbeiten meist die SuS zusammen, die auch nebeneinander sitzen. Hierbei habe ich beobachtet, dass die SuS, welche sich zusammensetzen, eine ähnlich hohe Selbstwirksamkeitserwartung kommunizieren und auch ihr methodisches Handeln zur Lösung von Arbeitsaufträgen aufeinander abstimmen, was sich insbesondere darin zeigt, dass die SuS, welche eine

⁸² Jerusalem 1993, S.14.

⁸³ Ebd., S. 13.

Aufgabe zusammen bearbeitet haben, sich zu den gleichen Aufgaben in jeweils gleichen Detailgraden Notizen machen bzw. ihre Ergebnisse durch die Nutzung gleicher Handlungsmuster / Inszenierungsformen und im gleichen Maße detailliert festhalten. Hierbei hat sich die Klasse in zwei Dreier- und drei Zweiergruppen aufgeteilt, welche sich in ihrer Selbstwirksamkeit und ihrer Performanz weitgehend homogenisiert haben.

Trotz der Unterschiede, welche zwischen den methodischen Handlungsweisen der Gruppen beobachtbar sind, zeigten alle SuS in den vorangegangenen Unterrichtsstunden die Bereitschaft, sich mit dem Gegenstand der Reihe auseinanderzusetzen. Da ein Großteil der SuS eine besondere Affinität zur praktischen Informatik und dabei insbesondere zur Anwendung des von ihnen Gelernten in produktorientierten Bezügen zeigte, bemühte ich mich insbesondere in der Konzeption der Aufgaben, diesem Interesse Folge zu leisten. Hierfür ritualisierte ich die Aufgabenfolge für das jeweils zu erarbeitende Sprachelement / die jeweils zu erarbeitende Gruppe von Sprachelementen in der Form, als dass die ersten Aufgaben der Erarbeitung der grundlegenden Kenntnisse bzw. des grundlegenden Verständnisses über die Struktur und Funktionsweise der Sprachelemente dienen, während die weiterführenden Aufgaben Implementierungen der Sprachelemente in konkreten Anwendungskontexten zum Gegenstand hatten, welche die SuS für die besonderen Spezifika / Ausnahmen bei der Anwendung des jeweiligen Elements sensibilisieren sollten. Die weiterführenden Aufgaben bestanden entweder aus Übungsaufgaben, wobei die SuS eine Implementierung nach spezifischen Vorgaben erstellen und debuggen sollten, oder aus Analyseaufgaben, in denen die SuS eine vorgegebene Implementierung unter bestimmten Aspekten / Fragestellungen analysieren und zum Teil auch evaluieren sollten. Da sich einige Schüler regelmäßig mit Online-Spielen beschäftigen und sogar auf verschiedenen Niveaus, differenziert nach der Komplexität der von ihnen erstellten Applikation und der dabei von ihnen verwendeten Technologie / Sprache, eigene Spiele entwickeln, betonte ich dabei den Einsatz der erarbeiteten Sprachelemente in der Entwicklung von Spielen.⁸⁴ Weiterhin wurde versucht, diesem Interesse nachzukommen, indem in die Implementierungsaufgaben oftmals der Entwurf bzw. die Programmierung grafischer Oberflächen anteilig integriert wurde. Neben der Herstellung von Beziehungen zwischen diesen persönlichen Interessen der SuS und dem Gegenstand der Unterrichtsreihe bemühte ich mich in den vorangegangenen Stunden weiterhin, Bezüge zwischen dem jeweils behandelten Stundenthema und der zuvor behandelten Unterrichtseinheit zu Betriebssystemen herzustellen, indem ich beispielsweise mit den SuS erläutert habe, was eigentlich im Hintergrund passiert, wenn sie über eine Methode in C# das Betriebssystem ansprechen, wobei als Grundlage hierfür in der ersten Stunde ein Überblick über das .Net Framework gegeben wurde. Die Berücksichtigung des Vorwissens und der persönlichen Interessen der SuS in der Konzeption der Auseinandersetzung mit dem Unterrichtsgegenstand intendiert die Aufrechterhaltung des bei den SuS anfänglich bestehenden Interesses am Gegenstand selbst, wobei diese Bemühungen umso wichtiger werden, je komplexer (und somit vermutlich auch je schwieriger) der Gegenstand wird. Das Handlungsmuster des Gruppenpuzzles, welches in der letzten Stunde für die Erarbeitung der Kontrollstrukturen angewendet und dessen letzter Schritt durch das gegenseitige Erklären der Kontrollstrukturen in dieser Stunde durchgeführt wurde, kennen die SuS in dieser Form bereits aus der ersten Stunden meiner Reihe. Ebenfalls erhielten die SuS in der Reihe mehrmals die Aufgabe, sich mit der MSDN-Dokumentation auseinanderzusetzen, sodass auch die Vervollständigung der vorgegebenen Dokumentation für die SuS ohne methodische Schwierigkeiten durchführbar sein sollte. Das zentrale Handlungsmuster des Dreischritts der Analyse, Planung und Implementierung ist den SuS in der Form, wie er

⁸⁴ Ein Schüler hat mir sogar gezeigt, wie er ein in der Stunde zuvor behandeltes Sprachelement in seinem Spiel verwendet hat.

für die Erarbeitung des Projekt gefordert wird, nicht bekannt, weshalb er auch zunächst exemplarisch im Plenum durchgeführt werden soll.

Die Konzeption der Unterrichtsstunde wird nicht durch das Sprachvermögen der SuS beeinträchtigt, da lediglich ein Schüler einen Migrationshintergrund hat, wobei alle SuS in Deutschland geboren sind und Deutsch als Muttersprache sprechen.

2.3.3 Lehrvoraussetzungen

Da ich mir C# und die damit konnotierten Technologien und Konzepte über mehrere Jahre hinweg angeeignet und dabei mit mehreren mittelständischen Unternehmen Softwareprojekte in dieser Sprache umgesetzt habe, verfüge ich über ein umfassendes Sachwissen im Bezug auf den zu unterrichtenden Gegenstand, wobei ich mich ebenso in der Lage sehe, diesen Gegenstand in seiner Komplexität so zu reduzieren, dass er durch die SuS angeeignet werden kann.

Weiterhin empfinde ich mein Handlungs- und Methodenrepertoire als dahingehend hinreichend ausgebildet, als dass ich die in der Planung der Stunde bereits getroffenen Ziel- und Inhaltsentscheidungen mit adäquaten Handlungs- und Methodenentscheidungen verbinden und auf unerwartete Ereignisse während der Stundendurchführung mit einer adäquaten Anpassung der Handlungs- und Methodenentscheidungen reagieren kann, wie es beispielsweise in der zweiten Unterrichtsstunde meiner Reihe notwendig war.⁸⁵ Diese Einschätzung über die Angemessenheit meiner Handlungs- und Methodenentscheidungen ergibt sich aus dem Umstand, dass diese Entscheidungen in der objektiven Beurteilung der bisher vorgelegten Unterrichtsentwürfe durch meine Fachlehrer ebenso als angemessen bzw. passend evaluiert wurden. Der überwiegende Teil der Rückmeldungen hierzu stammt jedoch aus der Geschichte und nicht aus der Informatik. Ebenso hatten meine Unterrichtskonzeptionen mit Schwerpunkten auf speziellen didaktischen Themen wie der Inklusion oder der Binnendifferenzierung bis jetzt geschichtsdidaktische Bezüge. Weiterhin habe ich insbesondere während meines Geschichtspraktikums und der vorhergehenden Stunden innerhalb dieses Praktikums bemerkt, dass ich bezüglich der Sozialformen und Handlungsmuster bestimmte Präferenzen entwickelt habe, wobei ich die SuS oftmals in Paaren oder Gruppen zusammenarbeiten lasse und dabei häufig solche Handlungsmuster nutze, welche das kooperative Lernen zum Gegenstand haben.⁸⁶ Obwohl ich mich in Fragen des Sach- sowie des Handlungs- und Methodenwissens (und damit der Unterrichtsplanung) als durchaus kompetent ansehe, fällt es mir aufgrund meiner geringen Praxiserfahrung schwer, meine sozial-kommunikative Kompetenz oder die Güte meiner Klassenführung zu evaluieren. Diese beiden Aspekte sollen daher die Schwerpunkte darstellen, unter denen die Hospitanten meinen Unterricht beobachten sollen.

In diesem Zusammenhang möchte ich mir selbst folgende Entwicklungsaufgaben stellen:

Ich möchte lernen,

- meinen Unterricht / meine Planung ggf. an die von den SuS eingebrachten Impulse anzupassen.⁸⁷
- bestehende oder von mir eingeführte Rituale und Regeln während des Unterrichts konsequent durchzusetzen.

⁸⁵ Vgl. hierzu das Protokoll im Praktikumsbericht von Herrn S. zu meiner Stunde vom 11.09.2014.

⁸⁶ Vgl. <http://www.klett.de/web/uploads/ce7c91814c3556284411ed4feaf1e46fc53e87b3.pdf> [08.10.2014].

⁸⁷ Dies ist notwendig, da, wie Meyer es formuliert, „nicht nur die planende Lehrkraft, sondern auch die Schülerinnen und Schüler ‚Subjekte‘ der didaktischen Strukturierung sind“ (Meyer 2012, S. 177).

- die Komplexität meiner Sprache soweit zu reduzieren, dass sie für die jeweilige Klassenstufe angemessen ist und zum Verständnis des Unterrichtsgegenstands beiträgt.

2.3.4 Institutionelle Rahmenvorgaben, materielle Lernvoraussetzungen und Raumausstattung

Neben der bereits erwähnten Umbenennung der Jahrgangsstufe stellt die Stundeneinteilung an der Humboldt-Schule ein weiteres, schulspezifisches Charakteristikum dar. Hierbei wird in der Oberstufe nicht in den klassischen Einzel- (45 Minuten) oder Doppelstunden (90 Minuten) unterrichtet, der Unterricht findet ausschließlich in 75-Minuten-Stunden statt.

Der Raum, in dem die hier vorgestellte Unterrichtsstunde durchgeführt wird, stellt für 24 SuS Arbeitsplätze mit Desktoprechnern zur Verfügung, wobei 18 Arbeitsplätze entlang den Wänden U-förmig angeordnet und die übrigen sechs Arbeitsplätze als „Insel“ in der Mitte des Klassenraums fixiert sind, wobei sich jeweils zwei der sechs Schüler gegenüber sitzen. Die U-förmige Anordnung ist dahingehend problematisch, als dass die Tische, auf denen die Rechner stehen, direkt an der Wand entlang gehen, sodass die SuS beim Arbeiten nicht zur Mitte des Klassenraums schauen, sondern davon weg zu der Wand, an der ihr Tisch steht. Weiterhin sind die Informatikräume für die Anzahl der Arbeitsplätze verhältnismäßig klein, weshalb nach einer Unterrichtsstunde die Belüftung oftmals schlecht ist. Auf den Arbeitsplätzen der SuS stehen dabei keine einzelnen Workstations, sondern „graphische Terminalarbeitsplätze mit Farbbildschirm, Tastatur und Maus, die keine Festplatte besitzen und keiner lokalen Software-Installation bedürfen. Sie sind platzsparend und besitzen in der Regel auch keine Lüfter, so dass sie - ideal für den Unterrichtseinsatz - absolut geräuschlos sind.“⁸⁸ Da die Rechnerarbeitsplätze also mit dem Schulnetzwerk verbunden sein müssen, ist eine Änderung der Tischanordnung nur schwer realisierbar. Neben den festen Rechnern verfügt die Schule desweiteren über 24 Laptops, auf denen neben Ubuntu, UNIX, Linux und Windows XP, welche auch für die Desktoprechner zur Verfügung stehen, ebenso Windows 7 installiert ist. Da die SuS in meiner Reihe mit der Microsoft IDE „Visual Studio“ arbeiten, wählte ich zu Beginn der Reihe die Laptops und das Betriebssystem Windows 7 als Arbeitsumgebung, wobei ich die IDE als Administrator auf jedem Laptop manuell vorinstallieren musste. Die Laptops besitzen dabei jeweils eine andere ID und ein anderes Passwort für den Administratorzugang. Hieraus ergibt sich der Umstand, dass vor der Stunde für jeden ausgeteilten Laptop ein anderes Passwort eingegeben werden muss, wobei die Administratoranmeldung notwendig ist, da das Visual Studio nur in der Rolle des Administrators ausgeführt werden kann. Die Laptops müssen daher rechtzeitig vor Stundenbeginn ausgeteilt und hochgefahren werden. Als weitere Präsentationsmaterialien stehen dem Lehrer ein portabler Beamer und ein Whiteboard zur Verfügung. Hierbei muss die Lehrperson darauf achten, dass das Bild des Beamers nicht allzu hoch über die Tafel projiziert wird, da die durchgehenden, ebenfalls U-förmig angeordneten Halogenlampen in dem Raum relativ tief hängen und dadurch den SuS ins Bild ragen würden, welche im Klassenraum an der hinteren, d. h. der am weitesten vom Lehrer entfernten Wand sitzen.

2.4 Lernziele

Bei der Konzeption von Lernzielen unterscheide ich zwischen Grob- und Feinzielen. Zwar geben die Grobziele lediglich an, was im Rahmen einer Unterrichtsreihe insgesamt erreicht werden soll,⁸⁹ und gehen daher nicht direkt auf die einzelne Unterrichtsstunde als solche ein, andererseits leiten sich aus den Grobzielen die

⁸⁸ <http://www.humboldtschule-berlin.de/index.php/unterricht/faecher/informatik/105-informatik-ausstattung> [08.10.2014].

⁸⁹ Vgl. https://www.wida.wiwi.uni-due.de/fileadmin/fileupload/BWL-WIDA/PDF-Dokumente/Leitfaden_Unterrichtsentwurf_2013-03.pdf, S. 13 [08.10.2014].

Feinziele der einzelnen Unterrichtsstunde ab.⁹⁰ Aus diesem Grund sollen vor der Aufzählung der durch die Stunde verfolgten Feinziele diejenigen Grobziele der Reihe enumeriert werden, aus denen sich die Feinziele generieren. Um Wiederholungen zu vermeiden, werden in diesem Kapitel die Feinziele lediglich benannt. Die Begründung der Zielauswahl sowie die Zuordnung der Feinziele zu den geplanten Inhalten der Unterrichtsstunde werden im Zuge der didaktischen Strukturierung vorgenommen.

2.4.1 Durch die Unterrichtsstunde berührte Grobziele der Reihe

Kognitive Lernziele:

Zur Hierarchisierung der kognitiven Lernziele beziehe ich mich auf die kognitive Lernzieltaxonomie der Arbeitsgruppe Bloom.⁹¹ Der Name der Stufe steht dabei jeweils in Klammern am Ende des jeweiligen Lernziels, dem diese Stufe zugeordnet wurde.

Die SuS sollen...

- lernen, die verschiedenen Strukturen, in die der Programmcode unterteilt ist, zu benennen und ihre Bedeutung für den Code zu erläutern. (Verständnis)
- lernen, die Struktur und die Funktion unbekannter Elemente einer Programmiersprache zu analysieren, indem sie sich selbständig in vorgegebenen Dokumentationen über diese informieren und ihre Erkenntnisse durch exemplarisches Implementieren der Elemente in verschiedenen Kontexten verifizieren. (Analyse)
- lernen, Lösungen für komplexere Probleme zu realisieren, indem sie die sich aus dem Problem ergebenden Anforderungen analysieren, aus neuen und bekannten Sprachelementen Algorithmen zur Umsetzung jener Anforderungen formulieren und diese Algorithmen programmatisch umsetzen. (Synthese)
- lernen, eigenständig die Güte ihres Programms zu beurteilen, indem sie unter Generierung verschiedener Anwendungsfälle testen, ob die funktionalen Anforderungen an das Programm auch durch die konkrete Implementierung geleistet werden können. (Begründung)
- lernen, Sprachelemente in einem kurzen, zeitlichen Rahmen und reduziert auf die wesentlichen Eigenschaften, d. h. solche Eigenschaften, welche den Aufbau oder die Funktionsweise des Elementes beschreiben, gegenüber ihren MitschülerInnen zu vermitteln, indem sie jene wesentlichen Eigenschaften aus ihrem bisher gesammelten Wissen über das Element selektieren und diese anschließend in einer strukturgebenden Darstellungsreihenfolge organisieren. (Analyse)

Affektive Lernziele:

Um die affektiven Lernziele bezüglich ihres Grades der Internalisierung zu hierarchisieren, beziehe ich mich auf Krathwohls Internalisierungsstufen der affektiven Taxonomie.⁹² Der Name der Stufe steht dabei jeweils in Klammern am Ende des jeweiligen Lernziels, dem diese Stufe zugeordnet wurde.

Die SuS sollen...

⁹⁰ Vgl. http://ddi.informatik.hu-berlin.de/hausarbeiten/ha_mis3.html [08.10.2014].

⁹¹ Bloom (zit. nach Sitte 2001, S. 474 f.).

⁹² Vgl. Meyer (zit. nach http://www.zfsl-recklinghausen.nrw.de/Seminar_GyGe/Seminarprogramm/Lernzielkonzept/Anhang/Anhang_5/ [08.10.2014]).

- lernen, in ähnlich beschaffenen bzw. ähnlich aussehenden Codeabschnitten auf geringfügige Unterschiede zu achten und dabei zu prüfen, ob bzw. wie sich die Unterschiede auf die Funktionalität des jeweiligen Abschnitts auswirken, indem sie die Funktionalitäten der Codeabschnitte vergleichend gegenüberstellen und auch kleinere Abweichungen als mögliche Folgen der Unterschiede wahrnehmen. (Aufmerksamwerden / Beachten)
- erkennen lernen, dass die Zusammenarbeit mit anderen SuS einen Mehrwert für ihren eigenen Wissensstand generiert. (Organisation)
- lernen, Ergebnisse / Erzeugnisse und Positionen / Haltungen ihrer Mitschülerinnen und Mitschüler kritisch zu reflektieren und ihren eigenen Ergebnissen / Erzeugnissen bzw. Positionen / Haltungen als mögliche Alternative unvoreingenommen, d. h. anhand objektiver Bewertungskriterien, gegenüberzustellen. (Charakterisierung durch einen Wert oder eine Wertstruktur)
- die Bereitschaft entwickeln, neue Impulse aufzunehmen und in ihren bisher erarbeiteten Wissenskontext zu integrieren bzw. diesen Wissenskontext neu zu organisieren. (Aufmerksamwerden / Beachten)
- lernen, ein Interesse daran zu entwickeln, dass ihre Mitschülerinnen und Mitschüler die Mindestanforderungen erfüllen. (Werten)
- lernen, bewusst Verantwortung für den Lernfortschritt ihrer Mitschülerinnen und Mitschüler zu übernehmen. (Werten)

2.4.2 Aus den Grobzielen abgeleitete Feinziele der Unterrichtsstunde

Kognitive Lernziele:

Zur Hierarchisierung der kognitiven Lernziele beziehe ich mich auf die kognitive Lernzieltaxonomie der Arbeitsgruppe Bloom.⁹³ Der Name der Stufe steht dabei jeweils in Klammern am Ende des jeweiligen Lernziels, dem diese Stufe zugeordnet wurde.

Die SuS sollen...

- lernen, die Struktur der if-else-Verzweigung als solche zu benennen und ihren Aufbau sowie ihre Funktionsweise in C# zu erläutern. (Verständnis)
- lernen, die Struktur der switch-case-Verzweigung als solche zu benennen und ihren Aufbau sowie ihre Funktionsweise in C# zu erläutern. (Verständnis)
- lernen, die Struktur der while-Schleife als solche zu benennen und ihren Aufbau sowie ihre Funktionsweise in C# zu erläutern. (Verständnis)
- lernen, die Struktur der for-Schleife als solche zu benennen und ihren Aufbau sowie ihre Funktionsweise in C# zu erläutern. (Verständnis)
- lernen, welche Funktionen die auf dem Arbeitsblatt „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“⁹⁴ angeführten Methoden der File-Klasse in C# besitzen, wobei sie selbständig in der Dokumentation des Microsoft Developer Network (MSDN) nach diesen suchen. (Anwendung)
- lernen, aus einer vorgegebenen Dokumentation auf die Funktionen und Anwendungskontexte der dort dargestellten Elemente zu schließen. (Verständnis)

⁹³ Bloom (zit. nach Sitte 2001, S. 474 f.).

⁹⁴ Vgl. Anhang 1.

- lernen, wie sie über das Visual Studio sowie über das Microsoft Developer Network (MSDN) auf die C#-Dokumentation zugreifen und diese nutzen können, um den Aufbau und die Funktionsweise von Sprachelementen zu untersuchen. (Anwendung)
- lernen, den in der Stunde eingeführten Dreischritt der Anforderungsanalyse, der Planung der Anforderungsumsetzung und der Implementierung als Methodik zur Problemlösung für solche Aufgaben zu erkennen und anzuwenden, bei denen die SuS aus einer gegebenen, textuell codierten Anforderungsbeschreibung ein Softwareprojekt realisieren sollen. Hierzu sollen die SuS lernen, die textuell codierten Anforderungen soweit zu elementarisieren, dass die einzelnen Elemente der Anforderung jeweils auf ein oder wenige Sprachelemente gleichen Typs in C# abgebildet werden können. Weiterhin sollen die SuS lernen, die aus den Anforderungen ermittelten, zu implementierenden Sprachelemente unter Beachtung von möglichen Abhängigkeiten in eine konsistente Reihenfolge zu bringen, deren Abarbeitung eine widerspruchsfreie Implementierung ermöglicht. (Synthese)
- lernen, dass die Manipulation des Dateisystems mithilfe des von ihnen realisierten Programms an denselben Orten der Verzeichnisstruktur dieselben Rechte benötigt, wie sie bei der manuellen Manipulation des Dateisystems durch den Nutzer vom Betriebssystem gefordert werden würden. (Begründung)
- lernen, ihr Wissen über die if-else-Verzweigung in C# so zu organisieren, dass sie in der Lage sind, deren Aufbau und Funktionsweise gegenüber ihren Mitschülerinnen und Mitschülern in einem fünfminütigen Kurzreferat zu vermitteln und dabei zu sichern, dass sie das Dargestellte auch verstanden haben. (Analyse)
- lernen, ihr Wissen über die switch-case-Verzweigung in C# so zu organisieren, dass sie in der Lage sind, deren Aufbau und Funktionsweise gegenüber ihren Mitschülerinnen und Mitschülern in einem fünfminütigen Kurzreferat zu vermitteln und dabei zu sichern, dass sie das Dargestellte auch verstanden haben. (Analyse)
- lernen, ihr Wissen über die for-Schleife in C# so zu organisieren, dass sie in der Lage sind, deren Aufbau und Funktionsweise gegenüber ihren Mitschülerinnen und Mitschülern in einem fünfminütigen Kurzreferat zu vermitteln und dabei zu sichern, dass sie das Dargestellte auch verstanden haben. (Analyse)
- lernen, ihr Wissen über die while-Schleife in C# so zu organisieren, dass sie in der Lage sind, deren Aufbau und Funktionsweise gegenüber ihren Mitschülerinnen und Mitschülern in einem fünfminütigen Kurzreferat zu vermitteln und dabei zu sichern, dass sie das Dargestellte auch verstanden haben. (Analyse)
- durch die zielbezogene Reflexion ihres eigenen Handelns bei der Realisierung des Dateikopierer-Projekts lernen, dass die Realisierung eines Softwareprojekts nicht mit seiner Implementierung gleichzusetzen ist, sondern dass die verschiedenen Aktivitäten, welche die Implementierung vor- bzw. nachbereiten, ebenso Gegenstand der Realisierung und somit des Softwareentwicklungsprozesses sind.

Affektive Lernziele:

Um die affektiven Lernziele bezüglich ihres Grades der Internalisierung zu hierarchisieren, beziehe ich mich auf Krathwohls Internalisierungsstufen der affektiven Taxonomie.⁹⁵ Der Name der Stufe steht dabei jeweils in Klammern am Ende des jeweiligen Lernziels, dem diese Stufe zugeordnet wurde.

Die SuS sollen...

- lernen, je zwei nur geringfügig voneinander abweichende Codeabschnitte, welche als Folge der Anwendung der Dreischrittmethodik auf zwei nur geringfügig voneinander abweichende Anforderungsbeschreibungen entstanden sind, vergleichend gegenüberzustellen und dabei die funktionalen Unterschiede auf die Unterschiede im Code sowie die Unterschiede im Code auf die Unterschiede in der Anforderungsbeschreibung zurückzuführen. (Aufmerksamwerden / Beachten)
- erkennen lernen, dass der Austausch über die in der Stunde gelernte Methodik mit dem Partner einen möglichen Mehrwert für sie darstellt, der zu einem besseren Verständnis der Methodik oder zu besseren bzw. vielfältigeren Arbeitsergebnissen führen kann. (Organisieren)
- durch den Austausch über die Kontrollstrukturen mit den anderen Gruppenmitgliedern erkennen lernen, dass der Austausch oder die Kooperation mit anderen Menschen einen Mehrwert für ihre eigene Wissensbasis generiert und Lernen oftmals bedeutet, von und mit anderen Menschen zu lernen. (Organisieren)
- während der Anwendung des Dreischritts in Partnerarbeit lernen, die von ihrem Partner eingebrachten Lösungskonzepte / -ideen / -vorschläge für die Analyse, Planung und Implementierung im Bezug auf lösungsrelevante Bewertungskriterien (wie dem angemessenen Detailgrad, der Korrektheit der Wahl der Lösungselemente oder der Berücksichtigung von Abhängigkeiten) kritisch zu reflektieren und ihren eigenen Lösungen als mögliche Alternative unvoreingenommen gegenüberzustellen. (Charakterisierung durch einen Wert oder eine Wertstruktur)
- während der Anwendung des Dreischritts in Partnerarbeit lernen, die von ihrem Partner eingebrachten Lösungskonzepte / -ideen / -vorschläge in ihrem Mehrwert als Innovationen, Lösungsfortschritte bzw. als wertvolle Beiträge für die Aufgabenbewältigung zu erkennen und anzuerkennen (Aufmerksamwerden / Beachten)
- lernen, aus der Prämisse, dass das Wissen und die Beiträge ihrer Mitschülerinnen und Mitschüler zukünftig (beim Ausführen des Dreischritts) einen Mehrwert für ihren eigenen Wissensstand generiert bzw. dieser bis zu einem gewissen Grad vom Wissen und den Beiträgen ihrer Mitschülerinnen und Mitschüler abhängig ist, heraus ein Interesse daran zu entwickeln, dass ihre Mitschülerinnen und Mitschüler die Mindestanforderungen (bei den Vorbereitungsaufgaben für die Projektrealisierung) erfüllen. (Werten)
- während ihrer Rolle als Experten für eine bestimmte Kontrollstruktur lernen, dass sie sich mit ihrer Wissensbasis in einer Monopolstellung befinden und sie dabei die ungeteilte Verantwortung tragen, jene Monopolstellung so gut wie möglich aufzulösen. (Werten)

⁹⁵ Vgl. Meyer (zit. nach http://www.zfsl-recklinghausen.nrw.de/Seminar_GyGe/Seminarprogramm/Lernzielkonzept/Anhang/Anhang_5/ [02.10.2014]).

2.5 Didaktische Strukturierung

Die Frage nach der „Sache“ dieser Unterrichtsstunde führt schnell zu einer These Meyers, nach der die Auswahl und die Strukturierung der geplanten Unterrichtsinhalte nicht aus fachwissenschaftlichen Vorgaben abgeleitet werden können, sondern von Beginn an aus einem didaktischen Begründungszusammenhang heraus erfolgen müssen: „Es gibt keine reine Sachanalyse! Was die ‚Sache‘ des Unterrichts ist, ergibt sich erst im Horizont didaktischer Anfragen an das Thema der Stunde.“ Hierbei wurde bereits gezeigt, dass die Unterrichtsreihe zu C# sowohl in der horizontalen als auch in der vertikalen Dimension einen ausgeprägten Transfercharakter besitzt,⁹⁶ welcher sich aufgrund des gewählten Stundenthemas der Realisierung eines Dateikopierers in C# und der damit verbundenen, bereits geschilderten Aufgabenstellung insbesondere in dieser Unterrichtsstunde offenbart:

So ist die Anforderungsbeschreibung für den Dateikopierer in ihrer Gesamtheit in der Art konzipiert, dass die SuS Vorwissen aus jedem Inhalt, welcher im Rahmen der Auseinandersetzung mit C# behandelt wurde,⁹⁷ aktivieren müssen, wobei im Rahmen der Aufgabenlösung bereits gezeigt wurde, dass die in den vorangegangenen Stunden der Unterrichtseinheit gesammelten Kenntnisse über die Sprachgrundlagen noch nicht einmal hinreichend sind, um das vorgegebene Problem in C# zu lösen, und daher in der Stunde der Projektrealisierung um weitere grundlegende (Kontroll-)Strukturen und spezifische, für die Realisierung der konkreten Anforderungen bedeutende Klassen, Methoden und Eigenschaften ergänzt werden muss. Das in der Aufgabenstellung skizzierte Projekt stellt die SuS somit vor die kognitive Herausforderung, ihre während der gesamten Unterrichtsreihe gewonnenen Erkenntnisse über C# in die Synthese einer Lösung für ein komplexes Softwareproblem einzubringen, wobei die „Fähigkeit der eigenständigen Zusammenführung unterschiedlicher Informationen zu einem neuen Ganzen“ Sittes Definition der Synthese als vorletzter Stufe in Blooms Taxonomie entspricht.⁹⁸ Ebenso entspricht diese Transferleistung der Zusammenführung von grundlegenden Elementen der Sprache C# zu einem komplexen WPF-Projekt Meyers Anforderung eines vertikalen Transfers, welcher Meyer zufolge dann vorliegt, „wenn von basalen Kenntnissen aus ‚hochgestiegen‘ wird zu anspruchsvolleren Fragen und hin und wieder zu Vorgriffen in der Ordnung des Lehrplans.“⁹⁹ Da die Reihe, wie bereits in 2.1.2 von mir erläutert, als Bindeglied zwischen dem Themenbereich „Grundlagen informatischen Arbeitens“, welcher durch die Auseinandersetzung mit dem Thema „Betriebssysteme“ vor meiner Reihe behandelt wurde, und dem Themenbereich „Algorithmen und Softwareentwicklung“, welcher durch die Auseinandersetzung mit dem Thema „PHP“ nach meiner Reihe behandelt wird, konzipiert wurde, stellt die Einführung und Anwendung einer Methodik zur Softwareentwicklung, wie sie in der Stunde mit der Einführung und Anwendung des Dreischritts der Anforderungsanalyse, Planung und Implementierung zur Realisierung von Softwareprojekten vollzogen wurde, einen Vorgriff auf den vom Rahmenlehrplan für die Qualifikationsphase formulierten, dritten Themenbereich der „Softwareentwicklung“ dar, in dem die SuS „Kenntnisse über das methodische Vorgehen zur modellhaften Entwicklung von Softwaresystemen“ erwerben sollen.¹⁰⁰ Durch die Einführung des Dreischritts werden dabei einzelne Phasen des Softwareentwicklungsprozesses angedeutet, sodass der Fachlehrer später bei der ausführlichen Behandlung der Softwareentwicklung als Themenfeld auf den

⁹⁶ 2.1 Erläuterungen zum Thema der Unterrichtsreihe.

⁹⁷ Vgl. Anhang 2 sowie 2.1.3 Struktur der Unterrichtsreihe.

⁹⁸ Sitte 2001, S. 474.

⁹⁹ Meyer 2012, S. 202.

¹⁰⁰ Vgl. https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1406800775&file=sek2_informatik.pdf [10.10.2014], S. 20 f.

Ausblick bzw. die Vorwegnahme, welche(r) im Rahmen der Realisierung des Dateikopierers als vertikaler Transfer geleistet wurde, rekurrieren kann. Aus den eben beschriebenen, vertikalen Transfers ergeben sich folgende Intentionen für die Unterrichtsstunde und die ihr zugrundeliegende Aufgabenstellung: Einerseits stellt die Realisierung des Dateikopierers als Projekt den Schlusspunkt der Reihe dar und dient dabei der Wiederholung und Anwendung des in der Reihe Gelernten in einem für die SuS sinnstiftenden Zusammenhang. Andererseits soll die Stunde jedoch nicht nur bereits vorhandenes Wissen wiederholen und einüben, sondern weiterhin einen Mehrwert für die Wissensbasis der SuS darstellen, welcher von diesen als solcher auch wahrgenommen und reflektiert wird. Hierin liegt auch der Grund, aus dem die SuS (unter Vorgriff auf die Softwareentwicklung) die ihnen vorgegebenen Anforderungen nicht einfach direkt implementieren, sondern durch die exemplarische Anwendung des Dreischritts auf den Gegenstand der Dateikopierer-Software an die Konzeption des Softwareentwicklungsprozesses als Modell phasenhaften Vorgehens herangeführt werden sollen.

Die Formulierung des sinnstiftenden Zusammenhangs, welche ich soeben für den Gegenstand des Dateikopierer-Projekts verwendet habe, besitzt in diesem Kontext zwei intendierte, miteinander verknüpfte Bedeutungen: Zunächst ist das Durchführen des Projekts für die SuS sinnstiftend, da am Ende dieser Durchführung eine in sich abgeschlossene, funktionsfähige Software steht. Dabei ist der Funktionsumfang der Software den SuS aus der Aufgabenstellung der Stunde sowie der im Arbeitsauftrag des Projekt-Arbeitsblatts formulierten Anforderungsbeschreibung bekannt, sodass die Software von den SuS genutzt werden kann und somit für diese einen sofort sichtbaren Mehrwert generiert. Weiterhin ergibt sich die Sinnstiftung im Kontext des durch die Stunde geleisteten, horizontalen Transfers, wobei aus der Aufgabenstellung ersichtlich wird, dass es sich bei dem „Wissen und Können [, welches] aus dem einen in den anderen Bereich ‚hinübergeschaufelt‘ und dort sinnvoll angewandt wird“,¹⁰¹ um das im Rahmen der Auseinandersetzung mit dem Thema Betriebssysteme Gelernte handelt, welches, wie im späteren Verlauf dieser didaktischen Strukturierung noch gezeigt wird, notwendig ist, um der Aufgabe der Vervollständigung der File-Klassendokumentation nicht nur auf der Kompetenzstufe der Anwendung, sondern ebenso auf der Stufe der Analyse begegnen und um die auf dem Arbeitsblatt „Projekttest“¹⁰² formulierte Beobachtungs- und Analyseaufgabe durchführen zu können. Die in der Unterrichtsstunde behandelte „Sache“ liegt also nicht nur in der wiederholten Thematisierung der Sprache C#, sondern ebenso in ihrer Einbettung in den methodischen Kontext der Softwareentwicklung sowie den inhaltlichen Kontext der Manipulation des Dateisystems durch die „Ansteuerung“ des Betriebssystems über C#.

Eine solche programmatische „Ansteuerung“ des Betriebssystems ist in C# über zwei Wege möglich, welche aus der Struktur der Programmiersprache und ihrer Genese als Teil des .Net-Framework hervorgehen. Beim .Net Framework handelt es sich Doberenz & Gewinnus zufolge „gleichermaßen um eine Entwurfs- wie um eine Laufzeitumgebung, in welcher Windows- und Web-Anwendungen erstellt und verteilt werden können.“¹⁰³ Als Hauptkomponenten des .Net Framework führt Microsoft selbst die Common Language Runtime (CLR) sowie die .NET Framework-Klassenbibliothek an,¹⁰⁴ wohingegen in der Literatur oftmals die Base Class Library (BCL) statt der gesamten Klassenbibliothek als Bestandteil angegeben wird.¹⁰⁵ Tatsächlich stellt die BCL nur eine Teilbibliothek der gesamten .NET Framework-Klassenbibliothek dar. Sie enthält solche Basisklassen, welche von jeder .Net-Anwendung, unabhängig von ihrem Anwendungskontext

¹⁰¹ Meyer 2012, S. 202.

¹⁰² Vgl. Anhang 3.

¹⁰³ Doberenz & Gewinnus 2013, S. 75.

¹⁰⁴ Vgl. <http://msdn.microsoft.com/de-de/library/zw4w595w%28v=vs.100%29.aspx> [12.10.2014].

¹⁰⁵ Vgl. Wenger 2010, S. 7.

(d. h. unabhängig davon, ob die Anwendung einen Webservice, eine Windows- (Phone-) App, eine ASP.NET- oder eine Desktopanwendung darstellt), benötigt werden. Die anderen Teilbibliotheken der .NET Framework-Klassenbibliothek stellen dem Entwickler solche Klassen zur Verfügung, welche ihm das Generieren einer .Net-Applikation für seinen konkreten Anwendungsfall erleichtern sollen: So gibt es eine zusätzliche Teilbibliothek für die Netzwerkkommunikation („Communication and Workflow“), für die Datenverwaltung („Data, XML, and LINQ“) sowie für verschiedene Nutzerschnittstellen (wie die „Windows Presentation Foundation“, „Windows Forms“ oder „ASP.NET“).¹⁰⁶ Zwar weisen Doberenz & Gewinnus richtigerweise darauf hin, dass ein großer Teil der .NET-Klassenbibliotheken selbst in C# programmiert wurde, es scheint jedoch fraglich, inwieweit eine Erhebung von C# zur „Systemsprache von .Net“, welche den Autoren zufolge „gewissermaßen die gleiche Rolle wie C++ für Windows“ spiele, zulässig ist.¹⁰⁷ Der Vorteil des .Net-Framework besteht ja gerade darin, dass über verschiedene .Net-Programmiersprachen auf die Klassen der Base Class Library zugegriffen werden kann, sodass der Zugriff nicht, wie es bei Java der Fall ist, auf die Sprache beschränkt ist, in der das Framework geschrieben wurde.¹⁰⁸ Der gemeinsame Zugriff von .Net-Sprachen wie Visual C#, Visual Basic oder Visual C++ auf das .Net-Framework wird durch die spezifischen Strukturen ermöglicht, welche bezüglich der Kompilierung und Ausführung eines .Net-Programm bestehen: Hierbei wird der in einer .Net-Sprache geschriebene Code zunächst in einen Zwischencode, den Microsoft-Intermediate-Language-Code (MSIL-Code), kompiliert, welcher noch von der konkreten Struktur des Prozessors unabhängig ist und somit keinen Maschinencode darstellt. Für die Genese des MSIL-Codes spielt es keine Rolle, in welcher .Net-Sprache der Quellcode ursprünglich geschrieben wurde, weshalb aus der Beschaffenheit des MSIL-Codes auch keine Rückschlüsse auf die verwendete .Net-Sprache möglich sind.¹⁰⁹ Nach dem Kompilieren des Quellcodes wird der MSIL-Code dem Nutzer als Assembly-Datei zur Verfügung gestellt, welche entweder in Form einer ausführbaren Anwendungsdatei, d.h. einer .exe-Datei, oder einer dynamischen Programmbibliotheksdatei, einer dynamic-link-library-Datei (.dll-Datei) vorliegt. Der in der Assembly-Datei enthaltene MSIL-Code jedoch nicht maschinenlesbar ist, kann dieser nicht unmittelbar vom Prozessor ausgeführt werden. Stattdessen wird der Code in der sogenannten Common Language Runtime (CLR) ausgeführt, welche eine virtuelle Maschine bzw. Laufzeitumgebung für die in MSIL-Code kompilierten Quellcodes der .Net-Sprachen darstellt.¹¹⁰ Die Runtime enthält einen Just-in-Time(JIT)-Compiler, welcher den MSIL-Code in einen für die Maschine lesbaren Code umwandelt.¹¹¹ Microsoft beschreibt seinen Entwicklern die Common Language Runtime und ihre Aufgaben wie folgt:

*The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that promote security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.*¹¹²

¹⁰⁶ Vgl. http://download.microsoft.com/download/4/a/3/4a3c7c55-84ab-4588-84a4-f96424a7d82d/NET_35_Namespaces_Poster_JAN08.pdf [12.10.2014].

¹⁰⁷ Doberenz & Gewinnus 2013, S. 75.

¹⁰⁸ Vgl. Wenger 2010, S. 6.

¹⁰⁹ Vgl. Doberenz & Gewinnus 2013, S. 74.

¹¹⁰ Die Bezeichnung „common“ ergibt sich also aus dem Umstand, dass alle .Net-Sprachen eine gemeinsame Laufzeitumgebung nutzen (vgl. Wenger 2010, S. 9).

¹¹¹ Vgl. Doberenz & Gewinnus 2013, S. 75.

¹¹² <http://msdn.microsoft.com/de-de/library/zw4w595w%28v=vs.100%29.aspx> [12.10.2014].

Wenger weist darauf hin, dass der in der CLR ausgeführte „managed code“ zwar nicht mit dem „unmanaged code“ nativer Sprachen wie C oder C++, deren Code ohne Zwischenschritte oder Laufzeitumgebungen unmittelbar in Maschinencode kompiliert wird, kompatibel ist, jedoch ermöglicht die Interoperabilität des .Net-Frameworks die Integration von „unmanaged code“ in .Net-Projekte:

NET stellt nämlich eine weite Palette von Möglichkeiten zur Verfügung, die Interoperabilität zwischen der nativen und der verwalteten Welt zu gewährleisten. Unter dem Namen PInvoke (platform invoke) definiert die Interoperabilität zunächst Mechanismen zur reinen Nutzung aller APIs mit C-Bindung. Da sind natürlich die WIN32-Funktionen zu nennen, aber auch die vielen Bibliotheken von Firmen, die im Laufe der Jahre entwickelt wurden. Als weiteren Mechanismus finden wir die Managed Extensions for C++, die es C++-Entwicklern ermöglichen, ihre objektorientierten Bibliotheken für die Nutzung unter .NET fit zu machen. Letztendlich ist dann noch die Interoperabilität mit COM zu erwähnen, die Microsoft so elegant in .NET integriert hat, dass die Nutzung, ja sogar die Erstellung neuer COM-Komponenten, noch nie so einfach war wie unter .NET.¹¹³

Durch die Interoperabilität bezüglich der WIN32-API und .Net ergibt sich somit die Möglichkeit, das Betriebssystem nicht nur über den „managed code“ einer .Net-Sprache in der CLR anzusprechen, sondern auch über den „unmanaged“ C-Code der WIN32-API. Für das in der Unterrichtsstunde geplante Projekt des Dateikopierers hätte also statt der „Copy“-Methode der File-Klasse, welche als Klasse im Namensraum *System.IO* (und damit als Klasse der Base Class Library)¹¹⁴ „managed code“ darstellt, *PInvoke* zum Import der „kernel.dll“, welche „unmanaged code“ enthält, und dem anschließenden Aufruf der „CopyFile“-Methode verwendet werden können.¹¹⁵ Dies hätte jedoch vorausgesetzt, dass den SuS das Konzept der Interoperabilität über *PInvoke* bereits bekannt ist und dass sie in der Lage sind, dieses Konzept anzuwenden und sich selbständig über geeignete Quellen (wie beispielsweise *pinvoke.net*) über die Vorgehensweise beim Dll-Import sowie der Integration des „unmanaged code“ in den „managed code“ zu informieren, jedoch waren weder *PInvoke* noch die Integration des „unmanaged code“ Gegenstand der Unterrichtseinheit.¹¹⁶ Das hier ausgeführte Konzept der zweistufigen Kompilierung des .Net-Codes dagegen wurde gegenüber den SuS bereits in der ersten Stunde der Einheit in einer vereinfachten Form erläutert.

Für den Sachgegenstand der Softwareentwicklung bzw. des Softwarelebenszyklus ist eine solche didaktische Reduktion durch die hier vorgestellte Unterrichtsstunde und die ihr zugrundeliegende Aufgabenstellung intendiert. Die mit der Softwareentwicklung verbundene Problematik ergibt sich aus dem von Booch formulierten Charakteristikum, nach dem „es für den einzelnen Entwickler sehr schwierig, wenn nicht gar unmöglich ist, alle Feinheiten des Designs [industriell einsetzbarer Software] zu verstehen. Einfach ausgedrückt, überschreitet die Komplexität solcher Systeme die Kapazität der menschlichen Intelligenz.“¹¹⁷ Diese Komplexität nahm in den 1960er Jahren so stark zu, dass am Ende dieses Jahrzehnts die Softwareentwicklungskosten die Hardwarekosten überstiegen.¹¹⁸ Um der sich hieraus entwickelnden Softwarekrise zu begegnen, fand zwischen dem 7. und dem 11. Oktober 1968 im bayerischen Garmisch eine vom NATO Science Committee finanzierte Konferenz zum Thema „Software Engineering“ statt,¹¹⁹ welche Aspray & Ensmenger als „perhaps the earliest and best-known attempt to rationalize the production of

¹¹³ Wenger 2010, S. 16.

¹¹⁴ Vgl. http://download.microsoft.com/download/4/a/3/4a3c7c55-84ab-4588-84a4-f96424a7d82d/NET_35_Namespaces_Poster_JAN08.pdf [12.10.2014].

¹¹⁵ Vgl. <http://www.pinvoke.net/default.aspx/kernel32/CopyFile.html> [12.10.2014].

¹¹⁶ Vgl. 2.1.3 Struktur der Unterrichtsreihe.

¹¹⁷ Booch 1995, S. 18.

¹¹⁸ Vgl. Aspray & Ensmenger 2002, S. 150-153.

¹¹⁹ Vgl. <http://www.scrummanager.net/files/nato1968e.pdf> [13.10.2014].

software development along the lines of traditional industrial manufacturing“ beschreiben.¹²⁰ Die Konnotation des „Software Engineering“-Begriffs mit der Intention einer rationalisierten Softwareproduktion findet sich ebenso in der IEEE-Definition: „[Software Engineering is the] application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.“¹²¹ Die Systematik des Software Engineering ergibt sich aus den verschiedenen Phasen bzw. Aktivitäten, in die der Softwareentwicklungsprozess eingeteilt wird. Hierbei versucht Michael J. Pont den Fokus von der eigentlichen Programmierungs- bzw. Implementierungsarbeit abzulenken, indem er betont, dass andere Phasen der Softwareentwicklung, wie beispielsweise die Phasen bzw. Aktivitäten der Spezifikation oder des Entwurfs der Software, genauso wichtig seien.¹²² Die einzelnen Aktivitäten werden in sogenannten Vorgehensmodellen systematisiert, worunter ein Entwicklungsplan zu verstehen ist, in dessen Folge festgelegt wird, „welche Aktivitäten in welcher Reihenfolge von welchen Personen in welchen Rollen erledigt und welche Ergebnisse dabei entstehen und wie diese überprüft werden.“¹²³ Hierbei existiert eine Vielzahl von Vorgehensmodellen, welche sich in der Häufigkeit und in der Reihenfolge der Anwendung der beim Softwareentwicklungsprozess durchgeführten Aktivitäten unterscheiden, wobei auch das Tätigkeitsfeld und der zu leistende Arbeitsaufwand, welcher jeweils mit einer bestimmten Aktivität konnotiert wird, zwischen den Modellen variiert. Floyd & Züllighoven identifizieren jedoch sieben Aktivitäten, welche in der von ihnen beschriebenen bzw. in einer nur geringfügig abgewandelten Form den meisten Vorgehensmodellen zugrundeliegen: (1) Anforderungsermittlung (Analyse), (2) Systemdefinition, (3) Entwurf, (4) Implementierung, (5) Validation, (6) Systemeinführung und (7) Wartung.¹²⁴ Diese finden sich ebenso im klassischen Wasserfallmodell, wobei die Aktivitäten der „Anforderungsermittlung“ sowie der „Systemdefinition“ zur Aktivität der „Analyse und Definition“ und die Aktivitäten der „Systemeinführung“ und „Wartung“ zur Aktivität „Einsatz und Wartung“ zusammengefasst werden.¹²⁵ Neben dem klassischen Phasenmodell / dem klassischen Wasserfallmodell enumeriert Bothe weiterhin das iterative Phasenmodell (Lebenszyklus), das Spiralmodell, das Prototyping (evolutionäre Softwareentwicklung), die inkrementelle Softwareentwicklung (z.B. als Rational Unified Process (RUP)) sowie das V-Modell als weitere, oft verwendete Vorgehensmodelle.¹²⁶ Hierbei fällt auf, dass er die Begriffe „Vorgehensmodelle“ und „Phasenmodelle“ als Synonyme verwendet,¹²⁷ während Floyd & Züllighoven die Phasenmodelle als Teilmenge der Vorgehensmodelle betrachten, da sie weder das Spiralmodell noch das Prototyping als Derivat des klassischen Wasserfallmodells als grundlegendem Phasenmodell anerkennen, weshalb diese beiden Modelle auch nicht den Phasenmodellen zugeordnet werden.¹²⁸ Der im Unterricht behandelte Dreischritt der Anforderungsanalyse, der Planung der algorithmischen Umsetzung dieser Anforderungen und der Implementierung des Algorithmus orientiert sich am Wasserfallmodell, wobei ein Vergleich der im Arbeitsauftrag beschriebenen Vorgehensweisen¹²⁹ bzw. des Erwartungsbilds¹³⁰ mit den

¹²⁰ Aspray & Ensmenger 2002, S. 150.

¹²¹ <http://www.idi.ntnu.no/grupper/su/publ/ese/ieee-se-glossary-610.12-1990.pdf> [13.10.2014], S. 67.

¹²² Vgl. Pont 1996, S. 3 f.

¹²³ <http://waste.informatik.hu-berlin.de/Lehre/ws0910/pet/entwicklung-20091111.pdf> [13.10.2014], S. 4.

¹²⁴ Aufgrund des Umfangs der Arbeit wird auf eine Wiedergabe der von Floyd & Züllighoven geleisteten Definitionen der einzelnen Aktivitäten verzichtet werden. Stattdessen soll auf ihren Beitrag im Sammelband „Handbuch Informatik“ verwiesen (vgl. Floyd & Züllighoven 1997, S. 656 f.).

¹²⁵ Vgl. <https://www2.informatik.hu-berlin.de/swt/lehre/GdP/fohlen/I.8-1s.pdf> [13.10.2014], S. 20.

¹²⁶ Vgl. ebd., S. 29.

¹²⁷ Vgl. ebd., S. 19.

¹²⁸ Vgl. Floyd & Züllighoven 1997, S. 657-660.

¹²⁹ Vgl. Anhang 2.

Definitionen der Aktivitäten bei Floyd & Züllighoven zeigt, dass die von den SuS in der Anforderungsanalyse erwarteten Handlungen nicht mit den Handlungen in einer realen Anforderungsanalyse und die von den SuS in der Planung erwarteten Handlungen nicht mit den Handlungen bei einem realen Entwurf kongruent sind: So verstehen Floyd & Züllighoven unter der Anforderungsanalyse die Ableitung von Anforderungen aus den Erfordernissen der in dem Anwendungsbereich, in dem die Software eingesetzt werden soll, relevanten technischen Prozesse bzw. Arbeitsprozesse über formale Informationsanalyse oder informelle Kundengespräche. In der Aktivität der Systemdefinition werden mithilfe der abgeleiteten Anforderungen anschließend „die grundlegenden technischen, planerischen und organisatorischen Entscheidungen über die Software und ihre Einbettung in den Einsatzkontext getroffen. Dies umfasst den fachlichen Funktionsumfang, die zu modellierenden Aspekte des Gegenstandsbereichs, die Grundzüge der Mensch-Rechner-Interaktion und das Basissystem.“¹³¹

Die Komplexität der im Wasserfallmodell benannten Aktivität der „Analyse und Definition“ wird für die SuS deutlich reduziert. Hierbei werden ihnen die konkreten Anforderungen an das zu realisierende Softwareprojekt bereits vorgegeben. Wenn in der Aufgabenstellung sowie im Arbeitsauftrag von der Anforderungsanalyse gesprochen wird, so wird dabei der Schwerpunkt auf den zweiten Teil der Aktivität gelegt, nämlich der Definition. Dieser Gegenstand wird für die SuS ebenfalls reduziert: Durch C# und WPF werden ihnen die Programmiersprache, die Technologien und die damit verbundenen Entwurfskonzepte, mit denen sie das Projekt realisieren sollen, bereits vorgegeben. Die SuS müssen daher lediglich beachten, ob es Interdependenzen zwischen den Anforderungen untereinander bzw. den Anforderungen und der vorgegebenen Technologie gibt, welche es beim Entwurf bzw. bei der Planung der Software zu beachten gibt. Dabei trennt der Arbeitsauftrag, wieder zur Vereinfachung der Komplexität des Problems, unscharf zwischen den Aktivitäten der „Analyse und Definition“ und des „Entwurfs“. Unter letzterem verstehen Floyd & Züllighoven die Genese der „Architektur des Softwaresystems. Das bedeutet die Zerlegung in Komponenten und die Festlegung ihrer Verknüpfung [...]. Anliegen sind die Bewältigung der Komplexität, die Gewährleistung von Verständlichkeit, Änderbarkeit und Wiederverwendbarkeit der Komponenten sowie die Vorbereitung einer arbeitsteiligen Programmierung.“¹³² Die Abbildung der im Arbeitsauftrag formulierten Anforderungen auf Sprachelemente in C# sowie deren Organisation in Strukturen des logischen Codes und des Oberflächencodes, welche von den SuS im ersten Schritt der Analyse geleistet werden sollen, stellen also solche Tätigkeiten dar, welche im Wasserfallmodell eher im Entwurf verortet werden müssten. Da die Realisierung des Projekts jedoch eine erste Annäherung an die Softwareentwicklung darstellen sollte, durch die lediglich das Schaffen eines Bewusstseins für die grundlegende Vorgehensweise der Unterteilung des Softwareentwicklungsprozesses in Aktivitäten intendiert war, sah ich mich nicht gezwungen, mich bei der Aufgabenformulierung strikt an den Aktivitäten des Wasserfallmodells zu orientieren, einerseits, da dieses zuvor nicht als Referenz eingeführt wurde, andererseits ist es fraglich, inwieweit die geringe Komplexität eine solch strikte Abgrenzung der einzelnen Aktivitäten überhaupt rechtfertigt, zumal die Anforderungen selbst, wie bereits erwähnt, in einer natürlichen Sprache vorgegeben waren. Aus diesem Grund wurden die Phasen der Analyse und Definition sowie des Entwurfs in Teilen integrativ behandelt, wodurch den SuS die Möglichkeit gegeben werden sollte, sinnzusammenhängend mit den Anforderungen zu arbeiten, ohne dass die (aufgrund der geringen Komplexität des Problems) einfachen Gedankengänge zu den einzelnen

¹³⁰ Da das Erwartungsbild eine Projektmappe enthält, wurde es nicht diesem Praktikumsbericht beigelegt sondern ist unter <http://randco.square7.ch/Dateikopierer.zip> einsehbar (vgl. <http://randco.square7.ch/Dateikopierer.zip> [13.10.2014]).

¹³¹ Vgl. Floyd & Züllighoven 1997, S. 657-660.

¹³² Ebd., S. 657.

Anforderungen durch die Trennung der Aktivitäten künstlich unterbrochen worden wären. Der Schritt der Planung der Arbeitsschritte zur Implementierung wiederum stellt eine integrative Aktivität zwischen dem Entwurf und der Implementierung im Wasserfallmodell dar.

Zusammengefasst stellt sich der bereits mehrmals erläuterte Dreischritt also als Maßnahme der Vereinfachung bzw. der didaktischen Reduktion der ersten vier allgemeinen, von Floyd & Züllighoven identifizierten Aktivitäten der Vorgehensmodelle (Anforderungsermittlung, Systemdefinition, Entwurf und Implementierung) dar. Diese Reduktion geht aus dem didaktischen Gesamtzusammenhang hervor, welcher mit der Realisierung des Projekts intendiert und in den der Dreischritt eingebettet war. Hierbei bildete das didaktische Prinzip des exemplarischen Lehrens und Lernens den zentralen normativen Grundsatz, welcher bei der Ziel- und Themenauswahl sowie der Gestaltung des Unterrichts berücksichtigt wurde.¹³³ Klafki beschreibt die Idee, welche sich hinter diesem Prinzip verbirgt, wie folgt:

Bildendes Lernen, das die Selbständigkeit des Lernenden fördert, als zu weiterwirkenden Erkenntnissen, Fähigkeiten, Einstellung führt [...], wird nicht durch reproduktive Übernahme möglichst vieler Einzelkenntnisse, -fähigkeiten und -fertigkeiten gewonnen, sondern dadurch, dass sich der Lernende an einer begrenzten Zahl von ausgewählten Beispielen [...] aktiv allgemeine, genauer: mehr oder minder weitreichend verallgemeinerbare Kenntnisse, Fähigkeiten, Einstellung erarbeitet, m. a. W.: Wesentliches, Strukturelles, Prinzipielles, Typisches, Gesetzmäßigkeiten, übergreifende Zusammenhänge. [...] Der Lernende gewinnt über das am Besonderen erarbeitete Allgemeine Einsicht in einen Zusammenhang, einen Aspekt, eine Dimension seiner naturhaften oder kulturell-gesellschaftlich-politischen Wirklichkeit, und zugleich damit gewinnt er eine ihm bisher nicht verfügbare neue Strukturierungsmöglichkeit, eine Zugangsweise, eine Lösungsstrategie, eine Handlungsperspektive.¹³⁴

Unter Klafkis Verständnis des Prinzips des „exemplarischen Lehrens und Lernens“ wurde diese Stunde konzipiert. Der Zusammenhang ihrer kulturell-gesellschaftlich-politischen Wirklichkeit, in die die SuS Einsicht gewinnen sollen, wurde mit dem Gegenstand des Softwareentwicklungsprozesses bereits benannt. Weiterhin wurde gerade erläutert, dass es hierbei nicht um die Erarbeitung und Abgrenzung der grundlegenden Aktivitäten oder die Erarbeitung eines bestimmten Vorgehensmodells, in der diese ausgeführt werden, ging. Vielmehr war intendiert, dass die SuS ihr Vorgehen bei der Bearbeitung des exemplarischen Gegenstands des Dateikopierer-Projekts reflektieren und hieraus ein allgemeines Verständnis für die Strukturierungsmöglichkeit des Softwareentwicklungsprozesses gewinnen, m. a. W.: Die SuS sollen durch die zielbezogene Reflexion ihres eigenen Handelns¹³⁵ bei der Realisierung des Dateikopierer-Projekts lernen, dass die Realisierung eines Softwareprojekts nicht mit seiner Implementierung gleichzusetzen ist, sondern dass die verschiedenen Aktivitäten, welche die Implementierung vor- bzw. nachbereiten, ebenso Gegenstand der Realisierung und somit des Softwareentwicklungsprozesses sind. Das hier formulierte, mit dem exemplarischen Lehren und Lernen verbundene Lernziel bewegt sich also auf einer höheren Abstraktionsebene als das Wissen und die Kenntnisse über die einzelnen Aktivitäten, weshalb es auch zulässig ist, dass die im Dreischritt definierten Aktivitäten inhaltlich nicht mit den von Floyd & Züllighoven definierten Aktivitäten übereinstimmen. Vielmehr ergibt sich die Genese der im Dreischritt definierten Aktivitäten aus der Inszenierung des exemplarischen Lehrens und Lernens als handlungsorientierte Projektarbeit. In diesem Zusammenhang wurde bei der Erläuterung der Aufgabenlösung bereits auf Deweys methodischen Dreischritt zur

¹³³ Vgl. Meyer 2012, S. 181.

¹³⁴ Klafki 1996, S. 143 f.

¹³⁵ Dies entspricht der 3. Stufe der Reflexionskompetenz in Meyers pragmatischem Kompetenzstufenmodell (vgl. Meyer 2012, S. 157).

Problemlösung, bestehend aus der Hypothesenbildung, dem Entwurf einer Lösung und dem Ausprobieren dieser, verwiesen.¹³⁶ Ebenso wurde hierbei ausgeführt, dass die im Arbeitsauftrag des Arbeitsblatts „Projekt-Dateikopierer“¹³⁷ definierten Aktivitäten auf Deweys Dreischritt abbilden. Hieraus folgt also, dass die in der Unterrichtsstunde genutzten Aktivitäten primär an einem Handlungsmuster zur Problemlösung (nach Dewey) orientiert sind, wohingegen der Bezug auf die von Floyd & Züllighoven definierten Aktivitäten eher eine nachgeordnete Position einnahm. Die didaktische Begründung hierfür ergibt sich aus der in der Bedingungsanalyse beschriebenen Schwierigkeit, das erreichte Niveau der Problemlösekompetenz der SuS verlässlich einzuschätzen. Da diese Kompetenz jedoch im Zentrum der Stunde steht, muss darauf geachtet werden, die SuS nicht zu überfordern.¹³⁸ Jene Gefahr der Überforderung würde jedoch zunehmen, wenn die SuS gezwungen wären, einem vorgeschriebenen zweidimensionalen Raster zu folgen, in dem sie auf jede von Floyd & Züllighoven definierte Aktivität Deweys methodischen Dreischritt zur Problemlösung anwenden müssten. Stattdessen wurde versucht, die Dimension des methodischen Dreischritts nach Dewey auf die Dimension der Aktivitäten nach Floyd & Züllighoven durch die Definition eigener Aktivitäten abzubilden, mit der bereits beschriebenen Folge, dass die neuen Aktivitäten inhaltlich nicht mit denen von Floyd & Züllighoven übereinstimmten. Als weitere Präventivmaßnahme gegen die Gefahr der Überforderung wurde bei der Stundenkonzeption darauf geachtet, dass möglichst nur solche Vorkenntnisse und Vorerfahrungen vorausgesetzt werden, welche ich mit den SuS selbst erarbeitet habe oder für die ich zumindest zuverlässig aussagen kann, inwieweit sie von den SuS inkorporiert wurden bzw. welche Kompetenzstufe die SuS erreicht haben. Da über die erreichte Stufe der Problemlösekompetenz jedoch keine verlässliche Aussage getroffen werden kann, eignet sich der in der Stunde definierte Dreischritt als „Gleichmacher“, d. h. als neu einzuführendes Handlungsmuster des Problemlösens, über die, da sie in der Stunde eingeführt wird, keine Kenntnisse bei den SuS vorausgesetzt werden. Es werden bei den SuS lediglich die über C# und damit in Verbindung stehenden Technologien und Entwurfskonzepte erarbeiteten Kenntnisse sowie die Fähigkeiten und Fertigkeiten erwartet, welche mit dem Implementierungsanteil der Problemlösekompetenz konnotiert sind (das einfache Erweitern einer vorgegebenen Problemlösung in geübtem Zusammenhang, das planvolle Einsetzen bekannter Informatiksysteme zur Lösung einer neuen Problemstellung aus einem bekannten Bereich sowie das Nutzen vorhandener Programmbibliotheken für die eigene Problemlösung). Letztere wurden im Rahmen der Bedingungsanalyse als bei den SuS beobachtbare bzw. gesicherte Fähigkeiten und Fertigkeiten ausgewiesen und können daher für die Durchführung des Projekts vorausgesetzt werden. Gleiches gilt für die beobachteten Fähigkeiten und Fertigkeiten bezüglich der Kompetenz der „Algorithmen“.

139

Der Dreischritt als „Gleichmacher“ ist dabei zugleich der größte Vor-, aber auch der größte Nachteil der Unterrichtsstunde. Die Vorteile ergeben sich in erster Linie für die leistungsschwächeren SuS. Mit der Einführung und Anwendung des Dreischritts innerhalb der Stunde verbinde ich die Hoffnung, dass die leistungsschwächeren SuS und dabei insbesondere die beiden Mädchen, auf deren schwach ausgeprägte Selbstkompetenz bereits in der Bedingungsanalyse hingewiesen wurde, reflektieren, dass sie bezüglich der Dreischritt-Methodik über denselben Kenntnisstand verfügen wie die leistungsstarken SuS, somit in ihrem Selbstkonzept bzw. ihrer schulbezogenen Selbstwirksamkeitserwartung bezüglich dieses konkreten Gegenstands bestärkt werden und sich infolgedessen trauen, den Prozess der Anwendung des Dreischritts in den verschiedenen Durchläufen aktiv mitzugestalten, d. h. selbst zu handeln, anstatt nur die Handlungen

¹³⁶ Vgl. 2.2.2 Mit der Aufgabenstellung verbundene Lernziele und Aufgabenlösung.

¹³⁷ Vgl. Anhang 2.

¹³⁸ Vgl. 2.2.2 Mit der Aufgabenstellung verbundene Lernziele und Aufgabenlösung.

¹³⁹ Vgl. 2.3.2 Lernvoraussetzungen der Schülerinnen und Schüler.

ihrer Mitschülerinnen und Mitschüler zu rezipieren bzw. bestenfalls zu reflektieren. Dabei werden die leistungsschwächeren SuS, deren Tendenz, den Wissenserwerb stark an den Vorgaben des Lehrers auszurichten, bereits in der Bedingungsanalyse ausgeführt wurde, nicht ins kalte Wasser geworfen, sondern über die verschiedenen Durchläufe sukzessiv zum selbständigen Arbeiten hingeführt.

Da die erste Anwendung des Dreischritts im Plenum stattfindet und die Produkte bzw. Arbeitserzeugnisse der drei Schritte als Ergebnisse eines Aushandlungsprozesses zwischen den SuS und dem Lehrer in Form eines fragend-entwickelnden Unterrichtsgesprächs entstehen, sind der Ablauf ihrer Genese sowie die hierfür notwendigen Arbeitsschritte für die SuS transparent. Gleichzeitig stellt ihnen die im Plenum erarbeitete Musterlösung ein Erwartungsbild für die eigenständige Anwendung des Dreischritts zur Verfügung. Mit der Bereitstellung eines Erwartungsbilds wird für die leistungsschwächeren SuS eine Orientierungshilfe intendiert, welche sie bei der Genese ihrer eigenen Arbeitserzeugnisse dabei unterstützen soll, einen adäquaten Detailgrad für ihre Ausführungen zu finden und dabei zwischen Wichtigem und Unwichtigem zu unterscheiden. Für die leistungsstärkeren Schüler besteht diese normierende Funktion in die andere Richtung. Sie sollen durch die Musterlösung veranlasst werden, ihre Gedanken zur Anforderungsanalyse und zur Planung explizit sowie in einem zur Musterlösung adäquaten Detailgrad zu formulieren und diese nicht nur als Zwischenschritte implizit mitzudenken.

Zusammengefasst sollen durch die Projektarbeit also die leistungsschwächeren SuS dazu hingeführt werden, eigenaktiver und autarker zu arbeiten, während die leistungsstärkeren SuS lernen sollen, vom Lehrer definierte Regeln und Vorgehensweisen zu befolgen. Vergleicht man diese Ziele mit dem in der Bedingungsanalyse beschriebenen Arbeitsverhalten, so zeigt sich, dass die leistungsschwächeren SuS in einem zu hohen Maß der bei den leistungsstärkeren vermissten Lehrerorientierung nachgehen bzw. dass die leistungsstärkeren SuS die bei den leistungsschwächeren SuS vermisste Fähigkeit, selbstständig bzw. selbstreguliert zu arbeiten, so weit treiben, dass sie durch das Einfügen bzw. Weglassen von Arbeitsschritten die Intention, die Zielstellung bzw. den Sinn der ihnen gestellten Aufgaben verfälschen, m. a. W.: Was der eine zu viel macht, macht der andere zu wenig. Dies kann, in der didaktischen Konzeption richtig integriert, einen Vorteil für den Unterricht bedeuten. Aus diesem Grund muss darauf geachtet werden, dass bei der Realisierung des Projekts immer ein(e) leistungsschwache(r) und ein(e) leistungsstarke(r) Schüler(in) paarweise zusammenarbeiten, um dabei ihre Schwächen, welche die Stärken des jeweils anderen darstellen, gegenseitig auszugleichen. Die theoretische Grundlage für die Überlegung findet sich in einem Beitrag Adamskis zur inneren Differenzierung im Geschichtsunterricht. Hierin weist der Autor auf verschiedene Differenzierungsfallen hin, welche auch bei der Konzeption dieses Entwurfs mit bedacht wurden:

Innere Differenzierung und Individualisierung von Lernen sind unmittelbar aufeinander bezogen und machen nur in ihrer Verknüpfung Sinn. Begreif man Binnendifferenzierung verkürzt, führt das tendenziell dazu, dass leistungshomogene Gruppen – hier die Stärkeren, dort die Schwächeren – sich im Leistungsspektrum immer weiter voneinander entfernen, nicht mehr voneinander und miteinander lernen können und sich nicht mehr den Anregungen von Anderen stellen müssen.¹⁴⁰

Die Arbeit in leistungsheterogenen Paaren zielt dagegen gerade auf die Genese von Synergien zwischen den Partnern, bei denen sie mithilfe der Anregungen des jeweils Anderen lernen, ihre eigenen Schwächen auszugleichen, indem sie die Handlungsweisen ihres Partners reflektieren und dabei bewerten, inwiefern

¹⁴⁰ Adamski 2009, S. 5.

eine Übernahme dieser einen Vorteil für das eigene Handeln bedeuten würde.¹⁴¹ Die Partnerarbeit in leistungsheterogenen Paaren führt dabei zu der Problematik, dass die Partner mit unterschiedlichen Arbeitstempi in die Partnerarbeit hineingehen, wobei voraussichtlich der leistungsstärkere Partner sein Tempo zugunsten des leistungsschwächeren Partners drosseln wird, wobei dieser Umstand zwei Gefahren impliziert: Für den leistungsstärkeren Partner besteht die Gefahr, dass er aufgrund der Entschleunigung seiner Arbeitsweise nach kurzer Zeit beginnt, sich zu langweilen. Dieser Gefahr wird zum Teil dadurch entgegengesteuert, dass er gezwungen wird, seine Gedanken zu formalisieren, wobei diese zusätzliche Aufgabe den leistungsstarken SuS nicht grundlos als „Beschäftigungstherapie“ vorgegeben werden darf, sondern über die Hervorhebung des Mehrwerts der Formalisierung dieser Gedanken für den Softwareentwicklungsprozess sinnstiftend dargelegt werden muss. Als weitere Maßnahme gegen diese Gefahr muss die geplante Interdependenz zwischen den Partnern, welche dadurch entsteht, dass jeder Partner gleichzeitig Schüler (bezogen auf seine Schwächen) und Lehrer (bezogen auf seine Stärken) des jeweils anderen Partners ist, gegenüber den SuS als solche explizit formalisiert werden. Während der leistungsstärkere Partner also als Tutor des leistungsschwächeren Schülers im Bezug auf inhaltliche Probleme fungiert, wird die tutorielle Funktion des leistungsschwächeren Schülers im Bezug auf die Formalisierungsaspekte intendiert. Insbesondere dem leistungsstärkeren Schüler soll also explizit bewusst gemacht werden, dass er sich nach der Bearbeitung der jeweiligen Aktivität nicht zurücklehnen, sondern als Tutor des leistungsschwächeren Schülers fungieren soll. Die von den SuS erwarteten Formalisierungen sollen, wie bereits formuliert, nicht jedes Detail erfassen, sondern so beschaffen sein, dass ihr Detailgrad zu den Lösungsanforderungen der Aufgabenstellung adäquat sind. Die SuS sollen durch die Reflektion ihres eigenen Handelns und dem ihrer Mitschüler(innen), insbesondere durch die Reflektion der Musterlösung des ersten Durchlaufs, selbst ein geeignetes Maß finden. Um zu verhindern, dass sich die leistungsschwächeren SuS in die (wie in der Bedingungsanalyse festgestellt) oft von ihnen praktizierte Detailarbeit stürzen, werden gegenüber den SuS für den zweiten und dritten Durchlauf anspruchsvolle Zeitvorgaben formuliert,¹⁴² welche sie dazu zwingen, ihr methodisches Vorgehen mit ihrem Partner auszuhandeln und ökonomisch zu strukturieren: Eine exemplarische, methodische Hilfestellung, die der Lehrer gegenüber einem Schülerpaar, welches Probleme hat, zeitadäquat zu arbeiten, formulieren könnte, wäre die Aufforderung, dass der leistungsstärkere Schüler einen Arbeitsschritt der zunächst gemeinsam ausgehandelten Planung formuliert, während der leistungsschwächere Schüler diesen zeitgleich in der Implementierung realisiert. Somit wäre gewährleistet, dass die SuS zeitadäquat arbeiten und dabei trotzdem an ihren leistungsspezifischen Schwächen arbeiten.

Aus dem eben geleisteten Überblick über das Projekt sowie die dahinter stehenden, didaktischen Überlegungen und Intentionen ergeben sich gewisse Charakteristika, welche Meyer zur Beschreibung der Grundform des kooperativen Unterrichts verwendet: Als dessen wichtigste Variante betrachtet Meyer das Projekt, in dessen Mittelpunkt er die Bearbeitung einer vom Lehrer und den SuS gemeinsam vereinbarten Aufgabe setzt, wobei sich die handelnden Akteure bei der Bearbeitung dieser Aufgabe gegenseitig unterstützen.¹⁴³ Der für die Stunde formulierte Projektgegenstand des Dateikopierers entspricht zwar Meyers Forderung nach einem persönlichen Nutzen der Aufgabe für die SuS,¹⁴⁴ er wurde jedoch seitens der Lehrperson vorgegeben und stellt daher nicht das Ergebnis einer gemeinsamen Zielvereinbarung dar. Die

¹⁴¹ Dies entspricht der 3. Stufe der Reflexionskompetenz in Meyers pragmatischem Kompetenzstufenmodell (vgl. Meyer, S. 157).

¹⁴² Vgl. 2.6 Tabellarischer Stundenverlaufplan

¹⁴³ Vgl. Meyer 2012, S. 62.

¹⁴⁴ Vgl. ebd., S. 62.

einseitige Zielvorgabe ergibt sich aus der Intention des Lehrers, die im Eingang der didaktischen Strukturierung beschriebenen Transferleistungen durch die Wahl eines adäquaten Projektgegenstands zu gewährleisten. Aus den horizontalen und vertikalen Transfers ergibt sich dabei die Zukunfts- und Gegenwartsbedeutung des in der Stunde behandelten Gegenstands für die SuS. Da sich diese zum größten Teil aus der bereits identifizierten Zukunfts- und Gegenwartsbedeutung der Unterrichtsreihe ergeben, werden jene Aspekte, welche sowohl für die einzelne Stunde als auch für die Reihe zutreffend sind, hier nur noch einmal stichpunktartig wiederholt. Der zentrale Gegenwartsbezug der Unterrichtsstunde ergibt sich für die SuS aus der Gelegenheit bzw. der Notwendigkeit, das bisher angeeignete Wissen über die Strukturen und Elemente der Sprache C# sowie die mit dieser konnotierten Technologien und Entwurfskonzepte im sinnstiftenden Zusammenhang eines Softwareprojekts anzuwenden.¹⁴⁵ Dabei hat das Projekt mit dem Dateikopierer eine Software zum Gegenstand, welche durch die Interaktion mit dem Betriebssystem dessen Dateisystem manipuliert. Da den SuS, wie oben in der Sachanalyse zu C# erläutert, in den vorhergehenden Stunden die Hintergrundvorgänge, welche beispielsweise durch das Aufrufen von Methoden in der File-Klasse ausgelöst werden, erläutert wurden, stellt das in der Stunde zu bearbeitende Projekt auch für das Themenfeld der Betriebssysteme einen exemplarischen Anwendungsfall dar. Die Gegenwartsbedeutung ergibt sich jedoch nicht nur aus der Integration des Wissens über Betriebssysteme als horizontalen Transfer an sich, im Rahmen der Begründung der Themenwahl für die Unterrichtsreihe wurde desweiteren bereits darauf verwiesen, dass mit der Befähigung der SuS, das Betriebssystem über C# programmatisch zu manipulieren, neue Zugänge zum Themenfeld der Betriebssysteme für diese geschaffen werden, wobei die Ergänzung der Entwicklersicht die SuS zur multiperspektivischen Auseinandersetzung mit dem Gegenstand der Betriebssysteme befähigt.¹⁴⁶ Weitere Aspekte der Gegenwartsbedeutung ergeben sich bezüglich der Reaktivierung der Java-Kenntnisse der SuS, wobei diese Gegenwartsbedeutung auf jede Stunde dieser Reihe anwendbar ist, in der hier vorgestellten Stunde jedoch insbesondere durch die erneute Erarbeitung der Kontrollstrukturen in C#, welche ebenso in Java vorkommen, gefördert wird.

Die Zukunftsbedeutung des in der Unterrichtsstunde behandelten Themas ergibt sich in erster Linie natürlich aus dem Vorgriff auf das Themenfeld „Softwareentwicklung“, welcher (wie oben erläutert) durch die Realisierung des Projekts mithilfe der neu eingeführten Dreischritt-Methodik in der Stunde verwirklicht wird, mit dem Ziel, ein grundlegendes Verständnis dafür zu schaffen, dass Software Engineering mehr ist als Implementierung, sondern diese in bestimmten Phasen bzw. Aktivitäten vor- und nachbereitet werden muss, um somit ein qualitatives Mindestmaß der Software bezüglich bestimmter, zuvor definierter Gütekriterien zu sichern. Desweiteren wird über die im Arbeitsauftrag des Arbeitsblatts „Projekt Dateikopierer“ gegebenen Hinweise zur Anforderungsanalyse versucht, die SuS für das Konzept der Trennung von Design- und logischem Code zu sensibilisieren. Hierbei wurde bereits in der Begründung der Themenwahl für die Unterrichtsreihe ausgeführt, dass jenes Konzept ebenso in anderen Sprachen und auf anderen Plattformen, wie bspw. bei Java in Verbindung mit Android, Anwendung findet. In diesem Zusammenhang wurde ebenso darauf hingewiesen, dass das Thema Android-Programmierung am Humboldt-Gymnasium innerhalb der 12. Jahrgangsstufe behandelt wird, weshalb durch die Thematisierung der Code-Trennung in C# ein weiterer vertikaler Transfer als Vorgriff auf die Android-Programmierung stattfand.¹⁴⁷ Ebenso ist es wahrscheinlich, dass dieses Konzept den SuS, falls sie sich nach der Schule weiterhin mit Informatik auseinandersetzen, im höheren Bildungswesen bzw. in ihrem Beruf begegnet, da neben Android vor allem die neuen Technologien und UI-Konzepte bei Microsoft (wie WPF, Windows Phone,

¹⁴⁵ Hierin wurde ein vertikaler Transfer identifiziert.

¹⁴⁶ Vgl. 2.1.2 Begründung der Themenwahl für die Unterrichtsreihe.

¹⁴⁷ Vgl. ebd.

Silverlight oder Windows Store) die Trennung von Oberfläche und Logik beinhalten.¹⁴⁸ Durch die Wiederholung der aus Java bekannten Strukturen in C# wird weiterhin das nächste große Themenfeld, die Programmierung mit PHP vorbereitet, wobei diese Zukunftsbedeutung wieder jeder der vier Stunden in dieser Reihe beigemessen werden muss.¹⁴⁹

Aus den bisher geleisteten Ausführungen zur Aufgabenstellung und ihrer Lösung sowie zur didaktischen Analyse wurde bereits zum Teil ersichtlich, inwiefern auf solche von dem Rahmenlehrplan und den EPA formulierten Standards Bezug genommen wird, welche im Rahmen der Bedingungsanalyse als für die Stunde relevant identifiziert wurden.¹⁵⁰ So entspricht die den SuS gestellte Anforderung, ihr in den letzten Stunden über die Sprache C# erarbeitetes Wissen zur Durchführung des Schritts der Implementierung zu reaktivieren, der Anforderung der EPA, welche als „Wiedergabe von bekannten Sachverhalten aus einem abgegrenzten Gebiet im gelernten Zusammenhang“ definiert und die dem Anforderungsbereich I zugeordnet wurde.¹⁵¹ Für den Themenbereich „Algorithmen“ wurde die Anforderung mit dem „Wiedergeben von Begriffsdefinitionen, Regeln, Zusammenhängen, bekannten Verfahren, einfachen Algorithmen, einfachen Modellierungen und Strukturen in einer im Unterricht behandelten Darstellungsform“ spezifiziert.¹⁵² Die wiederholte Anwendung des in der Stunde eingeführten Dreischritts wiederum rekurriert auf „die Anwendung bekannter Verfahren, Methoden und Prinzipien der Informatik zur Lösung eines neuen Problems aus einem bekannten Problemkreis“ sowie „die selbständige Übertragung des Gelernten auf vergleichbare neue Situationen, wobei es entweder um veränderte Fragestellungen oder um veränderte Sachzusammenhänge oder um abgewandelte Verfahrensweisen gehen kann“, wobei eine solche selbständige Übertragung im Rahmen der selbständigen Anwendung des Dreischritts auf die zweite und dritte Anforderung geleistet wird.¹⁵³ Dabei entsprechen die für die Stunde formulierten Feinziele, nach denen die SuS lernen sollen, den in der Stunde eingeführten Dreischritt der Anforderungsanalyse, der Planung der Anforderungsumsetzung und der Implementierung als Methodik zur Problemlösung für solche Aufgaben zu erkennen und anzuwenden, bei denen die SuS aus einer gegebenen, textuell codierten Anforderungsbeschreibung ein Softwareprojekt realisieren sollen, und diese durch die zielbezogene Reflexion ihres eigenen Handelns bei der Realisierung des Dateikopierer-Projekts einsehen, dass die Realisierung eines Softwareprojekts nicht mit seiner Implementierung gleichzusetzen ist, sondern dass die verschiedenen Aktivitäten, welche die Implementierung vor- bzw. nachbereiten, ebenso Gegenstand der Realisierung und somit des Softwareentwicklungsprozesses sind, dem im Anforderungsbereich III formulierten EPA-Standard der „planmäßige[n] Verarbeitung komplexer Gegebenheiten mit dem Ziel, zu selbständigen Gestaltungen bzw.

¹⁴⁸ Vgl. <http://msdn.microsoft.com/de-de/library/windows/apps/xaml/hh465340.aspx> [15.10.2014].

¹⁴⁹ Die Möglichkeiten, wie im Themenfeld PHP durch den Fachlehrer auf die Auseinandersetzung mit C# Bezug genommen werden kann, wurden bereits bei der Begründung der Themenwahl für die Unterrichtsreihe dargelegt und sollen deshalb hier nicht noch einmal wiederholt werden (vgl. 2.1.2 Begründung der Themenwahl für die Unterrichtsreihe).

¹⁵⁰ Vgl. 2.3.1 Richtlinien- und Fachvorgaben.

¹⁵¹ Vgl. http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01_EPA_Informatik.pdf [15.10.2014], S. 10.

¹⁵² Hierbei wurde die Zuordnung dieses EPA-Standards aus dem Anforderungsbereich I durch Fothe vorgenommen (vgl. https://ddi.informatik.uni-erlangen.de/EN/service/events/FachgruppeDDI2008/vortragsfolien/Fothe_giws08.pdf [15.10.2014], S. 14).

¹⁵³ Standards aus dem Anforderungsbereich II der EPA (vgl. http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01_EPA_Informatik.pdf [15.10.2014], S. 10 f.).

Deutungen, Folgerungen, Begründungen, Wertungen zu gelangen.“¹⁵⁴ Die Realisierung des Dateikopierers mithilfe bekannter Programmierwerkzeuge (d. h. dem Visual Studio) und -technologien (d. h. C#, .Net, WPF und XAML) sowie die Verwendung bekannter Namensräume, Klassen und Methoden aus dem .Net Framework zur Implementierung des Projekts rekurren wiederum auf das planvolle „Einsetzen bekannter Informatiksysteme zur Lösung einer neuen Problemstellung aus einem bekannten Bereich“¹⁵⁵ sowie das „Nutzen vorhandener Programmbibliotheken für die eigene Problemlösung“.¹⁵⁶ Weiterhin wurde hervorgehoben, dass die SuS (insbesondere die leistungsstärkeren) das Dokumentieren ihres Arbeits- und Lernprozesses üben sollen, indem sie nach jeder Aktivität des Dreischritts ein zu dieser adäquates Produkt anfertigen, was dem „Dokumentieren einer Problemlösung mit angemessenen Darstellungsmitteln“ entspricht.¹⁵⁷

Nachfolgend soll der in 2.6 tabellarisch dargestellte Stundenverlauf näher erläutert werden. Bei der Erläuterung orientiere ich mich an Meyers didaktischem Sechseck,¹⁵⁸ wobei ich den Schwerpunkt auf den methodischen Gang sowie die dahinterstehenden didaktischen Überlegungen lege:

Vorbereitung des Lernorts

Damit Klassen- und Fachräume als „vorbereitete Umgebung“ nach Meyer gelten, müssen einige Bedingungen erfüllt sein. Dies umfasst unter anderem die funktionale Einrichtung des Klassenraums, eine adäquate Belüftung, Beleuchtung und Akustik, genügend Platz zur Befriedigung der Bewegungsbedürfnisse der SuS, das Vorhandensein von ausreichendem Lernwerkzeug in ausreichender Zahl, genügend Arbeitsplätze sowie eine gute Ordnung, „gekennzeichnet durch klare Regeln, gemeinsam vereinbarte Rituale und entlastende Routinen.“¹⁵⁹ Aus der Betrachtung der materiellen Lernvoraussetzungen und der Raumausstattung im Rahmen der Bedingungsanalyse geht hervor, dass insbesondere im Bereich der Lernwerkzeuge erfahrungsgemäß nicht die für den Unterricht notwendigen Bedingungen vorherrschen.¹⁶⁰ So stehen die Laptops, auf denen die Arbeitsumgebung für die SuS mit der Installation von Windows 7 und dem Visual Studio 2012 bereits eingerichtet wurde, normalerweise in einer Ladestation im Vorbereitungsraum des Fachkollegiums. Sie müssen daher vor der Stunde in den Klassenraum transportiert, an den Strom angeschlossen und insoweit rechtzeitig gestartet werden, als dass der Fachlehrer das bei allen Laptops anders lautende Administratorpasswort eingeben kann und die SuS vor dem Beginn der Unterrichtsstunde dann noch in der Lage sind, das Visual Studio vollständig zu starten. Auch wenn die SuS im ersten Teil der Stunde noch nicht mit der Programmierumgebung arbeiten sollen, ist es dennoch sinnvoll, diese vor dem Beginn der Stunde starten zu lassen. Der Grund hierfür liegt in der Größe und Ressourcenintensivität des Programms, welche dazu führen kann, dass das Visual Studio lange Ladezeiten benötigt, abstürzt oder auf manchen Laptops nicht korrekt gestartet werden kann. Wenn man die SuS das Visual Studio erst kurz vor

¹⁵⁴ Ebd., S. 11. Hierbei wurde bereits erwähnt, dass die SuS über die Durchführung des Projekts ansatzweise in die Lage versetzt werden sollen, den Softwareentwicklungsprozess als Vorgehensweise zur Sicherung eines bezüglich festgelegter Gütekriterien geforderten, qualitativen Mindestmaßes zu werten und anzuerkennen.

¹⁵⁵ Diesem EPA-Standard aus dem Anforderungsbereich II ordnete Fothe der Problemlösekompetenz zu (vgl. https://ddi.informatik.uni-erlangen.de/EN/service/events/FachgruppeDDI2008/vortragsfolien/Fothe_giws08.pdf [15.10.2014], S. 17.).

¹⁵⁶ Diesem EPA-Standard aus dem Anforderungsbereich II ordnete Fothe der Problemlösekompetenz zu (vgl. ebd., S. 17.).

¹⁵⁷ Diesem EPA-Standard aus dem Anforderungsbereich II ordnete Fothe der Problemlösekompetenz zu (vgl. ebd., S. 17.).

¹⁵⁸ Vgl. Meyer 2012, S. 191-214.

¹⁵⁹ Ebd., S. 213.

¹⁶⁰ Vgl. 2.3.4 Institutionelle Rahmenvorgaben, materielle Lernvoraussetzungen und Raumausstattung.

dessen Einsatz starten lassen würde, würde man riskieren, dass solche Probleme bzw. ungeplanten Ereignisse den Unterricht in seinem geplanten Verlauf aufhalten. Da die SuS mindestens zu zweit an dem Dateikopierer-Projekt arbeiten, werden maximal lediglich sechs Laptops mit Windows 7 und Visual Studio 2012 benötigt. Desweiteren muss sichergestellt werden, dass die in den Anhängen 1-3 abgebildeten Arbeitsblätter in ausreichender Stückzahl vorliegen, wobei neben den 12 SuS, dem Fachlehrer und dem Lehrenden auch noch eventuell hinzukommende Hospitanten bedacht werden müssen. Die Arbeitsblätter sollten den anwesenden Personen ebenfalls vor Stundenbeginn ausgeteilt werden, wobei gegenüber den SuS die Bitte formuliert werden sollte, die Blätter zunächst in der ihnen ausgeteilten Reihenfolge liegenzulassen. Neben den Lernwerkzeugen muss insbesondere auf die Vorbereitung der Lehrwerkzeuge geachtet werden. Dabei muss der Beamer so positioniert werden, dass das projizierte Bild zum einen groß genug ist und zum anderen die weit unten hängenden Lampen nicht in dieses hineinragen. Weiterhin sollte der Beamer schon an den Laptop des Lehrers angeschlossen sein. Auch beim Lehrer-Laptop ist es wichtig, die Programmierumgebung frühzeitig zu starten, da den SuS zum Beginn der Stunde das lauffähige Projektergebnis (d. h. die Dateikopierer-Software) über diesen gezeigt werden soll. Weiterhin sollten die Erzeugnisse vorhergehender Stunden, wie beispielsweise Tafelbilder, entfernt oder, wenn diese noch gebraucht werden, möglichst vor den SuS verborgen werden, um diese nicht zu verwirren bzw. um nicht Kontexte zu diesen Erzeugnissen zu suggerieren. In der Bedingungsanalyse wurde bereits darauf hingewiesen, dass die Ausrichtung der Bänke im Klassenraum weitgehend fixiert ist und sich daher eine Anpassung der Bestuhlung als schwierig gestaltet.

10:45-10:50 – Einstieg / Informieren über den Stundenablauf

Mit dem Beginn der Stunde beherzige ich Meyers Ratschlag eines „informierenden Unterrichtseinstiegs“, bei dem die SuS über den Ablauf der Stunde informiert werden.¹⁶¹ Hierfür eignet sich die in 2.2.1 formulierte Aufgabenstellung der Stunde,¹⁶² welche zwar nicht wortwörtlich wiedergegeben werden muss, die Vorstellung des Stundenablaufs sollte sich jedoch inhaltlich mit der Aufgabenstellung decken und auch die durch die Aufgabenstellung vorgegebene Struktur sollte bei der Stundenvorstellung eingehalten werden. Die Eignung der Aufgabenstellung ergibt sich dabei nicht nur aus dem Umstand, dass den SuS durch diese die Arbeitsschritte vermittelt werden, welche sie in der Stunde durchzuführen haben, sondern den SuS für die einzelnen Arbeitsschritte ebenfalls transparent gemacht wird, welche Sozialformen, welche Handlungsmuster und welche Hilfsmittel mit dem jeweiligen Schritt konnotiert werden. Hierzu gehört insbesondere die Transparenz über den durch die Stunde intendierten vertikalen Transfer, welcher durch die Zusammensetzung einfacher C#-Elemente zu einer komplexeren Problemlösung geleistet wird. Den SuS soll bewusst gemacht werden, dass sie neben dem in den vergangenen drei Stunden erworbenen Wissen insbesondere jene Kenntnisse benötigen, welche sie sich in der heutigen Stunde vor dem Beginn des Projekts erarbeiten, um dieses realisieren zu können. Dabei soll im besonderen Maße die Bedeutung der in der vorhergehenden Stunde erarbeiteten Kontrollstrukturen betont werden. Jene Hervorhebung liegt darin begründet, dass die Erarbeitung dieser Kontrollstrukturen erst innerhalb dieser Stunde zum Abschluss kommt. Für diese Erarbeitung ist es dabei aus psychologischer Sicht womöglich hinderlich, dass mit der hier vorgestellten Stunde die Unterrichtsreihe zu C# endet und die Programmiersprache in den folgenden Reihen nicht weiter thematisiert wird. Die Frage, welche sich die SuS stellen könnten, wäre also möglicherweise, warum sie sich in der letzten Stunde noch Theoriewissen über eine Sprache aneignen sollten, welche sie für den weiteren Unterricht womöglich eh nicht mehr brauchen? Aus diesem Grund soll durch die

¹⁶¹ Vgl Meyer 2012, S. 48.

¹⁶² Vgl. 2.2.1 Ausformulierung der Aufgabenstellung.

Hervorhebung des theoretischen Wissens über Kontrollstrukturen mit der eher motivierenden, praktischen Anwendung eines benutzbaren bzw. funktionsfähigen Programms verknüpft werden. Da die Bedeutung jener Kontrollstrukturen für das Dateikopierer-Programm für die SuS womöglich nicht in dem gleichen Maße ersichtlich ist wie die Verwendung der Klasse File für dieses, soll diese Bedeutung explizit bereits zum Beginn der Stunde kenntlich gemacht werden. Hierzu präsentiert der Lehrer den SuS mithilfe seines Laptops und dem Beamer eine durch ihn vorbereitete Musterlösung des Programms,¹⁶³ welches die SuS innerhalb der Stunde realisieren sollen. Neben dem Programm öffnet der Lehrer einen Ordner, beispielsweise „Dokumente“, als Zielordner für den Kopiervorgang. Anschließend wählt er über den Button „Quelldatei“ eine Datei aus, welche kopiert werden soll, und über den Button „Ziel“ den Ordner, den er bereits geöffnet hat. Danach gibt er eine beliebige Anzahl an Kopien ein, welche zu Veranschaulichungszwecken jedoch groß gewählt, in jedem Fall jedoch größer 1 sein sollte, drückt „Starte Kopiervorgang“ und richtet das Augenmerk der SuS auf den zuvor geöffneten Zielordner, in dem plötzlich so viele Kopien der gewählten Datei erscheinen, wie der Lehrer als Zahl vorgegeben hat. Der bis zu diesem Zeitpunkt andauernde Lehrervortrag geht nun in ein fragend-entwickelndes Unterrichtsgespräch über, welches der Lehrer damit beginnt, dass er die SuS nach der Bedeutung der Schleife als Struktur zur Wiederholung einer Anweisung bzw. eines Anweisungsblocks fragt. Zwar wurde bereits in der Aufgabenlösung erläutert, dass sich nur eine der drei Vierergruppen in der vorhergehenden Stunde mit der for- und der while-Schleife in C# beschäftigt hat, da diese Strukturen jedoch allen SuS bereits aus Java bekannt sind, halte ich es dennoch für legitim, die Frage nach ihrer Bedeutung für dieses Programm bereits zu diesem Zeitpunkt gegenüber dem gesamten Plenum zu formulieren. Das Ziel dieses fragend-entwickelnden Unterrichtsgesprächs ist es, dass die SuS erkennen, dass eine Schleife notwendig ist, um den eigentlichen Kopiervorgang beliebig oft zu wiederholen. Die Dauer des Gesprächs hängt davon ab, wie schnell die SuS die Schleife als Mittel zur Wiederholung des Kopiervorgangs begreifen. Dabei sollte vor allem darauf geachtet werden, dass nicht die leistungstärkeren SuS, vor allem nicht die beiden C#-Experten, die Antworten geben, sondern in erster Linie die leistungsschwächeren SuS durch ihre Performanz zeigen, dass sie ein Grundverständnis für die Funktionalität von programmatischen Schleifen besitzen.

10:50-11:10 Ergebnissicherung / Fortsetzung der Erarbeitung der letzten Stunde

Nach der Einleitung wird die in der vorangegangenen Stunde begonnene Erarbeitung der Kontrollstrukturen, welche in der Einleitung dieser Stunde gerade motiviert wurde, fortgesetzt. In der Aufgabenlösung wurde bereits dargestellt, wie sich die Erarbeitung der Kontrollstrukturen in der vorangegangenen Stunde ereignete, weshalb jener Ablauf zur Vermeidung der Redundanz hier nicht noch einmal rezipiert, sondern der Fokus auf die dahinterstehende, didaktische Methode gerichtet werden soll: Hierbei handelt es sich um die sogenannte Puzzle-Methode, welche Hartmann, Näf & Reichert wie folgt charakterisieren:

Es werden Gruppen von 3-6 Schülerinnen und Schülern gebildet: Jede Gruppe erhält eine Aufgabe und das nötige Material und muss sich in der Expertenrunde mit der Aufgabe vertraut machen. Die Schüler erarbeiten einen Teil des Stoffes selber. Anschließend werden die Gruppen neu zusammengestellt, und die Experten geben ihr Wissen in der Unterrichtsrunde anderen weiter [...]. Die Lernenden bearbeiten bei der Puzzle-Methode selbstständig ein überschaubares Thema und bilden sich zu Experten aus. Genau diese Fähigkeit wird später im Umgang mit dem Computer gefragt sein. [...] Wichtig beim Einsatz der Puzzle-Methode ist, dass alle Schülerinnen und Schüler ein Erfolgserlebnis haben. Deshalb empfiehlt es sich, den schwächeren Schülern

¹⁶³ Vgl. <http://www.randco.square7.ch/Dateikopierer.zip> [16.10.2014].

einfachere Themen zuzuteilen. Besonders geeignet sind Themen, die sich in ungefähr gleich große Teile aufteilen lassen.¹⁶⁴

Dieser Konzeption bin ich gefolgt, mit dem Unterschied, dass ich die Gruppen für die Unterrichtsrunde als erstes festgelegt habe. Der Grund hierfür war, dass ich mir bereits in der letzten Stunde bewusst war, dass wir in jener Stunde zeitlich nur die Expertenrunde schaffen werden und die Unterrichtsrunde auf die nächste, d. h. die hier vorgestellte Stunde verlagert werden muss. Die Gruppenbildung für die Unterrichtsrunde bringt mehrere Vorteile mit sich: Zunächst muss in dieser Stunde nicht wieder Organisatorisches geklärt werden, da die Gruppen bereits in der letzten Stunde gebildet wurden und ich die SuS gebeten habe, sich aufzuschreiben, mit wem sie in einer Gruppe sind. Noch viel wichtiger ist, dass der Gruppenbildung Gedanken zur inneren Differenzierung zugrunde liegen. In diesem Zusammenhang wurde bereits in der Darstellung der Aufgabenlösung erläutert,¹⁶⁵ dass die vier zuvor gegründeten, leistungsheterogenen Dreiergruppen für die spätere Unterrichtsrunde mit jeweils einem Vertreter, in Abhängigkeit vom Leistungsstand, auf die drei in sich leistungshomogenen Vierergruppen aufgeteilt wurden. Ebenfalls wurde ausgeführt, dass die innere Leistungsdifferenzierung über die unterschiedliche Komplexität des von der Gruppe zu bearbeitenden Gegenstands sowie der Arbeitsaufträge (bezogen auf die zur Verfügung stehenden Hilfsmittel sowie ihrem Kompetenzniveau bzw. ihrer Verortung in Blooms Taxonomie) erfolgte.¹⁶⁶ Hierbei wurde bereits im Rahmen der Bedingungsanalyse darauf hingewiesen, dass (vor allem mit Blick auf die leistungsschwächeren SuS) die Arbeitsaufträge zur Erarbeitung der Sprachelemente in C# (und somit auch der Kontrollstrukturen der Sprache) in der Form gestaltet wurden, dass sich die SuS die jeweilige Struktur zunächst in ihrem Aufbau und ihrer Funktionsweise erarbeiten und diese anschließend in diversen Kontexten bzw. Szenarien anwenden mussten. Hierdurch sollte sichergestellt werden, dass die SuS die jeweilige Kontrollstruktur nach der erneuten Erarbeitung in C# auch anwenden können und somit für die jeweilige Struktur mindestens die dritte Kompetenzstufe erreichen. Dies ist notwendig, damit die SuS, wenn sie als Experten für ihre jeweilige Struktur in die Gruppen der Unterrichtsrunde gehen, in der Lage sind, ihre Struktur nicht nur im Bezug auf ihren Aufbau und ihre Funktionsweise gegenüber den anderen Gruppenmitgliedern zu erläutern, sondern ebenfalls die Übungsaufgaben mit diesen durchgehen, ihre Lösungen erläutern bzw. mit den anderen Gruppenmitgliedern Lösungen erarbeiten können. Die Lehrperson muss bei der Formulierung des Arbeitsauftrags darauf achten, dass sie sich an die SMART-Kriterien hält. Ein solcher Arbeitsauftrag könnte lauten:

In der letzten Stunde habt ihr euch in Expertengruppen eine oder zwei Kontrollstrukturen der Sprache C# erarbeitet. Vor der Expertenrunde habe ich euch in Dreiergruppen eingeteilt und euch gebeten, zu notieren, mit wem ihr in einer Dreiergruppe gewesen seid. Bitte findet euch wieder in diesen Dreiergruppen ein. [Pause, bis die SuS sich sortiert haben] Jedes Mitglied eurer Gruppe hat in der letzten Stunde in einer anderen Expertengruppe gearbeitet. Das bedeutet, dass ihr in eurer Gruppe jeweils die einzigen Experten für eure Struktur seid. Eure Aufgabe besteht nun darin, euren Gruppenmitgliedern mithilfe eurer Notizen aus der letzten Stunde den Aufbau und die Funktionsweise eurer Struktur zu erklären. Versucht dabei, eure Erläuterungen immer an einem Beispiel zu veranschaulichen. Hierfür sollt ihr die Übungsaufgaben verwenden, welche ihr in der letzten Stunde in eurer Expertengruppe bearbeitet habt. Erarbeitet anschließend zusammen mit den anderen Gruppenmitgliedern Lösungen für die Übungsaufgaben, welche ihr nicht bereits in der Erklärung verwendet habt. Ziel ist es, dass eure Mitschüler nach der Besprechung der jeweiligen Struktur in der Lage sind, diese in ihrem Aufbau und in ihrer Funktionsweise zu verstehen und selbständig in den Übungsaufgaben anzuwenden. Für die Erklärung und die Lösung der Übungsaufgaben habt ihr für jede Struktur

¹⁶⁴ Hartmann, Näf & Reichert 2007, S. 75.

¹⁶⁵ Vgl. 2.2.2 Mit der Aufgabenstellung verbundene Lernziele und Aufgabenlösung.

¹⁶⁶ Vgl. ebd.

fünf Minuten Zeit. Ich sage an, wann die fünf Minuten um sind und ihr zur nächsten Struktur übergehen sollt. Die Experten für die Schleifen sollen sowohl die for- als auch die while-Schleife in jeweils fünf Minuten erklären.

Mit diesem Arbeitsauftrag sind verschiedene Feinziele für die Unterrichtsstunde konnotiert. Trivial ist dies ersichtlich bei den mit der Aufgabe verbundenen kognitiven Feinzielen, nach denen die SuS lernen sollen, die Struktur der if-else-Verzweigung, der switch-case-Verzweigung bzw. der for- und der while-Schleife als solche zu benennen und ihren Aufbau sowie ihre Funktionsweise in C# zu erläutern. Diese Ziele ergeben sich aus fachlichen bzw. curricularen Intentionen, welche auf den vertikalen Transfer und die Zukunftsbedeutung des Stundenthemas rekurren: So wurde bereits bei der Beschreibung der Einleitung dargestellt, dass die Zielbegründung, nach der die hier erarbeiteten Kontrollstrukturen zur Realisierung des nachfolgenden Softwareprojekts benötigt werden, gegenüber den SuS transparent dargelegt wird. Weiterhin treten die behandelten Kontrollstrukturen auch in PHP in sehr ähnlicher Syntax auf, sodass die Erarbeitung der Strukturen in C# unter gleichzeitiger Reaktivierung der Java-Kenntnisse eine wiederholende Vorbereitung auf die Unterrichtseinheit zur Skriptsprache darstellt und dem Fachlehrer erspart, jene Strukturen in PHP nochmals ausführlich zu behandeln. Als weitere Feinziele für diesen Unterrichtsabschnitt wurde formuliert, dass die SuS lernen sollen, ihr Wissen über die if-else-Verzweigung, die switch-case-Verzweigung bzw. die for- und die while-Schleife in C# so zu organisieren, dass sie in der Lage sind, deren Aufbau und Funktionsweise gegenüber ihren Mitschülerinnen und Mitschülern in einem fünfminütigen Kurzreferat zu vermitteln und dabei zu sichern, dass sie das Dargestellte auch verstanden haben. Mit diesen Lernzielen wird eine Aktivierung bzw. eine Verbesserung der Methodenkompetenz der SuS angestrebt. Diese besteht Meyer zufolge „aus der Fähigkeit, das eigene Lernen bewusst, zielorientiert, ökonomisch und kreativ zu gestalten und dabei auf ein Repertoire geeigneter Methoden und Lernstrategien zurückzugreifen.“¹⁶⁷ Diese bei Meyer definierten Transferleistungen müssen von den Experten erbracht werden, da sie unter Beachtung einer strikten Zeitvorgabe (5 Minuten) und unter Zuhilfenahme ihres angeeigneten Wissens sowie eines vorgegebenen Materialpools (den Übungsaufgaben und Arbeitsaufträgen aus der letzten Stunde) kreativ ein Lehrkonzept selbst erstellen bzw. organisieren müssen, welches zielorientiert und (zeit-)ökonomisch gewährleistet, dass ihre Mitschülerinnen und Mitschüler den von ihnen vorgestellten Gegenstand verstanden haben und anwenden können. Gelingt es ihnen, das eigene Lernen und das ihrer Mitschülerinnen und Mitschüler bewusst, zielorientiert, ökonomisch und kreativ zu gestalten, so kann „[d]urch das Vermitteln des erarbeiteten Wissens an andere [...] das Selbstwertgefühl gesteigert“ werden, wie es Hartmann, Näf & Reichert formulieren.¹⁶⁸ Als weitere mögliche positive Auswirkung, die Schüler zu Lehrern zu machen,¹⁶⁹ führen die Autoren an, „dass Erklärungen von Schülern in der Sprache der Schüler formuliert werden und deshalb oft besser verständlich und weniger abstrakt sind als die Erklärungen des Lehrers.“¹⁷⁰ Mit der Wahl der Methodik wurde genau diese Auswirkung erhofft bzw. intendiert, weshalb auch das affektive Feinziel formuliert wurde, nach dem die SuS durch den Austausch über die Kontrollstrukturen mit den anderen Gruppenmitgliedern erkennen lernen sollen, „dass der Austausch oder die Kooperation mit anderen Menschen einen Mehrwert für ihre eigene Wissensbasis generiert und Lernen oftmals bedeutet, von und mit anderen Menschen zu lernen.“¹⁷¹ Dieses Feinziel intendiert somit die Stärkung der sozial-kommunikativen Kompetenz der SuS. Diese soziale Begründung ergibt sich ebenfalls für ein weiteres affektives Feinziel, nach dem die SuS in ihrer Rolle als Experten für eine bestimmte

¹⁶⁷ Meyer 2012, S. 154.

¹⁶⁸ Hartmann, Näf & Reichert 2007, S. 75.

¹⁶⁹ Vgl. Meyer 2012, S. 126.

¹⁷⁰ Hartmann, Näf & Reichert 2007, S. 75.

¹⁷¹ Vgl. 2.4.2 Aus den Grobzielen abgeleitete Feinziele der Unterrichtsstunde.

Kontrollstruktur lernen sollen, dass sie sich mit ihrer Wissensbasis in einer Monopolstellung befinden und sie dabei die ungeteilte Verantwortung tragen, jene Monopolstellung so gut wie möglich aufzulösen.¹⁷² Aus diesem Grund wird im Arbeitsauftrag gegenüber den SuS auch explizit hervorgehoben, dass es in ihrer Gruppe keinen weiteren Experten für ihre jeweilige Kontrollstruktur gibt.

11:10-11:13 Erarbeitung von benötigten C#-Sprachelementen

Nachdem die SuS ihre Kenntnisse über die Kontrollstrukturen in C# über die Puzzle-Methode gesichert bzw. erweitert haben, erhalten sie den Auftrag, sich die auf der Vorderseite des Arbeitsblatts „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“ dargestellte Dokumentation zu den dort ausgewählten Sprachelementen von C# durchzulesen und dabei sicherzustellen, dass sie wissen bzw. verstanden haben, welche Funktion die Elemente jeweils besitzen bzw. in welchen Anwendungskontexten diese eingesetzt werden. Falls dies für ein Element unklar ist, sollen sich die SuS dieses Element mit der entsprechenden Frage nach der Funktion bzw. dem Anwendungskontext notieren. Das damit verbundene Feinziel, nach dem die SuS lernen sollen, „aus einer vorgegebenen Dokumentation auf die Funktionen und Anwendungskontexte der dort dargestellten Elemente zu schließen“, ergibt sich dabei eher aus einer fachlichen als aus einer methodischen Intention: Die fachliche Begründung für die Auseinandersetzung mit den Sprachelementen ergibt sich aus dem bereits erwähnten Umstand, dass diese Elemente für die spätere Implementierungslösung benötigt werden. Eine methodische Intention wurde mit diesem Arbeitsauftrag dagegen nicht verfolgt, da hier keine spezielle Methodik bzw. kein spezielles Handlungsmuster trainiert wird, sondern die SuS lediglich die elementaren Handlungen des Lesens und Notierens ausführen müssen. Da die Dokumentation bereits auf das Wichtigste reduziert wurde, müssen die SuS auch keine komplexeren kognitiven Leistungen bezüglich der Bearbeitung des Textes erbringen, welche ein bestimmtes Maß an Methodenkompetenz verlangen würden. Sie sollen lediglich verstehen, was der Text fachlich aussagt, weshalb diesem Lernziel lediglich die Stufe des Verstehens in Blooms Lernzieltaxonomie zugeordnet wurde.¹⁷³ Die Arbeit am Text soll zu diesem Zeitpunkt in Einzelarbeit erfolgen. Die SuS sollen dabei zunächst nur auf der Grundlage ihrer eigenen Wissensbasis reflektieren, welche Elemente sie bereits kennen, ob sie die dargelegte Funktionsbeschreibung verstehen und welche Elemente ihnen gänzlich unbekannt sind.

11:13-11:18 Sicherung der Sprachelemente

Zwar traten viele der in der Dokumentation vorgestellten Elemente bereits in Beispielen aus den vorhergehenden Unterrichtsstunden der Einheit auf, da diese jedoch nur beiläufig thematisiert wurden und es sich hierbei nicht um grundlegende Elemente der Sprache C# handelt, sondern diese spezielle Funktionen umfassen, rechne ich damit, dass die SuS Fragen zu diesen Elementen haben. Aus diesem Grund sollen die SuS, welche zuvor ihr Wissen über diese Elemente unter Zuhilfenahme der Dokumentation reflektiert haben, offengebliebene Fragen an das Plenum stellen. Im Rahmen eines offenen Unterrichtsgesprächs¹⁷⁴ sollen die SuS also die Fragen ihrer Mitschülerinnen und Mitschüler beantworten, wobei der Lehrer lediglich als Moderator des Unterrichtsgesprächs fungiert. Hiermit wird wieder die bereits bei der Erarbeitung der Kontrollstrukturen angewendete Idee verknüpft, die SuS zu Lehrern zu machen. Hierbei bietet sich vor allem für die leistungstärkeren SuS, insbesondere für die beiden Schüler, welche sich C# autodidaktisch angeeignet haben, die Möglichkeit, ihr weiterführendes Expertenwissen (ihre Kompetenz) für das

¹⁷² Vgl. ebd.

¹⁷³ Bloom (zit. nach Sitte 2001, S. 474).

¹⁷⁴ Vgl. http://studienseminar.rlp.de/fileadmin/user_upload/studienseminar.rlp.de/gskus/bilder/Das_Unterrichtsgespraech.pdf [17.10.2014], S. 4 f.

Unterrichtsgeschehen förderlich einzubringen, wobei die Weitergabe des Expertenwissens der leistungsstärkeren SuS (Performanz) vor allem für die Kompetenz der leistungsschwächeren SuS gewinnbringend ist, weshalb diese (wie auch schon bereits in der Gruppenarbeit zu den Kontrollstrukturen) lernen, dass der Austausch oder die Kooperation mit anderen Menschen einen Mehrwert für ihre eigene Wissensbasis generiert und Lernen oftmals bedeutet, von und mit anderen Menschen zu lernen.

Sollten sich die SuS gegenseitig keine Fragen stellen oder vor dem Ablauf der hierfür eingeplanten Zeit ihre Diskussion beendet haben, behalte ich mir vor, vom Handlungsmuster des offenen in das des fragend-entwickelnden Unterrichtsgesprächs überzuleiten, wobei ich die in der Dokumentation vorgegebenen Sprachelemente mithilfe meines Laptops und des Beamers im Visual Studio in einfache Kontexte setze und die SuS nach der Funktion der jeweiligen Elemente in diesem Kontext frage.

11:18-11:25 Erarbeitung ausgewählter Methoden der File-Klasse

Wurde die methodische Begründung als Intention für die Erarbeitung der Vorderseite des Arbeitsblatts „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“ noch negiert, stellt sie für die Aufgabenstellung auf der Rückseite die grundlegende Motivation dar, mit der diese konzipiert wurde. Durch die Bearbeitung des Arbeitsauftrags zu den „Methoden der File-Klasse“ sollen die SuS lernen, welche Funktionen die angeführten Methoden der File-Klasse in C# besitzen, wobei sie selbständig in der Dokumentation des Microsoft Developer Network (MSDN) nach diesen suchen. Die selbständige Suche nach C#-Sprachelementen in der MSDN-Dokumentation unter der Bearbeitung von bestimmten Arbeitsaufträgen wurde mit den SuS in den vorangegangenen Unterrichtsstunden mehrmals praktiziert, sodass die SuS in diesem Arbeitsauftrag nicht mit einem völlig neuen Handlungsmuster konfrontiert werden, jedoch damit ein erst kürzlich erlerntes Handlungsmuster, nämlich die zielgerichtete Suche in vorgegebenen Dokumentationen, einüben und somit ihre Methodenkompetenz verbessern. Aus diesem Grund wurde dem Lernziel die Stufe des Anwendens auf Blooms Lernzieltaxonomie zugeordnet. Die Aufgabe besitzt in zwei Aspekten jedoch auch eine sozial-kommunikative Komponente: Diese ist zum einen dadurch gegeben, dass jeweils zwei Partner auf genau eine gemeinsame Ressource, nämlich den Laptop, zugreifen müssen. Hierbei wird intendiert, dass zwischen den Partnern bereits während der Recherche ein Aushandlungs- bzw. Diskussionsprozess stattfindet, in dem die gerade betrachteten Informationen auf ihre Wichtigkeit bzw. Verwendbarkeit für bzw. auf das zu formulierende Ergebnis evaluiert werden. Die SuS sollen also mithilfe ihres Partners üben, sich kritisch und zielorientiert mit einem vorgegebenen Informationsangebot und dessen Güte auseinanderzusetzen. Hierbei sollen die SuS jedoch nicht nur die vorgegebenen Informationen beurteilen, sondern ebenso lernen, die Wertungen ihres Partners über diese kritisch zu reflektieren, ihren eigenen Positionen / Haltungen als mögliche Alternativen unvoreingenommen, d. h. anhand objektiver Bewertungskriterien, gegenüberzustellen und, wenn sich die Wertungen ihres Partners als objektiv besser bzw. stichhaltiger erweisen, auch zu übernehmen.¹⁷⁵ Dieses sozial-kommunikative Feinziel soll auch durch die Notwendigkeit einer Einigung auf ein gemeinsames Ergebnis erreicht werden, wobei in diesem Fall der Fokus der sozialen Begründung im höheren Maße auf der Stärkung der Selbstkompetenz liegt. Gerade die leistungsschwächeren SuS sollen hierbei üben, ihre objektiv zieladäquaten, d. h. die Aufgabenstellung lösenden Vorschläge gegenüber den leistungsstärkeren SuS als eigene Position vorzustellen und diese auch zu behaupten. Damit gewährleistet ist, dass sich jedes Paar aus einem Schüler bzw. einer Schülerin aus der leistungsschwächeren und einem Schüler bzw. einer Schülerin aus der leistungsstärkeren Hälfte der Klasse zusammensetzt, werden diese Paare vom Lehrer festgelegt. Dabei wurde bereits geschildert, dass gegenüber

¹⁷⁵ Vgl. 2.4.2 Aus den Grobzielen abgeleitete Feinziele der Unterrichtsstunde.

den SuS vermittelt werden soll, dass sie in dieser Zusammensetzung für den Rest der Stunde zusammenarbeiten werden, womit intendiert wird, dass die SuS reflektieren, dass der Partner, mit dem sie die das Projekt vorbereitende Aufgabe lösen, auch der Partner ist, mit dem sie später das Projekt durchführen müssen und somit ein gewissen Eigeninteresse daran entwickeln, dass der Partner einen bestimmten Wissensstand über die später benötigten Sprachelemente erreicht. Aus dieser sozialen Begründung generiert sich das affektive Feinziel für diesen Arbeitsauftrag: Die SuS sollen „lernen, aus der Prämisse, dass das Wissen und die Beiträge ihrer Mitschülerinnen und Mitschüler zukünftig (beim Ausführen des Dreischritts) einen Mehrwert für ihren eigenen Wissensstand generiert bzw. dieser bis zu einem gewissen Grad vom Wissen und den Beiträgen ihrer Mitschülerinnen und Mitschüler ist, heraus ein Interesse daran zu entwickeln, dass ihre Mitschülerinnen und Mitschüler die Mindestanforderungen (bei den Vorbereitungsaufgaben für die Projektrealisierung) erfüllen.“¹⁷⁶

11:25-11:30 Sicherung der Methodenbeschreibung

Nach der Erarbeitung der File-Dokumentation durch die Partner sollen diese ihre Ergebnisse im Plenum vergleichen. Der Vergleich soll im Rahmen eines gelenkten Unterrichtsgespräches stattfinden, wobei der Lehrer die Partner der Reihe nach auffordert, ihre Dokumentation für jeweils eine Methode vorzustellen. Für diese Ergebnispräsentation, bei der von den Partnern lediglich erwartet wird, dass sie die Funktion sowie den Rückgabewert ihrer Methode benennen, sind insgesamt drei Minuten eingeplant. In den darauffolgenden beiden Minuten hält die Lehrperson einen kurzen Lehrervortrag, in dem sie den SuS mithilfe ihres Laptops und dem Beamer zeigt, welche Möglichkeiten es gibt, die Dokumentation eines Sprachelements im Visual Studio abzurufen. Hierbei verweist sie sowohl auf den Dokumentationscache, welcher von der IDE als Popup angezeigt wird, wenn man im Code über das entsprechende Sprachelement fährt, sowie auf den Objektkatalog, welcher die Dokumentationen der Sprachelemente aller Bibliotheken enthält, welche mit dem Visual Studio verbunden sind, auch wenn diese nicht originaler Bestandteil des .Net Frameworks. Hiermit wird das methodisch begründete Feinziel verfolgt, dass die SuS lernen, wie sie über das Visual Studio sowie über das Microsoft Developer Network (MSDN) auf die C#-Dokumentation zugreifen und diese nutzen können, um den Aufbau und die Funktionsweise von Sprachelementen zu untersuchen. Die SuS sollen also erkennen, dass für eine Bibliothek eine Vielzahl von Dokumentationen existieren, welche sich in ihrer Beschaffenheit variieren, weshalb es die Aufgabe der SuS ist, aus dem Dokumentationsangebot die für sie bzw. für ihr Problem adäquate Dokumentation auszuwählen und sich über diese zu informieren.

11:30-11:40 Erarbeitung des ersten Programmabschnitts

Aufgrund der fortgeschrittenen Seitenzahl und da der Ablauf der Projektrealisierung bereits in der Aufgabenlösung sowie im obigen Teil dieser didaktischen Strukturierung ausführlich dargestellt wurde, soll sich die Darstellung des Projekts lediglich auf den methodischen Gang sowie die Erläuterung der damit verbundenen Lernziele beschränken.

Es wurde bereits ausgeführt, dass die erste Anforderung auf dem Arbeitsblatt „Projekt Dateikopierer“ im Plenum über das Handlungsmuster eines fragend-entwickelnden Unterrichtsgespräches erarbeitet werden soll. Die Wahl des Handlungsmusters und der Sozialform ergab sich aus dem Umstand, dass der Dreischritt der Analyse, Planung und Implementierung erst mit der Erarbeitung dieser ersten Anforderung eingeführt wird. Dabei ist es das Ziel des Projekts, dass die SuS lernen sollen, den Dreischritt als Methodik zur Problemlösung für solche Aufgaben zu erkennen und anzuwenden, bei denen die SuS aus einer gegebenen,

¹⁷⁶ Vgl. ebd.

textuell codierten Anforderungsbeschreibung ein Softwareprojekt realisieren sollen. Hierzu sollen die SuS lernen, die textuell codierten Anforderungen soweit zu elementarisieren, dass die einzelnen Elemente der Anforderung jeweils auf ein oder wenige Sprachelemente gleichen Typs in C# abgebildet werden können. Weiterhin sollen die SuS lernen, die aus den Anforderungen ermittelten, zu implementierenden Sprachelemente unter Beachtung von möglichen Abhängigkeiten in eine konsistente Reihenfolge zu bringen, deren Abarbeitung eine widerspruchsfreie Implementierung ermöglicht. Dieses Feinziel ist sehr komplex und verlangt eine hohe methodische Kompetenz der SuS, welche in der Form von den SuS zu Beginn des Projekts nicht einfach erwartet werden kann, sondern über die Variation der für die einzelnen Durchläufe bestimmten Sozialformen, Handlungsmuster und Inhalte sukzessive aufgebaut werden muss. Die Wahl der Sozialform des Plenumsunterrichts mit dem fragend-entwickelnden Unterrichtsgespräch als Handlungsmuster intendiert die Schaffung eines sicheren Lernraums für die SuS: „Sicher“ bedeutet in diesem Zusammenhang, dass die SuS erkennen sollen, dass es für sie keine negativen Konsequenzen hat, wenn sie einen Fehler begehen und dass es bei der Erarbeitung der drei Schritte für die erste Antwort nicht nur eine mögliche, richtige Antwort gibt, sondern die Erzeugnisse beispielsweise in dem Detailgrad ihrer Formulierungen variieren können. Ziel ist es dabei, dass die SuS motiviert werden, sich an dem Gegenstand zu probieren, indem sie aktiv an der Plenumsdiskussion zur Genese des geforderten Gegenstands teilnehmen. Daher steht am Ende dieses ersten Durchlaufs des Dreischritts keine ausschließlich vom Lehrer vorgegebene Musterlösung, sondern ein mit den SuS im Rahmen des Unterrichtsgesprächs ausgehandeltes Erzeugnis. Hieraus ergibt sich auch die Aufforderung des Lehrers an die SuS, welche einen Vorschlag zur Umsetzung einer bestimmten Anforderung gemacht haben, nach vorne zu kommen und ihre Idee am Laptop des Lehrers zu formalisieren, sodass die an die Wand geworfene Lösung nicht vom Lehrer, sondern in erster Linie von den SuS (mit Unterstützung des Lehrers) generiert wurde. Die gemeinsame Erarbeitung soll für die SuS eine Grundlage darstellen, an der sie sich in den weiterführenden Durchläufen orientieren können. Aus diesem Grund ist es besonders wichtig, dass sich der vom Lehrer zum Stundenbeginn formulierte Gedanke, nach dem die Implementierung nur eine von mehreren Aktivitäten ist, wobei die anderen Aktivitäten, welche die Implementierung vor- bzw. nachbereiten, ebenso wichtig oder gar noch wichtiger sind, als die Implementierung selbst, auch im Ergebnis des ersten Durchlaufs widerspiegelt. Die Bedeutung ergibt aus dem methodischen intendierten Feinziel, wonach die SuS, „durch die zielbezogene Reflexion ihres eigenen Handelns bei der Realisierung des Dateikopierer-Projekts lernen [sollen], dass die Realisierung eines Softwareprojekts nicht mit seiner Implementierung gleichzusetzen ist, sondern dass die verschiedenen Aktivitäten, welche die Implementierung vor- bzw. nachbereiten, ebenso Gegenstand der Realisierung und somit des Softwareentwicklungsprozesses sind.“ Die SuS sollen diese Wertung der Aktivitäten also nicht nur (als Kompetenz) inkorporieren, sondern diese Verinnerlichung auch durch ihr eigenes Handeln (ihre Performanz) erkennbar machen.

11:40-11:45 Erarbeitung des zweiten Programmabschnitts

Eine erste Möglichkeit hierzu erhalten die SuS durch die Realisierung der zweiten Anforderung des Arbeitsblatts. Dabei sollen die drei Schritte der Realisierung nicht mehr im Plenum im Rahmen eines fragend-entwickelnden Unterrichtsgesprächs erarbeitet, sondern als Ergebnis eines Aushandlungsprozesses (Diskussion bzw. Schülergespräch) zwischen den Partnern generiert werden. Die größere Freiheit, welche den SuS durch den Wechsel der Sozialform und des Handlungsmusters für die Erarbeitung des zweiten Durchlaufs eingeräumt wird, geht jedoch auch mit dem Verlust der durch den Lehrer und das Plenum gegebenen Sicherheit einher. Durch den Wegfall eines Großteils der vorher sichernden Kommunikationspartner rückt die eigene Kompetenz der SuS deutlicher in den Fokus. Hierbei wurde bereits dargestellt, dass sich durch den Methodenwechsel nun nicht mehr passiv im Plenum verstecken kann,

sondern durch als Diskussion konzipierte Partnerarbeit gezwungen wird, aktiv Beiträge zur Genese der Anforderungsrealisierung zu leisten. Auf der anderen Seite führt der Methodenwechsel dazu, dass der jeweilige Partner nun die einzige Bezugsperson darstellt, mit der die SuS über die Aufgabenrealisierung reden können. Dies hat zur Folge, dass die SuS gezwungen sind, auf die Beiträge ihrer Partner einzugehen. Hiermit wird intendiert, dass die SuS infolge einer ertragreichen Diskussion über den Dreischritt und dessen Anwendung auf den konkreten Anforderungsabschnitt erkennen lernen, dass der Austausch über die in der Stunde gelernte Methodik mit dem Partner einen möglichen Mehrwert für sie darstellt, der zu einem besseren Verständnis der Methodik oder zu besseren bzw. vielfältigeren Arbeitsergebnissen führen kann. Dieses Ziel erwächst also wieder aus der Intention, dass die SuS durch die Reflektion ihres eigenen Handelns erkennen, dass Lernen oftmals bedeutet, von und mit anderen Menschen zu lernen und infolgedessen immer mehr dazu befähigt werden, selbst zu entscheiden, welche Sozialformen für welche Lernsituationen geeignet sind.¹⁷⁷ Die Lernzielbegründung enthält somit sowohl eine soziale als auch eine methodische Komponente, welche auch die Grundlage für das damit verwandte, affektive Feinziel bildet, nach dem die SuS lernen sollen, die von ihrem Partner eingebrachten Lösungskonzepte / -ideen / -vorschläge in ihrem Mehrwert als Innovationen, Lösungsfortschritte bzw. als wertvolle Beiträge für die Aufgabenbewältigung zu erkennen und anzuerkennen. Dieses Lernziel stellt eine Voraussetzung für das damit eng verbundene, kognitiv-methodische Feinziel dar, nach dem die SuS während der Anwendung des Dreischritts in Partnerarbeit lernen, die von ihrem Partner eingebrachten Lösungskonzepte / -ideen / -vorschläge für die Analyse, Planung und Implementierung im Bezug auf lösungsrelevante Bewertungskriterien (wie dem angemessenen Detailgrad, der Korrektheit der Wahl der Lösungselemente oder der Berücksichtigung von Abhängigkeiten) kritisch zu reflektieren und ihren eigenen Lösungen als mögliche Alternative unvoreingenommen gegenüberzustellen. Die methodische Charakteristik dieses Lernziels ergibt sich aus der Anforderung an die SuS, dass diese in der Lage sind, die Beiträge ihres Partners aus methodischer Sicht auch tatsächlich objektiv zu werden, d. h., dass sie in der Lage sind, selbstständig adäquate Bewertungskriterien zu wählen und diese auch methodisch adäquat anzuwenden. Daneben nimmt das Lernziel ebenso Bezug auf die Sozial- und die Selbstkompetenz der SuS. Der Bezug auf die Sozialkompetenz liegt dabei in der Fähigkeit, dass, wenn die SuS den Beitrag ihres Partners gegenüber ihren eigenen als den objektiv besseren betrachten, sie dies auch gegenüber ihren Partner veräußern bzw. in der Lage sind, ihren eigenen Beitrag zurückzustellen. Der Bezug auf die Selbstkompetenz ergibt sich aus dem umgekehrten Fall, dass, wenn die SuS ihren eigenen Beitrag gegenüber dem ihres Partners als objektiv besser betrachten, sie auch selbstkompetent genug sind, um ihre Position gegenüber ihrem Partner zur Disposition zu stellen und diese zu verteidigen.

Die zweite Anforderung soll von den Partnern innerhalb von fünf Minuten vollständig realisiert werden. Die SuS sollen den Dreischritt auf die zweite Anforderung also in der Hälfte der Zeit anwenden, welche für die erste Anforderung vorgesehen war. Die Gründe hierfür wurden im Rahmen dieses Berichts bereits mehrfach erläutert und sollen daher an dieser Stelle nur zusammenfassen enumeriert werden: Dabei ergibt sich die Verkürzung der Zeit einerseits aus dem Umstand, dass die Veränderung der Sozialform und des Handlungsmusters eine zügigeres Arbeiten ermöglichen, da die Aushandlungsprozesse, aus denen letztlich die Ergebnisse der drei Schritte generiert werden sollen, nur zwischen zwei Schülern und nicht im Rahmen fragend-entwickelnden Unterrichtsgesprächs im Plenum stattfinden. Der Wechsel der Sozialform und des Handlungsmuster verringert jedoch nicht nur die Komplexität des Aushandlungsprozesses, sie räumt den SuS ebenfalls die Möglichkeit ein, ihr Handeln zu koordinieren bzw. aufeinander abzustimmen und dabei

¹⁷⁷ Dies entspricht der vierten Stufe der Reflexionskompetenz in Meyers pragmatischem Kompetenzstufenmodell. Vgl. Meyer, S. 157.

bestimmte Tätigkeiten aufzuteilen, m. a. W.: Die Reduktion der Komplexität des Sozialgefüges soll den SuS dabei helfen, methodisch und sozial-kommunikativ kompetent und ökonomisch zu agieren. Als Beispiel hierfür wurde angeführt, dass der leistungsstärkere Partner einen Arbeitsschritt der zunächst gemeinsam ausgehandelten Planung formuliert, während der leistungsschwächere Partner diesen zeitgleich in der Implementierung realisiert. Somit wäre gewährleistet, dass die SuS zeitadäquat arbeiten und dabei trotzdem an ihren leistungsspezifischen Schwächen arbeiten, welche durch die Partnerarbeit trainiert werden sollen. Diese Schwächen wurden dahingehend formuliert, dass die leistungsschwächeren SuS in einem zu hohen Maß der bei den leistungsstärkeren vermissten Lehrerorientierung nachgehen bzw. dass die leistungsstärkeren SuS die bei den leistungsschwächeren SuS vermisste Fähigkeit, selbstständig bzw. selbstreguliert zu arbeiten, so weit treiben, dass sie durch das Einfügen bzw. Weglassen von Arbeitsschritten die Intention, die Zielstellung bzw. den Sinn der ihnen gestellten Aufgaben verfälschen. Neben dem das Arbeitstempo fördernden Wechsel des Handlungsmusters und der Sozialform wird die kurze Zeitvorgabe desweiteren dadurch legitimiert, dass die SuS mit der zweiten Anforderung einen im Grunde bekannten Gegenstand bearbeiten, da dieser nur in geringem Maße von dem gerade im Plenum realisierten ersten Anforderungsgegenstand abweicht. Da die Anforderungen jedoch nur nahezu, aber nicht vollständig identisch sind, müssen die SuS gerade den Unterschieden besondere Aufmerksamkeit zukommen lassen und dabei insbesondere reflektieren, welche Veränderungen die Unterschiede im Ergebnis des Dreischritts bewirken, damit die Veränderungen im Vergleich mit dem Ergebnis des ersten Dreischritts nicht fälschlicherweise als Fehler identifiziert werden. Die SuS sollen daher „lernen, je zwei nur geringfügig voneinander abweichende Codeabschnitte, welche als Folge der Anwendung der Dreischrittmethodik auf zwei nur geringfügig voneinander abweichende Anforderungsbeschreibungen entstanden sind, vergleichend gegenüberzustellen und dabei die funktionalen Unterschiede auf die Unterscheide im Code sowie die Unterscheide im Code auf die Unterschiede in der Anforderungsbeschreibung zurückzuführen.“ Sie sollen somit ihre Methodenkompetenz mit Blick auf die Präzision ihrer Arbeitsweise trainieren.

11:45-11:47 Sicherung des zweiten Programmabschnitts

Nach der Erarbeitung des zweiten Programmabschnitts soll diese gesichert werden, indem ein Paar das Ergebnis bzw. die Produkte des Dreischritts im Plenum in Form eines Schülervortrags präsentieren. Die Ergebnissicherung fungiert dabei als Aktivität des „Tests“, wobei (ähnlich wie beim V-Modell) nach jeder Aktivität geprüft wird, ob das generierte Produkt auch den Anforderungen, welche mit der Aktivität verbunden werden, genügt und dieses adäquat zum Problemgegenstand ist.¹⁷⁸ Die Betrachtung der Ergebnissicherung als Test muss durch den Lehrenden auch gegenüber den SuS kommuniziert werden, da der Lehrer anderenfalls Gefahr läuft, dass die SuS nur die Aktivitäten,¹⁷⁹ welche die Implementierung darstellen oder vorbereiten, auch als solche erkennen.

11:47-11:57 Erarbeitung des dritten Programmabschnitts

Verglichen mit der Erarbeitung der zweiten Anforderung ergibt sich für die Realisierung des dritten Programmabschnitts keine Veränderung der Sozialform oder der Handlungsmuster, dafür variiert jedoch der Gegenstand inhaltlich wesentlich stärker als zwischen der ersten und zweiten Anforderung. In diesem Zusammenhang wurde bereits erläutert, dass die im Rahmen der ersten Anforderung erarbeitete „Musterlösung“ nicht mehr in dem Maße als Vorlage genutzt werden kann, wie dies bei der Realisierung der zweiten Anforderung der Fall war, sondern nun wesentlich stärker abstrahiert werden bzw. auf einer

¹⁷⁸ Vgl. <http://waste.informatik.hu-berlin.de/Lehre/ws0910/pet/entwicklung-20091111.pdf> [19.10.2014], S. 8.

¹⁷⁹ Hier sind die Aktivitäten nach Floyd & Züllighoven gemeint (vgl. Floyd & Züllighoven 1997, S. 656 f.).

methodischen Ebene angewendet werden muss. Aus diesem Grund wird den SuS für die Erarbeitung dieses Programmabschnitts mehr Zeit eingeräumt.

11:57-12:00 Sicherung des dritten Programmabschnitts

Wie in „11:45-11:47 Sicherung des zweiten Programmabschnitts“.

12:00 Hausaufgabe

Als Hausaufgabe sollen die SuS den auf dem Arbeitsblatt „Projekttest“ hinterlegten Arbeitsauftrag einzeln bearbeiten.¹⁸⁰ Die im Arbeitsauftrag formulierte Schrittfolge verhält sich dabei ähnlich zu den Schritten, welche der Lehrer zum Stundenbeginn bei der Präsentation des Programms ausgeführt hat. Dabei werden zwei Intentionen verfolgt, welche sich in ihrem Kompetenzstufenniveau bezüglich der Dimension der Reflexionskompetenz in Meyer pragmatischen Kompetenzstufenmodell unterscheiden: Auf der zweiten Stufe der Reflexionskompetenz erkennen die SuS, dass die Rechte, welche ihr Programm besitzt, nicht ausreichen, um das Dateisystem an jeder Stelle zu verändern, sondern beispielsweise für das Erstellen einer Kopie im Laufwerkordner C: Administratorrechte benötigt werden. Auf der dritten Stufe der Reflexionskompetenz sollen die SuS „lernen, dass die Manipulation des Dateisystems mithilfe des von ihnen realisierten Programms an denselben Orten der Verzeichnisstruktur dieselben Rechte benötigt, wie sie bei der manuellen Manipulation des Dateisystems durch den Nutzer vom Betriebssystem gefordert werden würden.“ Damit gewährleistet ist, dass die SuS auch über die notwendigen Hilfsmittel verfügen, um die gleichen Resultate wie im Erwartungsbild zu erzielen und infolgedessen die richtigen Schlüsse zu ziehen, sollen die SuS nicht ihr eigenes, womöglich noch nicht einwandfrei funktionierendes oder unfertiges Programm verwenden, sondern eine vom Lehrer hochgeladene Musterlösung.

Weichenstellungen und Differenzierungsaufgaben

Die Unterrichtsstunde ist in ihrem Verlauf und ihrem geplanten Inhalt so konzipiert, dass der Lehrer sowohl auf den Fall, dass die SuS länger brauchen als erwartet, als auch auf den Fall, dass sie schneller fertig werden, adäquat reagieren kann.

Hierbei entspricht der Umstand, dass das Projekt in drei Teilprojekte modularisiert wurde, auf welche nacheinander der methodische Dreischritt angewendet, der Empfehlung Meyers, Weichenstellungen in die Planung einzubauen. So kann für den Fall, dass sich der Stundenablauf verzögert, beispielsweise die Anwendung des Dreischritts auf den letzten Projektteil als in sich abgeschlossenes Modul auf den Beginn der nächsten Stunde verschoben, ohne dass den SuS dabei der Sinnzusammenhang verloren geht.

Für den Fall, dass ein Paar einen Projektabschnitt bereits vor Ablauf der Zeit realisiert hat, besteht die Möglichkeit, dass dieses Paar, orientiert an ihrer Interessenlage, mit dem Lehrer zusätzliche Programmfunktionen aushandelt, welche das Paar realisieren kann, während die anderen SuS noch an der eigentlichen Anforderung arbeiten. Aus diesem Grund wurde sich bei der Erarbeitung der File-Klasse auch nicht nur auf die für die Realisierung der gegebenen Anforderungen notwendigen Methoden beschränkt, sondern eine Vielzahl von Methoden betrachtet, welche das Dateisystem in unterschiedlicher Art und Weise manipulieren. Den SuS wird somit die Möglichkeit gegeben, inhalts- und interessendifferenziert aus diesem durch die erarbeitete Dokumentation der File-Klasse terminierten Methoden- bzw. Funktionenpool zu wählen und ihre Wahl als Additum in ihrem Programm zu realisieren. Dieses Szenario ist vor allem im

¹⁸⁰ Vgl. Anhang 3.

Rahmen der Realisierung der dritten Anforderung denkbar, bei dem die Paare doppelt so viel Zeit für die Realisierung der Anforderung haben, wie dies bei der zweiten Anforderung der Fall war, weshalb es auch sehr wahrscheinlich ist, dass in diesem Unterrichtsabschnitt einige SuS zeitiger fertig sind.

2.6 Tabellarischer Stundenverlaufsplan

Zeit / Phase	Lehrerverhalten	Schülerverhalten	Sozialform / Handlungsmuster / Medien
10:45-10:50 Einstieg / Informieren über den Stundenablauf	<p>L. begrüßt die SuS, präsentiert die Anwendung „Dateikopierer“ als Stundenziel und führt diese aus.</p> <p>L. vermittelt den SuS, dass die in der vorherigen Stunde erarbeiteten Strukturen notwendig sind, um das Projekt realisieren.</p> <p>L. fragt nach der Notwendigkeit einer Schleife.</p>	<p>SuS hören aufmerksam zu.</p> <p>SuS erkennen, dass eine Schleife notwendig ist, um den eigentlichen Kopiervorgang beliebig oft zu wiederholen.</p>	<p>Plenumsunterricht/ Lehrervortrag/ Laptop u. Beamer</p> <p>Plenumsunterricht/ Unterrichtsgespräch/ Laptop u. Beamer</p>
10:50-11:10 Ergebnissicherung / Fortsetzung der Erarbeitung der letzten Stunde	<p>L. fordert die SuS dazu auf, sich in die Gruppen der letzten Stunde zu setzen.</p> <p>L. erklärt den SuS, dass sie in ihrer Gruppe nun die Experten für die in der vorangegangenen Stunde jeweils von ihnen erarbeitete Struktur sind und fordert sie auf, den jeweils anderen SuS der Gruppe ihre Struktur zu erläutern, wobei sie sich auf die Funktionsweise ihrer Struktur sowie deren Bedeutung für das Programm fokussieren sollen.</p> <p>L. macht transparent, dass für das Vorstellen einer</p>	<p>SuS finden sich in den Gruppen aus der letzten Stunde zusammen.</p> <p>SuS stellen sich in der Gruppe gegenseitig ihre erarbeiteten Strukturen in je 5 Minuten vor und fokussieren sich dabei auf die Funktionsweise ihrer Struktur sowie deren Bedeutung für das Programm.</p>	<p>Gruppenarbeit/ Schülergespräch (Puzzle-Methode)/ Aufzeichnungen und Arbeitsaufträge aus der letzten Stunde</p>

	Struktur ca. 5 Minuten vorgesehen sind und er die SuS nach 5 Minuten auffordert, sich mit der nächsten Struktur auseinanderzusetzen.		
11:10-11:13 Erarbeitung von benötigten C#-Sprachelementen	L. fordert die SuS auf, sich das AB „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“ durchzulesen und sich für sie unverständliche oder unklare Sprachelemente zu notieren.	SuS lesen das Arbeitsblatt durch und notieren sich bei Unklarheiten Fragen zum jeweiligen Sprachelement.	Einzelarbeit/ Textarbeit/ AB „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“
11:13-11:18 Sicherung der Sprachelemente	L. fragt, ob die Sprachelemente verstanden wurden und bittet die SuS, sich bei Unklarheiten zu melden. Falls die SuS Fragen stellen, bittet L. andere SuS, diese zu beantworten.	SuS stellen Fragen zu den Sprachelementen im Text bzw. beantworten die Fragen ihrer Mitschüler.	Plenumsunterricht/ offenes Unterrichtsgespräch/ AB „Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften“
11:18-11:25 Erarbeitung ausgewählter Methoden der File-Klasse	L. fordert die SuS auf, in Partnerarbeit den Arbeitsauftrag auf dem AB „Methoden der File-Klasse“ zu bearbeiten, wobei sie sich auf die Funktion der jeweiligen Methode sowie ihren Rückgabotyp konzentrieren sollen. Dabei sollen je zwei Partner einen Laptop benutzen und sich eine gemeinsame Lösung notieren.	SuS finden sich in Partnergruppen zusammen und bearbeiten den Arbeitsauftrag, wobei sie sich eine gemeinsame Lösung notieren.	Partnerarbeit/ Textarbeit/ AB „Methoden der File-Klasse“ sowie Laptop (MSDN)
11:25-11:30 Sicherung der Methodenbeschreibung	11:25-11:28 L. fordert die Partnergruppen auf, jeweils eine Methode zu beschreiben.	Die Partner beschreiben jeweils eine Methode.	Plenumsunterricht/ gelenktes Unterrichtsgespräch/ AB „Methoden der

	<p>11:28-11:30</p> <p>L. zeigt den SuS, wie die von ihnen verwendeten Microsoft-Definitionen im Visual Studio abgerufen werden können und erläutert kurz die Bestandteile der Beschreibung.</p>	<p>SuS vollziehen den Aufbau der Methodenbeschreibungen im Visual Studio nach.</p>	<p>File-Klasse“</p> <p>Plenumsunterricht/ Lehrervortrag/ Beamer und Laptop</p>
<p>11:30-11:40</p> <p>Erarbeitung des ersten Programmabschnitts</p>	<p>L. verschafft einen Überblick über die Anforderungen. Nach jeder Anforderung fordert L. einen S. auf, diese am Lehrer-PC zu realisieren.</p> <p>L. fordert SuS, die nicht präsentieren, auf, den Code in ihr Programm in Partnerarbeit einzufügen.</p>	<p>SuS rezipieren die Anforderungen, entwickeln aus ihrem Vorwissen heraus Ideen zu deren Umsetzung und präsentieren ihren Lösungsvorschlag über den Beamer.</p> <p>SuS implementieren die Musterlösung in ihrem Programm in Partnerarbeit.</p>	<p>Plenumsunterricht/ fragend - entwickelndes Unterrichtsgespräch/ Laptop und Beamer</p> <p>Partnerarbeit/ Programmieren/ Laptop</p>
<p>11:40-11:45</p> <p>Erarbeitung des zweiten Programmabschnitts</p>	<p>L. fordert SuS auf, den zweiten Anforderungsblock in Partnerarbeit selbständig zu implementieren.</p>	<p>SuS implementieren den zweiten Anforderungsblock selbständig in Partnerarbeit.</p>	<p>Partnerarbeit/ Schülergespräch/ Laptop und AB „Projekt Dateikopierer“</p>
<p>11:45-11:47</p> <p>Sicherung des zweiten Programmabschnitts</p>	<p>L. fordert ein Paar auf, ihr Ergebnis über den Beamer vorzustellen.</p>	<p>Partner stellen ihr Ergebnis vor.</p>	<p>Plenumsunterricht/ Schülervortrag/ Laptop und Beamer</p>
<p>11:47-11:57</p> <p>Erarbeitung des dritten Programmabschnitts</p>	<p>L. fordert SuS auf, den dritten Anforderungsblock in Partnerarbeit selbständig zu implementieren und gibt während der Bearbeitungszeit Lösungsimpulse bei auftretenden Problemen.</p>	<p>L. fordert SuS auf, den dritten Anforderungsblock in Partnerarbeit selbständig zu implementieren.</p>	<p>Partnerarbeit/ Schülergespräch/ Laptop und AB „Projekt Dateikopierer“</p>
<p>11:57-12:00</p>			

Sicherung des dritten Programmabschnitts	L. fordert ein Paar auf, ihr Ergebnis über den Beamer vorzustellen.	Partner stellen ihr Ergebnis vor.	Plenumsunterricht/ Schülervortrag/ Laptop und Beamer
12:00 Hausaufgabe	L. fordert die SuS auf, sich die Bearbeitung des Arbeitsblatts „Projekttest“ als Hausaufgabe zu notieren und zeigt ihnen den Server, auf dem sie die für die Hausaufgabe notwendige Datei finden.	SuS notieren sich die Hausaufgabe.	Plenumsunterricht/ Lehrervortrag/ Laptop und Beamer

2.7 Vorüberlegungen zur Unterrichtsauswertung

Bevor nun gleich die eben vorgestellte Unterrichtsstunde reflektiert wird, soll zuvor erörtert werden, was der Gegenstand dieser Reflexion sein und wie diese vonstattengehen soll. Für die Klärung dieser Punkte schlägt Meyer eine Auswertungs-Checkliste mit W-Fragen vor, welche im Rahmen dieses Abschnitts beantwortet werden sollen.¹⁸¹

Zunächst stellt Meyer die Frage nach der Zielstellung, welche mit der Unterrichtsauswertung konnotiert wird, wobei er u. a. die Verbesserung der eigenen Unterrichtsführung als ein solches Ziel benennt.¹⁸² Mit diesem Ziel soll auch die folgende Reflexion formuliert werden, wobei die spezifischen Zielstellungen bereits im Rahmen der Lehrbedingungen formuliert wurden. Hierbei erwähnte ich bereits, dass ich mich in Fragen des Sach- sowie des Handlungs- und Methodenwissens (und damit der Unterrichtsplanung) zwar als durchaus kompetent ansehe, es mir aufgrund meiner geringen Praxiserfahrung jedoch schwerfällt, meine sozial-kommunikative Kompetenz oder die Güte meiner Klassenführung zu evaluieren, wobei ich mir selbst folgende Entwicklungsaufgaben stellte:

Ich wollte lernen,

- meinen Unterricht / meine Planung ggf. an die von den SuS eingebrachten Impulse anzupassen.
- bestehende oder von mir eingeführte Rituale und Regeln während des Unterrichts konsequent durchzusetzen.
- die Komplexität meiner Sprache soweit zu reduzieren, dass sie für die jeweilige Klassenstufe angemessen ist und zum Verständnis des Unterrichtsgegenstands beiträgt.

Neben diesem Schwerpunkt muss in der Reflexion insbesondere darauf geachtet werden, inwiefern die in den verschiedenen Phasen des Unterrichts bezüglich der Dimensionen des didaktischen Sechsecks getroffenen Entscheidungen auch noch unter dem Eindruck der gehaltenen Unterrichtsstunde als adäquat gewählt gelten können und wo die Konzeption, womöglich aufgrund von falschen Annahmen über die Bedingungen oder einer missglückten Formulierung der aus den im didaktischen Sechseck getroffenen Entscheidungen generierten Aufgabenstellung und der Arbeitsaufträge, nicht zu den intendierten Ergebnissen führte.

Aus dieser Zielstellung heraus ergibt sich die Genese des Gegenstands der Unterrichtsauswertung aus dem Vergleich zwischen der Unterrichtsplanung sowie den damit intendierten Lernzielen und Inhalten einerseits und dem tatsächlichen Unterrichtsverlauf samt den tatsächlich erreichten Lernzielen und behandelten Inhalten andererseits sowie aus der Betrachtung solcher Ereignisse, welche Rückschlüsse darauf zulassen, inwiefern die mit der Verbesserung der sozial-kommunikativen Kompetenz des Lehrers einhergehenden Zielstellungen erreicht wurden. Die dabei verwendeten Bewertungskriterien bzw. Bezugssysteme ergeben sich dabei schwerpunktmäßig aus den im Stundenentwurf formulierten Ansprüchen und Hypothesen sowie aus den allgemein- sowie fachdidaktischen Kriterien.¹⁸³ Die Schwerpunktsetzung auf diese beiden Bezugssysteme ist insoweit sinnvoll, als dass sie ausschlaggebend für die im Rahmen der Aufgabenstellung sowie der didaktischen Strukturierung getroffenen Entscheidungen waren.

¹⁸¹ Vgl. Meyer 2012, S. 217.

¹⁸² Vgl. ebd., S. 217 f.

¹⁸³ Vgl. ebd., S. 219.

Die Reflexion der Stunde wird durch mich vorgenommen, wobei ich jedoch nicht allein meine subjektive Sicht auf den Unterrichtsverlauf rezipiere, sondern ebenso das Feedback meines Fachlehrers, meines Seminarleiters sowie die Arbeitserzeugnisse der SuS aus dem Dreischritt bei der Stundenauswertung berücksichtige.

Für die Unterrichtsauswertung empfiehlt Meyer, auf sogenannte Praktikertheorien einzugehen. Hierbei handelt es sich um selbst formulierte Theorien, welche auf der Grundlage eigener Erfahrungen und der gezielten Aneignung von Theoriewissen formulierte, kritisch reflektierte Annahmen über Grundlagen und Effekte unterrichtlichen Handelns sind.¹⁸⁴ Im Rahmen der didaktischen Reduktion wurden einige solcher Praktikertheorien formuliert, welche an dieser Stelle jedoch nicht wiederholt, sondern zu zwei globaleren Theorien zusammengefasst werden sollen, deren Stichhaltigkeit unter realen Unterrichtsbedingungen im Rahmen dieser Auswertung evaluiert werden soll:

Das Projekt wurde mit der Intention konzipiert, dass der zunächst vom Lehrer vorgegebene Dreischritt als Handlungsmuster von den SuS reflektiert und inkorporiert wird. Dabei sollte durch die sukzessive Wegnahme von Orientierungshilfen, welche in den Anforderungsrealisierungen durch die Variation der Sozialformen, Handlungsmuster und Inhalte erreicht wurde, dazu führen, dass sich die SuS mit jedem Durchlauf weniger an diesen Hilfen orientieren, sondern ihre Ergebnisse im zunehmenden Maße das Produkt einer reflektierten, kompetenten Anwendung dieses Handlungsmusters darstellen.

Die zweite große Theorie war, dass durch den Austausch zwischen den leistungsschwächeren und den leistungsstärkeren SuS in der Partnerarbeit deren jeweilige Schwächen durch eine Angleichung des methodischen Arbeitens ausgeglichen werden, wobei der leistungsschwächere Partner den leistungsstärkeren nicht durch sein Arbeitstempo aufhält, da der leistungsstärkere Partner aus der Reflexion der Arbeitsweise des leistungsschwächeren Partner gewinnbringende Erkenntnisse für seine eigene Arbeitsweise erwirbt.

¹⁸⁴ Vgl. ebd., S. 225.

3. Reflexion der Unterrichtsstunde

3.1 Besonderheiten beim realen Stundenverlauf gegenüber dem geplanten

Verlauf

Die Vorbereitung der Stunde verlief leider nicht wie von mir geplant. Ein großes Problem dabei war, dass ich vor der Stunde einen außerschulischen Termin hatte, wobei ich aufgrund der Verzögerungen im ÖPNV erst eine viertel Stunde vor dem Unterrichtsbeginn in der Schule eintraf. Diese Zeit nutzte ich zunächst, um die variierbaren Raumbedingungen zu verbessern, indem ich das Licht anschaltete sowie für eine bessere Belüftung sorgte. Anschließend transportierte ich sechs Laptops, auf denen Windows 7 und das Visual Studio bereits installiert waren, in den Klassenraum, schloss sie an den Strom an und startete diese mit Windows 7. Währenddessen betraten die SuS den Klassenraum, wobei sie durch die geringe Anzahl der Laptops verwirrt waren und nachfragten, wie sie sich setzen sollen. Ich erklärte ihnen, dass wir die Laptops im ersten Teil der Stunde nicht benötigen und wir diese später für die Partnerarbeiten verwenden würden. Aus diesem Umstand heraus ergab sich die Gelegenheit, die sozial-kommunikative Kompetenz der SuS zu beobachten, da die von mir informierten SuS ihr bisher gesammeltes Wissen über den Unterrichtsverlauf an die später eintreffenden SuS weitergaben. Ebenso waren viele SuS bemüht, ihre Arbeitsumgebung für den Unterricht selbständig vorzubereiten, wobei sie sich, nachdem der Fachlehrer das Administratorpasswort eingegeben hatte und Windows hochgefahren war, kurz bei mir rückversicherten, ob sie das Visual Studio starten sollen und dann die Entwicklungsumgebung für die Arbeit mit C# und XAML selbständig konfigurierten. Gegenüber diesen SuS musste ich diesen Vorbereitungsschritt also gar nicht als Arbeitsauftrag formulieren, da die Mehrheit der SuS das Starten des Visual Studio bereits als Ritual zur Stundenvorbereitung aufgenommen hatte. Den Rest der Vorbereitungszeit verbrachte ich damit, das Visual Studio sowie die Musterlösung des Dateikopierers auf meinem Rechner lauffähig zu machen sowie den an meinem Laptop angeschlossenen Beamer so zu positionieren, dass das Bild auf einer glatten Fläche angezeigt und nicht verzerrt wird, wobei ich ebenfalls darauf achtete, dass die Deckenleuchten nicht in das Bild hineinragen. Leider habe ich es vor dem Beginn der Stunde nicht mehr geschafft, die Arbeitsblätter auszudrucken und auszuteilen. Stattdessen habe ich die Materialien auf meinen Server geladen und die SuS im Verlauf der Stunde angewiesen, die Arbeitsblätter von diesem herunterzuladen. Während der Vorbereitung habe ich die Tafel als Unterrichtsmedium des Klassenraums vollständig ignoriert, da ich so sehr auf die Vorbereitung der Medien fokussiert war, deren Einsatz ich in meinem Stundenverlaufsplan vorgesehen hatte, dass ich erst während der Stunde bemerkte, dass die Tafel nicht abgewischt war, was mein Seminarleiter dahingehend als mögliche Gefahr identifizierte, dass das auf der Tafel stehende die SuS verwirren könnte, da sie versuchen könnten, dies mit dem Stundenthema in einen Zusammenhang zu bringen.

Da ich auf das Ausdrucken der Arbeitsblätter verzichtete, konnte ich pünktlich mit der Stunde beginnen. Ich begrüßte die SuS mit „Hallo“ und wiederholte dieses Wort mehrmals, da ich zunächst nicht die Aufmerksamkeit aller SuS hatte. Dies wurde von meinem Fachlehrer als unfreundlich empfunden, wobei er mir nahelegte, den Unterricht mit einer persönlicheren bzw. freundlicheren Anrede zu beginnen und dann zu warten, bis mir die SuS durch ihre Körperhaltung und ihren Blickkontakt die Bereitschaft zur Teilnahme am Unterricht signalisieren. Nachdem ich die Aufmerksamkeit der SuS bekommen hatte, verlief der Einstieg wie geplant. Nach meinem Lehrervortrag zu den Inhalten der Unterrichtsstunde sowie der Präsentation des Dateikopierers als Stundenziel konnten mir die leistungsschwächeren SuS erläutern, in welchem funktionalen Kontext die Struktur der Schleife für die Anwendung von Bedeutung ist. Mit Bezug auf meine sozial-kommunikative Kompetenz wies mich mein Mentor darauf hin, dass ich mich während meines

Lehrervortrags häufig zum Beamerbild gewendet, an diesem etwas gezeigt und dann weiterhin in Richtung des Bilds gesprochen habe, da ich mich nicht wieder dem Plenum zuwendete. Deshalb hat er mir nach der Stunde die Vortragsweise des „Touch and Turn“ nahegelegt, bei der der Vortragende zunächst eine demonstrierende Handlung an der Tafel oder am Beamerbild vornimmt, während dieser Handlung innehält und erst dann weiterspricht, wenn die Handlung finalisiert ist. Sprechen und Zeigen sind in dieser Vortragsweise also sequenziell aufeinanderfolgende, dabei aber voneinander abgegrenzte Handlungen. Diese Technik erachte ich für mich als sehr sinnvoll, da ich durch das Schweigen während der zeigenden Handlung an der Wand des Klassenraums wieder daran erinnert werde, mich wieder dem Plenum zuzuwenden und nicht meiner schlechten Angewohnheit zu folgen, mich so stark auf den durch das Tafel- oder Beamerbild visualisierten Gegenstand zu konzentrieren, dass ich dabei mein Publikum vergesse.

Die Phase des Abschlusses der Erarbeitung der Kontrollstrukturen verlief didaktisch wie geplant, allerdings gab es zu Beginn der Phase das organisatorische Problem, dass nicht alle SuS meinem Arbeitsauftrag aus der letzten Stunde gefolgt sind und sich notiert haben, in welcher Dreiergruppe sie sind. Nachdem ich den Arbeitsauftrag wie in der von mir in der didaktischen Strukturierung geplanten Form erteilt habe, mussten einige SuS im Plenum nachfragen, wer mit ihnen in einer Gruppe war, weshalb es eine halbe Minute dauerte, bis sich die Gruppen wieder zusammengefunden haben. Die Umsetzung des eigentlichen Arbeitsauftrags des gegenseitigen Erklärens der Strukturen erfolgte innerhalb der Gruppen jedoch sehr diszipliniert, engagiert und (bezüglich des vom Lehrer vorgegebenen Anforderungen) zielgerichtet. Hiervon überzeugte ich mich, indem ich, während sich die SuS gegenseitig die Strukturen erklärten, zu den einzelnen Gruppen ging und mir die Erklärungen anhörte. Diese Handlungsweise wurde von meinem Seminarleiter als zu passiv empfunden. Er empfahl mir, in das Schülergespräch hineinzugehen, indem ich nach bestimmten Spezifika der Strukturen frage bzw. die Aufmerksamkeit der Gesprächsteilnehmer auf diese leite: „Könnt ihr mir erklären [...]?“. Nach der Hälfte der vorgesehenen Zeit war eine Gruppe bereits mit dem gegenseitigen Erklären der Strukturen fertig, weshalb ich den Ratschlag meines Seminarleiters, welchen er mir für die vorangegangene Stunde gegeben hatte, befolgte und die fertigen SuS anwies, sich auf die restlichen Gruppen aufzuteilen und sich an den dort stattfindenden Schülergesprächen über die Strukturen zu beteiligen. Nach ca. 15 Minuten waren hatten alle SuS die Erarbeitung der Strukturen beendet. Den Rest der für diese Phase eingeplanten Zeit nutzte ich, um mit den SuS die als Übungsaufgaben konzipierten Stolpersteine der Strukturen im Plenum zu sichern, wobei ich auf meinem Rechner die Arbeitsblätter öffnete und die SuS aufforderte, die von mir angezeigte Aufgabe zu lösen, wobei ich darauf achtete, vor allem die leistungsschwächeren SuS dranzunehmen. Hierbei wurde wieder (diesmal jedoch von meinem Seminarleiter) angemerkt, dass ich mich lange Zeit über der Beamerprojektion zugewandt und dabei gar nicht bemerkt habe, dass ein Teil der SuS nicht nach vorne guckt, sondern auf ihren Laptop. Der Grund hierfür könnte zum Teil darin liegen, dass die SuS die Projektion nicht gut erkennen konnten. So meldeten mir sowohl mein Mentor als auch mein Seminarleiter zurück, dass die projizierte Schrift zu klein war, als dass sie in den hinteren Reihen lesbar gewesen wäre. Inhaltlich gestaltete sich die improvisierte, als katechetischer Frageunterricht im Plenum realisierte Sicherung als ertragreich, da ich hierüber sicherstellen konnte, dass auch die leistungsschwächeren SuS die Strukturen verstanden haben.

Nach der eingeschobenen Sicherung begann die Erarbeitung von benötigten C#-Strukturen wie geplant um 11:10. Aus der Umsetzung dieser Phase wurde erkennbar, dass sie bezüglich des für diese formulierten Arbeitsauftrags didaktisch nicht adäquat konzipiert wurde. So zeigte sich die Aufforderung an die SuS, sich die gegebene Dokumentation durchzulesen und Unklarheiten über dort beschriebenen Elemente sowie deren Eigenschaft und Funktionen als Fragen zu notieren, als unklarer Arbeitsauftrag. Dies wurde besonders gut dadurch ersichtlich, dass die SuS, nachdem sie die Dokumentation gelesen hatten, keine Notizen

gemacht hatten und meine Frage, ob es Unklarheiten gibt, durchgehend negierten. Als ich jedoch begann, die Strukturen zu sichern, indem ich die in der Planung vorgesehene alternative Sicherung bemühte und die in der Dokumentation vorgegebenen Sprachelemente mithilfe meines Laptops und des Beamers im Visual Studio in einfache Kontexte setzte und die SuS nach der Funktion der jeweiligen Elemente in diesem Kontext fragte, musste ich feststellen, dass die Funktionsweisen der Elemente nicht gänzlich von allen SuS inkorporiert wurden, weshalb ich die Dokumentation in einem fragend-entwickelnden Unterrichtsgespräch noch einmal Schritt für Schritt mit den SuS durchgegangen bin. Mein Mentor bemerkte daher folgerichtig nach der Stunde, dass es besser gewesen wäre, gleich dieses fragend-entwickelnde Unterrichtsgespräch durchzuführen und die SuS nicht vorher ziellos lesen zu lassen. Wenn andererseits ein Leseauftrag gegeben wird, dann sollte er auch unter einer konkreten Zielstellung stattfinden, beispielsweise der Herstellung eines Produkts oder der Lösung von Testfragen. Hierbei schlug mit mein Seminarleiter vor, mit den SuS nach dem Lesen der Dokumentation einen Multiple-Choice-Test durchzuführen und über den Vergleich des Tests im Plenum zu sichern, dass die SuS die dargestellten Elemente auch verstanden haben. Ebenso wäre es denkbar, die Dokumentation, wie nachfolgend für die File-Klasse realisiert, von den SuS erarbeiten zu lassen. Diese Erarbeitung gestaltete sich unhandlicher als geplant, da die SuS aufgrund des Fehlens der Arbeitsblätter gezwungen waren, sich dieses zunächst von meinem Server herunterzuladen und die dort dargestellten Methoden anschließend auf ein Blatt Papier zu übertragen. Für die vorhergehende Aufgabe zur Erarbeitung der notwendigen Sprachelemente folgte hieraus, dass diese nicht als Einzelarbeit durchgeführt werden konnte, sondern, da den zwölf SuS lediglich sechs Rechner bereitgestellt wurden, mindestens zwei SuS mit demselben Medium arbeiten mussten. Nachdem ich feststellen musste, dass einige SuS nach der Erarbeitung der notwendigen Sprachelemente diese nicht sicher beherrschten, wandelte ich meine Konzeption für die File-Klasse dahingehend ab, als dass ich für die Erarbeitung dieser die Schüler noch nicht in Paare einteilte, sondern diese wieder in den Dreiergruppen arbeiten ließ, welche innerhalb der Stunde bereits die Erarbeitung der Kontrollstrukturen finalisiert hatten. Der Grund dafür lag in der während der Erarbeitung der Kontrollstrukturen gemachten Beobachtung, dass die SuS in dieser Gruppenkonstellation sehr diszipliniert und zielgerichtet arbeiten. Ich musste jedoch die Erfahrung machen, dass sich dieser Effekt bei der Erarbeitung der File-Klassendokumentation nicht einstellte: Als Hauptgrund hierfür sehe ich die Knappheit der zur Verfügung stehenden Medien an: Während die SuS für die Erarbeitung der Kontrollstrukturen jeweils auf ihre eigenen Arbeitsblätter und ihre Notizen aus der letzten Stunde zurückgreifen konnte, mussten sich die SuS bei der Erarbeitung der File-Klasse ein oder zwei Laptops teilen. Dies führte dazu, dass nicht alle SuS aktiv an der Erarbeitung teilnahmen, sondern oftmals nur diejenigen SuS die Formulierungen für die Dokumentation erstellten, welche an den Laptops arbeiteten, während die anderen SuS diese lediglich rezipierten und in ihre Notizen übernahmen. Da ich dieses Vorgehen noch während der Phase beobachten konnte, versuchte ich der Passivität bestimmter SuS entgegenzusteuern, indem ich mir in der darauffolgenden Sicherung der Dokumentation im Plenum die Methoden von denjenigen SuS präsentieren ließ, welche nicht am Computer gearbeitet haben.

Nachdem ich also bemerkte, dass das Abweichen von der geplanten Sozialform, welche ich als Reflexion-in-der-Handlung aufgrund der Erfahrungen aus der Erarbeitung der Kontrollstrukturen vornahm,¹⁸⁵ nicht den gewünschten Erfolg mit sich brachte, kehrte ich bei der Realisierung des Projekts wieder zur ursprünglich geplanten Partnerarbeit zurück, wobei ich die Zusammensetzung der Paare leistungsdifferenziert festlegte. Die Erarbeitung des Dreischritts am Gegenstand des ersten Programmabschnitts im Plenum verlief etwas träger als erwartet. Dies war vor allem dem Umstand geschuldet, dass ich versucht habe, möglichst oft die leistungsschwächeren SuS zu aktivieren und in das fragend-entwickelnde Unterrichtsgespräch einzubinden

¹⁸⁵ Vgl. Meyer 2012, S. 224.

und nicht nur die leistungsstärkeren SuS dranzunehmen, welche sich während des gesamten Gesprächs durchgängig meldeten. Diese Intention machte es jedoch oftmals notwendig, dass ich, wenn ich merkte, dass die leistungsschwächeren SuS meine Frage nicht verstanden haben oder ihnen keine entsprechende Antwort einfällt, weitere, hierauf bezogene, kleinschrittige Fragen formulierte, um für die leistungsschwächeren SuS eine Hinführung zu einer adäquaten Antwort zu gewährleisten. Hierdurch dauerte das Unterrichtsgespräch jedoch länger als ursprünglich geplant, sodass der letzte Schritt der Implementierung erst nach 15 Minuten abgeschlossen war und wir bis zum Stundenklingeln nur noch weitere 15 Minuten Zeit hatten. Ich beschloss daher, mit dem Erreichen der nächsten Weichenstellung die Stunde abubrechen, sodass ich mit den SuS in dieser Stunde nur noch die Erarbeitung und Sicherung des zweiten Anforderungsabschnitts vornahm und die Erarbeitungszeit für die Partner von fünf auf zehn Minuten erhöhte. Dieser Schritt erwies sich als notwendig, da auch nach diesen zehn Minuten erst zwei Paare mit der Anwendung des Dreischritts auf den Gegenstand des zweiten Anforderungsabschnitts fertig waren. Trotzdem beendete ich nach den zehn Minuten die Erarbeitung dieses Programmteils, da ich in den letzten fünf Minuten noch einen Vergleich im Plenum vornehmen wollte. Hierzu bat ich eine der beiden Gruppen, welche bereits fertig waren, den SuS ihre Vorgehensweise bei der Bewältigung der Aufgabe zu erläutern und ihr Produkt am Lehrerlaptop nachzustellen. Von der Erläuterung der Vorgehensweise erhoffte ich mir, dass die fertigen SuS berichten, wie sie die methodischen Spielräume, welche ihnen der Dreischritt lässt, nutzen, um in der ihnen zur Verfügung stehenden Zeit effektiv zu arbeiten.¹⁸⁶ Die schrittweise Reproduktion sollte den anderen SuS ermöglichen, Fehlerquellen in ihrer eigenen Arbeit zu identifizieren bzw. diejenigen Schritte zu ergänzen, mit denen sie noch nicht fertig waren. Die Frage nach dem methodischen Vorgehen in den verschiedenen Aktivitäten wurde von dem präsentierenden Paar beantwortet, wobei ich gezielt Zwischenfragen stellen musste, um die Spezifika in ihrem Vorgehen hervorzuheben. Die schrittweise Reproduktion der Erzeugnisse der einzelnen Aktivitäten stellte sich dahingehend als ungünstig heraus, als dass das erneute formulieren viel Zeit in Anspruch nahm und wir zwei Minuten überziehen mussten. Hierbei wäre es besser gewesen, zumindest die am Laptop produzierten Ergebnisse nicht noch einmal neu generieren zu lassen, sondern diese vom Schüler auf den Lehrerlaptop zu übertragen, sodass die Partner ihr Vorgehen schrittweise an der bereits von ihnen produzierten Lösung erklären können. Die eigentliche Intention des schrittweisen Erklärens, d.h. die Fehlerkorrektur und der Ergänzung der Lösungen der anderen SuS, blieb hiervon jedoch unberührt und erfolgte daher wie erwartet.

Wie bereits erwähnt, war eine Behandlung der dritten Anforderung leider nicht mehr möglich. Da mein Mentor in der nächsten Stunde gleich mit PHP einsteigen wollte, wurde die Fertigstellung des Produkts auch in der darauffolgenden Stunde nicht realisiert. Die SuS mussten also, wenn sie ein verwendbares Produkt mit der Funktionalität des Dateikopierers haben wollten, die dritte Anforderung zu Hause selbst realisieren. Ich gab den SuS den Hinweis, dass sie sich zu Hause (als fakultative Übungsaufgabe) an dem letzten Anforderungsabschnitt versuchen sollten. Dies tat ich in dem Wissen, dass die Realisierung des dritten Anforderungsabschnitts Gegenstand der Klausur sein würde, wobei ich die Aufgabe selbst formuliert habe. Aus Zeitgründen musste die Aufgabe jedoch insoweit reduziert werden, als dass nur die Implementierung als explizit zu formulierender Gegenstand erwartet wurde, wohingegen die Analyse und Planung nicht explizit gefragt wurde und nur implizit von den SuS mitgedacht wurde.¹⁸⁷

¹⁸⁶ Als Beispiel hierfür wurde die Möglichkeit angeführt, dass sich ein Paar in Partnerarbeit über die Planung des Projektabschnitts verständigen und die Partner dann in parallel verlaufender Einzelarbeit die Planung als solche formalisieren und diese gleichzeitig implementieren.

¹⁸⁷ Vgl. Anhang 7.

3.2. Fazit zur Stunde – Was habe ich gelernt?

Wollte ich den Erfolg der Stunde an den Zielen messen, welche ich mit dieser verband, so würde das Ergebnis durchwachsen ausfallen. Für den Hauptgegenstand der Stunde, d. h. die Anwendung des Dreischritts auf den Dateikopierer, muss festgehalten werden, dass dieses nicht vollständig bearbeitet werden konnte. Das inhaltliche Ziel der vollständigen Erstellung einer Software konnte in der Stunde leider nicht abgeschlossen werden. Dafür konnte ich beobachten, dass sich meine Praktikertheorien in der Anwendung des Dreischritts auf die zweite Anforderung bestätigten: So nahmen die SuS den Dreischritt nach der gemeinsamen Erarbeitung der ersten Anforderung im Plenum als Handlungsmuster an und verwendeten dies für die zweite Anforderung. Einschränkend muss jedoch erwähnt werden, dass der Gegenstand der zweiten Anforderung von den SuS noch kein hohes Maß an Abstraktion bezüglich der Anwendung dieses Handlungsmusters verlangte und somit während der Stunde nicht gesichert werden konnte, ob der Problemumfang, welchen die SuS als durch das Handlungsmuster des Dreischritts lösbar ansehen, auch dem intendierten Umfang entspricht. Diese Frage wird sich wohl erst langfristig über solche Aufgaben klären lassen, bei denen die SuS ihre Problemlösekompetenz unter Beweis stellen müssen, indem sie für ein vorgegebenes Problem die richtigen Handlungsmuster wählen und dabei auch das Handlungsmuster des Dreischritts situativ adäquat verwenden. Der Erfolg der zweiten Praktikertheorie war noch deutlicher sichtbar. Die leistungsheterogenen Partner eines Paares glichen ihre Arbeitsweise wie erwartet an, wobei sie die Ergebnisse in einem adäquaten Detailgrad formulierten. Der hierfür notwendige Gesprächs- und Aushandlungsprozess führten hierbei leider dazu, dass einige SuS mit der Realisierung der zweiten Anforderung auch nach den zehn Minuten nicht fertig waren. Rückblickend nehme ich diese Verzögerung jedoch gerne in Kauf, da der Umstand, dass sich die SuS mehr Zeit nehmen konnten, sich positiv auf das Erreichen der mit dem Dreischritt verknüpften, affektiven Ziele ausgewirkt hat. Das Aufgehen der mit der Konzeption dieser Phase intendierten, affektiven Ziele ist deswegen besonders wichtig, weil das Erreichen der affektiven Ziele in den Phasen der Erarbeitung der für die Realisierung des Dateikopierers notwendigen Sprachelemente durch die Fehlplanung bezüglich der damit verbundenen Aufgabenstellung und der File-Klasse durch eine unzureichend durchdachte Abkehr von der geplanten Sozialform stark beeinträchtigt wurde. Aus der Fehlplanung habe ich gelernt, dass es notwendig ist, einen Arbeitsauftrag mit einem konkreten Produkt bzw. einem konkreten, sinnstiftenden Ziel zu verbinden. Ein solcher Arbeitsauftrag war mit der Aufforderung, den Text zu lesen und unklare Elemente zu notieren, jedoch nicht gegeben. Der unzureichend durchdachte Wechsel der Sozialform rief mir schmerzlich eine Formulierung Meyers ins Gedächtnis, nach der „Entscheidungen in der einen Ecke des Sechsecks [...] immer – und nicht nur hin und wieder – Rückwirkungen auf die fünf anderen Ecken“ haben.¹⁸⁸ Bei dem Wechsel der Sozialform hatte ich mich vor allem die Zeitstruktur im Blick, ich wollte erreichen, dass die SuS effektiv arbeiten. Dabei habe ich jedoch nicht die Auswirkungen beachtet, welche dieser Wechsel auf die anderen Ecken hat. So wurde bereits erwähnt, dass es durch die Arbeit in Dreiergruppen nicht zu dem erwünschten Austausch zwischen den SuS gekommen ist, welcher durch die Partnerarbeit intendiert wurde, sondern einige SuS passiv die Arbeit ihrer Mitschülerinnen und Mitschüler rezipierten. Ich habe somit gelernt, dass auch scheinbar kleinere Veränderungen im methodischen Verlauf den tatsächlichen Unterrichtsverlauf massiv beeinträchtigen können und solche Änderungen als Handlung-in-der-Reflexion nur dann legitim sind, wenn die Konsequenzen für alle Ecken des Sechsecks bedacht wurden. Dies ist vor allem dann der Fall, wenn die Änderungen bereits bestehende Phasen betreffen, deren Planung mit bestimmten Lernzielen konnotiert wird. Weniger schwerwiegend waren solche Änderungen im Unterrichtsablauf, aus denen kurze, neue, zwischen zwei bestehenden Phasen eingeschobene Unterrichtsabschnitte entstanden sind, da solche

¹⁸⁸ Meyer 2012, S. 180.

Einschübe vor allem (gesehen auf den gesamten Stundenverlauf) in erster Linie die Ecken der Inhalts- und der Zeitstruktur betreffen. So hat sich beispielsweise mein eingeschobener Lehrervortrag über die C#-Elemente auf der Vorderseite der Dokumentation, welcher lediglich als optionale Phase in der Planung kurz umrissen wurde, sich nicht negativ auf das weitere Unterrichtsgeschehen oder die damit verbundenen Intentionen ausgewirkt. Der Umstand, dass die vorhergehende Phase des Fragestellens, da die SuS keine Fragen stellten, zugunsten des Lehrervortrags abgekürzt wurde, führte dazu, dass der Einschub der Phase nicht mit einer Verzögerung des weiteren Stundenverlaufs einherging. Stattdessen konnte ich durch solche Einschübe sichern, dass die kognitiven Feinlernziele auch tatsächlich erreicht werden. So konnte ich den Dreischritt in dem Wissen einführen, dass die SuS sowohl die Kontrollstrukturen als auch die für die Realisierung des Dateikopierers notwendigen Sprachelemente (inklusive der File-Klassenmethoden) verstanden haben. Ebenso erwiesen sich die Ziele des methodischen Umgangs mit der MSDN-Dokumentation sowie dem Dreischritt als von den SuS inkorporiert. Dabei bestätigte mir auch mein Mentor, dass ich sowohl in dieser Stunde als auch in den vorangegangenen Stunde dafür sorgte, dass die SuS das methodische Rüstzeug sowie fundierte Kenntnisse über die Sprachelemente an Hand bekommen haben, welche für die Realisierung des Softwaregegenstands notwendig sind.

Die Durchführung der Stunde habe ich mir selbst durch die unzureichende Vorbereitung etwas schwer gemacht. Hieraus habe ich gelernt, an solchen Tagen, an denen ich unterrichte, auch möglichst ganztägig an der Schule zu bleiben und nicht zwischendurch außerschulische Termine wahrzunehmen. Ebenso werde ich zukünftig Vorbereitungsaufgaben wie das Ausdrucken der Arbeitsblätter, welche bereits einige Zeit vor dem Unterrichtsbeginn erledigt werden können, auch einige Zeit vorher in Angriff nehmen, sodass ich nicht kurz vor Stundenbeginn alle Punkte der Vorbereitung abdecken muss. Weiterhin habe ich gesehen, dass es sinnvoll ist, eine Checkliste für die Unterrichtsvorbereitung mache, da einem auch solche einfachen Sachen wie das Abwischen der Tafel entfallen können und ich dann auch nicht vergesse zu prüfen, ob die Schriftgröße der Beamerfolien angemessen ist.

Bezüglich der Verbesserung meiner Unterrichtsführung habe ich das Ziel formuliert, die Komplexität meiner Sprache soweit zu reduzieren, dass sie für die jeweilige Klassenstufe angemessen ist und zum Verständnis des Unterrichtsgegenstands beiträgt. Dies ist mir zwar in solchen Situationen gelungen, in denen ich den SuS einen inhaltlichen Sachverhalt erklärt habe, andererseits habe ich durch die implizite Annahme bestimmter Aufgabenschritte und das ausschließlich mündliche Formulieren von Arbeitsaufträgen künstlich die Komplexität bei der Aufgabenformulierung gesteigert. Ich habe daher gelernt, dass es sinnvoll ist, auch mündlich bereits gegebene Arbeitsaufträge nochmals schriftlich an der Tafel (bzw. an einem für alle SuS gut sichtbaren Ort) festzuhalten und alle Arbeitsaufträge vorher auszuformulieren. Dabei ist es sinnvoll, die einzelnen, mit der Aufgabe verbundenen, vorzunehmenden Handlungen den SuS als Checkliste zur Verfügung zu stellen. Dies unterstützt die SuS einerseits in ihrem methodischen Arbeiten und erleichtert andererseits die Auswertung anhand von zuvor festgelegten Kriterien. Als weitere organisatorische Maßnahme werde ich in Zukunft Transparenzpapiere für meine Unterrichtsstunden erstellen, in denen ich die SuS nicht nur über die Inhaltsstruktur der Stunde informiere, sondern alle Ecken des didaktischen Sechsecks sichtbar mache. Die SuS wissen also dann, wann welche Phase beginnt, in welchen Sozialformen und Handlungsmustern sie in diesen Phasen arbeiten und welche Leistungen von ihnen erwartet werden. Das Transparenzpapier für eine Stunde sollte dabei zum Ende der vorhergehenden Stunde ausgegeben werden, damit die SuS auf das, was auf sie zukommt, mental vorbereitet sind, und sich nicht durch vom Lehrer vorgenommene Modifikationen der Lehr- und Lernumgebung beirren lassen. Als weitere Lernziele habe ich für mich formuliert, meinen Unterricht / meine Planung ggf. an die von den SuS eingebrachten Impulse anzupassen sowie bestehende oder von mir eingeführte Rituale und Regeln während des

Unterrichts konsequent durchzusetzen. Die Anpassung der Planung ist mir teilweise gut gelungen. So wurde bereits erwähnt, dass es mir durch den Einschub von Sicherungsphasen gelang, sicherzustellen, dass alle SuS die zur Realisierung der Software notwendigen Erkenntnisse erworben haben. Andererseits bin ich zu wenig auf kreative und alternative Lösungsvorschläge der SuS eingegangen, da ich fürchtete, dass wir sonst in zeitliche Probleme kommen. Daher bestätigte ich den SuS nur die Korrektheit ihrer Ideen, ohne diesen in den Unterricht zu integrieren. Das Durchführen von Ritualen und Regeln habe ich leider nicht konsequent durchgesetzt, da ich oftmals selbst während des Unterrichtsgeschehens nicht an diese Gedacht habe. So habe ich beispielsweise vergessen, die SuS zu bitten, die Laptops während des Lehrervortrags zuzuklappen. Bei anderen Handlungen wie dem selbständigen Starten des Visual Studio durch die SuS habe ich hingegen gar nicht gemerkt, dass dieser Vorgang bei den SuS bereits rituell verankert ist. Ich habe daher gelernt, dass es sinnvoll ist, auf den Spickzettel bzw. auf den Verlaufsplan zu schreiben, wann ich welche Regeln und Riten durchführen wollte und dass ich solche Vorgänge, welche sich allmählich zu Riten entwickeln, auch explizit als Riten einführe, damit sie von allen SuS durchgeführt werden. Als wichtigen Ritus für mich selbst nehme ich mit, beim Zeigen nicht zu sprechen, wobei ich dies zukünftig innerhalb der Vortragstechnik des „Touch and Turn“ umsetzen werde.

4. Hospitationsprotokolle

Die nachfolgenden Hospitationsprotokolle haben zwei Unterrichtsstunden von Thomas S. zum Gegenstand, welche er aufeinanderfolgend in derselben Klasse gehalten hat. Die Protokolle bestehen dabei aus je zwei Teilen: Im ersten Teil stelle ich meine Beobachtungen zu der jeweiligen Phase des Unterrichts von Herrn S. dar. Dabei werden positive Beobachtungen grün, negative Beobachtungen rot und beschreibende Beobachtungen gar nicht markiert. Im zweiten Teil ziehe ich ein Fazit über die gesamte Stunde und schildere, was ich aus dieser gelernt habe.

4.1 Hospitationsprotokoll I

Referendar: Thomas S.

Schule: Einstein Gymnasium Neuenhagen

Klassenstufe: 8

Fachlehrerin: Frau Kretschmer

Thema der Reihe: E-Mail nur für mich

Thema der Stunde: Transferprotokolle beim Versenden und Empfangen von E-Mails

Datum: 15.09.2014

Zeit: 7:50-8:35

Beobachtungsschwerpunkt: Frage- und Aufgabenstellung

Zeit: 7:50-7:51

Phase: Begrüßung / Einleitung

Sozialform: Frontalunterricht

Handlungsmuster: Lehrervortrag

Beobachtungen / tatsächliche Unterrichtsinhalte:

- L. begrüßt die SuS und informiert sie über den weiteren Stundenverlauf.
- L. hält den Stundenverlauf nicht schriftlich fest.
- L. nutzt die vorhandene Zeit effektiv und ist bereits nach einer Minute mit der Phase fertig, obwohl für diese fünf Minuten eingeplant waren.

Zeit: 7:51-8:02

Phase: Erarbeitung des Sendens einer E-Mail mit SMTP

Sozialform: Frontalunterricht

Handlungsmuster: Lehrervortrag

Medien: Powerpoint-Präsentation (Beamer / Lehrercomputer), AB1_E-Mail.docx, AB2_E-Mail.docx

Beobachtungen / tatsächliche Unterrichtsinhalte:

- Unterrichtsgeschehen:
 - L. teilt die Arbeitsblätter AB1_E-Mail.docx und AB2_E-Mail.docx aus und fordert die SuS auf, dass auf AB1_E-Mail.docx abgebildete Schaubild mithilfe der aus dem Lehrervortrag gewonnenen Informationen zu vervollständigen.
 - SuS hören dem Lehrervortrag aufmerksam zu und vervollständigen das Schaubild.
- Es ist fraglich, ob für diese Phase mit dem Frontalunterricht und dem Lehrervortrag die richtige Sozialform / das richtige Handlungsmuster gewählt wurde. Immerhin steht den SuS mit dem Arbeitsblatt ein Darstellungstext zur Verfügung, welcher die gleichen Informationen enthält wie der Lehrervortrag, sodass sich die SuS das Schaubild selbst hätten erarbeiten können.
- Die auf dem Arbeitsblatt formulierten Arbeitsaufträge
 1. Aufgabe: „Lies dir den folgenden Informationstext aufmerksam durch.“

2. Aufgabe: „Trage in die Skizze die entsprechenden Fachbegriffe ein.“

stehen im Widerspruch zum vom Lehrer formulierten, mündlichen Arbeitsauftrag. Diese Aufgaben wurden vermutlich für die Parallelklasse konzipiert. Sie hätten jedoch für diese Klasse vom Arbeitsblatt genommen werden müssen, damit keine Verwirrung bei den SuS entsteht. In der Unterrichtsstunde hätte der Lehrer die SuS auch auffordern können, die Aufgabe durchzustreichen.

- Der von der Lehrperson erstellte Darstellungstext über das „Senden“ von E-Mails wurde auf die Klasse zugeschnitten und in einer einfach verständlichen, prägnanten Sprache verfasst.
- Das Arbeitsblatt weist kleinere orthografische Mängel auf.
- Da u. a. das Schaubild in der Powerpoint-Präsentation sukzessive vervollständigt wird, können die SuS sichern, dass sie auch die richtigen Begriffe eingetragen haben.
- Aufgrund der Lichtverhältnisse ist die Powerpoint-Präsentation leider nicht gut sichtbar.
- L. spricht in einem angemessenen Tempo und mit ruhiger Stimme.

Zeit: 8:02-8:15

Phase: Erarbeitung von POP3 und IMAP4

Sozialform: Partnerarbeit

Handlungsmuster: Schülergespräch (Lernaufgabe)

Medien: AB2_E-Mail.docx

Beobachtungen / tatsächliche Unterrichtsinhalte:

- Unterrichtsgeschehen:
 - L. fordert die SuS auf, in den kommenden zehn Minuten den Text „Empfangen“ auf dem Arbeitsblatt AB2_E-Mail.docx in Partnerarbeit zu erschließen, indem sie die auf dem Arbeitsblatt stehenden Arbeitsaufträge ausführen:
 1. Aufgabe: „Lies dir den Informationstext zum POP3-Protokoll oder zum IMAP4-Protokoll aufmerksam durch. (5 min)“
 2. Aufgabe: „Erkläre deinem Gruppenpartner das Protokoll, über das du dich informiert hast. (5 min)“
 - SuS stellen selbständig Paare zusammen, lesen jeweils einen der beiden Texte und erklären ihrem Partner den von ihnen gelesenen Text.
- Sowohl auf dem Arbeitsblatt als auch im Lehrervortrag ist von Gruppenarbeit die Rede, obwohl die Sozialform der Partnerarbeit angewendet wird.
- Hinter jedem Arbeitsauftrag steht die vom Lehrer jeweils zugeordnete Zeit.
- Die erste und zweite Aufgabe verlangen von den SuS jeweils eine Handlung, wobei die Handlungen nur als Kette sinnstiftend sind. Hierbei besteht die Gefahr, dass die SuS nach dem Lesen der ersten nicht gleich die zweite Aufgabe lesen, weshalb sie den Text ohne eine konkrete Zielstellung lesen.
- Das Arbeitsblatt weist kleinere orthografische Mängel auf.
- L. geht auf die SuS zu und beantwortet ihre Fragen mit sehr ruhiger, verständnisvoller Stimme.
- Der von der Lehrperson erstellte Darstellungstext über das Empfangen wurde auf die Klasse zugeschnitten und in einer einfach verständlichen, prägnanten Sprache verfasst.
- Der Umfang des Textes steht in einem guten Verhältnis zur dafür vorgesehenen Zeit
- Während der Partnerarbeit steigt der Lärmpegel im Plenum zunehmend. Da Herr S. seine ganze Aufmerksamkeit auf die Beantwortung der Verständnisfragen der SuS richtet, unterbindet er dies nicht, weshalb die Fachlehrerin die Klasse diszipliniert.
- Bereits fertige SuS wissen nicht, was sie machen sollen und stören den Unterricht.
Binnendifferenzierende, durch Fundamentum und Additum gekennzeichnete Aufgaben fehlen.

Zeit: 8:15-8:23

Phase: Sicherung von POP3 und IMAP4

Sozialform: Plenumsunterricht

Handlungsmuster: fragend-entwickelndes Unterrichtsgespräch

Medien: Powerpoint-Präsentation (Beamer / Lehrercomputer), AB2_E-Mail.docx

Beobachtungen / tatsächliche Unterrichtsinhalte:

- Unterrichtsgeschehen:
 - L. zeigt den SuS über den Beamer ein Schaubild zum Empfangen von E-Mails und stellt den SuS Fragen, welche sich sowohl auf den Text als auch auf das Schaubild beziehen, um die Funktionsweise von POP3 und IMAP4 herzuleiten.
 - SuS beantworten die Fragen des Lehrers und verstehen die Funktionsweise von POP3 und IMAP4.
- Mit dem Text und dem Schaubild legt Herr S. den SuS ein mediendifferenziertes Angebot vor, welches den gleichen Gegenstand verständlich / zugänglich machen soll.
- L. geht auf Schülerfragen individuell ein, indem er einen Bezug zu ihrer Lebenswelt herstellt.
- Die von den SuS gegebenen Antworten werden nicht schriftlich festgehalten (kein Tafelprodukt bzw. kein Produkt als Sicherung).
- L. wird beim Vortragen zunehmend leiser.
- Wenn die SuS bestimmte Inhalte nicht gleich erfassen oder Fragen nicht beantworten können, versucht L., die SuS über suggestive Fragen zur richtigen Lösung hinzuleiten.
- L. befragt SuS stichprobenartig, um sicherzustellen, dass der erarbeitete Inhalt verstanden wurde.

Zeit: 8:23-8:30

Phase: Zweite Sicherung von POP3 und IMAP4

Sozialform: Einzelarbeit

Handlungsmuster: Textarbeit

Medien: AB2_E-Mail.docx

Beobachtungen / tatsächliche Unterrichtsinhalte:

- Unterrichtsgeschehen:
 - L. fordert SuS auf, die aus dem Text entnommenen Vor- und Nachteile von POP3 und IMAP4 in die dafür vorgesehene Tabelle auf der Rückseite von AB2_E-Mail.docx einzutragen.
 - SuS tragen die aus dem Text entnommenen Vor- und Nachteile von POP3 und IMAP4 in die dafür vorgesehene Tabelle ein.
- Die Rückseite hat die Überschrift „Hausaufgabe“, obwohl der Arbeitsauftrag in der Stunde bearbeitet werden soll.
- Das Arbeitsblatt weist kleinere orthografische Mängel auf.
- L. geht diesmal nicht durch die Klasse, sondern bleibt am Pult stehen. Er hat somit keinen allgemeinen Überblick darüber, wie weit die SuS sind und ob sie mit der Aufgabe zurechtkommen.
- Schnelle SuS langweilen sich, da auch für diese Phase keine Aufgabendifferenzierung stattfand.

Zeit: 8:30-8:34

Phase: Vergleich der zweiten Sicherung von POP3 und IMAP4

Sozialform: Frontalunterricht

Handlungsmuster: fragend-entwickelndes Unterrichtsgespräch

Medien: Powerpoint-Präsentation (Beamer / Lehrercomputer), AB2_E-Mail.docx

Beobachtungen / tatsächliche Unterrichtsinhalte:

- L. nimmt vorwiegend solche SuS dran, welche sich in der Stunde bisher wenig am Unterrichtsgeschehen beteiligt haben, lobt sie und ergänzt / verbessert ggf. deren Beiträge.
- Falls die SuS einen Vor- oder Nachteil nicht erkennen, versucht L., die SuS über Suggestivfragen zu diesen hinzuführen.
- Die über den Beamer projizierte Musterlösung des Lehrers ist von hinten schlecht sichtbar.

Zeit: 8:34-8:35

Phase: Zusammenfassung des Gegenstands der Unterrichtsstunde

Sozialform: Frontalunterricht

Handlungsmuster: fragend-entwickelndes Unterrichtsgespräch

Beobachtungen / tatsächliche Unterrichtsinhalte:

- L. lässt die SuS die heute gesammelten Erkenntnisse zusammentragen und achtet dabei darauf, möglichst oft verschiedene SuS dranzunehmen, insbesondere solche, welche sich in der Stunde selten am Unterrichtsgeschehen beteiligt haben.

Fazit – Was habe ich gelernt?

Assoziiert man den Erfolg der Unterrichtsstunde mit den mit dieser konnotierten Lernzielen, so stellt sich die hospitierete Unterrichtsstunde von Herrn S. als sehr erfolgreich dar. So zeigte die abschließende Sicherung zum Stundenende, dass die SuS die Lernziele, welche im Kern das Inkorporieren und Verstehen der Empfangs- und Sendeprotokolle zum Gegenstand hatten, erreicht haben. Der von dem Herrn S. eingeführte Ritus der zusammenfassenden Wiederholung des in der Stunde Gelernten stellt sich für mich als sehr sinnvolle Form des fragend-entwickelnden Unterrichtsgesprächs zur Sicherung im Plenum dar, da dem Lehrer (vorausgesetzt das Unterrichtsgespräch wird von ihm in einer zum Unterrichtsgegenstand adäquaten Art und Weise konzipiert und inszeniert) hierdurch noch vor dem Ende der Stunde, d. h. unmittelbar, ein Feedback darüber gegeben wird, ob der Unterrichtsgegenstand von den SuS inkorporiert bzw. die damit verbundenen, direkt beobachtbaren Lernziele erreicht wurden. Aus diesem Grund möchte ich diese Form der Sicherung zukünftig in meinen eigenen Unterricht integrieren. Der Erfolg der Unterrichtsstunde ergibt sich aus mehreren Faktoren: Besonders beeindruckt haben mich die sozial-kommunikative Kompetenz und das hohe Maß an Empathie, welche Herr S. während der Durchführung der Unterrichtsstunde an den Tag legt. So wurde in der Darstellung der Unterrichtsstunde darauf verwiesen, dass Herr S. es schafft, die Komplexität seiner Sprache auf ein für SuS der achten Klasse angemessenes Niveau zu reduzieren, wobei er Sachverhalte durch die exemplarische Bezugnahme auf die Lebenswelt der SuS veranschaulicht. Bei auftretenden Problemen während der Erarbeitungsphasen geht Herr S. eigenaktiv auf die SuS zu und beantwortet deren Fragen in seiner ihn charakterisierenden, ruhigen Art, wobei die ruhige und verständnisvolle Stimme, welcher sich Herr S. während des gesamten Unterrichts bedient, den SuS dahingehend ein gewisses Maß an Sicherheit vermittelt, als dass den SuS suggeriert wird, dass es nicht schlimm ist, Fragen zu stellen oder etwas nicht zu verstehen. Durch die Hospitation bei Herr S. kann man sehr gut lernen, wie man seine eigene Präsenz als Lehrkraft nutzt, um ein lernförderliches Klima zu generieren. Hierzu gehört auch, dass sich Herr S. Zeit für die SuS nimmt, sodass er beispielsweise (wie im Verlauf bemerkt) in solchen Fällen, in denen die SuS eine Frage nicht beantworten können oder etwas nicht verstehen, in fragend-entwickelnden Unterrichtsgesprächen versucht, die SuS zum Unterrichtsgegenstand bzw. zur Antwort auf seine ursprüngliche Frage hinzuführen. Dies kostet zwar Zeit, jedoch schafft es Herr S. hierdurch sicherzustellen, dass alle SuS die mit der jeweiligen Phase konnotierten Lernziele erreichen. Hierbei versucht der Praktikant, für die einzelnen Phasen Zeitpuffer einzuplanen, um unerwartete Beiträge oder Fragen der SuS in den Unterricht integrieren zu können. Ein Beispiel hierfür ist die Phase der Einleitung bzw. Begrüßung, für die Herr S. im Stundenverlaufsplan fünf Minuten vorgesehen hat, er jedoch bereits wusste, dass sein Lehrervortrag nicht diese Zeitspanne einnehmen wird. Diese Vorgehensweise werde ich mir für meine eigene Stundenplanung aneignen, zumal ich die Neigung habe, oftmals sehr viel Stoff in eine Stunde packen zu wollen. Herr S. hat für seine Stunde ein geeignetes Maß gefunden. Hierbei fand ich es sehr gut, dass er nach wenigen Unterrichts- und Hospitationsstunden die Leistungsfähigkeit seiner SuS soweit einschätzen konnte, dass er in der Lage war, die Komplexität und die Länge seiner Texte an die in diesem Bereich gesammelten Erfahrungen adäquat anzupassen. Der Mehrwert für den Unterricht bestand darin,

dass alle SuS die Texte in der vorgegebenen Zeit unter den entsprechenden Arbeitsaufträgen bearbeiten konnten. Dennoch stellten die Medien, welche Herr S. zur Erarbeitung und Sicherung der „Sache“ der jeweiligen Phase verwendet hat, den größten Schwachpunkt der Stunde dar. Bei den Arbeitsblättern stand die Intention im Mittelpunkt, möglichst effizient Lernmaterialien und Lernaufgaben für die SuS zu produzieren. So passte Herr S. zwar die Texte auf die achte Klasse an, die ausgegebenen Arbeitsblätter differenzierten jedoch weder zwischen den beiden achten Klassen, welche Herr S. unterrichtete, noch legten sie den SuS ein binnendifferenziertes Angebot dar. Da die Konzeption der Arbeitsblätter am für beide Klassen erarbeiteten Stundenverlaufsplan des Lehrers orientiert war und nicht am Lernfortschritt der Klasse oder am für die Klasse zugeschnittenen methodischen Gang, kam es zu den bereits beschriebenen Phänomenen, dass manche schriftliche Arbeitsaufträge den in der Stunde gegebenen, mündlichen Aufträgen widersprachen und andere als Hausaufgabe gekennzeichnet wurden, obwohl sie Inhalt der Stunde waren. Dabei folgte die Struktur der Arbeitsaufträge eher der Logik des Lehrers, als dass diese schülerorientiert formuliert wurden. So wurden die einzelnen Handlungen einer Handlungsfolge in separate Arbeitsaufträge gepackt, mit der bereits beschriebenen Gefahr, dass sie nicht, wie vom Lehrer intendiert, zusammenhängend gelesen werden. Statt die Arbeitsaufträge separat zu formulieren, halte ich es für geeigneter, den SuS eine Checkliste für die Handlungsfolge zur Verfügung zu stellen. Neben den Arbeitsaufträgen wurden auch die Medien zur Ergebnissicherung eher aus der Lehrerperspektive konzipiert. Hierbei verwendete Herr S. Powerpoint-Präsentationen mit vorgefertigten Antworten, welche eine Integration von Beiträgen bzw. Formulierungen in das Sicherungsprodukt nicht zuließen. Die Erarbeitungsphasen haben mir bewusst gemacht, wie wichtig das Einführen und konsequente Durchsetzen und Anwenden von Ritualen ist und wie groß die Notwendigkeit ist, den SuS während solcher Phasen ein binnendifferenziertes Angebot zu unterbreiten. So wussten die SuS, welche die Arbeitsaufträge bereits vorzeitig vollständig bearbeitet hatten, nicht, was sie anschließend tun sollen, weshalb sie den Unterricht störten. Durch ein Additum an Aufgaben hätten diese SuS weiter sinnstiftend beschäftigt werden können. Ein weiteres Problem war, dass für die SuS nicht immer transparent war, wie viel Zeit ihnen für die Bearbeitung eines Arbeitsauftrags zur Verfügung steht. Trotz der verpassten Möglichkeiten, die Stunde in der Planung durch binnendifferenzierende Maßnahmen zu individualisieren, finde ich den methodischen Gang der Stunde bezüglich der intendierten Lernziele insofern angemessen, als dass durch das inkrementelle Durchlaufen der Phasen der Erarbeitung und der Ergebnissicherung sichergestellt werden konnte, dass die mit dem Inhalt der jeweiligen Phase verbundenen Lernziele auch tatsächlich erreicht wurden, wobei die Zusammenfassung am Ende der Stunde das Gelernte in einem globaleren Zusammenhang sichert. Die Phasenkonzeption der Einleitung, der inkrementellen Erarbeitung und Sicherung sowie der abschließenden Sicherung nehme ich für mich ebenfalls mit, wobei ich beobachtet habe, dass die Phaseneinteilung zumindest die Klasse von Herrn S. bei der Aneignung der „Unterrichtssache“ unterstützt hat.

4.2 Hospitationsprotokoll II

Referendar: Thomas S.

Schule: Einstein Gymnasium Neuenhagen

Klassenstufe: 8

Fachlehrerin: Frau Kretschmer

Thema der Reihe: E-Mail nur für mich

Thema der Stunde: Versenden von E-Mails

Datum: 23.09.2014

Zeit: 12:55-13:40

Beobachtungsschwerpunkt: Frage- und Aufgabenstellung

Zeit: Vor dem Stundenbeginn

Phase: Feststellung der Anwesenheit

Sozialform: Frontalunterricht

Handlungsmuster: Lehrervortrag

Beobachtungen / tatsächliche Unterrichtsinhalte:

- L. erklärt, dass er im Folgenden eine Anwesenheitsliste herumgibt und fordert die SuS auf, auf dieser ihre Anwesenheit zu bestätigen. Er klärt somit das Organisatorische vor dem Beginn der eigentlichen Stunde (mehr Lernzeit in der Stunde).

Zeit: 12:55-13:00

Phase: Begrüßung / Einleitung / Sicherung der Ergebnisse der letzten Stunde / Hausaufgabenkontrolle

Sozialform: Frontalunterricht

Handlungsmuster: Lehrervortrag und katechetischer Frageunterricht

Medien: Musterlösung der Hausaufgabe als Powerpoint-Präsentation (Beamer / Lehrercomputer)

Beobachtungen / tatsächliche Unterrichtsinhalte:

- Unterrichtsgeschehen:
 - L. begrüßt die SuS.
 - L. fordert eine Schülerin auf, den Lerngegenstand der vorhergehenden Stunde zusammenzufassen.
 - Nach der Zusammenfassung durch die Schülerin informiert L. die SuS über den weiteren Stundenverlauf.
 - L. sichert mit den SuS die Hausaufgaben in Form eines katechetischen Frageunterrichts.
- L. begrüßt die SuS und informiert sie über den weiteren Stundenverlauf.
- L. hält den Stundenverlauf nicht schriftlich fest.
- L. hat seine Begrüßung und die Vorstellung des Stundenthemas zuvor durchgesprochen und kann daher sehr genau vorhersagen, wie viel Zeit er dafür benötigt, und hat die Begrüßung auch innerhalb des von ihm im Stundenverlauf festgesetzten Zeitrahmens absolviert.
- L. versucht, die Sicherung des Gelernten am Anfang und Ende einer Stunde als Ritus zu etablieren.
- L. sichert die Hausaufgabe im Plenum und achtet dabei darauf, möglichst alle SuS dranzunehmen.
- Die vom Lehrer präsentierte Musterlösung ist vorgefertigt, weshalb die Formulierungen der SuS keinen Eingang in diese finden. Für die SuS ist somit nicht direkt erkennbar, inwieweit die von ihnen formulierte Antwort eine akzeptable Lösung darstellt. Hierfür müssen sie eine Transferleistung erbringen, in der sie die Formulierung des Lehrers auf ihre eigene abbilden und inhaltlich vergleichen.
- L. stellt bei der Kontrolle der Hausaufgabe Bezüge zum Stundenthema in der Form her, dass für die SuS nicht mehr eindeutig erkennbar ist, ob das gerade Gesagte auf die Hausaufgabe oder auf die heutige Stunde zielt.

- L. sammelt mehrere Beiträge der SuS zur Hausaufgabenlösung und sagt, dass zwei wichtige Fakten genannt wurden, ohne diese zu wiederholen, sodass die SuS nicht wissen, welche Fakten der Lehrer genau meint, da diese auch nicht während des Unterrichts in die Musterlösung integriert werden.

Zeit: 13:00-13:10

Phase: Arbeiten mit dem alten Stoff / Wiederholung

Sozialform: Einzel- und Partnerarbeit

Handlungsmuster: Geben und Nehmen

Beobachtungen / tatsächliche Unterrichtsinhalte:

- Unterrichtsgeschehen:
 - L. fordert SuS auf, ein A4-Blatt zu nehmen und dies in 3x3 Felder zu falten.
 - L. gibt den SuS vor, was sie in die einzelnen Kästchen schreiben sollen. Dabei nennt er nicht die konkreten Begriffe, sondern umschreibt diese. Die Aufgabe der SuS ist es dabei, den Begriff für die Umschreibung des Lehrers zu finden und diese in das entsprechende Feld einzutragen.
 - Nachdem der Lehrer die Umschreibungen für alle neun Felder verlesen hat und die SuS die entsprechenden Begriffe eingetragen haben, fordert L. die SuS auf, sich zu Paaren zusammenzufinden und die Ergebnisse ihrer neun Felder zu vergleichen bzw. diese zu ergänzen.
- Wie bereits in der vorangegangenen Stunde spricht der Lehrer auch in dieser wieder von einer Gruppenarbeit, obwohl es sich bei der Sozialform um Partnerarbeit handelt.
- Obwohl die SuS angewiesen wurden, zu zweit zu arbeiten, arbeitet ein Schüler allein, wohingegen drei SuS in einer Gruppe arbeiten. Der Lehrer unterbindet dies nicht.
- L. macht nicht transparent, wie viel Zeit die SuS für die Partnerarbeit haben.
- Als Alternative für den Vergleich in Partnerarbeit könnte ebenfalls das Handlungsmuster des Tandembogens angewendet werden.

Zeit: 13:10-13:17

Phase: Problemorientierter Einstieg in das Versenden von E-Mails

Sozialform: Frontalunterricht

Handlungsmuster: Lehrervortrag

Beobachtungen / tatsächliche Unterrichtsinhalte:

- Unterrichtsgeschehen:
 - L. versucht, die SuS für eventuell auftretende Probleme beim Versenden einer E-Mail zu sensibilisieren.
 - Hierzu zeigt er zunächst einen Film, welcher dieses Thema behandelt und die SuS zur Auseinandersetzung mit diesem motivieren soll.
 - Anschließend hält L. einen kurzen Lehrervortrag über die Probleme beim E-Mail-Verkehr, wobei er u. a. Bezüge zum Film herstellt.
- Die Beleuchtung des Unterrichtsraums ist für die Filmvorführung nicht angemessen. Die SuS müssen daher selbständig eine geeignete Umgebung schaffen, indem sie die Vorhänge zuziehen.
- Der Einstieg mit dem Film ist für die SuS motivierend. Der Film dient jedoch nicht als Selbstzweck, sondern wird in den nachfolgenden Lehrervortrag integriert.
- Die Phase beginnt später als vom Lehrer geplant und dauert länger als erwartet.

Zeit: ----

Phase: Erarbeitungsphase zu den Merkmalen von E-Mails

Sozialform: Frontalunterricht

Handlungsmuster: fragend-entwickelndes Unterrichtsgespräch

Beobachtungen / tatsächliche Unterrichtsinhalte:

- Diese Phase wurde zwar geplant, aus Zeitgründen hat L. auf eine Durchführung dieser in der Stunde

jedoch verzichtet.

Zeit: 13:17-13:37

Phase: Problemorientierter Einstieg in das Versenden von E-Mails

Sozialform: Frontalunterricht / Einzelarbeit

Handlungsmuster: Lehrervortrag / Textarbeit / Lernaufgabe / Produktherstellung

Beobachtungen / tatsächliche Unterrichtsinhalte:

- L. kündigt eine Übungsphase an, wobei er erklärt, dass das hierfür notwendige Arbeitsblatt noch nicht freigeschaltet ist. Die SuS fangen trotzdem schon an, ihre Rechner zu verwenden und in ihrem Verzeichnis nach dem Arbeitsblatt zu suchen. Es wäre sinnvoll gewesen, die Bildschirme während des Lehrervortrags auszuschalten bzw. ausschalten zu lassen.
- Während sich die SuS das Arbeitsblatt durchlesen, nutzt L. die Zeit, um die von ihm erstellten Test-E-Mail-Adressen an die SuS zu verteilen.
- Das Arbeitsblatt kann von den SuS digital bearbeitet werden.
- L. denkt daran, den SuS organisatorische Hinweise bezüglich der Bearbeitung des Arbeitsblatts auf dem Rechner zu geben.
- L. geht herum und rücker sich, dass die von den SuS vorgenommenen Handlungen auch zum gewünschten Ergebnis führen.
- L. beantwortet die Fragen der SuS mit ruhiger und verständnisvoller Stimme.
- L. übersieht eine sich meldende Schülerin für mehrere Minuten.
- Wie bereits in der letzten Stunde wurden auch in dieser Phase die unterschiedlichen Arbeitstempi im Arbeitsauftrag nicht berücksichtigt. Dabei hätte es sich gerade beim Schreiben von Mails angeboten, zwischen Fundamentum und Additum zu differenzieren.
- L. fordert SuS, die bereits fertig sind, auf, sich abzumelden. Einige SuS gehen dieser Forderung jedoch nicht nach, sondern surfen im Internet.
- L. schenkt solchen SuS, die den Unterricht stören, keine Aufmerksamkeit bzw. gibt ihnen keinen Raum zum Stören, sondern unterbindet dies, indem er sie vom Lehrerpult aus abmeldet.

Zeit: 13:37-13:40

Phase: Zusammenfassung des Gegenstands der Unterrichtsstunde

Sozialform: Frontalunterricht

Handlungsmuster: fragend-entwickelndes Unterrichtsgespräch

Beobachtungen / tatsächliche Unterrichtsinhalte:

- L. lässt die heute gesammelten Erkenntnisse von einer Schülerin zusammentragen, welche die Stunde über unauffällig war und sich nicht gemeldet hat.
- Die Fragen des Lehrers bezogen sich lediglich auf die älteren Inhalte, eine Sicherung der gerade durchgeführten Aufgabe fand nicht statt.

Fazit – Was habe ich gelernt?

Fragt man wieder nach dem Mehrwert der Stunde für den Wissens- und Kenntnisstand bzw. für die Fähigkeiten und Fertigkeiten der SuS, so kann für diese Stunde leider nicht ein so positives Fazit gezogen werden, wie für die vorhergehende. Als hauptsächliche Ursachen hierfür sehe ich einerseits an, dass aufgrund der Verzögerungen im Unterrichtsablauf eine wichtige Erarbeitungsphase nicht durchgeführt wurde. Desweiteren hat der Praktikant in dieser Unterrichtsstunde das vorher angewendete Konzeptionsmuster des Einstiegs, der inkrementellen Erarbeitung und Sicherung sowie der abschließenden Sicherung verlassen. Stattdessen wurde zunächst mehr als ein Viertel der Unterrichtsstunde zur Wiederholung und Sicherung des alten Stoffs verwendet. Hierauf folgte ein lehrerzentrierter Einstieg in das Thema, wobei in der Evaluierung der einzelnen Phasen bereits dargestellt wurde, dass der Film auf die SuS motivierend wirkte und in Verbindung mit dem Lehrervortrag den SuS viele Dimensionen bzw. Problemfelder bezüglich des Schreibens und Versendens von Mails aufzeigte, ob das vorgetragene Lehrerwissen jedoch auch bei den SuS angekommen ist, wurde nicht gesichert. Hierbei wäre es gut gewesen, wenn sich der Praktikant an seiner vorhergehenden Stunde orientiert hätte, in der er den SuS ein nicht ausgefülltes Schaubild zur Verfügung stellte, welches diese während seines Lehrervortrags vervollständigen sollten, wobei die Sicherungen im Verlauf des Vortrags durch das Einblenden der richtigen Eintragungen über den Beamer erfolgte. Stattdessen setzte der Praktikant die Stunde mit einer Übungsphase fort, wobei der Arbeitsauftrag lediglich in dem Schreiben einer E-Mail mit und ohne Anhang bestand. Die einzige Phase, in der die SuS mit neuem Wissen konfrontiert wurden, bestand also in dem Lehrervortrag über die Probleme beim Schreiben und Versenden von Mails, wobei die Aufgabe des Mail-Versendens keine hinreichende Sicherung darstellte. Folgerichtig war der Gegenstand der Auswertung bzw. der Zusammenfassung des in der Stunde Behandelten eher an dem bereits in der letzten Stunde erworbenen Wissen orientiert. Der Grund für die Verzögerung und den hieraus resultierenden Ausfall der Erarbeitungsphase liegt in der auf den Lehrenden fokussierten Vorbereitung, welche bereits in der vorhergehenden Unterrichtsstunde beobachtbar war. So plant Herr S. die Rolle des Lehrers, d. h. seine Aktionen im Unterrichtsgeschehen, sehr detailliert, wohingegen Phasen oder Phasenabschnitte mit Schülerbeteiligung nicht in gleichem Maße vorbereitet werden, was gut am Beispiel der Einleitung veranschaulicht werden kann: Herr S. hat seine Begrüßung und die Erläuterung zum Stundenverlauf vorher durchgesprochen und kann daher sehr genau voraussagen, wie viel Zeit er für diesen einführenden Lehrervortrag benötigt. Der hierauf folgende Vergleich der Hausaufgaben wird dagegen von improvisierten Einschüben unterbrochen, in denen Herr S. auf das heutige Stundenthema verweist, wobei für die SuS nicht immer ersichtlich wird, auf welchen Gegenstand sich der Lehrende gerade bezieht. Weiterhin wurde in der Darstellung des Stundenverlaufs bereits geschildert, dass der Lehrer nicht eigenaktiv versucht hat, die Lernumgebung für den Medieneinsatz (des Beamers) vorzubereiten und dies bei der Filmvorführung sogar erst auf Initiative der SuS erfolgte. Ebenso hat der Praktikant (wie bereits in der vorhergehenden Stunde) bei der Konzeption der Medien zur Übung und Sicherung wieder einen eher für den Lehrer ökonomischen Ansatz verfolgt. So war die Musterlösung der Hausaufgaben bereits vorgefertigt, weshalb wieder keine Integration der Beiträge und Formulierungen der SuS möglich war. Ebenso wenig stellte der Praktikant den SuS während der Übungsphase ein binnendifferenziertes Angebot in Aussicht, obwohl gerade der Inhalt des Schreibens von E-Mails viele Möglichkeiten zur Erweiterung dieses Gegenstands bietet, weshalb es wieder zu Unruhen in der Klasse kam. Hierauf hat Herr S. gut reagiert, indem er durch die Disziplinierung der störenden SuS die arbeitenden SuS nicht etwa gestört hat, sondern über das Lehrerpult die Störer lautlos mit dem Abmelden ihrer Rechner sanktionierte. Hierbei hat Herr S. die Ratschläge seiner Mentorin umgesetzt und sich die technischen Möglichkeiten zur Interaktion mit den Rechnern der SuS angeeignet. Generell konnte ich beobachten, dass

sich die Klassenführung von Herrn S. im Vergleich zur letzten Woche deutlich verbessert hat: So habe ich in meinem ersten Protokoll konstatiert, dass Herr S. die Beantwortung von Schülerfragen mit hoher fachlicher wie sozial-kommunikativer Kompetenz verfolgt, die starke Konzentration auf die Rolle des helfenden Beraters führte jedoch dazu, dass er in den Phasen, in denen er zu den SuS ging und ihre Fragen beantwortete, das Geschehen im Rest der Klasse ausblendete und bei Unterrichtsstörungen nicht eingriff. In dieser Stunde hat er es geschafft, zur selben Zeit beide Rollen wahrzunehmen, d. h. sowohl als helfender Akteur aufzutreten als auch als Klassenführer, welcher die Einhaltung von Regeln und Riten überwacht. Die damit verbundene Herausforderung, gleichzeitig auf Individualbedürfnisse einzugehen, ohne dabei das globalere Geschehen im Klassenraum aus dem Blick zu verlieren, hat der Praktikant durch die adäquate Ausbalancierung dieser beiden Aufgaben über lange Strecken gut gelöst. Das einzig kritische Vorkommnis in diesem Zusammenhang war, als sich mehrere, um Hilfe suchende SuS meldeten und Herr S. sich gedanklich eine Liste über die SuS machte, welche er nach einer bestimmten Zeit schloss und begann, diese abzuarbeiten.¹⁸⁹ Durch die Konzentration auf jene SuS übersah er, dass eine weiter hinten sitzende Schülerin für mehrere Minuten aufzeigte, ohne dass sie von der Lehrkraft beachtet worden wäre. Als ich dies beobachtete, erinnerte ich mich an einen Hinweis seiner Mentorin, welchen sie ihn sowohl nach der vorhergehenden als auch nach dieser Stunde gegeben hatte: Sie schlug ihm vor, während der schülerzentrierten Unterrichtsphasen einen Teil des Unterrichtsraums zu einem Frageraum zu erklären, in der die SuS sich, nach dem Vorbild einer Sprechstunde, beim Lehrer anmelden und ihm ihre Fragen stellen können. Diesen Vorschlag sehe ich dahingehend als sinnvoll an, als dass hierdurch vermieden wird, dass der Lehrer Schülerinnen oder Schüler übersieht, da diese ihn nun aktiv ansprechen und nicht passiv warten, bis der Lehrer auf sie zugeht. Weiterhin kann durch eine günstige Positionierung dieses Frageraum im Klassenzimmer gewährleistet werden, dass dem Lehrenden trotz seiner beratenden Tätigkeit Störungen oder andere Vorkommnisse im Plenum nicht entgehen. Auch wenn mit Bezug auf die Lernziele der Stunde das Fazit gezogen werden muss, dass aufgrund der fehlenden Sicherungen und des Wegfalls der Erarbeitungsphase einerseits nicht alle kognitiven Ziele erreicht wurden, da die hierfür notwendigen Inhalte schlicht nicht Gegenstand des Unterrichts waren, und andererseits für andere Lernziele, welche mit bestimmten, tatsächlich durchgeführten Unterrichtsinhalten konnotiert wurden, aufgrund des Fehlens von damit verbundenen Sicherungsphasen nicht sicher evaluiert werden kann, ob und inwieweit diese Ziele tatsächlich erreicht wurden, so zeigen sich Abseits dieser inhaltlichen Komponente mit stärkerem Bezug auf die Lehrer-Schüler-Kommunikation, insbesondere in der Kompetenz der Klassenführung, beim Praktikanten große Fortschritte im Vergleich zur letzten Woche. Als weiteres positives Charakteristikum (für beide hospitierten Stunden) ist das Bestreben des Herrn S. zu nennen, durch vielfältige Variationen im methodischen Gang, d. h. durch häufiges, aber sinnvolles Alternieren der Sozialformen und Handlungsmuster, den Unterricht abwechslungsreich zu gestalten. Hierbei nutzt er in dieser Stunde das Handlungsmuster des Gebens und Nehmens, welches ich für mich als neues Handlungsmuster aus der Stunde mitnehme.

¹⁸⁹ Diesen Eindruck hatte ich zumindest bei der Beobachtung der Interaktion mit den SuS während der Übung.

5. Reflexion über das Praktikum – Was habe ich gelernt?

In diesem Praktikum habe ich meine ersten Erfahrungen im Unterrichten des Fachs Informatik gesammelt. Hierbei hatte ich das Glück, einen kompetenten Mentor an die Hand bekommen zu haben, welcher meine Philosophie des „Learning by doing“ teilte, weshalb seine Unterrichtsstunden oftmals handlungs- und praxisorientiert konzipiert waren. Durch die Hospitation seines Unterrichts sowie die schwerpunktmäßige Betrachtung des methodischen Gangs habe ich sehr viel darüber erfahren, wie man solche Unterrichtsstunden konzipiert, in deren Zentrum die Vermittlung von Wissen und Fertigkeiten über Lernaufgaben mit handlungs- bzw. praxisorientierten Bezügen steht. Im Mittelpunkt des Unterrichts standen die SuS, welche als forschend-entdeckende Lerner über lange Phasen hinweg eigenständig den Lerngegenstand erarbeiteten und diesen inkorporierten. Der Lehrer selbst nahm in diesen Phasen lediglich eine beratende bzw. helfende Position ein. Die SuS sind es gewohnt, mehrere Lernaufgaben in einem Block, das heißt nacheinander in Einzel-, Partner- oder Gruppenarbeit zu bearbeiten, ohne dass diese Arbeit durch Zwischensicherungen im Plenum unterbrochen wird. Dabei werden die Lernaufgaben eines Blocks meist in der Art konzipiert, dass sie ein und denselben Lerninhalt zum Gegenstand haben und diesen sukzessive mindestens auf den ersten drei Stufen der Bloomschen Lernzieltaxonomie behandeln, wobei Analyseaufgaben ebenso häufig vertreten sind. Ein Aufgabenblock stellt also ein sinnzusammenhängendes Lernmodul dar, welches die SuS anregen soll, sich mit dem Lerngegenstand auf unterschiedlichen Abstraktionsniveaus auseinanderzusetzen. Eventuelle Zwischensicherungen innerhalb eines Blocks bürgen dabei die Gefahr in sich, dass der Sinnzusammenhang zwischen den Aufgaben aufgrund der Unterbrechungen durch die Sicherungen für die SuS nicht mehr oder nicht mehr so deutlich erkennbar bleibt. Die Strategie, die Aufgaben erst nach der vollständigen Bearbeitung eines Blocks zu sichern, ist jedoch ebenso risikoreich. In diesem Zusammenhang bin ich bereits mehrmals darauf eingegangen, dass die leistungsschwächeren SuS oft nur die Aufgaben der Stufen der „Kenntnis“ und des „Verstehens“ bearbeiten, für die sie zwar detaillierte Ergebnisse produzieren können, sie schaffen es jedoch nicht, den Anforderungen gerecht zu werden, welche die Stufen der „Anwendung“ und „Analyse“ an sie stellen. Bei den leistungsstärkeren SuS wurde als Problem identifiziert, dass sie ihre Arbeitsergebnisse oftmals nicht festhalten, selbst wenn im Arbeitsauftrag das Herstellen eines Produkts explizit gefordert wird. Diese Handlungsweisen bzw. Vermeidungsstrategien der SuS können für die Arbeitsaufträge nur durch Sicherungen entgegengesteuert werden. Mit der Strategie, eine solche Sicherung nur am Ende des jeweiligen Blocks vorzunehmen, beraubt sich der Lehrer jedoch der Möglichkeit sicherzustellen, dass die SuS für die Durchführung der Arbeitsaufträge die damit konnotierten Handlungsmuster und Arbeitstechnik auch in der intendierten Form anwenden. Dass viele SuS trotz der fehlenden Sicherungen einen hohen Lernfortschritt erzielten, lag nicht zuletzt an dem hohen Maß an Engagement, welches mein Mentor an den Tag legte: Er war ständig präsent, ohne im Mittelpunkt zu stehen, er versuchte, den Anteil der Lehrervorträge und der Phasen des Frontalunterrichts so gering wie möglich zu halten und ging in schülerzentrierten Erarbeitungsphasen aktiv auf die SuS zu, fragte sie, ob und wie er helfen kann, und gab ihnen permanent Impulse für die weitere Arbeit. Ich konnte durch die Hospitation seines Unterrichts sehr gut beobachten, wie man schülerzentrierten Unterricht so gestaltet, dass er den SuS auch wirklich Spaß macht und diese für die Auseinandersetzung mit dem Gegenstand motiviert werden. Dies erreichte mein Mentor nicht nur durch seine Person, sondern ebenso durch das konsequente Verfolgen der Schülerinteressen. Hierbei versuchte er oft wie möglich, durch die Wahl solcher Bezüge, welche möglichst dicht an der Lebenswelt der SuS sind, neue Zugänge zu schaffen, welche die SuS befähigen, sich den Unterrichtsgegenstand weitestgehend eigenständig anzueignen. Hierbei war es sehr hilfreich, dass die Vorliebe des Lehrers für die praktische Informatik auch von den SuS geteilt wurde. Eine solche praktische

Veranlagung habe ich jedoch nicht nur bei meinem Mentor beobachtet, vielmehr kann ich auf der Grundlage der von mir hospitierten Stunden konstatieren, dass das gesamte Kollegium die Handlungs- und Produktorientierung als maßgebliches Charakteristikum ihres Unterrichts sieht und auch dementsprechend handelt. Durch das Aktivieren der SuS als handelnde Akteure schaffen es die Fachlehrer, auch solche Themen, welche den SuS als eher trocken oder theoretisch begegnen, durch das Schaffen handlungsorientiert Zugänge aufzulockern. Dies zeigt sich im Unterricht von Herrn W. insbesondere durch die oftmalige Integration des Handelns auf der enaktiven Repräsentationsebene in das Unterrichtsgeschehen, wobei diese Form vom Lehrenden insbesondere in den Phasen des Einstiegs in die Stunde bzw. der Einführung in ein neues Thema angewendet wird. So stellte er zum Beginn der Unterrichtsstunde zur Erarbeitung des Zählens mit Binärzahlen acht Stühle in der Nähe der Tafel nebeneinander und bat acht SuS, nach vorne zu kommen und sich auf diese zu setzen. Er erklärte den SuS, dass sie nun acht Bit darstellen würden, wobei der Schüler auf dem Stuhl rechts außen das niederwertigste und der Schüler auf dem Stuhl links außen das höchstwertigste Bit einer Dualzahl repräsentiere. Hierbei bedeutete das Sitzen auf dem Stuhl, dass der Schüler den Wert 0 annimmt, steht der Schüler andererseits, so hat er den Wert 1. Mit diesen Regeln sollten die SuS die Dualzahl schrittweise inkrementieren, wobei die SuS, welche nicht Teil der Zahl waren, prüfen, ob die SuS auf den Stühlen die einzelnen Inkrementierungsschritte richtig darstellen. Der Lehrer hatte sein Ziel, die SuS für das Thema, welches für manche Lernenden doch etwas schwerer fassbar ist, zu motivieren, wobei eine Schülerin die durchgeführte Aktion mit dem Satz „Das ist ja voll lustig“ kommentierte. Der Mehrwert, welchen das Praktikum für mich generierte, ergibt sich jedoch nicht nur in den Gegenständen der Konzeption und der Inszenierung von Unterricht, ich habe weiterhin viele Erfahrungen bezüglich der Lehreraufgaben außerhalb des Klassenraums sammeln können. So habe ich durch unsere Exkursion zur Internationalen Funkausstellung am eigenen Leib erfahren, wie schwierig es ist, zwei 11.Klassen trotz der Anwesenheit von zwei Lehrern sowie zwei Praktikanten und trotz dem Umstand, dass die SuS sehr diszipliniert waren, zu koordinieren und darauf zu achten, dass die SuS zusammenbleiben und während der Führung keiner verloren geht. Hierbei haben die Fachlehrer den SuS für mehrere Stunden sowohl die Freiheit, das Messegelände eigenständig, d. h. ohne Aufsicht durch Lehrer, zu erkunden, als auch die Verantwortung, an dem vereinbarten Ort zur vereinbarten Zeit wieder zur Gruppe zurückzukehren, übertragen.

Enaktive Repräsentation

Aushandlungsprozess über Klausuraufgabe

Aufgaben außerhalb des Klassenraums

IFA

6. Stundenübersicht

6.1 Übersicht über die Hospitationsstunden

Wochentag, Datum (Uhrzeit)	Kursart, Jahrgangsstufe (Kurskennzeichnung)	Klassenlehrer, Reihenthema
Montag, 01.09.2014 (14:00-15:15)	Leistungskurs, 11 (1)	Herr P. , Betriebssysteme
Dienstag, 02.09.2014 (8:00-9:15)	Leistungskurs, 12 (2)	Herr D. , Datenbanksysteme
Dienstag, 02.09.2014 (10:45-12:00)	Leistungskurs, 11 (1)	Herr P. , Betriebssysteme
Mittwoch, 03.09.2014 (8:00-9:15)	Profilierungskurs, 10 (1)	Herr W. , Scratch
Mittwoch, 03.09.2014 (10:45-12:00)	Leistungskurs, 12 (2)	Herr D. , Datenbanksysteme
Donnerstag, 04.09.2014 (10:45-12:00)	Leistungskurs, 12 (1)	Herr P. , Rechnerarchitektur
Donnerstag, 04.09.2014 (12:30-13:45)	Leistungskurs, 11 (1)	Herr P. , Betriebssysteme
Freitag, 05.09.2014 (12:30-13:45)	Leistungskurs, 12 (1)	Herr P. , Rechnerarchitektur
Freitag, 05.09.2014 (14:00-15:15)	Grundkurs, 12 (1)	Herr W. , App-Entwicklung
Montag, 08.09.2014 (9:15-10:30)	Grundkurs, 12 (1)	Herr W. , App-Entwicklung
Montag, 08.09.2014 (12:30-13:45)	Grundkurs, 11 (1)	Herr W. , Rechnerarchitektur
Mittwoch, 10.09.2014 (8:00-9:15)	Profilierungskurs, 10 (1)	Herr W. , Scratch
Mittwoch, 10.09.2014 (10:45-12:00)	Leistungskurs, 12 (2)	Herr D. , Datenbanksysteme
Donnerstag, 11.09.2014 (12:30-13:45)	Leistungskurs, 11 (1)	Herr P. , Betriebssysteme
Freitag, 12.09.2014 (12:30-13:45)	Leistungskurs, 12 (1)	Herr P. , Rechnerarchitektur
Freitag, 12.09.2014 (14:00-15:15)	Grundkurs, 12 (1)	Herr W. , App-Entwicklung
Montag, 15.09.2014 (7:50-8:35)	Klassenstufe 8	Herr S. , E-Mail nur für mich
Montag, 15.09.2014 (12:30-13:45)	Grundkurs, 11 (1)	Herr W. , Rechnerarchitektur
Mittwoch, 17.09.2014 (8:00-9:15)	Profilierungskurs, 10 (1)	Herr W. , Scratch
Mittwoch, 17.09.2014 (10:45-12:00)	Leistungskurs, 12 (2)	Herr D. , Datenbanksysteme
Donnerstag, 18.09.2014 (10:45-12:00)	Leistungskurs, 12 (1)	Herr P. , Rechnerarchitektur
Donnerstag, 18.09.2014 (12:30-13:45)	Leistungskurs, 11 (1)	Herr P. , PHP
Freitag, 19.09.2014 (12:30-13:45)	Leistungskurs, 12 (1)	Herr P. , Rechnerarchitektur
Freitag, 19.09.2014 (14:00-15:15)	Grundkurs, 12 (1)	Herr W. , App-Entwicklung
Montag, 22.09.2014 (14:00-15:15)	Leistungskurs, 11 (1)	Herr P. , PHP
Dienstag, 23.09.2014 (12:55-13:40)	Klassenstufe 8	Herr S. , E-Mail nur für mich
Mittwoch, 24.09.2014 (8:00-9:15)	Profilierungskurs, 10 (1)	Herr W. , Scratch
Mittwoch, 24.09.2014 (10:45-12:00)	Leistungskurs, 12 (2)	Herr D. , Datenbanksysteme
Donnerstag, 25.09.2014 (10:45-12:00)	Leistungskurs, 12 (1)	Herr P. , Rechnerarchitektur
Donnerstag, 25.09.2014 (12:30-13:45)	Leistungskurs, 11 (1)	Herr P. , PHP

6.2 Außerschulische Aktivitäten

Ganztägige Exkursion zur IFA am 09.09.2014

6.3 Übersicht über die selbst gehaltenen Stunden

Vgl. 2.1.3 Struktur der Unterrichtsreihe.

Anhang 1:

Erklärung der zu verwendenden Klassen, Methoden und Eigenschaften

OpenFileDialog

Objekte der Klasse OpenFileDialog sind Dialogfenster, welche den Nutzer dazu auffordern, eine oder mehrere Dateien zum Öffnen auszuwählen. Eine Instanz *ofd* der Klasse OpenFileDialog wird wie folgt erzeugt:

```
OpenFileDialog ofd = new OpenFileDialog();
```

Um die Instanz anzuzeigen, muss die Methode ShowDialog() aufgerufen werden.

```
ofd.ShowDialog();
```

SaveFileDialog

Objekte der Klasse SaveFileDialog sind Dialogfenster, welche den Nutzer dazu auffordern, einen Speicherpfad für eine zu speichernde Datei festzulegen. Eine Instanz *sfd* der Klasse SaveFileDialog wird wie folgt erzeugt:

```
SaveFileDialog sfd = new SaveFileDialog();
```

Um die Instanz anzuzeigen, muss die Methode ShowDialog() aufgerufen werden.

```
sfd.ShowDialog();
```

null

Der null-Zeiger ist ein Verweis, der sich auf kein Objekt bezieht.

Double.NaN

Stellt einen Wert dar, der keine Zahl ist. Man kann NaN (Not a Number) mit anderen Werten vergleichen, um herauszufinden, ob diese eine Zahl darstellen.

Convert.ToInt32(string value)

Konvertiert einen String in einen Integer.

Methoden der File-Klasse

`File.Exists(string path):`

Bestimmt, ob die angegebene Datei vorhanden ist.

`File.Copy(string sourceFileName, string destFileName):`

Kopiert eine vorhandene Datei in eine neue Datei. Das Überschreiben einer gleichnamigen Datei ist nicht zulässig.

`File.Move(string sourceFileName, string destFileName):`

Verschiebt eine angegebene Datei an einen neuen Speicherort und ermöglicht das Angeben eines neuen Dateinamens.

`File.Create(string path):`

Erstellt oder überschreibt eine Datei im angegebenen Pfad.

`File.Delete(string path):`

Löscht die angegebene Datei.

`File.Encrypt(string path):`

Verschlüsselt eine Datei, sodass sie nur mit dem Konto, mit dem die Datei verschlüsselt wurde, entschlüsselt werden kann.

`File.Decrypt(string path):`

Entschlüsselt eine Datei, die vom aktuellen Konto mit der Encrypt-Methode verschlüsselt wurde.

Anhang 2:

Projekt Dateikopierer

Arbeitsauftrag:

Realisiert die folgende Anforderungsbeschreibung für das WPF-Projekt „Dateikopierer.sln“.

Analysiert hierfür zunächst die drei Anforderungen, indem ihr die Sprachelemente identifiziert, welche durch die einzelnen Unterpunkte der Anforderungen beschrieben werden, und ihre Beziehungen zueinander erläutert. Bei der Analyse solltet ihr...

- entscheiden, ob ein Sprachelement im Oberflächencode (XAML-Code) oder im logischen Code (C#-Code) implementiert werden muss.
- entscheiden, ob es ein Sprachelement gibt, welches dem gerade betrachteten Sprachelement übergeordnet ist bzw. in dessen Anweisungsblock sich das gerade betrachtete Sprachelement befindet.
- prüfen, welche euch bekannten Klassen und Methoden für die Implementierung der jeweiligen Anforderung erforderlich sind.
- prüfen, ob der jeweilige Unterpunkt die Implementierung neuer Klassen oder Methoden bedingt.
- prüfen, welche anderen Sprachelemente bereits vorhanden sein müssen, damit das im Unterpunkt beschriebene Sprachelement implementiert werden kann.

Nach der Analyse solltet ihr **planen**, wie ihr bei der Implementierung der einzelnen Unterpunkte einer Anforderung vorgehen wollt. Unterteilt dazu das von euch geplante Vorgehen in einzelne Arbeitsschritte und schreibt sie in der Reihenfolge untereinander, in der ihr sie bei der Implementierung durchführen wollt. Achtet dabei auf mögliche Abhängigkeiten zwischen den Schritten!

Beispiel:

Schritt eins: Erstellen eines neuen WPF-Projekts mit dem Namen „Dateikopierer“ im Visual Studio

Schritt zwei: Öffnen der MainWindow.xaml-Datei. Hinüberziehen des Button-Elements aus der Toolbox in das Designfeld

Schritt drei: Auswahl des eben erzeugten Buttons. Zuweisung des Strings „Quelldatei“ zur Eigenschaft „Content“ sowie des Strings „btn_quelldatei“ zur Eigenschaft „Name“ im Eigenschaftfenster des Buttons

Schritt vier...

Implementiert anschließend die jeweilige Anforderung, indem ihr die von euch geplante Schrittfolge abarbeitet.

Anforderung 1:

Für diese Anforderung führen wir die drei Schritte der Analyse, Planung und Implementierung gemeinsam in der Klasse durch.

- Die Klasse *MainWindow.xaml* enthält:
 - einen Button mit dem Text „Quelldatei“ und dem Namen „btn_quelldatei“ sowie
 - eine TextBox mit dem Namen „tb_quelldatei“.
- Wenn *btn_quelldatei* geklickt:
 - Erstelle eine *OpenFileDialog*-Instanz und zeige sie dem Nutzer.
 - Prüfe über die Eigenschaft *FileName*, ob der Nutzer eine Datei ausgewählt hat.
Hinweis: Wenn nicht, ergibt die Abfrage *FileName == null* den Wert *true*.
 - Falls der *FileName* nicht *null* ist:
 - Weise dem Text von *tb_quelldatei* den Wert von *FileName* zu.
 - Falls doch, zeige in einer *MessageBox*: „Es wurde keine Datei ausgewählt.“

Anforderung 2:

Für diese Anforderung sollt ihr die drei Schritte der Analyse, Planung und Implementierung in Partnerarbeit durchführen. Ihr werdet feststellen, dass die Anforderung und somit auch die Ergebnisse der drei Schritte eine hohe Ähnlichkeit zur Anforderung 1 aufweisen. Daher solltet ihr es während der Bearbeitung der drei Schritte vermeiden, eure Aufzeichnungen über die Anforderung 1 zu Hilfe zu nehmen. Diese solltet ihr lediglich als Vergleich für diejenigen Schritte verwenden, die ihr bereits eigenständig erarbeitet habt.

- Die Klasse *MainWindow.xaml* enthält:
 - einen Button mit dem Text „Ziel“ und dem Namen „btn_ziel“ sowie
 - eine TextBox mit dem Namen „tb_ziel“.
- Wenn *btn_ziel* geklickt:
 - Erstelle eine *SaveFileDialog*-Instanz und zeige sie dem Nutzer.
 - Prüfe über die Eigenschaft *FileName*, ob der Nutzer einen Dateinamen eingegeben hat.
Hinweis: Wenn nicht, ergibt die Abfrage *FileName == null* den Wert *true*.
 - Falls der *FileName* nicht *null* ist:
 - Weise dem Text von *tb_ziel* den Wert von *FileName* zu.
 - Falls doch, zeige in einer *MessageBox*: „Es wurde keine Datei ausgewählt.“

(nach Anforderung 2: PAUSE – wir vergleichen)

Anforderung 3:

Für diese Anforderung sollt ihr die drei Schritte der Analyse, Planung und Implementierung in Partnerarbeit durchführen.

- Die Klasse *MainWindow.xaml* enthält:
 - einen TextBlock mit dem Text „Anzahl der Kopien:“,
 - eine TextBox mit dem Namen „tb_anzahl“ sowie
 - einen Button mit dem Text „Starte Kopiervorgang“ und dem Namen „btn_kopieren“.
- Wenn *btn_kopieren* geklickt:
 - Prüfe, ob die in der TextBox *tb_quelldatei* angegebene Quelldatei existiert.
 - Wenn nein, zeige in einer *MessageBox*: „Die Quelldatei existiert nicht.“
 - Wenn ja, prüfe, ob der Text von *tb_anzahl* eine Zahl darstellt.

- Wenn nein, zeige in einer *MessageBox*: „Keine gültige Zahl.“
- Wenn ja, kopiere die Quelldatei n-mal, wobei n für die aus *tb_anzahl.Text* konvertierte Zahl steht. Die neuen Dateien sollen dabei *tb_ziel.Text 1*, *tb_ziel.Text 2*,... *tb_ziel.Text n* heißen.

Anhang 3:

Projekttest

Arbeitsauftrag:

Ladet die Datei „Dateikopierer.zip“ von <http://randco.square7.de/> herunter und entpackt sie auf dem Desktop.

Hierin befindet sich unser Programm *Dateikopierer.exe*. Führt das Programm einmal mit Administratorrechten und einmal ohne Administratorrechte aus.

Kopiert in beiden Fällen jeweils 10-mal die Datei „Test.txt“ in den Ordner, in dem sich auch die EXE befindet, sowie in C:

Beobachtet, wie sich das Betriebssystem in allen vier Fällen verhält und notiert eure Beobachtungen in der unten stehenden Tabelle.

Formuliert anschließend eine Vermutung über die Ursache des Verhaltens des Betriebssystems in allen vier Fällen.

	Ohne Administratorrechte	Mit Administratorrechten
Exe-Verzeichnis		
C:		

Anhang 4:

If-else-Verzweigungen

- Erarbeitet aus dem *Galileo Openbook* die Funktionsweise der if-else-Verzweigung. Arbeitet die generelle Struktur einer solchen Verzweigung heraus, indem ihr die Bedeutung der Schlüsselwörter if und else sowie der in Klammern stehenden Abfragen ermittelt und notiert. (15⁺)

http://openbook.galileo-press.de/visual_csharp_2010/visual_csharp_2010_02_006.htm#mj2efa666a6e6f567495ac5c855c465b45

- Nachfolgend seht ihr zwei Programmausschnitte mit if-else-Verzweigungen. Stellt Vermutungen darüber an, welche Werte die jeweils betrachtete Variable annehmen darf, damit die MessageBox eine bestimmte Meldung ausgibt. Haltet eure Vermutungen wie folgt fest:

Programmausschnitt 1:

Ausgabe der MessageBox	Mögliche Werte für produkt_gekauft
Das Produkt wurde gekauft.	
Das Produkt wurde nicht gekauft.	

Programmausschnitt 2:

Ausgabe der MessageBox	Mögliche Werte für schulklasse
Sekundarstufe II	
Sekundarstufe I	
Grundschule	

Hinweis: In manchen Fällen kann eine unendliche Menge an Werten zu einer bestimmten Ausgabe führen. Beschreibt in solchen Fällen, wie man diese Mengen bildet, indem ihr die Eigenschaften der jeweiligen Menge beschreibt oder mithilfe der logischen bzw. relationalen Operatoren eine Vorschrift zur Bildung dieser Menge formuliert. (10⁺)

- Überprüft eure Vermutungen. Geht hierzu genauso vor wie bereits bei den relationalen und logischen Operatoren. Lasst die Programmausschnitte auch diesmal schrittweise durchlaufen. (10⁺)

Programmausschnitt 1

```
bool produkt_gekauft = false;

if (produkt_gekauft == true)
{
    MessageBox.Show("Das Produkt wurde gekauft.");
}
else
{
    MessageBox.Show("Das Produkt wurde nicht gekauft.");
}
```

Programmausschnitt 2

```
int schulklasse = 0;

if (schulklasse > 10)
{
    MessageBox.Show("Sekundarstufe II");
}
else
{
    if (schulklasse > 6)
    {
        MessageBox.Show("Sekundarstufe I");
    }
    else
    {
        MessageBox.Show("Grundschule");
    }
}
```

Anhang 5:

Switch-case-Verzweigungen

- Nachfolgend seht ihr einen Programmausschnitt mit mehreren if-Verzweigungen. Testet den Ausschnitt im Visual Studio. Öffnet dazu ein neues WPF-Projekt mit C# und fügt anschließend den unten stehenden Code in die Methode *MainWindow()* der Klasse *MainWindow.xaml.cs* ein.

```
string gewählte_farbe = "Orange";

string farbwahl = "keine oder unbekannte Farbe gewählt";

if (gewählte_farbe == "Schwarz")
{
    farbwahl = "Schwarz";
}
if (gewählte_farbe == "Weiß")
{
    farbwahl = "Weiß";
}
if (gewählte_farbe == "Blau")
{
    farbwahl = "Blau";
}
if (gewählte_farbe == "Gelb")
{
    farbwahl = "Gelb";
}
if (gewählte_farbe == "Rot")
{
    farbwahl = "Rot";
}

MessageBox.Show("Gewählte Farbe: " + farbwahl);
```

Initialisiert den string *gewählte_farbe* mit verschiedenen Werten und haltet für jeden Wert die Ausgabe der *MessageBox* fest. Testet in jedem Fall die Werte in der nachfolgenden Beispieltabelle:

Werte von <i>gewählte_farbe</i>	Ausgabe der <i>MessageBox</i>
"Schwarz"	
"Weiß"	
"Blau"	
"Gelb"	
"Rot"	
"Orange"	

Setzt einen Haltepunkt in der Zeile *public MainWindow()* und führt das Programm aus, indem ihr solange F11 drückt, bis das Ende der Methode erreicht ist. Beobachtet bei jedem Durchlauf, in welche Zeilen der Debugger springt.

Formuliert aus euren Beobachtungen eine allgemeine Vermutung über die Funktion der if-Verzweigung. (15')

- Erarbeitet aus dem *Galileo Openbook* die Funktionsweise der switch-case-Verzweigung. Arbeitet die generelle Struktur einer solchen Verzweigung heraus, indem ihr die Bedeutung der Schlüsselwörter switch und case sowie der in Klammern stehenden Abfragen ermittelt und notiert. (15')

http://openbook.galileo-press.de/visual_csharp_2010/visual_csharp_2010_02_006.htm#mjf8e3e1ebdca9edcfe6304d232f700107

- Beobachtet im nachfolgenden Programmausschnitt, für welche Werte von gewählte_farbe der Debugger in welche Anweisung innerhalb der switch-case-Verzweigung springt, indem ihr den Programmausschnitt für verschiedene Werte von gewählte_farbe schrittweise ausführt. Inwiefern unterscheidet sich die Funktionalität der switch-case-Version des Ausschnittes von der Implementierung mit if-Verzweigungen? (5')

```
string gewählte_farbe = "Orange";
```

```
string farbwahl = "keine oder unbekannte Farbe gewählt";
```

```
switch (gewählte_farbe)
{
    case "Schwarz":
        farbwahl = "Schwarz";
        break;
    case "Weiß":
        farbwahl = "Weiß";
        break;
    case "Blau":
        farbwahl = "Blau";
        break;
    case "Gelb":
        farbwahl = "Gelb";
        break;
    case "Rot":
        farbwahl = "Rot";
        break;
}
```

```
MessageBox.Show("Gewählte Farbe: " + farbwahl);
```

Anhang 6:

For- und while-Schleife

- Erarbeitet euch die Funktionsweisen der for- sowie der while-Schleife, indem ihr den folgenden Code schrittweise im Debug-Modus ausführt.
Öffnet dazu das *VisualStudio* und erstellt ein neues WPF-Projekt mit C#. Fügt anschließend den unten stehenden Code in die Methode *MainWindow()* der Klasse *MainWindow.xaml.cs* ein. Setzt einen Haltepunkt in der Zeile *public MainWindow()* und führt das Programm aus, indem ihr solange F11 drückt, bis das Ende der Methode erreicht ist.
Notiert euch für jeden Schritt, wie sich das Programm verhält: In welche Zeile springt der Compiler? Wie verändern sich die Variablen? (25‘)

Hinweis: Ihr könnt eure Vermutungen über die Funktionsweise der beiden Schleifen im Kapitel „2.7 Programmschleifen“ des *Galileo Openbook* zu C# überprüfen.

http://openbook.galileo-press.de/visual_csharp_2010/visual_csharp_2010_02_006.htm#mjd3feeb6d2c6836721f38ccb9014ec214

```
//for-Schleife
MessageBox.Show("Ich zähle nun von 1 bis 10");
for (int i = 1; i <= 10; i = i + 1)
{
    MessageBox.Show(i.ToString());
}

//while-Schleife
MessageBox.Show("Ich zähle nun von 1 bis 10");
int j = 1;
while(j <= 10)
{
    MessageBox.Show(j.ToString());
    j = j + 1;
}
```

- Beim Ausführen des Programms werdet ihr festgestellt haben, dass es nicht das tut, was es eigentlich ankündigt. Verändert die Variablen und Vergleichsoperatoren so, dass in beiden Schleifen wirklich von 1 bis 10 gezählt wird. (10‘)

Anhang 7:

Klausuraufgabe

Gegeben sind eine TextBox mit dem Namen *tb_quelldatei*, eine TextBox mit dem Namen *tb_ziel* sowie ein Button mit dem Namen *btn_kopieren*.

Aufgabe: Schreiben Sie in der Programmiersprache C# eine Methode *private void btn_kopieren_Click()*, welche folgende Anforderungen erfüllt:

- Prüfe, ob die in der TextBox *tb_quelldatei* angegebene Quelldatei existiert.
 - Wenn nein, zeige in einer *MessageBox*: „Die Quelldatei existiert nicht.“
 - Wenn ja, kopiere die Quelldatei 5.000-mal. Beim *i*-ten Kopiervorgang soll die Zieldatei den Namen *tb_ziel.Text + i* erhalten.

Als Hilfe sind nachfolgend einige Methoden der Sprache C# dokumentiert

- `bool File.Exists(string path)`:
Gibt *true* zurück, wenn eine Datei unter dem angegebenen Pfad *path* vorhanden ist und *false*, wenn dies nicht der Fall ist.
- `void File.Copy(string sourceFileName, string destFileName)`:
Kopiert die Datei unter dem Pfad *sourceFileName* nach *destFileName*.
- Auf die Eigenschaft *Text* einer TextBox mit dem Namen *tb* wird in C# durch *tb.Text* zugegriffen.
- `void MessageBox.Show(string text)`:
Öffnet eine *MessageBox*, die den als Parameter übergebenen Text anzeigt.

Erwartungsbild / Benotung (max. 5 Punkte)

```
private void btn_kopieren_Click()
// 0.5 Punkte für Methodenbenennung/Klammersetzung
{
    if (File.Exists(tb_quelldatei.Text))
        // 0.5 Punkte für Verwendung von if und Klammersetzung
        // 0.5 Punkte für die Verwendung von File.Exists(tb_quelldatei.Text) als Bedingung
    {
        for (int i = 0; i < 5000; i = i + 1)
            //jeweils einen halben Punkt für die Verwendung von for, der Definition /
            //Initialisierung einer Zählvariablen, das Inkrementieren sowie die
            //Abbruchbedingung
        {
            File.Copy(tb_quelldatei.Text, tb_ziel.Text + "_" + i);
            //0.5 Punkte
        }
    }
    else // 0.5 Punkte für Verwendung von else und Klammersetzung
    {
        MessageBox.Show("Die Quelldatei existiert nicht.");
        //0.5 Punkte
    }
}
```

Gleiche Bewertung, wenn *while*-Schleife verwendet wird, d. h. jeweils 0.5 Punkte für die Definition / Initialisierung der Zählvariablen, der Verwendung von *while*, der Abbruchbedingung im Kopf sowie das Inkrementieren am Schleifenende.

Literaturverzeichnis

Adamski, P.: Auf vielen Wegen ins Land der Pharaonen. Innere Differenzierung im Geschichtsunterricht, in: Geschichte lernen. Jg. 22 (H. 131, 2009), S. 2-13. (= Adamski 2009).

Aspray, W. & Ensmenger, N.: Software as Labor Process, in: Hashagen, U., Keil-Slawik, R. & Norberg, A. (Hrsg.), History of Computing: Software Issues. Heidelberg 2002, S. 139-166. (= Aspray & Ensmenger 2002).

Booch, G.: Objektorientierte Analyse und Design . Mit praktischen Anwendungsbeispielen. 6. Aufl. Bonn 1995. (= Booch 1995).

Doberenz, W. & Gewinnus, T.: Visual C# 2012 - Grundlagen und Profiwissen. 1. Aufl. München 2013. (= Doberenz & Gewinnus 2013).

Floyd, C. & Züllighoven, H.: Softwaretechnik, in: Pomberger, G. & Rechenberger, P. (Hrsg.), Informatik Handbuch. München 1997, S. 641-667. (= Floyd & Züllighoven 1997).

Hartmann, W., Näf, M. & Reichert, R.: Informatikunterricht planen und durchführen. 2. Aufl. Berlin / Heidelberg 2007. (= Hartmann, Näf & Reichert 2007).

Humbert, L.: Didaktik der Informatik: mit praxiserprobtem Unterrichtsmaterial. 2. Aufl. Wiesbaden 2006. (= Humbert 2006).

Jerusalem, M.: Die Entwicklung von Selbstkonzepten und ihre Bedeutung für Motivationsprozesse im Lern- und Leistungsbereich. 1. Aufl. Berlin 1993. (= Jerusalem 1993).

Klafki, W.: Neue Studien zur Bildungstheorie und Didaktik. Zeitgemäße Allgemeinbildung und kritisch konstruktive Didaktik. 5. Aufl. Basel 1996. (= Klafki 1996).

Meyer, H.: Unterrichtsmethoden. 2. Praxisband. 5. Aufl. Frankfurt am Main 1994. (= Meyer 1994).

Meyer, H.: Leitfaden Unterrichtsvorbereitung . 6. Aufl. Berlin 2012. (= Meyer 2012).

Pont, M. J.: Software Engineering with C++ and CASE Tools. 1. Aufl. München 1996. (= Pont 1996).

Sitte, W.: Taxonomie, in Sitte, W. & Wohlschlägl, H. (Hrsg.), Beiträge zur Didaktik des „Geographie und Wirtschaftskunde“-Unterrichts. Wien 2001, S. 473-475. (= Sitte 2001).

Wenger, R.: Handbuch der .NET 4.0-Programmierung. Band 1. C# und .NET-Grundlagen. 1. Aufl. München 2010. (= Wenger 2010).