

# Übungen und Projektarbeit ohne Druck und Fristen

Erfahrungen aus der Lehrveranstaltung "Programming for Physicists" an der Uni Wien

Clara Kofler, Morris Weimerskirch, Manuel Längle

Jede\*r Studierende kennt es - Berge von Aufgaben, Zittern vor Präsentationen, Stress der droht, einen zu begraben. Dieser Härte-test ist so fest in unserem System verankert, dass man ihn leicht als Voraussetzung für Lernerfolg sehen kann: "No pain, no gain". Aber es muss doch möglich sein, eine Lehrveranstaltung so zu gestalten, dass weder der Lernerfolg noch das psychische Wohlbefinden der Studierenden beeinträchtigt wird. Hier stellen wir das System einer Programmierübung des Physikstudiums vor, bei der wir sowohl auf Deadlines während des Semesters, als auch auf Präsentationen der Studierenden verzichten, und gezielt versuchen, ein angenehmes Lernklima zu schaffen. Evaluationsdaten zeigen hauptsächlich positive Rückmeldungen im Bezug auf die oben genannten Punkte, jedoch führt das gewählte interaktive Übungskonzept auch zu Problemen, wie etwa zu langen Wartezeiten bei der Beantwortung von Studierendenfragen während der Übungseinheiten.

## 1 Einleitung

Programmieren ist einerseits Mittel zum Zweck: Datenaufnahme, -aufbereitung, -manipulation und -visualisierung setzen allesamt Programmierkenntnisse im breiteren Sinn voraus. Andererseits haben moderne Programmierkonzepte einen formativen Charakter: *separation of concerns* bildet den Unterschied zwischen der experimentellen Datenerfassung und der physikalischen Interpretation plastisch nach. Eine Dokumentation des Prozesses mittels Versionskontrollsystem (VCS) ist auch für eine gewissenhaft durchgeführte Forschung hilfreich. *Test Driven Development* - Hypothese, Antithese und überprüfbare Synthese sind als Begriffe auch prägend für den (Natur-) Wissenschaftsprozess an sich.

Die Einführung ins Programmieren sollte unserer Meinung nach nicht als theoretisches Konzept erfolgen. Gerade hier ist *learning-by-doing* beziehungsweise *learning-by-repeatedly-failing* ein wichtiger, wenn nicht sogar der wichtigste Aspekt des Lernprozesses. Wo es bei anderen Lehrveranstaltungen Sinn haben kann, die Theorie klar von den Anwendungen zu trennen - die Vorlesung von der Übung - sind andere Herangehensweisen für Programmierkurse eventuell sinnvoller. Hier wollen wir auf den Lehrveranstaltungsmodus des Moduls "Programming for Physicists" (P4P), das im dritten Semester des Physikstudiums an der Universität Wien vorgesehen ist und das Ziel hat, den Studierenden die Grundlagen des Programmierens beizubringen, eingehen.

## 2 Aufbau der Lehrveranstaltung

Der Kurs wird seit 2019 als Vorlesung mit prüfungsimmanenten Übung angeboten, die Leistungskontrolle erfolgt über eine Modulprüfung welche beide Bereiche abdeckt. Die Vorlesung setzt hierbei auf Demonstrationen und Beispiele, wobei die Studierenden explizit aufgefordert sind, den Beispielen am Gerät zu folgen. Der gesamte Vorlesungstoff und die Übungsaufgaben sind auf einer eigenen Website einsehbar. Ein Screenshot dieser ist in Abb. 1 zu sehen.

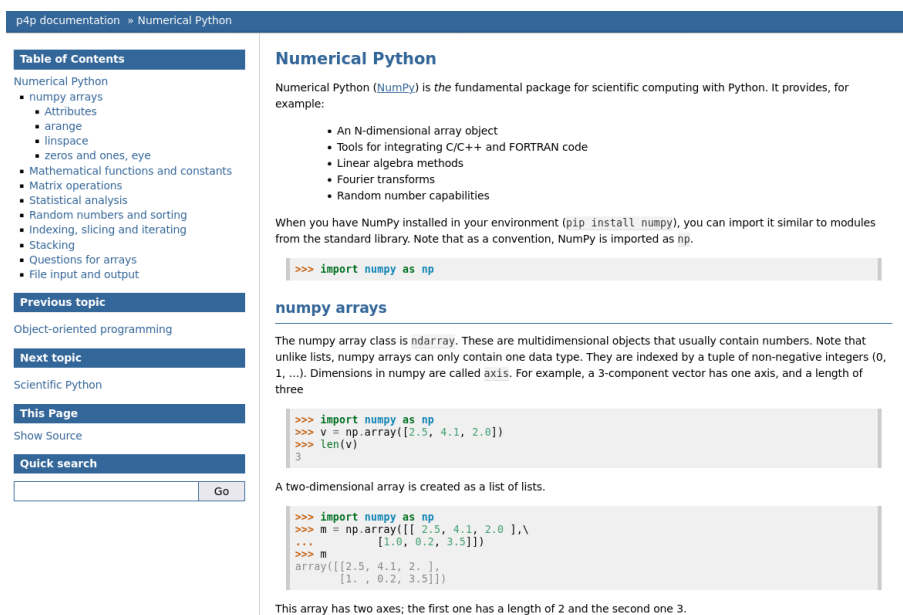


Abbildung 1: Ein Screenshot der P4P Kurswebsite, auf der alle Vorlesungsunterlagen und Übungsaufgaben gesammelt sind.

Auch wenn die Veröffentlichung der Übungen dem Themenzeitplan der Vorlesung folgt, wird auf Abgabetermine bis Ende des Semesters verzichtet. Dies soll allen Studierenden eine individuelle Aufgabenteilung und eigenes Arbeitstempo ermöglichen. Die Übungseinheiten bieten Raum für die Studierenden, um gemeinsam an ihren Lösungen zu arbeiten und den Lehrenden Fragen zu stellen. Die Lehrenden agieren hier als Coaches, die den Studierenden helfen sollen, ihre Probleme zu lösen, anstatt sie auf Basis selbiger schlechter zu beurteilen.

Zusätzlich gibt es ein wöchentliches Tutorium, das von zwei Studierenden begleitet wird. Auch dieses ist ungewöhnlich gestaltet, da, anstatt genauer auf die Übungen einzugehen, Konzepte aus der Lehrveranstaltung erneut anhand von Beispielen erklärt und zusätzliche, ähnliche jedoch kürzere Aufgaben samt Lösungen zur Verfügung gestellt werden.

Als Programmiersprache wurde Python3 gewählt, eine *all rounder* Sprache, die sehr weit verbreitet und beliebt ist. Jedoch hebt sich dieser Kurs vor allem durch die Wahl der Programmierumgebung von anderen ab. Anstatt sich ausschließlich auf das Erlernen von Python zu konzentrieren, ist das Ziel, auch Fähigkeiten im Umgang mit *open source* Werkzeugen zu erlernen, die im akademischen Rahmen viel verwendet werden. So wird während des gesamten Kurses im Linux Terminal (*Windows Subsystem for Linux* für Windows User) gearbeitet, die Übungen werden auf einen Server geladen und können über das *version control system git* heruntergeladen werden. Auf ihren eigenen Rechnern im Terminal bearbeiten Studierende die Dateien mit dem Texteditor *neovim* und pushen ihre Lösungen über *git* wieder auf den Server. Unter einem *push* wird in *git* das Hochladen auf

den Server verstanden. Zusätzlich wird die Jupyter Konsole in *neovim* verwendet, um den Code zu testen und sich mit einzelnen Funktionen auseinanderzusetzen. Studierenden wird die Programmierumgebungen freigestellt, jedoch wird hier nur bedingt Support garantiert.

### 3 Übungsmodus im Detail

Die Übungsaufgaben sind spielerisch aufgebaut. Jede Aufgabe besteht aus mehreren Levels, die unterschiedliche Anweisungen beinhalten. Nach Bearbeitung der Aufgaben eines Levels, beispielsweise dem Implementieren einer bestimmten Funktion, werden die Änderungen über *git* registriert und auf den Server gepushed, woraufhin mit dem nächsten Level weitergemacht wird. Meist gibt es vorgefertigte Dateien mit einer vorgegebenen Programmstruktur, die mit dem entsprechenden Code vervollständigt werden muss, und eine Reihe von Testdateien, um die Programme zu überprüfen. Bei jedem neuen *push* auf den Server werden ebendiese Tests mittels *pytest* für eine automatische Evaluation auf Serverseite verwendet und den Studierenden der aktuellen Status mitgeteilt. Es gibt keine Beschränkung für die Anzahl der hochgeladenen Lösungen, am Semesterende wird die aktuellste beurteilt.

Im letzten Drittel des Kurses finden sich die Studierenden in Kleingruppen zusammen und implementieren von Grund auf gemeinsam ein von den Übungsleiter:innen entworfenes Softwareprojekt. Diese Projekte können von Datenanalyse über Simulationen von Massesystemen bis hin zu einfachen Spielen und Animationen viele unterschiedliche Anwendungen von Programmen beinhalten. Anders als die Übungen werden diese Projekte von Hand kontrolliert und nicht über vorprogrammierte Tests. Während der Umsetzung der Projekte stehen die Studierenden in Kontakt mit den Lehrenden, die die Projekte entworfen haben. Wir haben durchaus positive Rückmeldungen von Studierenden bekommen, die das Projekt abgeschlossen haben, und zufrieden sowie stolz auf das Geleistete waren.

### 4 Rückmeldung der Studierenden

Gegen Ende des Kurses wurde eine anonyme Lehrveranstaltungsevaluation durchgeführt. Die Rückmeldungen zur Vorlesung waren vorrangig positiv in den vorgegebenen Fragen, jedoch gaben mehr als 40% von insgesamt 72 Studierenden an, dass die Lehrveranstaltung mehr Arbeitsaufwand erforderte, als durch die ECTS angedeutet wurde und dass die Voraussetzungen zu hoch angesetzt waren. Der Hauptkritikpunkt der Studierenden in der offenen Evaluation war das verwendete Setup. Es wurde bemängelt, dass viel Zeit benötigt wurde, um damit verbundene technische Probleme zu beheben und die Syntax zu erlernen.

Die offenen Antworten bei einer von Seiten der Universität durchgeführten Übungsevaluation waren sehr ausgeglichen, jedoch zeigte sich, dass vor allem das Verzichten auf eine Deadline den Studierenden geholfen hat, sich ihre Zeit besser einzuteilen. Auch das Fehlen des Präsentationsaspekts wurde meist positiv aufgefasst, da es den Stress einer Übung reduziert. Allerdings gab es auch Studierende, die der Meinung waren, dass Präsentationen ihr Verständnis verbessern würden. Laut Evaluation profitierten die Studierenden am meisten von der Bearbeitung der Übungsaufgaben und dem Gruppenprojekt am Ende des Semesters. Ein weiterer Punkt der oft positiv genannt wurde, in diesem wie auch in vergangenen Jahren, war die geförderte Teamarbeit unter den Studierenden, sowohl bei den Gruppenprojekten als auch bei Bearbeitung der Übungsaufgaben, und die entspannte Umgangsform zwischen den Lehrenden und den Studierenden.

Die Studierenden bemängelten an diesem Übungskonzept jedoch, dass es oft lange Wartezeiten gab, bis sie Unterstützung bekamen, vor allem als zu Beginn des Semesters viele

Fragen aufgekommen sind.

Aus den Fragen der Evaluation wurden die sechs relevantesten genauer betrachtet: (1) Gesamt gesehen halte ich die Lehrveranstaltung für ...; (2) Die Studierenden werden bei der Erreichung der Lernziele unterstützt; (3) Die Studierenden werden in der Entwicklung ihrer Vortragstechnik unterstützt; (4) Es wird darauf geachtet, dass sich alle Studierenden aktiv an der Lehrveranstaltung beteiligen; (5) Durch die Lehrveranstaltung habe ich ein tieferes Verständnis der grundlegenden Konzepte gewonnen; (6) Die gewählte Durchführungsform (Präsenz / hybrid / digital) funktioniert reibungslos.

Dabei konnten die Studierenden aus fünf Antworten von “sehr gut/trifft zu” (1) bis “sehr schlecht/trifft nicht zu” (5) auswählen. Das Ergebnis dieser Befragung ist in Abb. 2 visualisiert.

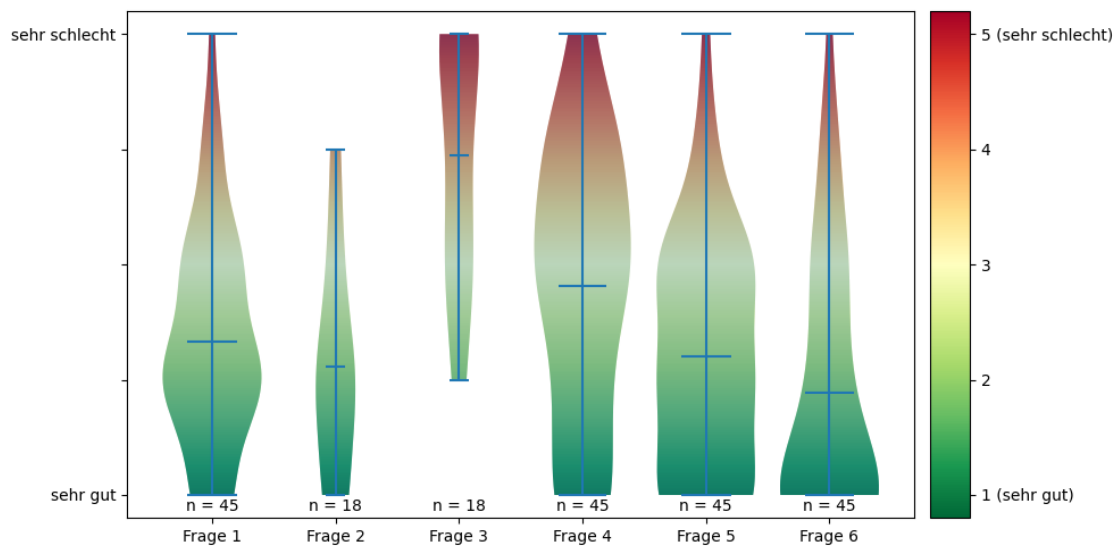


Abbildung 2: Violinengrafik der Studierendenbefragung zur Übung, wobei die Mittelwerte mit horizontalen blauen Linien gekennzeichnet sind. Deren Länge ist an die Graphik angepasst und hat keine Aussagekraft. Die Breite der Violinengrafik gibt Aufschluss über die Anzahl der Stimmen für die entsprechende Bewertung. Je tiefer der Schwerpunkt der Violine desto besser die Bewertung der entsprechenden Frage. Die Anzahl abgegebener Stimmen ist mit n angemerkt.

Wie in der Visualisierung ersichtlich, ist die Bewertung der Studierenden in den meisten Punkten positiv. Vor allem die Frage 6, also wie gut die Durchführungsform funktioniert, wurde mit einem Mittelwert von 1.89 und einem Median von 1 als sehr gut bewertet. Sowohl der Erwerb von grundlegendem Verständnis (Frage 5) als auch die Erreichung von Lernzielen (Frage 2) wurde positiv bewertet. Einzig die Unterstützung bei aktiver Beteiligung der Studierenden an den Übungseinheiten (Frage 4) und die Entwicklung der Vortragstechnik (Frage 3) fielen bei den ausgewählten Fragen mittelmäßig bzw. schlecht aus. Dieses Ergebnis deckt sich auch mit den offenen Fragen (wie zuvor besprochen) und resultiert aus dem Fehlen der Präsentationen in der Übung.

Gesamt wurde die Lehrveranstaltung mit einem Mittelwert von 2.3 und einem Median von 2 als gut bewertet, was auch durch die Ergebnisse von Frage 1 ersichtlich ist.

In Abb. 3 ist die Übungsaktivität (Änderungen der Studierenden an ihren hochgeladenen Lösungen) des Kurses im Jahr 2020, welcher im selben Modus abgehalten wurde,

gezeigt. In der Darstellung sind die Anteile an den Gesamtänderungen an jedem Tag des Semesters in Prozent farblich dargestellt. Die höchste Aktivität wurde dabei kurz vor Ende des Semesters verzeichnet. Dieses Wochenende vor der Deadline machte etwa 10% der Gesamtaktivität aus. Dennoch sieht man, dass mit Ausnahme der Weihnachtswoche (Woche 9) trotz des Verzichts auf Deadlines während des Semesters relativ kontinuierlich an den Übungen und Projekten gearbeitet wurde.

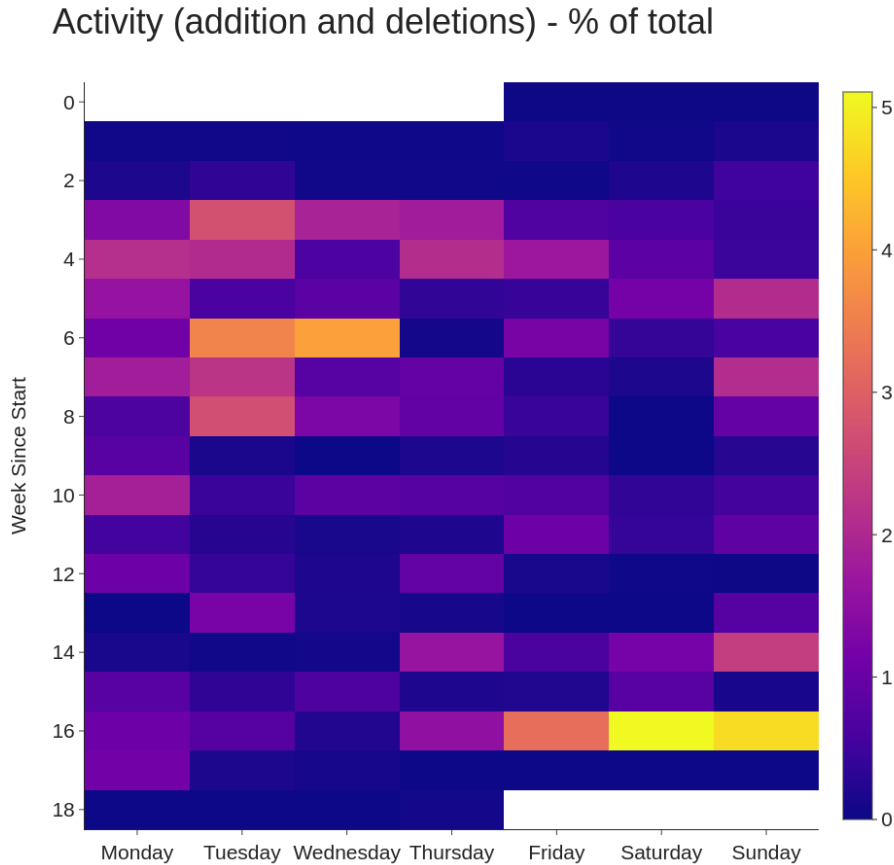


Abbildung 3: Aktivität der Studierenden während des Semesters, gemessen an Veränderungen ihrer hochgeladenen Übungslösungen als Anteil der gesamten Änderungen pro Tag in Prozent.

## 5 Prüfungsergebnisse

Die Prüfung am Ende besteht darin, dass Studierende in ausreichend Zeit ein einfaches Programm selbstständig schreiben. Die Korrelation der Übungsnoten mit den Gesamtnoten des Moduls ist in Abb. 4 dargestellt. Dabei ist zu Beachten, dass bei einer Beurteilung der Übung mit *sehr gut* (1) die Prüfungsnote des ersten angebotenen Prüfungsantritts um einen Notengrad verbessert wird, vorausgesetzt die Prüfung wurde positiv abgeschlossen.

Es ist ersichtlich dass jene Studierenden, welche die Übung mit *sehr gut* bestanden haben, zum Großteil auch das Modul mit guten Noten abgeschlossen haben, wohingegen schlechtere Übungsnoten quasi keine Auswirkung auf die erreichte Modulprüfungsnote hatten. Der schlechteste Schnitt von 4.3 wurden bei der Übungsnote 3 verzeichnet. Das

könnte daran liegen, dass einige Studierende, die schon Programmiererfahrung haben, die Übungen nicht besuchen, jedoch trotzdem die Prüfung bestehen, wohingegen Studierende, die keine Programmiererfahrung haben, aber die Übungen nicht oft besuchen oder nicht alle Übungsaufgaben bearbeiten, bzw. das Projekt nicht implementieren, die Übung zwar bestehen, jedoch nicht genug Praxis erworben haben, um die Prüfung positiv abzuschließen.

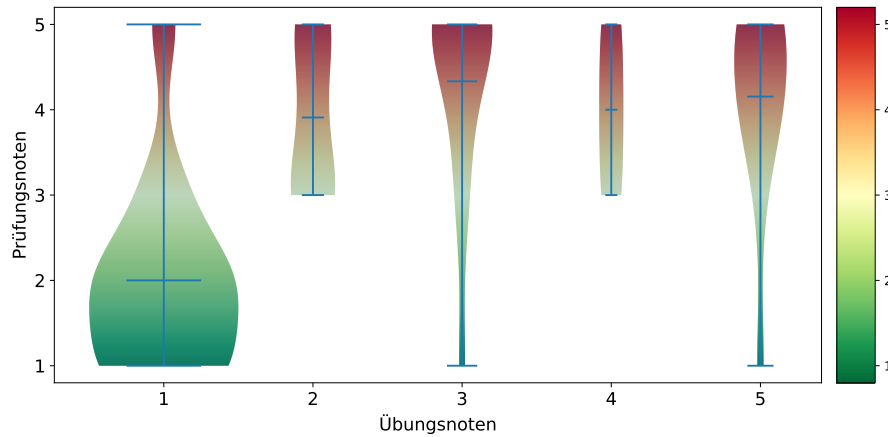


Abbildung 4: Übersicht der erreichten Prüfungsnoten im Zusammenhang mit den Übungsnoten des Kurses. Die Mittelwerte sind durch horizontale blaue Linien gekennzeichnet.

Es ist zu beachten, dass bei vergangenen Jahrgängen mit demselben System die Notenvergabe der Prüfung deutlich besser ausfiel. Etwa 60 % der Studierenden schlossen mit einem *sehr gut* ab.

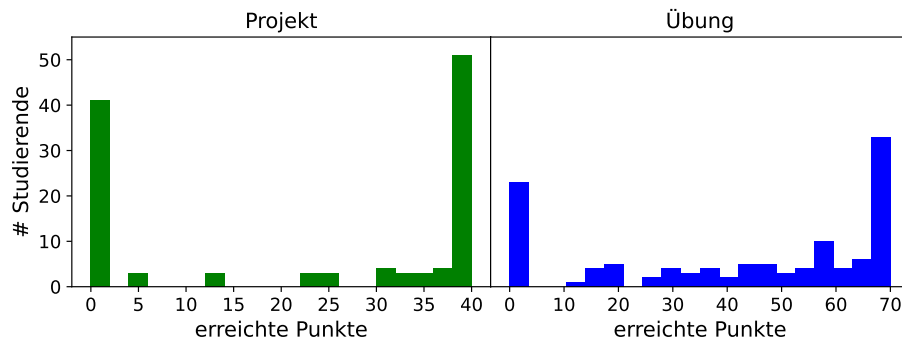


Abbildung 5: Histogramm, welches den Erfolg von Studierenden beim Projekt sowie bei den Übungsbeispielen zeigt. Sowohl bei dem Projekt als auch bei der Übung hat der Großteil der Studierenden entweder 0 oder fast alle Punkte erreicht.

## 6 Fazit

Alles in allem wurde der Übungsmodus größtenteils positiv aufgefasst. Vorrangig wurde der verminderte Druck durch das Verzichten auf eine Deadline für Übungsabgaben während des Semesters und der angenehme Umgang miteinander als auch die Zusammenarbeit im Zuge von Gruppenarbeiten gelobt. Die negativen Bewertungen bezogen sich hauptsächlich

auf die Wahl der Programmierumgebung und die Menge des Stoffs, jedoch trafen wir auch auf Probleme mit den interaktiven Übungen, da Studierende oft lange auf Beantwortung ihrer Fragen warten mussten. Obwohl der Kurs noch einige Schwierigkeiten aufweist, ist es uns dennoch gelungen, einen Übungsmodus einzuführen, der das psychische Wohl der Studierenden priorisiert - ein Aspekten der ebenso wichtig ist wie deren Lernerfolg.

## **Acknowledgement**

Beiträge der Autor:innen: Das Manuskript wurde in Zusammenarbeit aller Autor:innen erstellt. M.J. hat die Übungsbeispiele sowie das didaktische Konzept (mit-)entwickelt und war als technischer Support tätig. M.L. war an der Testung und Weiterentwicklung der Beispiele beteiligt sowie ein Übungsleiter. C.K. war als Tutorin tätig.

Wir möchten uns bei Jani Kotakoski, dem Leiter der Lehrveranstaltung für die Entwicklung des didaktischen Konzepts sowie aller Lehrveranstaltungs-materialien bedanken.

Wir möchten uns bei den Übungsleitern der anderen Gruppen Carsten Speckmann, Wael Joudi, Jacob Madsen und Arixin Bo bedanken. Sie waren an der Testung und Weiterentwicklung der Übungsbeispiele beteiligt. Ohne permanenten Austausch, sowie gemeinsame Evaluation der Lehrveranstaltung wäre diese Publikation nicht möglich gewesen.

Besonderer Dank gebührt auch dem technischen Support der Lehrveranstaltung, Maximilian Negedly, der den Großteil der technischen Probleme der Studierenden gelöst hat und es somit dem Rest des Teams erleichtert hat sich auf die Vermittlung des Lehrveranstaltungsinhalts zu konzentrieren.

Wir danken allen zuvor genannten Personen für ihre hilfreichen Kommentare und Anregungen im Bezug auf diese Publikation.