

Bericht zur Wissensvermittlung zum Tutorium der  
Lehrveranstaltung Objektorientierte Programmierung

Alexander Johr

u34584

m27007

---

Prüfer: Prof. Daniel Ackermann

Wernigerode, D-38855

31. August 2019

# Inhaltsverzeichnis

1	Report	4
1.1	Feedback der Studierenden . . . . .	4
1.2	Vorgehen . . . . .	5
1.3	Fazit und Ausblick . . . . .	6

# Abbildungsverzeichnis

1.1	Von interessierten Studierenden gefüllter Tutoriumsraum . . . . .	4
-----	---	---

# 1 Report

Die Wissensvermittlung im Tutorium für die Lehrveranstaltung Objektorientierte Programmierung verlief äußerst gut. Über das Semester kamen immer mehr und mehr Studenten in mein Tutorium. Auch Studierende, die ursprünglich einem anderen Tutor zugeteilt wurden, sahen sich mein Tutorium an und blieben dann bei mir (Abb. 1.1).



Abbildung 1.1: Von interessierten Studierenden gefüllter Tutoriumsraum

## 1.1 Feedback der Studierenden

Nach dem Tutorium holte ich mir Feedback von den Studierenden ein. Einige fanden das Tutorium so gut, dass sie sich wünschten, dass ich für sie noch mal einer Prüfungsvorbereitung anbiete. Bei Befragungen gaben die Studierenden an, dass sie das Tutorium für sehr gut befunden haben. Ich frage die Studierenden auch, ob aus ihrer Sicht noch Verbesserungspotenzial besteht. Die Studierenden waren der Meinung, dass es so, wie ich den Stoff anbiete, perfekt sein und ihnen keine Idee kämen, wie ich das Tutorium noch besser leiten könnte. Auch wenn mir diese Meinung schmeichelt, weiß ich um ein paar Punkte, die ich in Zukunft noch verbessern möchte.

In fast jedem Tutorium demonstrierte ich die Algorithmen, indem ich sie im Debugger vorführte. Ich zeigte den Studierenden den Umgang mit dem Debugger und setzte einige Mal sogar voraus, dass mir die Studierenden die Ergebnisse im Debugger zeigen. Erst wenn alle Studierenden den gleichen Wert in der Ausgabe demonstrieren konnten, kümmerten wir uns um die nächste Aufgabe. Die Studierenden gaben an, dass sie das als sehr positiv empfanden und sie mit dem Debugger im vorigen Semester wenig bis gar keine Erfahrung sammelten. Sie berichteten, nun das Gefühl zu haben, über den Debugger mehr Verständnis über den von ihnen geschriebenen Programmcode zu erlangen.

## 1.2 Vorgehen

Ich habe mir vor fast jedem Tutorium die Übungsaufgabe angesehen, welche die Studenten in dieser Woche bearbeiten müssen. Dann überlegte ich mir ein Minimalbeispiel, welches man in dem Tutorium schaffen kann.

Ich überlege mir die Übung aus mehreren Gründen:

Zum einen war es notwendig, sich die Aufgabe vorher anzusehen, um abzuschätzen, welche Grundlagen die Studenten für diese Aufgabe benötigen und welche die höchste Priorität haben.

Zum anderen, um sicher zu sein, dass während des Tutoriums keine Schwierigkeiten auftreten. Dafür programmiere ich die Übung zuvor selbst und überprüfe sie auf Lauffähigkeit und korrekte Ausgabe. Würde ich das nicht tun, ist die Wahrscheinlichkeit sehr groß, dass ich im Tutorium Fehler im eigenen Quellcode oder in meiner Entwicklungsumgebung finden muss. Aus privaten Gründen war ich einmal nicht in der Lage, die Übung vorzubereiten. Dann wurde mir auch klar, wie sinnvoll es war, die Übungen immer vorher zu testen. Denn während ich in dem Tutorium die Lösung für das Problem überlegte und die Studierenden mit mir gemeinsam entwickelten, fielen logische Fehler auf. Dadurch war es notwendig, Quellcode aus bereits geschriebenen Dateien zu modifizieren. Die Studierenden mussten das ebenso tun. Das führte zu Missverständnissen und unterschiedlichen, miteinander inkompatiblen, Versionen der Dateien. Für einige Studierenden war dann eine individuelle Betreuung notwendig, sodass ihr Quellcode wieder lauffähig war.

Dass alle Studierenden mitkommen, war mir auch besonders wichtig. Ich wollte, dass zu jedem Zeitpunkt alle Studierenden mitarbeiten können und niemand das Gefühl hat, abgehängt zu sein, da er nicht die nötige Erfahrung mitbringt. Daher fügte ich in regelmäßigen Abständen Momente ein, in denen ich mir die Ergebnisse aller Studierenden ansah. Gab es auch nur bei einem Studierenden Schwierigkeiten, die ihm das Weiterarbeiten behinderten, so lösten wir das Problem gemeinsam.

Einige Male befürchtete ich, dass das von den Studierenden im Minimalbeispiel gelernte Wissen nicht ausreichen könnte. Daher habe ich im Anschluss an das Tutorium angeboten, dass die Studierenden in einem weiteren Unterrichtsblock noch mehr Details erfahren können. Dies

wurde vom größten Teil der Studierenden angenommen. Lediglich Studierende, die terminlich gebunden waren, verließen das Tutorium.

### 1.3 Fazit und Ausblick

Ich programmierte alle Übungen mit den Studenten gemeinsam. Das von mir überlegte Minimalbeispiel konnten wir in fast allen Tutoriums Blöcken vollenden. Ich bin mir jedoch deutlich darüber im Klaren, dass diese Lernmethode einen Nachteil hat. Wenn ich Studierenden mit mir gemeinsam entwickeln, haben sie dabei die Möglichkeit, den aktuellen Code abzuschreiben, ohne darüber weiter nachzudenken. Dann lernen die Studierenden das Programmieren nicht, sie üben lediglich das Abschreiben von Code. Daher bezog ich das Publikum häufig mit ein, indem ich fragte, welchen Schritt wir wohl als Nächstes angehen sollten. Ich ließ mir dann von den Studierenden diktieren, was ich als Nächstes eingeben muss.

Mein Wunsch war es aber darüber hinaus, die Studenten häufiger Programmteile ganz selbstständig zu schreiben zu lassen. Durch die hohe Menge an Themen, die die Studenten bereits in diesem Semester lernen müssen, war mir das leider nicht möglich. Ich habe allerdings von Zeit zu Zeit die Studierenden gebeten, die letzte Teile des Quellcodes selbst zu schreiben und damit den Algorithmus zu vollenden.

Ein Großteil der Studenten schaffte das nicht, was mir zeigt, dass sie wohl schon im ersten Semester beim Erlernen der Grundlagen Schwierigkeiten hatten, hinterherzukommen. Fähigkeiten, die ich in meinem Tutorium lehrte und regelmäßig wiederholte, stellten für die Studierenden dagegen keine Schwierigkeiten dar. Daher wurde mein Wunsch umso größer, das Tutorium für Programmierung 1 im nächsten Semester zu übernehmen. So kann ich den gleichen Übungsstil auch bei den Grundlagen der Programmierung anwenden.