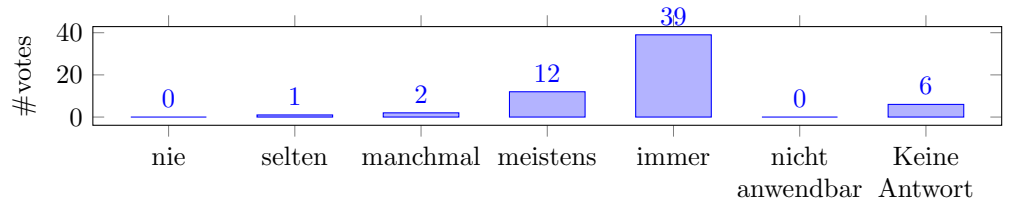


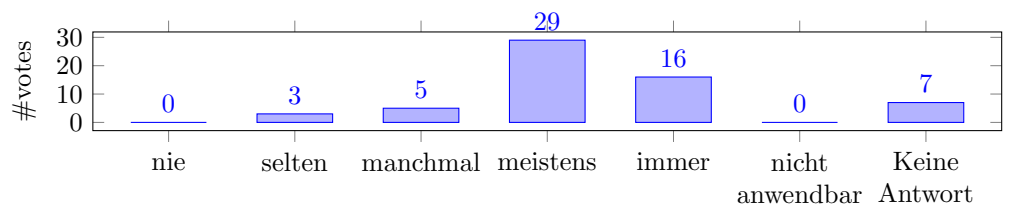
Ergebnis der Online-VLU. Die Umfrage fand in den letzten beiden Vorlesungswochen statt.

## 1 Bewertung der Vorlesung

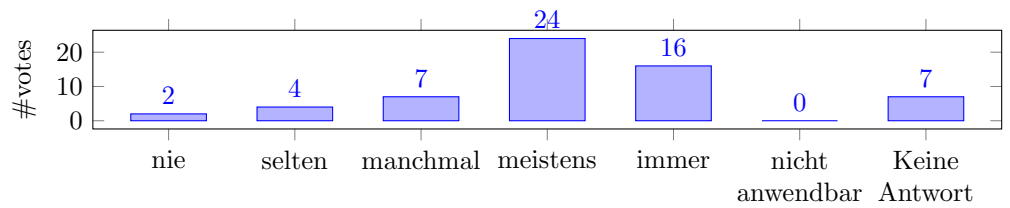
Wie oft hast du die Vorlesung besucht?



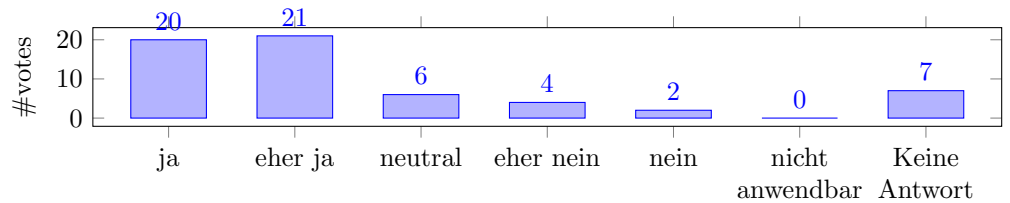
Wurden Themen durch Beispiele veranschaulicht?



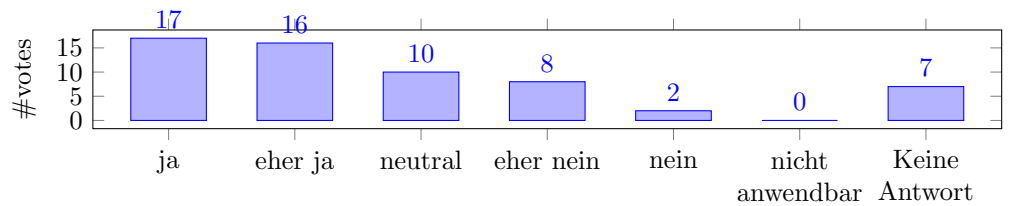
Wurden die Themen ausführlich genug erklärt?



War die Struktur der Vorlesung klar zu erkennen?

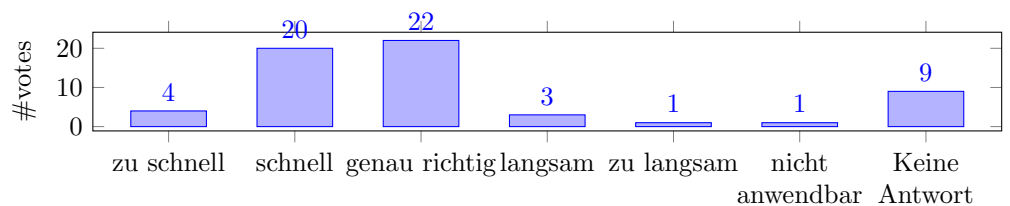


Waren die Folien/das Skript hilfreich?

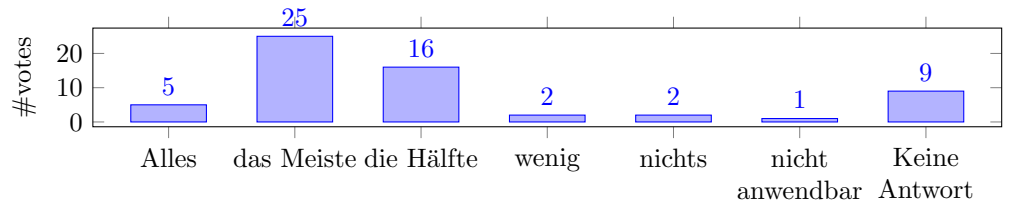


## 2 Bewertung der Dozierenden

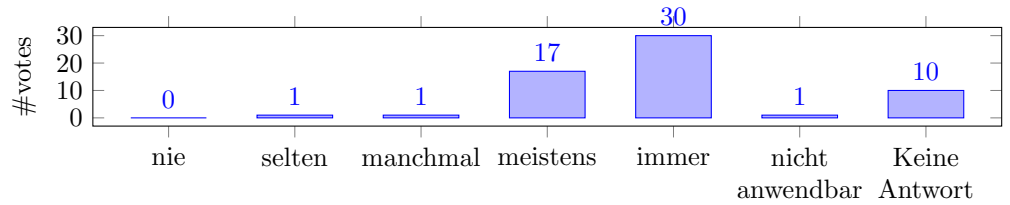
Die Geschwindigkeit der Vorlesung war...



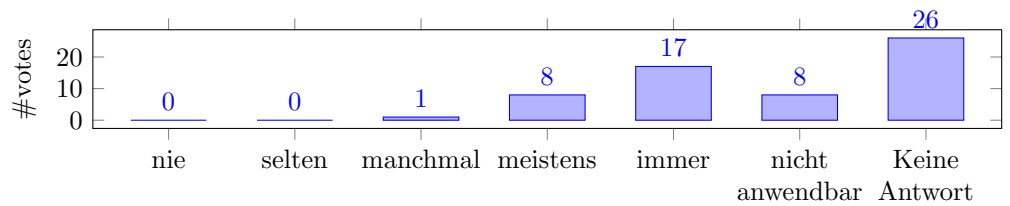
Wie viel verstehst du während der Vorlesung?



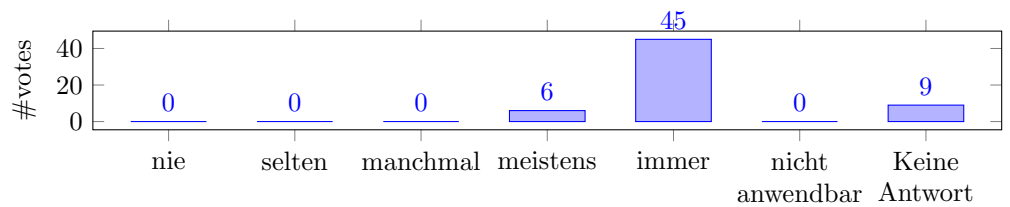
Ist der Dozent/die Dozentin gut auf Fragen eingegangen?



War der Dozent/die Dozentin außerhalb der Vorlesung für Fragen etc. erreichbar?

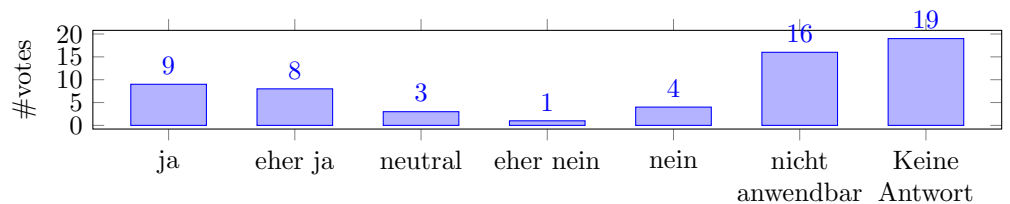


War die Dozentin / der Dozent akustisch gut zu verstehen?

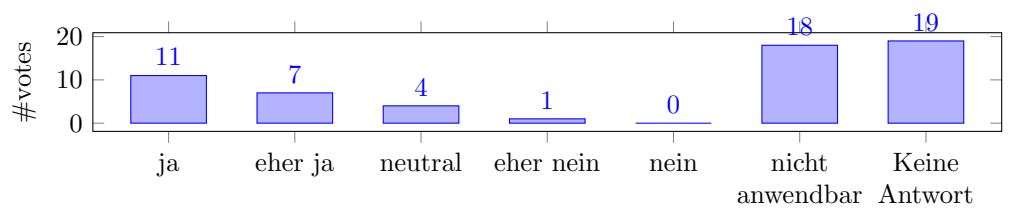


### 3 Bewertung des Moduls

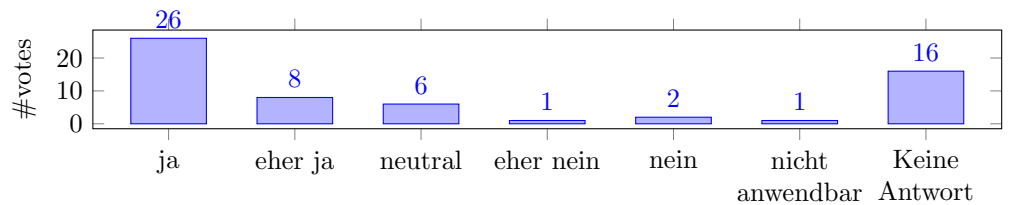
Helfen die verlangten Studienleistungen, das Modul erfolgreich abzuschließen?



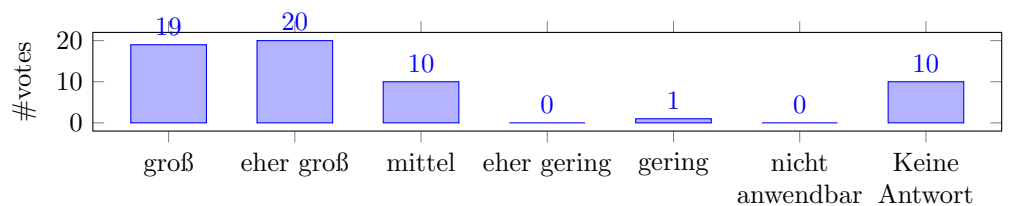
Findest du die verlangten Studienleistungen für dieses Modul angemessen?



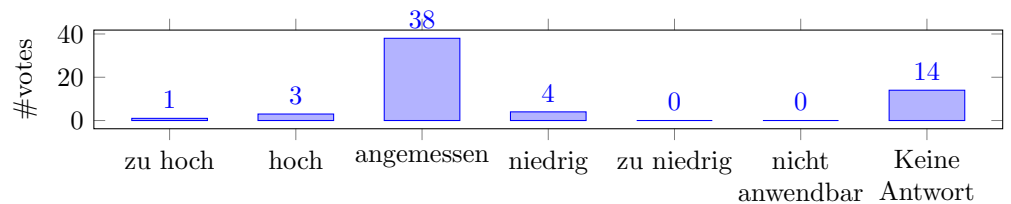
Würdest du das Modul weiterempfehlen?



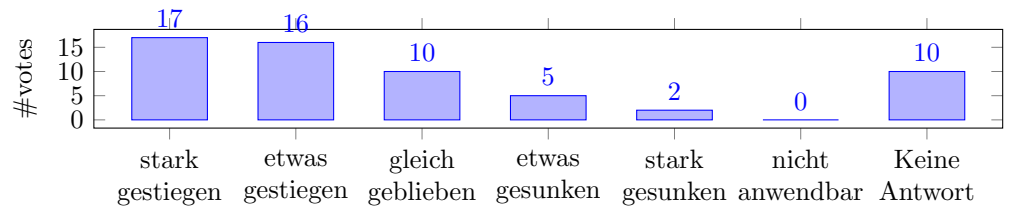
Der Praxisbezug war...



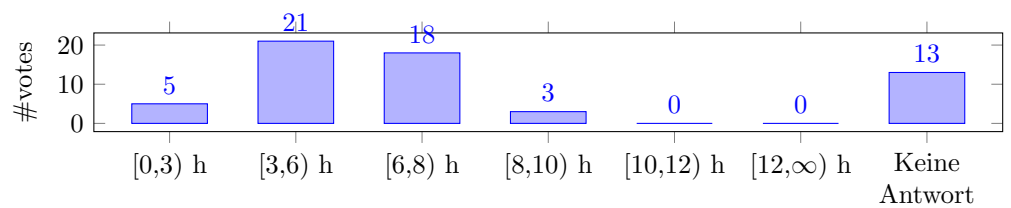
Ist der Arbeitsaufwand für dieses Modul im Hinblick auf die LP-Zahl angemessen?



Dein Interesse für dieses Thema ist...

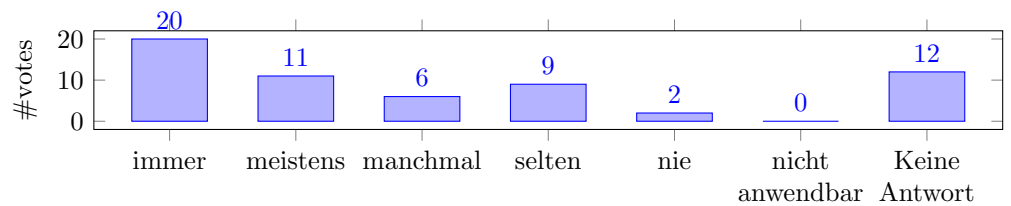


Wie viele Stunden hast du insgesamt, inkl. Vorlesung, Übung, Übungsaufgaben..., pro Woche für dieses Modul aufgewendet?

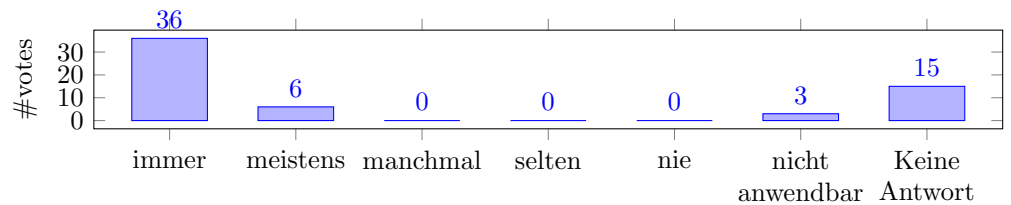


## 4 Bewertung der Übungsaufgaben

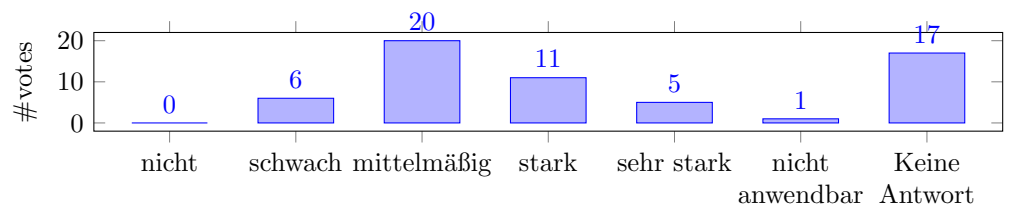
Wie oft hast du die Übungen besucht?



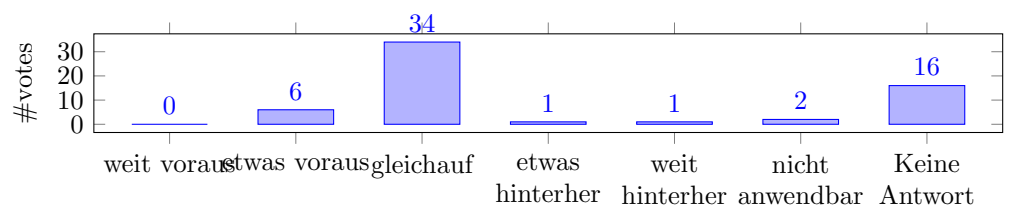
Wurden die Übungsaufgaben rechtzeitig zur Verfügung gestellt?



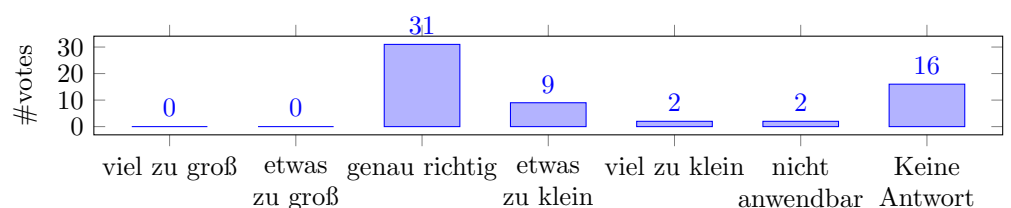
Die Schwierigkeit der Übungsblätter schwankte...



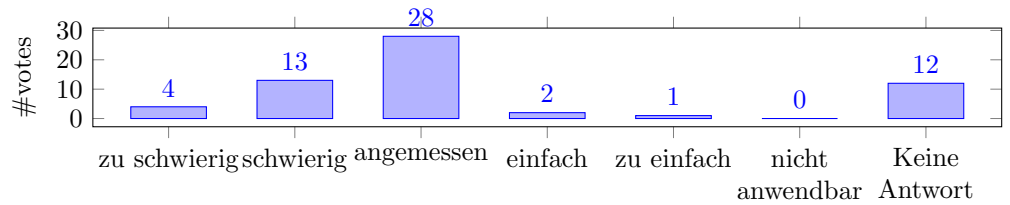
Die Vorlesung war...



Die Übungsgruppe war...

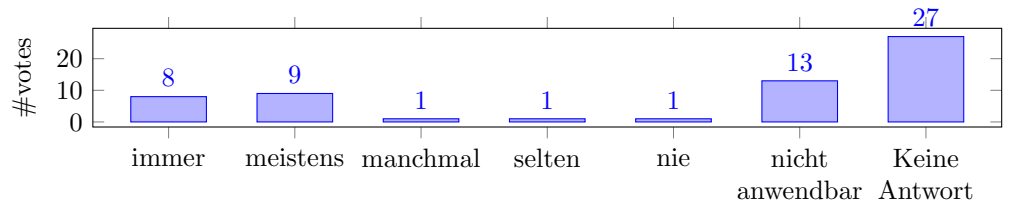


Die Übungsaufgaben waren meistens...

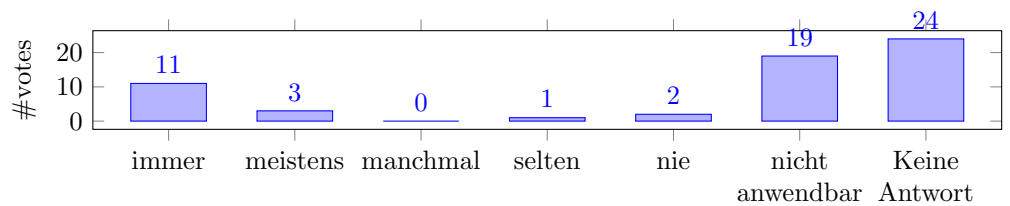


## 5 Bewertung des Tutoriums

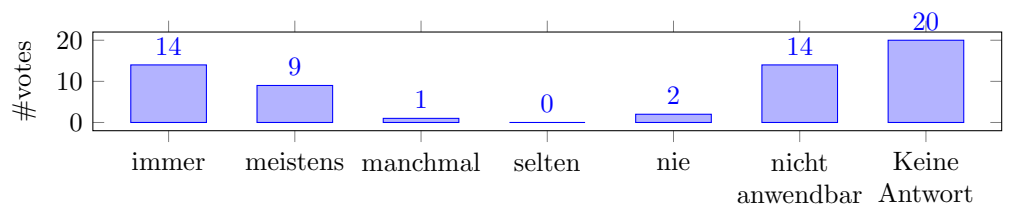
War der Tutor/die Tutorin außerhalb der Übung für Fragen etc. erreichbar?



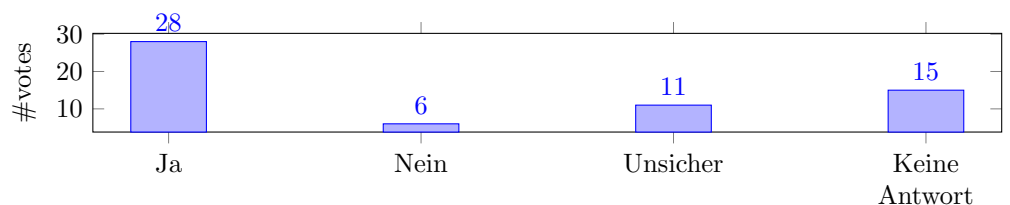
Waren die Korrekturen des Tutors/der Tutorin nachvollziehbar?



Wurde der Tutor/die Tutorin mit dem Stoff der Übung fertig?

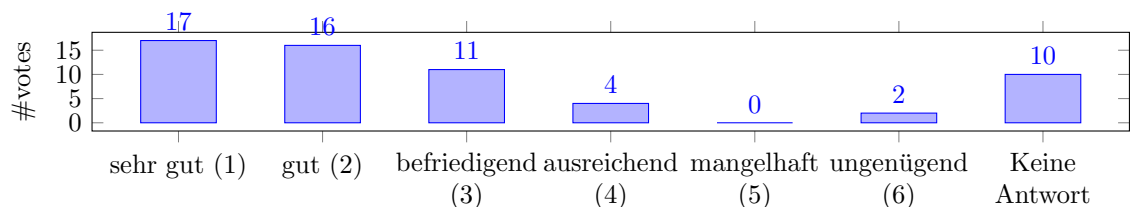


Lohnt sich der Besuch des Tutoriums?



## 6 Abschließende Bewertung des Moduls

Note:



## 7 Freitextkommentare

### 7.1 Was hat dir an dieser Lehrveranstaltung gefallen?

Artemis ist sehr cool

der Praxisbezug und die vielen live beispiele, die lernbetreuung, die vielen übungsaufgaben zum üben

Mir haben die Dozenten gefallen, ihr wart sehr sympatisch. Ich mochte dass ihr hin und wieder projekte gezeigt habt wie den Taschenrechner.

Der Einsatz von Artemis für die Übungen war eine gute Idee.  
Wenigstens Alex hatte ein einigermaßen Ahnung.  
Die Benennung des Moduls war angemessen ehrlich (das "Denken" aus dem Titel von AdiP wurde gestrichen).

Der hohe Bezug zur tatsächlichen Programmiertätigkeit, die wöchentlichen Aufgaben über Artemis.

Die Dozierenden waren immer gut zu erreichen. Die Vorlesung gleichzeitig auf YT zu streamen und da Zugang zu haben war eine sehr gute Lösung.

Ich hatte das Gefühl, dass dem Dozenten wichtig war, die behandelten Themen nachvollziehbar und anfängerfreundlich zu vermitteln. Gleichzeitig wurden interessante fachbezogene Themen und Methoden angerissen, die einen möglichen Ausblick auf das weitere Studium boten. Das Feedback der Hörer wurde regelmäßig eingeholt und konstruktiv umgesetzt.

Die Aufgaben "Discord-Bot" und "Simple TCP Chat" haben mir gut gefallen.

-Die Artemis-Übungsaufgaben fand im Großen und Ganzen ich extrem gut. Zwar war das Feedback der Tests nicht immer zu 100% hilfreich, aber zumindest etwas Feedback zu Programmier-Übungsaufgaben zu bekommen fand ich sehr gut. So konnte man das Anwenden des Stoffes gut üben, auch wenn einiges in der VL nicht angesprochen wurde,  
-Kontakt zu Dozenten/Tutoren sehr einfach und schnell, z.B. Fragen auf Artemis schnell und gut beantwortet, Lernbetreuung auch super Sache, man hatte das Gefühl dass die Dozenten trotz Corona usw. recht nah dran an den Studenten waren.

Engagement der Dozenten, Bereitschaft auf Fragen, Meinungen und Wünsche der Studierenden einzugehen, lockerer Umgangston

Praktische Übungen, nicht Lösungen besprechen

der Dozent ist recht jung und wir haben die Übungsaufgaben wie im echten Leben machen können: man kann googlen, hat eine IDE zur Hilfe

++ Artemis ++ Dozenten haben sehr guten Kontakt zu uns Studenten. Bei Problemen sofort zur Stelle und sogar auf dem Discord Channel aktiv

Sehr kompetente Dozenten, gute Überarbeitung des Moduls (vorher grauenhaft), super toller Praxisbezug

Nutzung von Artemis und Praxisübungen; Erreichbarkeit des Dozenten; Hochladen der Videos über Youtube; der Discord-Bot-Contest; Fragestunden donnerstags.

Mir hat am besten gefallen, dass die erklärten Dinge immer sofort in Beispielen im Code demonstriert wurden. Die Folien waren gut strukturiert. In Artemis wurden fragen zu Aufgaben immer sehr gut beantwortet. Viele interessante Dinge wurden behandelt. Den Discord Bot Contest fand ich auch sehr anregend und auch sinnvoll um das Interesse der Studenten zu wecken. Mir gefällt dass es ein Dokument gibt wo alle Dinge aufgelistet sind die man können muss, das erleichtert das Lernen mit externen Medien.

sympathische Lerhkräfte

Häufige Veranschaulichung durch Beispiele

Live-Coding

sehr guter Überblick über alles was man mit Java machen kann

## 7.2 Was könnte noch besser gemacht werden?

Die Tutorien könnten Pflicht sein, so kommt keiner

Bei C am Ende hat man gemerkt, dass es nicht eure Hauptprogrammiersprache ist und nicht die tiefen Einblicke und potentielle Fehlerquellen erwähnt habt. Besonders die Vorlesung am 31.1 hat mir nicht gefallen, da ihr meines Erachtens in ein viel zu komplexes Beispiel eingestiegen seid, bei dem schon ziemlich viel vor programmiert war. So hatte man selbst nicht so richtig die Zeit das Programm Stück für Stück nachzuvollziehen, vorallem, da man viele wichtigen Schlüsselwörter nicht kannte. Noch dazu war es verwirrend, dass viele Beispiele nicht funktioniert haben und ihr teilweise widersprüchliche Aussagen gesagt haben (pointer++ erhöht immer um Elementgröße, was Alexander am Mittwoch davor schon richtig erklärt hatte). Ich hatte allgemein den Eindruck, dass Alexander am besten erklären konnte und am meisten Ahnung hatte, vielleicht kann er ja nächstes Jahr mehr machen. Alles in Allem sind das aber nur kleine Kritikpunkte, das Modul hat mir sehr gefallen und war dieses Semester mein Lieblingsmodul.

Mir hat nicht gut gefallen, dass es häufig vorgekommen ist, dass Sachen auf den Folien beispielsweise Code nicht richtig war und in der Vorlesung korrigiert wurde. Ich hätte mir stark gewünscht wenn jemand vorher nochmal drüber geguckt hätte. Hätte mich auch gefreut, wenn der C Teil von jemandem gezeigt wurde, der das tagtäglich benutzt. Ansonsten gut!

Leute die Vorlesung halten lassen, die die Materie auch verstehen. Z. B. die, die man auch für die Vorkurse nimmt. Die können das.

Etwas durchstrukturiertere Vorlesung wäre wünschenswert, Live-Coding generell unübersichtlich oder relativ träge, es wäre schön, wenn manche der "Prompts" und Aufgabenstellungen der Artemis-Aufgaben allgemein noch etwas konsistenter und "eindeutiger" wären.

Im Prinzip 2 Dozierende zu haben von denen einer ständig rhetorische Fragen stellt war für mich so verwirrend, dass ich diese Vorlesungsteile oft mehrmals schauen musste um zu verstehen was gemeint ist. Nur ein Dozierender wäre sehr viel weniger verwirrend.

Gegen Ende des Semesters ist man dazu übergegangen die Vorlesung mit Live-Coding zu beginnen und das Vorgeführte im Anschluss zu erläutern und mit Folien zu ergänzen. Das hat mir persönlich mehr zugesagt, als die Vorlesung theoretischer zu gestalten und am Ende ein Beispiel einzubringen. Mir ist es so schwierig gefallen, der VL aufmerksam zu folgen.

-Folien manchmal wenig hilfreich, da sie ohne Erläuterung schwierig nachzuvollziehen waren. Sind eher hilfreich, wenn man schon relativ genau weiß, worum es geht.  
-Auch nicht so toll, wenn bspw. in der Vorlesung zu TCP/UDP erst angekündigt wird, das sei nicht klausurrelevant, um das dann ein paar Vorlesungen später zu revidieren. Nachträgliche Änderungen am Google-Doc gehen in die gleiche Ecke.  
-Im Tutorium war es bei uns häufiger so, dass wenn wir dem Tutor eine Frage stellten, dieser zunächst die Aufgabe gar nicht kannte und sich erstmal eine ganze Zeit in die Aufgabe und Musterlösung einarbeiten musste.

Die Folien könnten mehr Infos beinhalten, Mehr Graphiken, Mehr Erklärungstext,

Mehr Infos über die Prüfung.

der Stoff war gerade am Anfang sehr viel und danach kam nur gefühlt Java spezifische Themen. Bevor wir überhaupt die Basics richtig drauf hatten, wurden schon von komplizierten Dingen geredet

– In meinem Tutorium waren nie mehr als zwei Leute... Das wird überwiegend an der Uhrzeit liegen, aber das Tutorium war auch nicht sehr interaktiv. Anregen, dass man über Vorlesungsstoff oder ähnliches redet.

- Es gab stellenweise Aufgaben, die zu schwer waren. Beispiel Generics. Diese Aufgabe hätte man vielleicht etwas einfacher und weniger umfangreich gestalten können

Die Menge Inhalt ist durchaus immens.

Statt freiwillige Aufgaben wären verpflichtende Übungszettel besser. Denn der Schwierigkeitsgrad und der Umfang der freiwilligen Aufgaben haben enorm geschwankt bei den Aufgaben. Bei Zetteln wäre der Umfang gleichmäßig und so, dass alle mitkämen. Es kamen jede Woche sehr viele Aufgaben raus, für die man als absoluter Anfänger extrem viel Zeit aufwenden musste (enorm viel Zeit für externe Recherche und Methoden, die in der Vorlesung gar nicht angerissen wurden) und zusätzlich gab noch es die Tutoriumsaufgaben. Es war für mich persönlich zu viel und schwierig und ich kam überhaupt nicht mehr mit seit Dezember. Auch mit "easy" gekennzeichnete Aufgaben fand ich teilweise zu schwierig, vor allem ohne Anhaltspunkt in der Vorlesung oder Musterlösungen (die oft erst Wochen später hochgeladen wurden). Da auch so viele Aufgaben hochgeladen wurden, ist es sehr schwierig zu filtern, welche Aufgaben(typen) bspw. tatsächlich für die Klausur relevant sind oder welche Aufgaben nur für die Vertiefung sind. Die Lösungen sollten zeitnaher hochgeladen werden, damit man direkt abgleichen und seine Fehler korrigieren kann.

Viele (Meiner Meinung nach) wichtige Dinge wie z.B. Referenztypen, instanceof Operator, Scanner Klasse, Methoden von String und Character usw..., Testframeworks (JUnit), implizite Typumwandlung, interface Comparable ... wurden nicht ausreichend genug oder gar nicht behandelt. Auch waren die Übungsaufgaben in Artemis nicht immer ausreichend detailliert gestellt und ließen viel Interpretationsfreiraum, sodass man die Lösung die Artemis annimmt raten musste. Die Aufgabe Maps und Graphen war davon am stärksten betroffen, denn dort z.B. war nicht festgelegt wie die Klasse Grap die Daten im Detail halten soll, sodass ich mehrere mögliche Implementationen austesten musste, bis es dann angenommen wurde. Es wurde auch oft in den Übung verlangt unsicheren Code zu schreiben (also attribute auf public) obwohl das den Java Coding Guidelines widerspricht. Auch ist die Strukturierung nicht immer gerade anfängerfreundlich wie z.B. in der ersten Vorlesung arrays zu behandeln hat viele Anfänger verwirrt.

Struktur der Vorlesung sollte dringend verbessert werden. Der "Haupttutor" hat andauernd den Dozenten unterbrechen müssen, da dieser kleine Fehler gemacht hat und dann sind dadurch Diskussionen entstanden wo man als Student nicht mehr mithalten konnte und eher dazu eingeladen wurde abzudriften. Besser: Live-Beispiele lassen und den fertigen Code besser auf Folien zeigen. Damit kann man sicher sein, dass dieser korrekt ist und man kann Zeile für Zeile diesen in der Vorlesung durch gehen.

Vorlesungsstil: Dozent wirkte manchmal nicht gut vorbereitet/informiert, Vortrag stockte teilweise

Wenn möglich, Themen ausführlicher behandeln und nicht nur kurz anreißen. Für Anfänger ist es sonst ab einem gewissen Punkt schwer mitzukommen, weil tiefergehende Kenntnisse erforderlich sind, die so nicht vermittelt wurden

### 7.3 Hier hast du Platz für weitere Anmerkungen und Feedback zum Modul.

Diese Vorlesung war eine Katastrophe. Kein Plan, keine Struktur, ein einziges Chaos. Die Reihenfolge der Inhalte war nicht durchdacht, die Vorbereitung der einzelnen Vorlesungen ungenügend. Inhalte wurden un- oder nur halbverstanden hervorgestammelt, Zusammenhänge nicht vernünftig erklärt. Die Live-Coding-Sessions waren zwar eine gute Idee, aber in der Durchführung zu holprig.

Und als man dachte, schlimmer kann es nicht mehr kommen, kam der C-Teil... Es wurden einfach Folien vom "Crashkurs C" aus der Drittsemestervorlesung "Systemnahe Vorlesung" ausgeliehen und unverstanden abgelesen. Von der Machart her wirkte das nicht wie eine Vorlesung an einer Universität, sondern wie ein typisches schlechtes von Wikipedia abgeschrieben Schülerreferat.

Und wieso zum Teufel macht man überhaupt \*erst\* monatelang objektorientierte Konzepte und quetscht \*dann\* am Ende kurz noch n bisschen Pointerarithmetik in C rein? Wer hat sich diese Reihenfolge ausgedacht?

Trotz allen Schwierigkeiten denke ich eine klare Verbesserung zu den letzten Jahren, nach allem was man darüber hört!

Ich würde mir klarer Strukturen und vorgefertigte Muster beim Erklären des Stoffes wünschen, d.h. z.B mehr vorgefertigt Analogien. Außerdem würde ich mir weniger Live-Coding und dafür mehr Programmtext mit Kommentaren, welcher dann erklärt wird, auf den Folien wünschen.

Gut eigentlich

Als Programieranfänger war das Tempo der Vorlesung viel zu schnell, da bei einführenden Themen über die Folien hinweg geflogen wurde. Dadurch konnte man im späteren Verlauf der Vorlesung nicht mehr ganz folgen. Zudem waren die Beispiele die in der Vorlesung besprochen wurden nicht sehr hilfreich, wenn der von Troisdorf den Dozenten immer wieder unterbrochen hat und seine Verbesserungsvorschläge kundtun musste oder wo er den Code schöner und abgekürzt haben wollte, was mich als Programieranfänger total verwirrt hat. Es wäre dadurch um einiges besser gewesen den Dozenten das Beispiel auf Anfängerniveau erklären zu lassen ohne das jemand, der schon programmieren kann, das Beispiel zu einem neu besprochenen Thema auf den Stand von Fortgeschrittenen erklärt haben will. Die Hausaufgaben für die jeweilige Woche waren stets zu viel und die Aufgaben an sich teilweise zu schwierig. Selbst bei den leichten Aufgaben (wurden als diese deklariert) saß man selbst in einer Gruppe 2-3 Stunden dran. Dies ist vor allem dem Grund zu verschulden, dass am Anfang die Themen nicht genau genug erklärt worden waren. Die Tutoriumsaufgaben enthielten teilweise Sachen die man noch nicht in der Vorlesung besprochen hat und gingen manchmal über die Zeit der Tutoriums hinaus. Zudem hat man den Prof selbst nie gesehen oder gehört und der Dozent schien die Vorlesung ganz allein stemmen zu müssen.

Danke!

Alles im Allem ist zum aktuellen Zeitpunkt die Vorlesung ohne Vorwissen nicht oder bzw. schwierig schaffbar, sodass man hier angewiesen ist auf externe Quellen.

Siehe oben: weniger Diskussionen während der Vorlesung mit dem "Haupttutor", besser Strukturiert sein in dem der Dozent selbstsicherer die Vorlesung halten kann

Tutorium bei Dominik Behrent leider sehr inkonsistent und unverlässlich

Die Erklärungen allgemein und die von den gestellten Fragen waren leider nicht wirklich hilfreich.

### 7.4 Hier hast du Platz für Anmerkungen und Feedback zur Umfrage.

so ein modul auch für andere programmiersprachen

Keine Tutorenevaluation...?