Objektorientierte Programmierung in der Sekundarstufe II des Gymnasiums

Reflexion über Herangehensweisen zur Vermittlung grundlegender Programmierparadigmen

Schriftliche Arbeit für das Lehramt an Gymnasium im Fach Informatik

Hamburg, den . November 2017

Pamina Maria Berg

LiV
HS 16-08-Frö

Hauptseminarleitung	Dr. Sven Michael Fröhlich
Erstgutachter	Sven Alisch
Zweitgutachter	Christina von Bremen
Datum der mündlichen Prüfung	21.12.2017

Inhaltsverzeichnis

1 | Einleitung

> „Man kann sich Vorstellungen und Begriffe nicht in fertiger Form einverleiben. Man muss sie nachschaffen, nachkonstruieren.“
>
> *(Hans Aebli)*

Die Lehre des Programmierens ist stark von einem schrittweisen Abarbeiten von Programmierparadigmen in einer ausgewählten Programmiersprache geprägt. In Lehrbüchern werden beispielsweise anhand von Projekten die klassischen Begriffe der Objektorientierung *Klasse, Objekt, Methode, Parameter* vor- und zum Durcharbeiten am Computer bereitgestellt.

Informatische (Grund-)Bildung sollte als Teil der Allgemeinbildung () auch allgemeine Konzepte der Informatik vermitteln. HUBWIESER weist in seinem Standardwerk zur Informatik-Didaktik auf einen leider immer noch auftretenden Sachverhalt hin:

Zitat Breier 1994 raussuchen

> Beim Betrachten entsprechender Rahmenpläne entsteht der Eindruck, dass entweder produktbezogene Anwenderschulungen oder Programmierkurse im Kleinen in diesem Unterricht durchgefuührt werden. ([Hub07, S.40] aus [Koe93])

2 | Ausgangssituation

Es werden nun zunächst die systemischen Rahmenbedingungen, sowie die Voraussetzungen, die sich aus den inhaltlichen Lernzielen und den Lerngruppen ergeben, dargestellt.

2.1 Systemische Rahmenbedigungen

Der Unterrichtsplanung zugrunde liegen zum einen der Rahmenplan Informatik für die Gymnasiale Oberstufe (Vgl. [HH09]) sowie das schulinterne Curriculum des Gymnasium Ohmoor. Im folgenden werden die für die durchgeführte Unterrichtspraxis relevanten Inhalte kurz erläutert.

2.1.1 Rahmenplan

Der Rahmenplan Informatik für die Gymnasiale Oberstufe in Hamburg spezifiziert die *Objektorientierte Modellierung* als verbindlichen Inhalt, wobei eine explizite Forderung nach der „Erarbeitung der Sprachelemente der verwendeten objektorientierten Programmiersprache“ ([HH09, S. 17]) besteht. Des weiteren sind verschiedene Anforderungsbereiche definiert, durch die sowohl fachliche als auch überfachliche Kompetenzen erworben und überprüft werden sollen. Der für die zu untersuchende Unterrichtssituation relevante Kompetenz bezieht sich auf den Bereich des *Darstellen und Interpretieren*, in dem die Schülerinnen und Schüler „Modelle und Algorithmen sowohl grafisch als auch verbal“ beschreiben können sollen (Vgl. [HH09, S.16]). Ein großer Fokus wurde planungsbedingt auch auf die Kompetenz des *Kommunizieren und Kooperieren* gelegt (siehe hierzu Abschnitt **3.3**).

2.1.2 Curriculum des Gymnasium Ohmoor

Die Fachschaft Informatik des Gymnasium Ohmoor konkretisiert im Schulinternen Curriculum die allgemein formulierten Vorgaben aus dem Rahmenplan Informatik. So ist im zweiten Semester der Gymnasialen Oberstufe das Thema *Objektorientierte Modellierung/Programmierung von Grafiksystemen mit Java* angesiedelt (Vgl. [GyOhm16, S.6f.]). Als verbindlicher Inhalt ist hier unter anderem die „Erarbeitung von Sprachelementen: [...] Kontrollstrukturen“ ([GyOhm16, S.7]) genannt, zu denen auch das Programmierparadigma der Schleifenkonstrukte gehört.

2.2 Inhaltliche Ziele

Die inhaltlichen Ziele der Unterrichteinheit waren sowohl fachlicher als auch überfachlicher Art und lassen sich wie folgt zusammenfassen:

- Die SuS analysieren das BlueJ-Projekt, um sich mit den wesentlichen Merkmalen von Schleifen als Kontrollstrukturen in der Programmierung vertraut zu machen.
- Die SuS erläutern anhand eines Minimalbeispiels in Java-Syntax den Ablauf einer *for-, while-* oder *do-while*-Schleife, indem sie eine Kurz-Vorführung im Plenum vorbereiten und durchführen.

Die hierzu passenden Einzellernziele sind:

- Die SuS geben mindestens die prägnanten Merkmale des Quellcodes an und beschreiben den grundsätzlichen Programmablauf im BlueJ-Projekt und sind bestenfalls in der Lage, eigene Schleifenkonstrukte zu implementieren und die Geeignetheit des gewählten Konstrukts zu begründen.
- Die SuS beschreiben mindestens umgangssprachlich den Zusammenhang zwischen dem von ihnen gewählten Schleifenkonstrukt und ihrer Präsentation und beurteilen bestenfalls die Passung von Präsentation und Schleifenkonstrukt der eigenen und anderen Gruppen.
- Die SuS stellen mindestens auf Nachfrage die wesentlichen Unterschiede der drei Schleifenkonstrukte dar und vergleichen diese bestenfalls im Hinblick auf verschiedene Einsatzszenarien.

2.3 Lerngruppen

Der in dieser Arbeit untersuchte Lerngegenstand wurde in zwei Vergleichsgruppen mit methodisch variierten Vorgehensweisen unterrichtet. Diese werden im Folgenden als Vergleichsgruppe **A** und **B** bezeichnet.

Vergleichsgruppe **A**:
Der Informatik-Wahlpflichtkurs auf grundlegendem Niveau umfasst insgesamt 17 Schülerinnen und Schüler, davon sind vier weiblich und dreizehn männlich. Der Unterricht findet regelhaft donnerstags von 8:00 Uhr bis 9:30 Uhr statt. Es handelt sich um einen motivierten, kleinen Kurs, den ich zum zweiten Halbjahr übernommen habe. Die Leistungsspanne ist relativ groß, was sich in der kürzlich geschriebenen Klausur gezeigt hat – jedoch befindet sich die Mehrheit der SuS im oberen Leistungsdrittel. Einige SuS sind sehr programmieraffin und probieren eigenständig programmiertechnische Verfahren im Unterricht aus, die über das geforderte Maß hinausgehen. Schwächeren SuS fällt das Programmieren an sich noch etwas schwerer. Einer der SuS setzt sich leistungstechnisch deutlich von der Gruppe ab, da er durch sein Praktikum bereits umfassende Programmierkenntnisse in Java besitzt und diese selbstständig und mühelos im Unterricht umsetzen kann. Auch nimmt er immer wieder die Rolle eines Lernberaters ein und hilft seinen MitschülerInnen bei Schwierigkeiten. Bei Nachfragen antwortet er umfassend und adressatengerecht.

Nochmal umformulieren, da zu nah an U-Entwürfen!!!

Vergleichsgruppe **B**:
Die erste Vergleichsgruppe ist der PGW-Profil-Kurs, bestehend aus 26 Schülerinnen und Schülern, deren Leistungsniveau sich eher heterogen gestaltet, wobei sich im Vergleich zu Gruppe **B** keine echte Leistungsspitze abzeichnet. Das Vorwissen der SuS in Bezug auf die zu untersuchende Unterrichtseinheit war (bedingt durch verschiedene Projektphasen der SuS in anderen Fächern) etwas geringer als bei der Vergleichsgruppe **A**, jedoch nicht in diesem Maße, als dass eine Lernhürde bei der Analyse von Quelltext zu erwarten gewesen wäre.

Beschreibung des PGW-Profil-Kurses

Beide Vergleichsgruppen waren bereits sicher im Umgang mit der Entwicklungsumgebung *BlueJ* und hatten in unterschiedlichen Kontexten eigene Programmiererfahrungen mit Java machen können. Größere Schwierigkeiten beim eigenständigen Umgang mit Quellcode waren deshalb nicht zu erwarten.

3 | Die Praxissituation

Im folgenden werden die für die Planung der Praxissituation maßgeblichen Details und Entscheidungen, sowie die Vorgehensweisen für die Durchführung beschrieben.

3.1 BlueJ

Die Java-Entwicklungsumgebung BlueJ wurde an der Monash University in Australien mit dem Ziel entwickelt, eine Umgebung für Programmieranfänger mit einer einfachen Benutzerschnittstelle zu schaffen (Vgl. [Bar03, S.14]). BARNES und KÖLLING betonen die besondere Geeignetheit von BlueJ in der Lehre und führen diese auf die native Visualisierung der Klassenstruktur und die damit einhergehende Möglichkeit, mit den Objekten direkt zu interagieren, „ohne Testklassen schreiben zu müssen“ (Vgl. [Bar03, S.15]).

Hier wäre noch Spielraum...

3.2 Bausteine der Unterrichtsplanung und Didaktische Entscheidungen

Die Unterrichtseinheit zum Lerngegenstand Schleifenkonstrukte besteht aus zwei Bausteinen, die inhaltlich ineinandergreifen, jedoch methodisch andere lerntheoretische Ansätze verfolgen.

Der erste Baustein besteht aus einem BlueJ-Projekt, welches ich durch meine Tätigkeit als Übungsgruppenbetreuerin an der Universität Hamburg kennengelernt habe. Dieses Projekt ist an das Lehrbuch zur Objektorientierten Programmierung mit Java und BlueJ von BARNES und KÖLLING angelehnt und versucht, über kurze Methodenrümpfe, die von den Lernenden analysiert werden sollen, verschiedene Arten von Schleifenkonstrukten zu vermitteln.

Zusätzlich zu diesem Kurzprojekt haben die SuS ein Arbeitsblatt mit verschiedenen Aufträgen bekommen, welche sie schrittweise durch das Projekt führen sollten (Vgl. Anhang **A**).

Betrachtet man die Arbeitsaufträge, so sieht man, dass die spezifischen Vorteile von BlueJ genutzt werden, um die SuS mit dem Quellcode interagieren zu lassen: Sie sollen verschiedene Beispiele ausführen und dabei auf die jeweiligen Rückgaben des Programms achten. Die SuS werden demnach aufgefordert, die informatischen Vorgänge zu *beobachten* (s. hierzu auch [Aeb11, S.67ff.]).

Weiterführen

Den zweiten Baustein haben sich die SuS jeweils eigenständig erarbeitet: Auf Grundlage ihrer basalen Syntaxkenntnisse haben die SuS in Gruppen verschiedene „leere“ Schleifenkonstrukte in Java bekommen, deren Ablauf sie jeweils zunächst analysieren und dann in Form einer interaktiven Gruppenpräsentation darstellen sollten. Hierbei war es den SuS freigestellt, welche Hilfmittel sie dazu einsetzen und welche Aktivitäten vorgeführt werden können. Es sollte lediglich deutlich werden, wie die von ihnen vorgestellte Sequenz von Anweisungen mit der von ihnen ausgewählten *for-, while-* oder *do-while*-Schleife im Zusammenhang steht.

Unter „Programmieren“ im klassischen Sprachgebrauch wird immer ein Am-PC-Sitzen und Quellcode schreiben verstanden. Die Informatik und insbesondere der Teilbereich der Objektorientierten Programmierung hat in jedoch weiten Teilen der Lehre bereits den Anspruch, syntaxübergreifende Konzepte anstatt primitivem (Programmier-)Sprachenverständnis zu vermitteln.

Hier wäre eine Quelle schön

Aus diesem Grunde wurden die SuS bei der Unterrichtseinheit angeregt, den Abstraktionsschritt vom Quelltext zum spielerischen physischen Ausprobieren des Programmierparadigmas *Schleifen* zu vollziehen. Sie sollten das Schema eines Schleifenkonstrukts nicht nur verstehen, sondern auch nachempfinden und – mit einem Ausblick auf das folgende Semesterthema *Algorithmen und Datenstrukturen* – ein Gefühl dafür entwickeln, dass mehrere Einzelschritte für die Ausführung der Anweisung bis zum Ergebnis notwendig sind.

Entweder hier Aebli zitieren oder bei der Reflexion...?

3.3 Vorgehensweise und Methodische Entscheidungen

Die beiden Bausteine wurden zum Zweck eines produktiven Vergleichs in jeweils umgekehrter Reihenfolge mit den Vergleichsgruppen durchgeführt. So hat Vergleichsgruppe **A** zunächst

mit der interaktiven Vorführung, also dem enaktiven Ansatz, begonnen und sich danach mit dem BlueJ-Projekt beschäftigt – der Vergleichsgruppe **B** wurde zuerst der Quelltext und das Arbeitsblatt ausgegeben, mit deren Erarbeitung sie in Partnerarbeit und ohne weitere Einhilfen begonnen haben.

Zweck der Variation der Vorgehensweise war mein Interesse an der Effektivität und dem Einfluss des Arbeitsauftrags zur freien Präsentation der Schleifenkonstrukte, also des Lernens durch *Bewegungshandlungen* (Vgl. [Aeb11, S.183f.]) auf das Grundverständnis des Programmierparadigmas.

4 | Reflexion der Herangehensweisen

Bei der Reflexion der Herangehensweisen und der Praxissituationen wird es nun darum gehen, die beiden in ihrer Reihenfolge variierten Vorgehensweisen kritisch zu betrachten, in Bezug zueinander zu setzen und eine daraus resultierende Fragestellung zu entwickeln und untersuchen.

4.1 Kritische Betrachtung

Erarbeitung A: Die erste Vergleichsgruppe hat mit dem BlueJ-Projekt begonnen. Sie sollten hierbei ohne Einhilfen arbeiten und es wurden Fragen nur in einem sehr geringen Maße beantwortet. Die analysierenden Aufgabenteile (1–3) wurden von den SuS ohne Probleme bearbeitet, bei Aufgabe 4 trat jedoch eine Lernhürde auf: Die SuS wurden dazu aufgefordert, den Lernschritt vom *Verstehen* des Lerngegenstands zum *Anwenden* des Konzepts auf eine neue Situation zu tätigen. Viele der SuS haben sich an die Aufgabe herangewagt und Schritt für Schritt versucht, eine Lösung zu implementieren

- Lernhürden bei Start mit Programmierung aufgetreten
- Enaktiver Asnatz bietet viel Spielraum für sehr spezifische Randfälle
- EA fördert kreativen Umgang und Durcharbeiten von trockener Syntax
- Bei Vorstellung des EA durch SUS direkte Diskussionsmöglichkeiten -> Mehr Zeit für den Vergleich der Lösungen einplanen, um tieferes Verständnis zu fördern
- Reihenfolge (Präferenz) stark v individ. SuS abhängig
- EA von allen SuS als förderlich wahrgenommen
- PA setzt Syntaxkenntnisse voraus
- vorgegebene Konstrukte müssen analysiert und verstanden werden

4.2 Resultierende Fragestellung

Fällt der Umgang mit Programmiersprachen und -konzepten leichter, wenn neben der theoretischen Vermittlung auch eine Handlungsorientierung abseits des Computers stattfindet?

5 | Schlussfolgerungen für die Unterrichtseinheit

Aufgrund der Untersuchung und Reflexion der Praxissituation werden die daraus resultierenden Konsequenzen und Fragestellungen nun in einer Alternativkonzeption der Unterrichtsstunde zum Einen und mithilfe von Schlussfolgerungen für die gesamte Unterrichtseinheit der *OOP/OOM* zum Anderen verarbeitet

hier fehlt noch ein schöner Halbsatz

- statt autodidaktisch v. SuS beispielhaft gemeinsam im Vorwege oder verknüpft mit EA, so dass vorgegebene Schleifen nachgespielt werden sollen
- IDEE: Anknüpfen/Verknüpfen von EA mit Programmierprojekt
- Kontext Spiele/Möbelplaner -> Modellbildung auf Papier -> Ausschneiden, legen,...

6 | Fazit und Ausblick

Abbildung 6.1: Der NXT-Stein [**?**, S. 42]

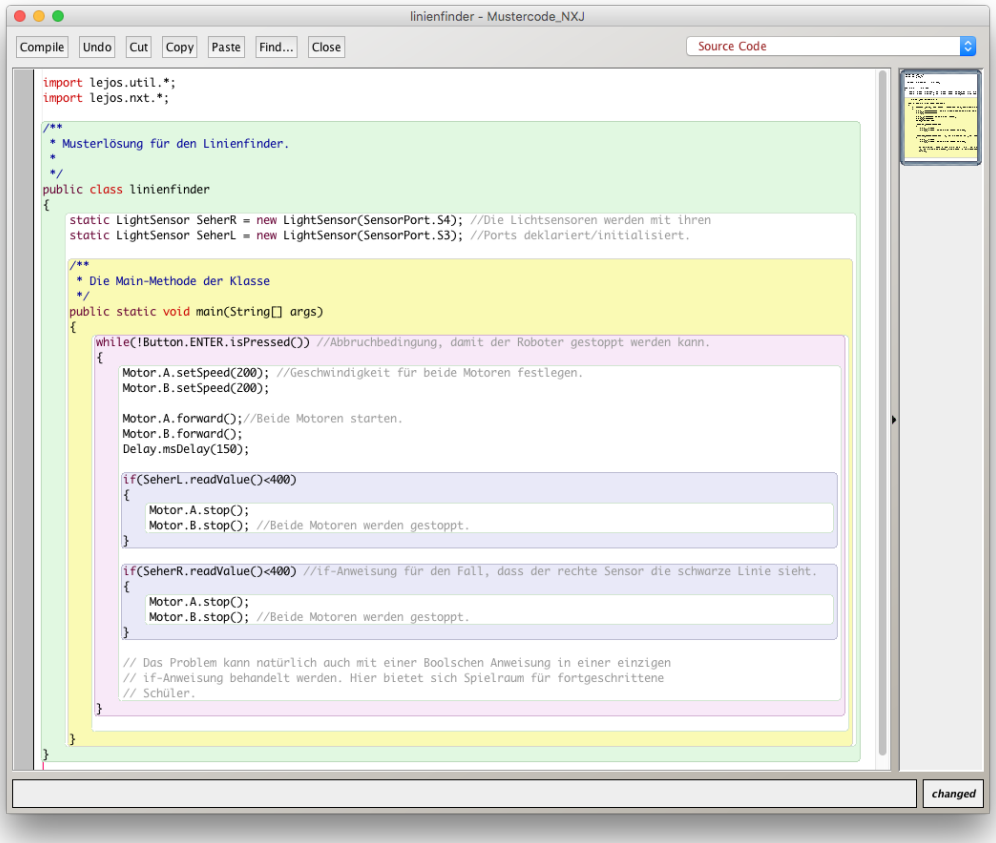

Abbildung 6.2: BlueJ-Beispiel zum Finden einer Linie

Literaturverzeichnis

[Abt15] Dietmar Abts. *Grundkurs JAVA. Von Grundlagen bis zu Datenbank- und Netzwerkanwendungen*, 8., überarbeitete und erweiterte Auflage, Springer Vieweg, Wiesbaden, 2015

[Aeb11] Hans Aebli. *Zwölf Grundformen des Lehrens. Eine Allgemeine Didaktik auf psychologischer GRundlage. Medien und Inhalte didaktischer Kommunikation, der Lernzyklus*, 14. Auflage, Klett-Cotta, Stuttgart, 2011

[Bar03] David J. Barnes, Michael Kölling. *Objektorientierte Programmierung mit Java. Eine praxisnahe Einführung mit BlueJ*, Übersetzt von Axel Schmolitzky, Pearson Studium, München, 2003

[Ehm09] Matthias Ehmann et al. *Duden Informatik - Sekundarstufe I / 9./10. Schuljahr - Objektorientierte Programmierung mit BlueJ*, Duden Schulbuchverlag Berlin Mannheim, 2009

[GyOhm16] Fachschaft Informatik. *Schulinternes Curriculum Informatik. Sekundarstufe II Wahlbereich und Profile*, Hamburg, Stand: 14.03.2016

[HH09] Behörde für Schule und Berufsbildung Hamburg (Hrsg.). *Informatik – Bildungsplan Gymnasiale Oberstufe*, Hamburg, 2009

[HH11] Behörde für Schule und Berufsbildung Hamburg (Hrsg.). *Informatik Wahlfplichtfach – Bildungsplan Gymnasium Sekundarstufe I*, Hamburg, 2011

[HH14] Behörde für Schule und Berufsbildung Hamburg (Hrsg.). *Informatik Wahlpflichtfach – Bildungsplan Stadtteilschule Jahrgangsstufen 7 – 11*, Hamburg, 2014

[Hub07] Peter Hubwieser. *Didaktik der Informatik*, 3. Auflage, Springer-Verlag Berlin Heidelberg, 2007

[Koe93] Koerber B., Peters I.R.: *Informatikunterricht und informationstechnische Grundbildung – ausgrenzen, abgrenzen oder integrieren?* In: Troitzsch, 1993, S. 108–115

Vornamen der Autoren

[Schwa07] Christine Schwarzer, Petra Buchwald. "Umlernen und Dazulernen.", In: Michael Göhlich, Christoph Wulf, Jörg Zirfas (Hrsg.): *Pädagogische Theorien des Lernens*, Beltz, Weinheim und Basel, S. 213–221, 2007

[Ull12] Christian Ullenboom. *Java ist auch eine Insel – Das umfassende Handbuch*, 10. Auflage, Galileo Press, Bonn, 2012

[Wag05] Oliver Wagner. "LEGO Roboter im Informatikunterricht. Eine Untersuchung zum Einsatz des LEGO-Mindstorms-Systems zur Steigerung des Kooperationsvermögens im Informatikunterricht eines Grundkurses (12. Jahrgang, 2. Lernjahr) der Otto-Nagel-Oberschule (Gymnasium)", *Schriftliche Prüfungsarbeit im Rahmen der zweiten Staatsprüfung für das Amt des Studienrats*, Berlin, 2005

Anhang A

Schleifenkonstrukte *OOP*

Schleifen

EINLEITUNG

Neben der Kontrollstruktur *Sequenz* gibt es noch die *Wiederholung*. Diese wird in Java durch *Schleifenkonstrukte* realisiert. Es gibt so genannte *Zählschleifen* und *bedingte Schleifen*.

ARBEITSAUFTRAG

1. Öffnet das Projekt *Iteration* und schaut euch die Klasse `Schleifendreher` an. Dort findet ihr Beispiele für die verschiedenen Schleifentypen in Java. Führt die Beispiele aus, um euch mit den Schleifen vertraut zu machen. Beachtet dabei die Ausgaben auf der Konsole.
2. Zeichnet ein Diagramm für den Ablauf von `for-`/`do-while-` und `while`-Schleifen und besprecht dieses mit eurem Sitznachbarn.
3. Schaut euch nun die Klasse Textanalyse des Projekts `Iteration` an. Dort gibt es eine vorgegebene Methode `istFrage(String text)`, die demonstriert, wie man die Länge eines Strings erhält und wie man auf einzelne Zeichen eines Strings zugreift. Probiert diese Methode interaktiv aus, indem ihr ein Exemplar von `TextAnalyse` erstellt und dann `istFrage` z.B. mit dem aktuellen Parameter „Wie geht's?“ aufruft.

   Worin unterscheiden sich die beiden Methoden `istFrage` und `istFrageKompakt`?
4. Schreibt nun eine eigene Methode `int zaehleVokale(String text)`, die für einen gegebenen Text als Ergebnis liefern soll, wie viele Vokale er enthält. Für den String „hallo“ soll die Methode beispielsweise eine 2 zurückgeben.

   Tipp: Verwendet in der Implementierung einen Schleifenzähler, der bei 0 beginnt und alle Positionen des Strings durchläuft. Erarbeitet euch ggf. die Funktionsweise einer `switch`-Anweisung, um die Aufgabe elegant zu lösen.

I

"Hiermit versichere ich an Eides statt, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe, die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht."

Hamburg, 8. März 2016 .....

Pamina Maria Berg