Ausgangslage

- **Beschreibung der Veranstaltung**
  Es geht um zwei Vorlesungsübungen für eine Informatikvorlesung. Die Übungen sind zur Vorbereitung der Tests der Klausurzulassung. Es werden fundamentale Programmierfähigkeiten im Vorlesungskontext abgefragt.
- **Format**: Übung als Präsenzveranstaltung mit vorher veröffentlichten Aufgabenpool an Programmieraufgaben.
- **Anzahl der Teilnehmenden**:
  Ca. 75 Personen in der Vorlesung und 10-30 in der Übung.
- **Zeitliche und räumliche Randbedingungen**:
  - **Zulassungsbedingungen Klausur**: Studierende müssen 2 von 3 praktischen Programmieraufgaben bestehen. Diese werden jeweils in Präsenz in einem 120 Minuten-Slot abgehalten.. Die Aufgaben werden vor Ort zufällig aus einem Aufgabenset gewählt, welches vorher veröffentlicht wird.
  - **Raum**: Container Hörsaal mit 115 Sitzplätzen
  - **Studierende/Teilnehmende**:
    - > 80% sind Mono-Informatik Studierende
    - Rest: Mix aus naturwissenschaftlichen Bereichen
  - **Zeit**: 2x2: 2x 90min Übungen pro Aufgabenset. Beide Übungen enthalten jeweils den gleichen Inhalt.
  - **Inhalt**:
    - Überwiegend frontal.
    - Aufgabenset der Übungsaufgaben vorstellen
      - Aufgabenset zwei Wochen vorher veröffentlicht
      - Zugehörige Prüfung eine Woche später
    - Eingehen auf Fragen von Studierenden
      - Nicht priorisiert, wegen 2 getrennten Frageterminen
    - Live lösen einer Aufgabe, die prüfungsrelevant ist.
    - Lösung der Studierenden wird hochgeladen

Zusammenfassung des Transferkonzepts

- Flipped-Classroom Coding Class vorweg als Vorbereitung
- Aufgabenübersicht (5 min)
- Classroom Response mit Cliqr mit Fragen zur Vorbereitung und Ausblick (5 min)
- Programmablaufvisualisierung in Murmelgruppen (5 min)
- Programmmodulsammlung als Brainstorming & aktivem Strukturieren aus der Visualisierung (5 min)
- Pseudocode Live-”Programmieren” nach der Fishbowl Methode (15 min)
  Python Implementation nach Think-Pair-Share, einen Programmteil nach Think+Pair erstellen und dann im Share vorstellen (50 min)
- Blitzlicht als Ablussstimmungsbild (5 min)

Beschreibung der Durchführung

Durch den Classroom Response wurde klar das nur exakt ein Studierender vorbereitet war und das alles deutlich länger dauern wird, und es wurden agil Teile des Konzepts zusammengefasst: Die Einzel- und Kleingruppenarbeiten funktionieren nicht ohne Vorbereitung, also wurde der Programmablauf als längeres Brainstorming direkt durchgeführt (20 min). Da sich Think+Pair auch auf nur ein “Pair” beschränken würde, wurden die beiden folgenden Programmierteile zusammengezogen. Das klappte dann durch den einen vorbereiteten Studierenden halbwegs gut, aber durch die ca. fehlenden 30min blieben am Ende noch einzelne Fehler über. Dabei wurde der Fokus auf das logische Verstehen der Aufgabe gezogen: Es wurden alle Programmteile logisch umgesetzt und nur formale Fehler waren noch vorhanden oder einzelne triviale Implementationen wurden nur textlich eingefügt. Damit sollte jeder mit Programmierkenntnissen - die zumindest teilweise vorausgesetzt werden - in der Lage sein, die Aufgabe alleine zu einem guten Ergebnis zu führen. Die textlichen Teile sind allerdings nicht ausführbar und die vorhandenen formellen Fehler - unpassende Variablen-Typen - führen direkt zu Fehlermeldungen, die relativ einfach verständlich und damit schnell behebbar sind. Daher ist die Lösung so nicht verwendbar und nur mit Nacharbeit verständlich, wenn man nicht in der Übung selbst war.

Darstellung der Ergebnisse der begleitenden Evaluation

Anmerkungen der begleitenden Evaluation durch kollegiale Hospitation:

- Geringe Vorbereitung der Studenten trotz unmittelbarer Prüfungsrelevanz
- Flexible Kürzung des Transferkonzepts
- Niedriger Schwierigkeitsgrad der Aufgabe (für Informatiker)
- Fehlende grundlegende Informatikerskills trotz vorwiegend Mono-Bachelor

Nachträgliche Analyse

Durchführung für weitere Übungsgruppe nach bisherigem Lehrkonzept im Vergleich:

- Auch nur ein Studierender vorbereitet
- Aktive Mitwirkung fällt gleichwertig aus
- Weniger Eigenleistung der Studierenden am Ergebnis
- Ergebnis: Funktionierende, vollständige Beispiellösung

Prüfung:
Studierende, die vermutlich gar nicht in der Übung waren, lernten die Beispiellösung teilweise ohne zu reflektieren. Sie nahmen fälschlicherweise an, es sei eine Musterlösung, obwohl die Beispiellösung so nicht ausführbar ist.

Reflexion Ihrer Erfahrungen

- Die aktivierenden Methoden funktionieren nur, wenn ein grundlegender Wissensstand vorhanden ist und sie kosten enorm Zeit.
- Ergebnisse haben eine deutlich schlechtere Qualität, da keine Zeit zur ausreichenden Korrektur bleibt. Darauf folgt, dass nachträglich der Übungsstoff nicht gut wiederholt werden kann.
- Kurze Anfangsumfragen zur Vorbereitung geben guten Einblick
- Die Komplexität der Aufgaben benötigt eine Vorbereitung für einen eigenen Beitrag eines Studierenden.

Fazit und Ausblick

Kurze Feedbackrunden bringen ohne großen Zeitaufwand einen großen Mehrwert. Je nach Raumbegebenheit können und sollten Lehrmethoden angepasst werden, was durch technische Hilfsmittel erleichtert werden kann. Aktivierende Lehrmethoden sind abhängig von der Vorbereitung der Studierenden und sollten daher auch abhängig von der jeweiligen Übungsgruppe und -stunde gewählt werden. Glücklicherweise kann die Übungsstunde in diesem Kurs meistens direkt dafür angepasst werden. Je größer die Vorbereitung, desto mehr ist eine Eigenleistung der Studierenden möglich. Dabei darf das klare Ziel eines Ergebnisses zum nachträglichen Lernen nicht verloren gehen. Dies konnte inzwischen erfolgreich umgesetzt werden.