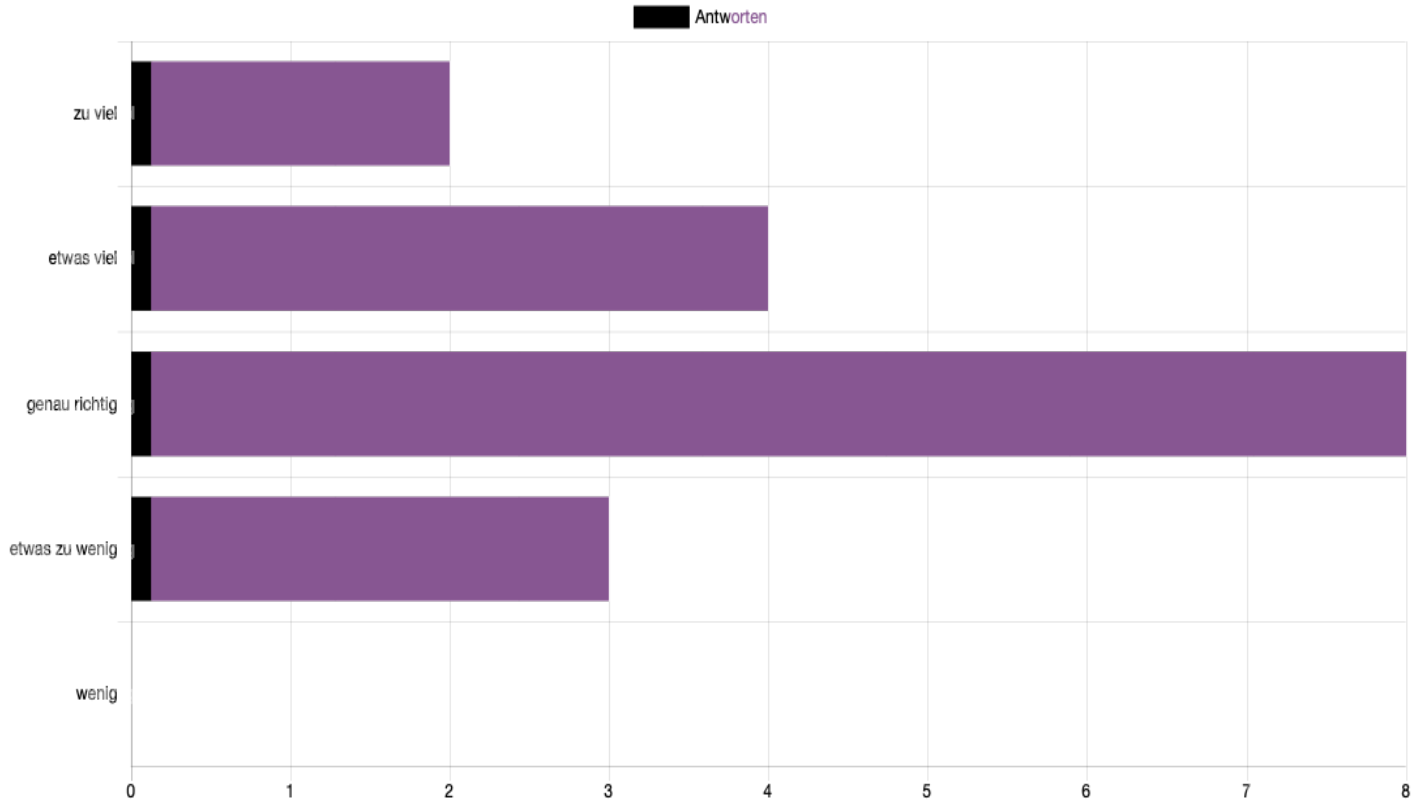
Programmieren 1, WS 2019, IB

Vorlesungsumfrage

Nach Excel exportieren

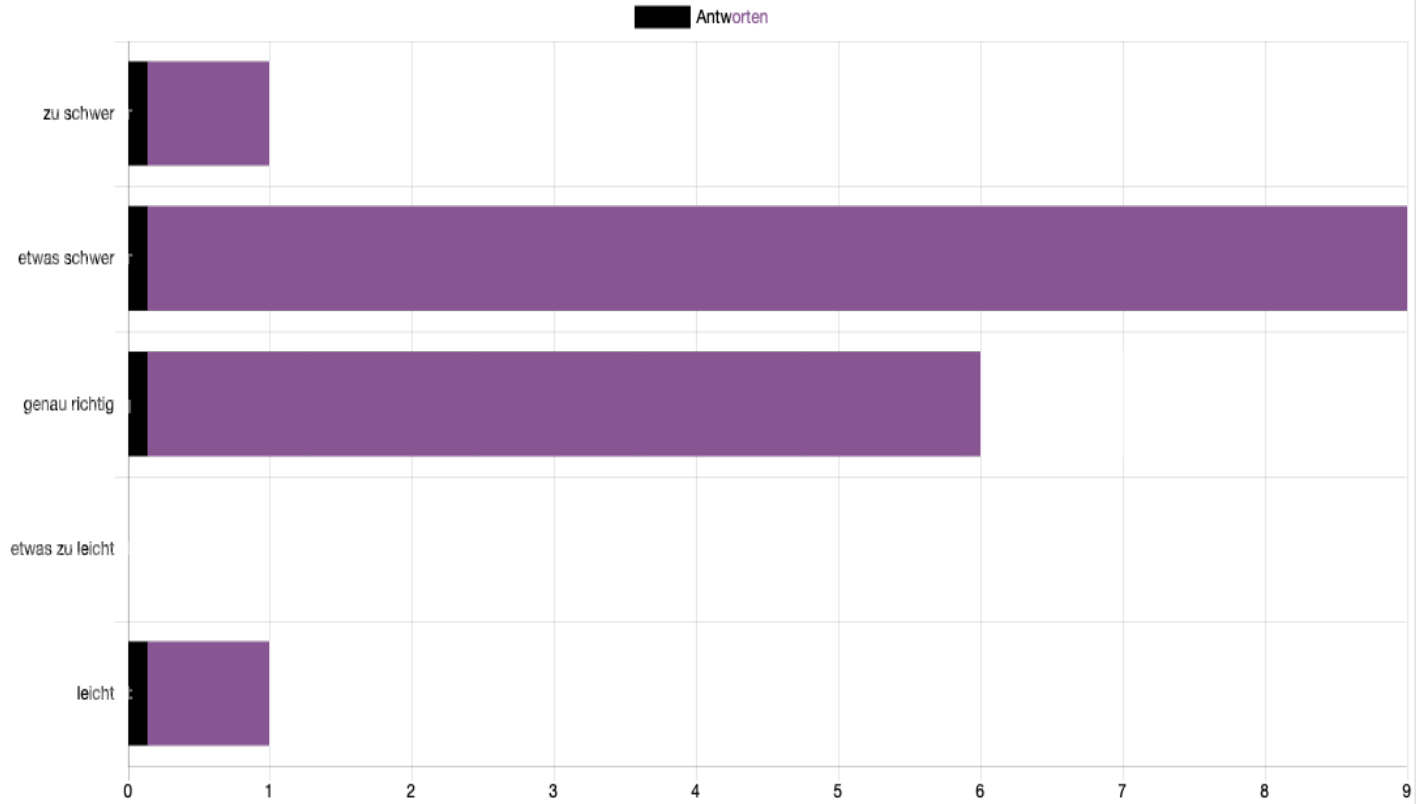
Ausgefüllte Feedbacks: 17
Fragen: 14

Wie beurteilen Sie die Stoffmenge der Veranstaltung?



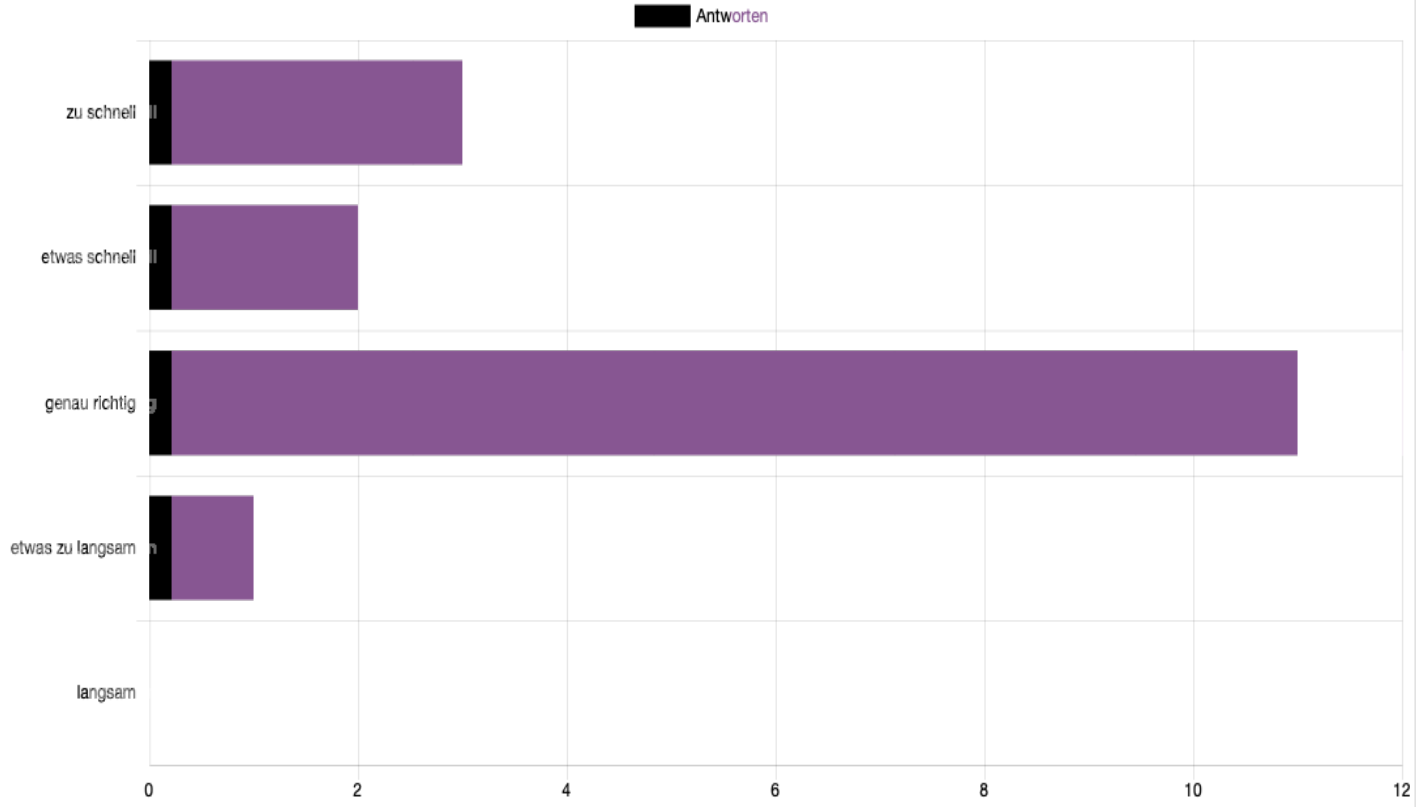
Grafikdaten anzeigen

Wie beurteilen Sie die Schwierigkeit der Veranstaltung?



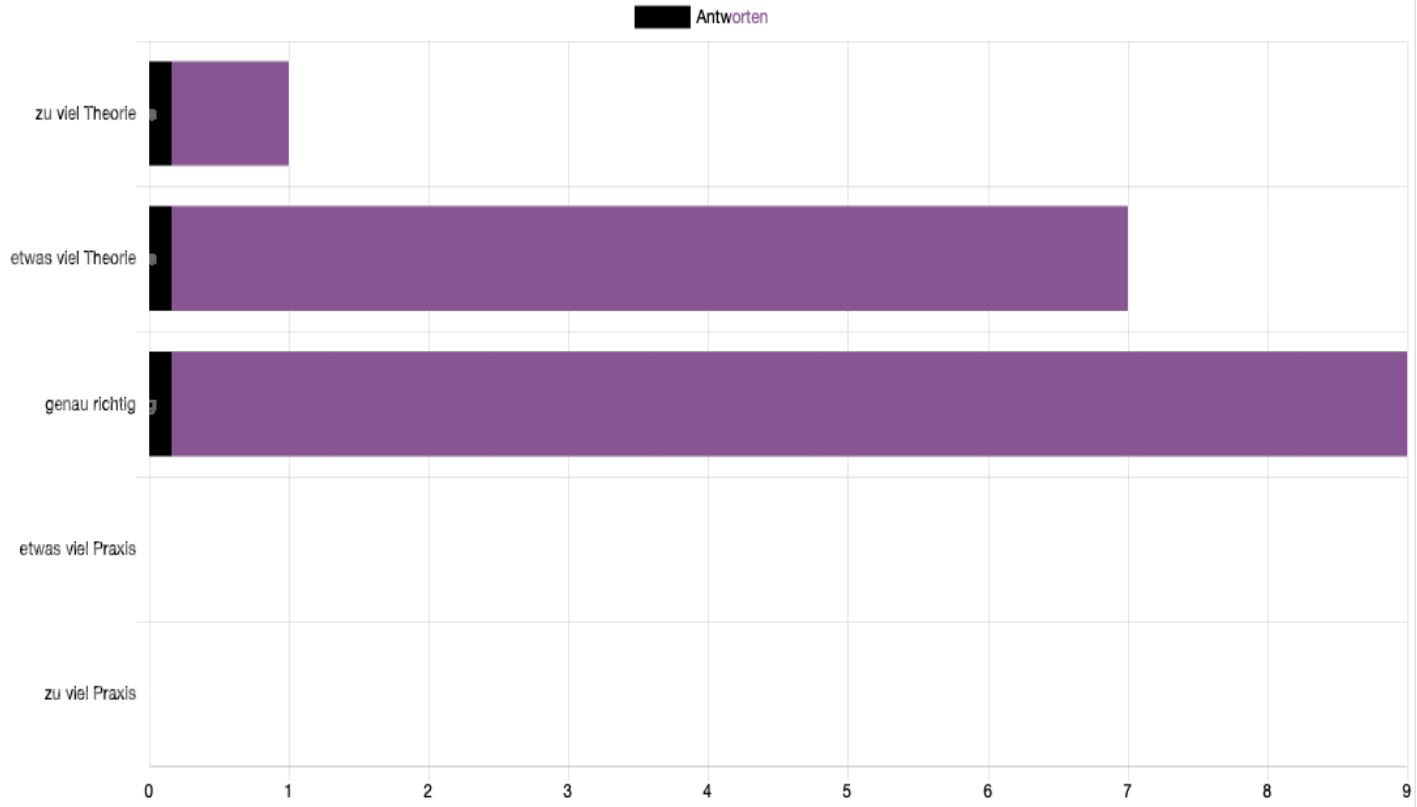
Grafikdaten anzeigen

Wie beurteilen Sie das Tempo der Veranstaltung?



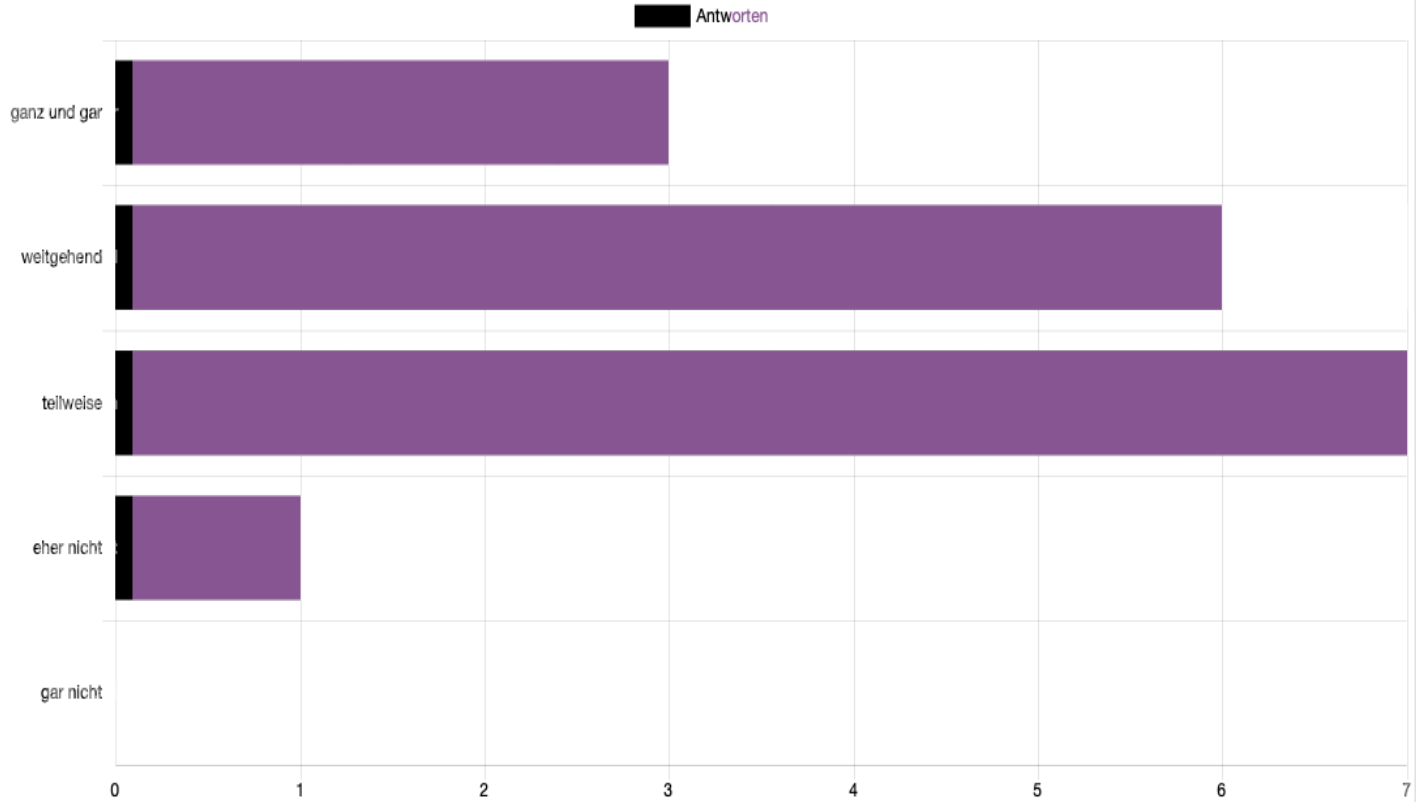
Grafikdaten anzeigen

Wie beurteilen Sie das Verhältnis von Theorie und Praxisbeispielen?



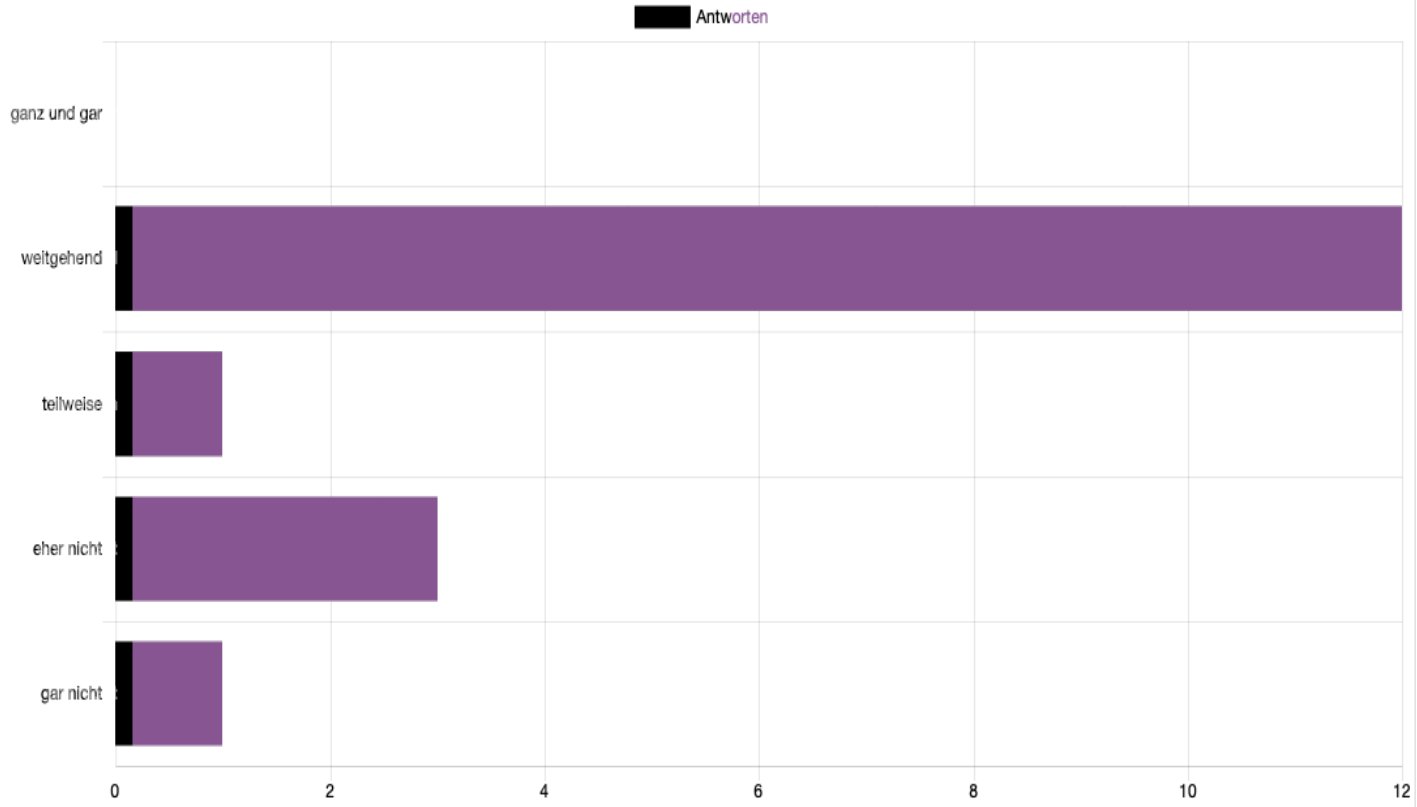
Grafikdaten anzeigen

Wie deutlich ist für Sie der "rote Faden" in der Veranstaltung erkennbar?



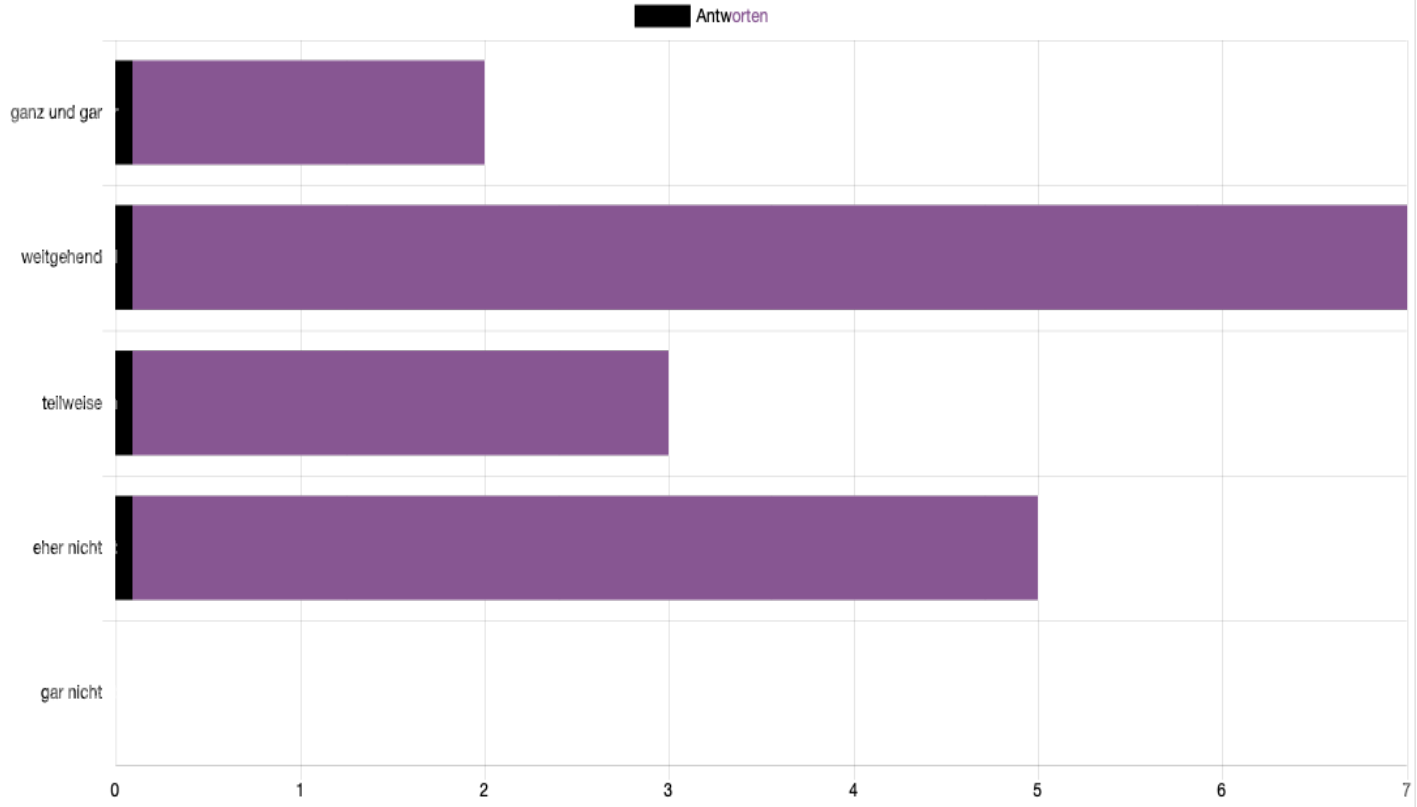
Grafikdaten anzeigen

Ist die Arbeitsatmosphäre in der Veranstaltung für Sie hilfreich?



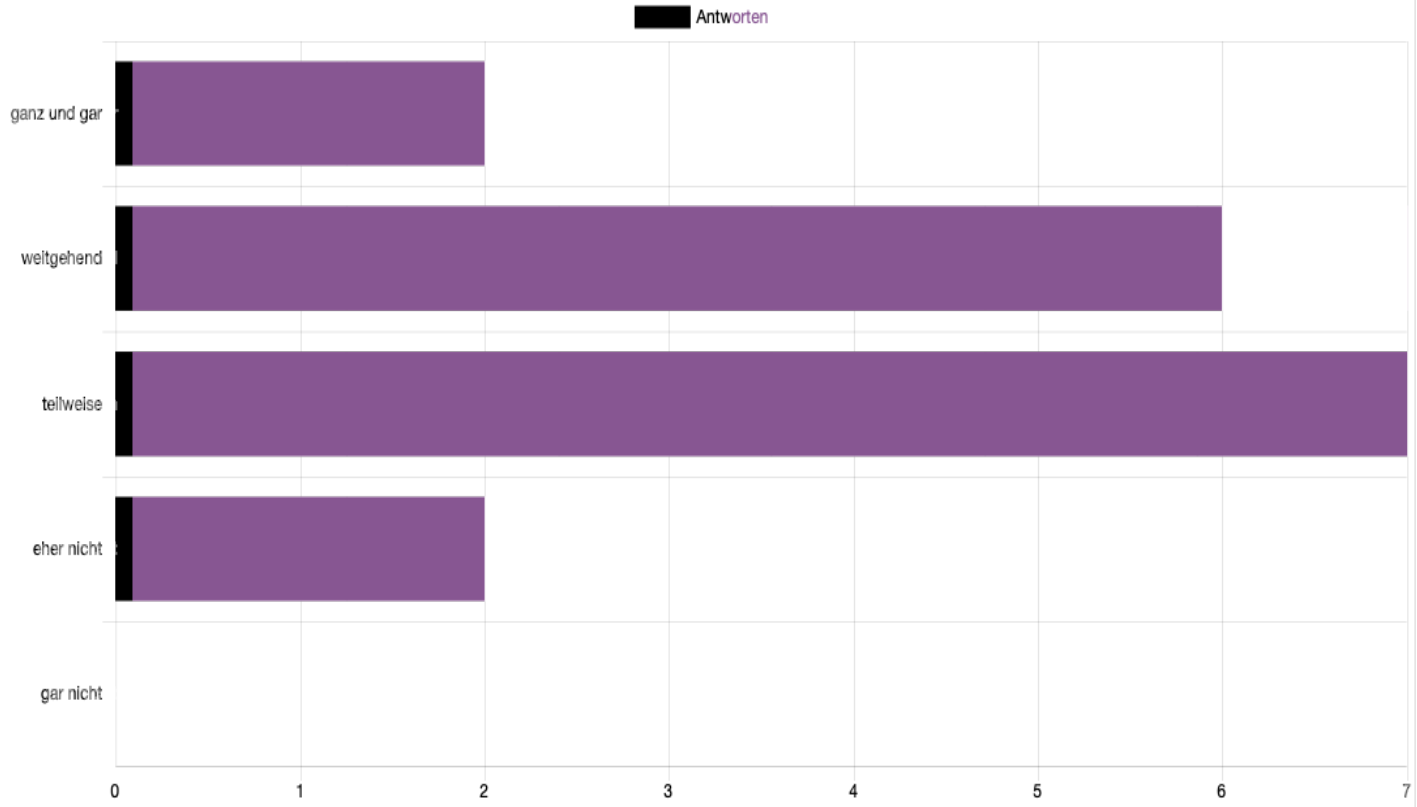
Grafikdaten anzeigen

Sind die eingesetzten Lehrmethoden (Vorlesung, Diskussion, Übung, etc) für Sie hilfreich?



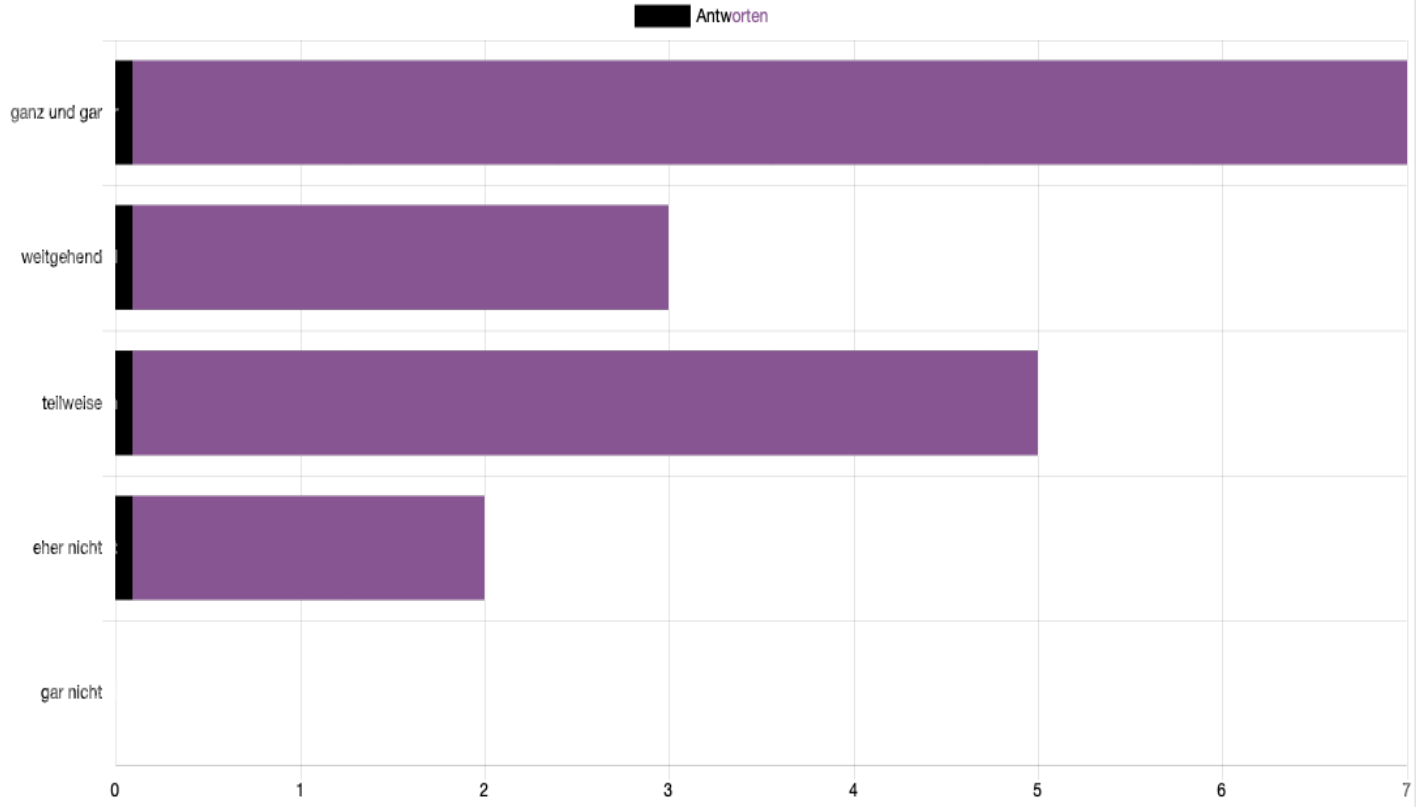
Grafikdaten anzeigen

Sind die angebotenen Arbeitsmaterialien für Sie hilfreich?



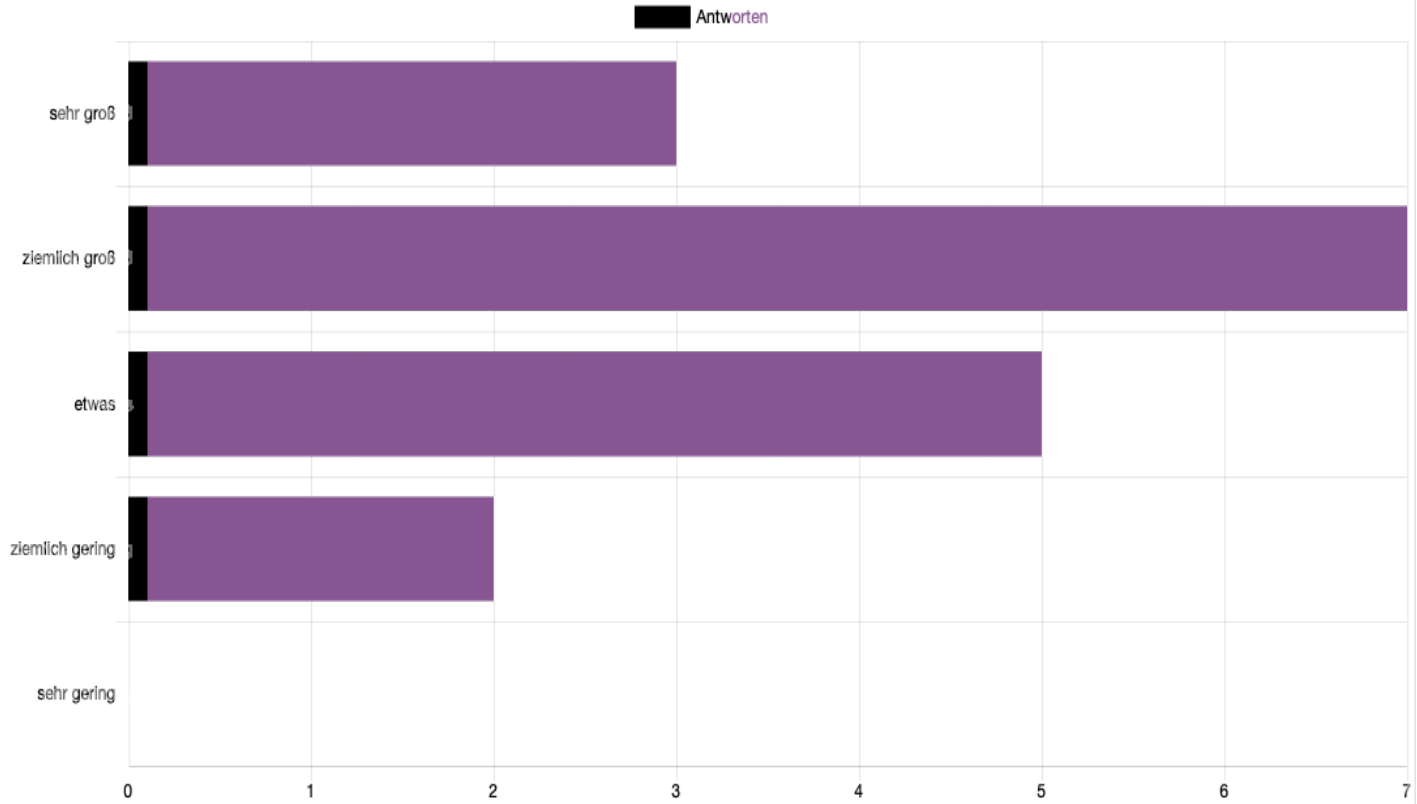
Grafikdaten anzeigen

Können Sie sich an der Veranstaltung aktiv beteiligen?



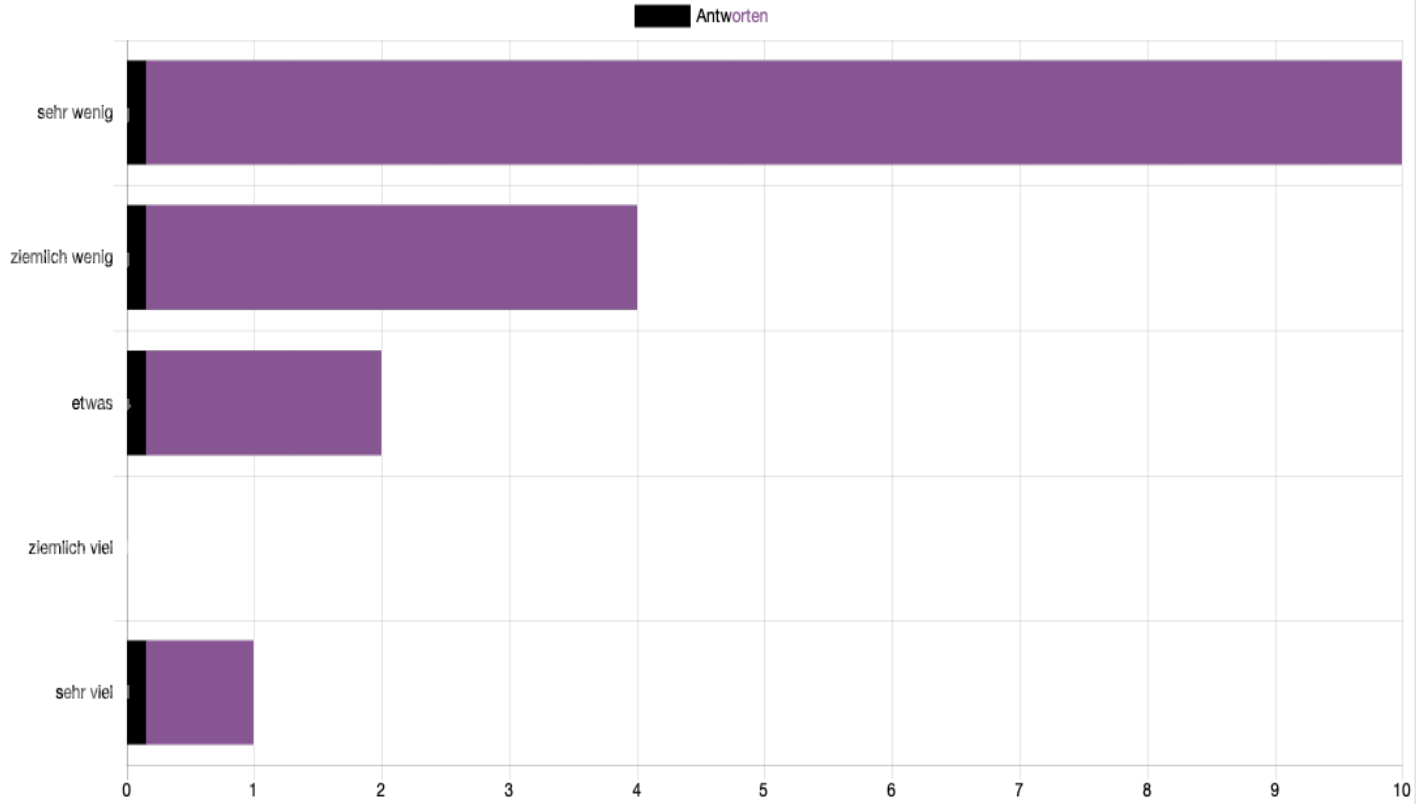
Grafikdaten anzeigen

Wie beurteilen Sie Ihre eigene Motivation, den Stoff nach der Vorlesung nachzuarbeiten und zu vertiefen?



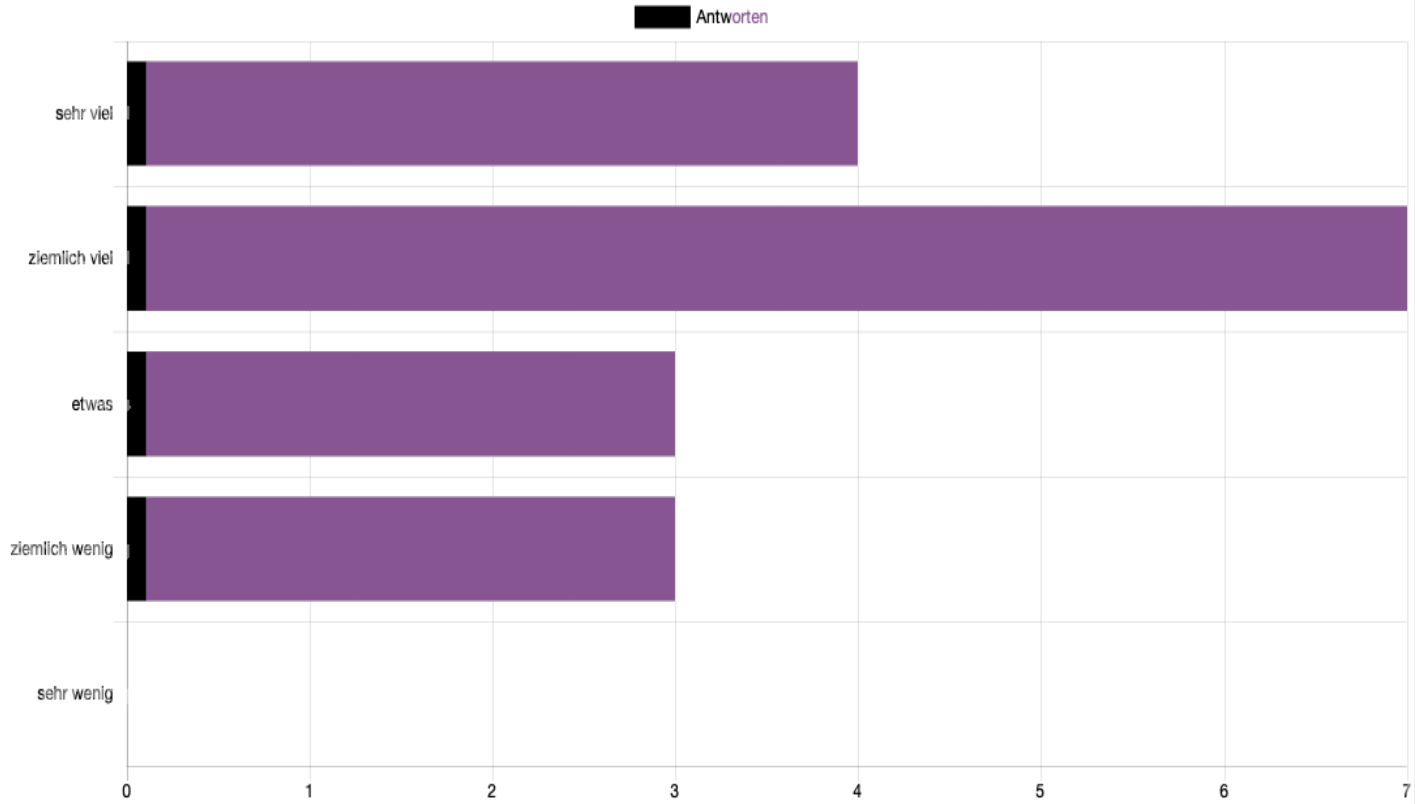
Grafikdaten anzeigen

Haben Ihnen Vorkenntnisse gefehlt, um in die Lehrveranstaltung einsteigen bzw. folgen zu können?



Grafikdaten anzeigen

Wie viel haben Sie bisher in der Veranstaltung gelernt?



Grafikdaten anzeigen

Wie groß ist insgesamt Ihr Aufwand für die Veranstaltung pro Vorlesungswoche? (Anwesenheit, Übungen, Vor-/Nachbereitung etc. pro Woche; Prüfungsvorbereitung auf 15 Wochen umgerechnet)

- 7
- 19
- 15
- 20
- 20
- 12
- 12
- 18
- 20
- 14
- 20
- 15
- 14
- 20
- 12,5

Mittelwert: 15,90

Sonstige Anmerkungen (Kommentare, Verbesserungsvorschläge)

- Die Beispielprogramme sollten mehr auf Eclipse gezeigt werden. Besser wäre noch, wenn jeder ein fertiges Programm bekommen würde und mit diesem Programm, werden dann die Skripte erklärt.

- Auch wenn es ein fester Bestandteil eines Studiums ist, auch Eigenarbeit zu leisten und es Laborstunden für praxis Fragen gibt, wäre es wünschenswert gewesen, wenn vor allem "komplexere" Themen gegen Ende des Semester von Grund auf live vorgeführt werden würden. Als Beispiel das Thema Listen. Bei den vorgeschriebenen Programmfragmenten fiel es mir persönlich schwer eine eigene Umsetzung zustande zu bringen und musste per Videos recherchiert werden. Zusätzlich würde ich es "als Programmieranfänger" vorziehen, "einfache" Themen die zu Beginn des Semesters vermittelt wurden miteinander in den Übungen zu verknüpfen, sodass z.B. das Thema Objektorientierte Programmierung über zwei Übungsaufgaben verteilt werden kann. Aus meiner Sicht hat die eine Pflichtübung in den Themen Objekte / Listen nicht gereicht, um diese auch wirklich zu verfestigen. Und da ein Studium generell ein zeitaufwändiges Unterfangen ist, war relativ wenig Platz um Übungen hierzu im "Eigenstudium" zu bewältigen. Ansonsten gilt noch

das Lob auszusprechen, dass die Veranstaltung vor allem für Programmieranfänger sehr Hilfreich war und man den Professor wirklich zu jedem Sonderfall ausfragen konnte und immer eine kompetente Antwort bekam.

- Es wäre evtl bei manchen hochgeladenen Folien hilfreich gewesen, wenn die Beispiele ausgefüllt wären.

- Gut aufgeteilter Stoff, Leider etwas durcheinander, viel Angesprochen aber vertagt auf spätere Vorlesungen. Besser erst erwähnen wenn es wichtig wird. Gerne etwas mehr Erklärung zu den Programmieraufgaben. Sie waren verständlich aber an mancher Stelle etwas verwirrend zb. beim Sudoku oder BigInt.

- Ich konnte schon vorher programmieren und war nicht immer da, daher kann ich keine all zu gute Bewertung abgeben.
Was ich mitbekomme habe, ist, dass einige Kommilitonen meinen, dass es ihnen geholfen hätte, wenn Sie in der Vorlesung häufiger auch programmcode ausgeführt hätten. Außerdem denke ich, dass es manchen geholfen hätte, wenn im Tutorium ein Tutor sich in der zweiten Hälfte hingestellt hätte & die Aufgaben vorprogrammiert hätte und den Leuten schrittweise erklärt hätte, was er sich denkt und wie er seine Gedanken dann in Programmcode umsetzt. Das hätte manchen vielleicht geholfen, mehr und besser in dieses Denken rein zu kommen.
Für mich war die Sache passend. Klar hätte ich viele Sachen mir nicht anhören müssen, da ich sie schon konnte, aber ich hätte ja auch einfach zu Hause bleiben können. Das Tempo war meines Erachtens ausgewogen.

- Zusammenarbeit mit Partner hat nicht funktioniert.

- Ohne Vorkenntnisse ist es schaffbar, wenn man sich viel damit beschäftigt.

- Allgemeines:

- weniger mit der Tafel arbeiten, sondern in Eclipse selbst arbeiten und den Screen sharen.

- Tafel für theoretischen Anteil, den man nicht in Eclipse zeigen kann. (Speicher Verweise, etc.)

- mehr tatsächlichen Code zeigen (wie die Screenshots, die sie verwendet in den Skripts)

Vorteil: man sieht schneller wie ein Code aussehen sollte, und die Struktur ist deutlicher nachzuvollziehen.

- bei den Folien andere Markierungen nehmen.

Bsp1: wenn sie ein wort markieren, nehmen Sie häufig einen Ellipse. Diese ist häufig zu dick und verdeckt dadurch andere Stellen.

Bsp2: Blasen mit Zusatinformationen sind häufig zu groß und schlecht angeordnet.

- selbst wenn wenig Inhalt auf der Folie steht, verwenden Sie häufig dennoch den gesamten Patz der Folie indem Sie die Schriftgröße sehr stark
erhöhen. (inheitliche Schriftgröße bei allen Folien wäre besser, auch wenn die Folie "leer" wirkt)

Bsp: "Haben Sie Fragen?" - Folie (wirkt sehr archaisch)

- bei längeren Texten auf Folien gerne Zeilen unterschiedlich einrücken, um den Text zusätzlich Struktur zu geben.

Allgemeiner Aufbau der Folien:

zu viel Text, keine optimale Struktur.
-> pro Folie brauch man zu lange den eigentlichen Inhalt nachzuvollziehen

- durch die vielen Animationen werden die Folien immer komplexer und immer unverständlicher.

Bsp: bei einem Programm wollen Sie zu mehreren Zeilen Zusatzinformation mittels animierter Blasen hinzufügen:
Gegen Ende sind eindeutig zu Blasen sprich die Folie ist allgemein zu voll.
Ohne vorherigen Kontext ist die Folie dann viel zu komplex. Besser wäre es währenddessen Blasen wieder zu entfernen, oder an sich kleiner Elemente nehmen die gut angeordnet sind.

konkretes Beispiel:

Foliensatz: Erweiterung am Beispiel - Folie 33

eigentliche Vorlesung:

- oft wurden Dinge zu umgangsprachlich erklärt. ("der compiler meckert"). Es wäre besser fachsprachlich solche Dinge auszudrücken.

- Verwendung von Begriffen die eigentlich falsch sind. (aktueller Parameter/ Vererbung)
Obwohl die Begriffe falsch sind, benutzen wir sie dennoch.

- die Aufgaben, die die Stundenten während der Vorlesung machen sollen, um das kürzlich Vermittelte zu vertiefen,verlangen oft nur den Stoff
wiederzugeben, den man kurz davor gehört hat. Man redet also prinzipell nur Ihnen nach. Optimaler hier wäre ein konkretes Anwendungsbeispiel, bei dem der vermittelte Stoff an ein komplett neues Beispiel angewendet werden soll. Dadurch verhindert man das
Nacherzählen und man muss den Stoff tatsächlich verstanden haben, um ihn anzuwenden.

Direkt zu:

Ankündigungen ▶

Dokumentation zu dieser Seite

Sie sind angemeldet als Prof Peter Knauber (Logout)
PR1 19
Laden Sie die mobile App