



## Auftrag Task 74 "Einstufungstest OOP TI Informatik"

### Ziel:

1. Für das HS 2020/21 soll im Fachbereich Informatik ein Einstufungstest im Self-Service-Modus entwickelt werden, im Moodle-Kurs <https://moodle.bfh.ch/course/view.php?id=23699>
2. Dieser Einstufungstest soll allen neu eintretenden Studierenden ins 1. Studienjahr Informatik die Möglichkeit geben, ihre Java-Kenntnisse unter Beweis zu stellen.
3. Bei einem genügend hohen Punktwert soll eine Zulassung zur OOP1-Modulvariante "Kotlin" anstatt "Java" erfolgen.
4. Bei einem stark ungenügenden Punktwert soll der Besuch des Vorkurses empfohlen werden.

### Fertigstellung:

- Vor Beginn HS 2020
- **Richtdatum 1. September 2020 bereit zum Ausrollen**

### Deliverables:

1. Vollständig eingerichteter, attraktiver Moodle-Kurs mit allen Elementen für alle Stakeholder (<https://moodle.bfh.ch/course/view.php?id=23699>), zweisprachig DE/FR
2. Fragesammlung mit Testfragen unterhalb der Kategorie 2020-Level-Test-OOP ([Link](#)), mit selbst entwickelter Unterstruktur
3. Prüfungsprozess (BPMN-Diagramm, mit bpmn.io erstellt und im Ordner von Task 74 ([Link](#)) als .bpmn sowie als .png im Moodle-Kurs einsehbar abgelegt, mit folgenden Elementen
  - a. Administration des Kurses
  - b. Anmeldung/Einschreibung in den Moodle-Kurs für Studierende
  - c. Vorbereitungs-Test, um die technischen Voraussetzungen zu prüfen
  - d. Realer Einstufungstest
  - e. Kommunikation der Ergebnisse an Studierende und Modulverantwortliche
  - f. Löschung der Testergebnisse nach nützlicher Frist (z. B. 90 Tage)
  - g. Erweiterung des Fragepools/Einstufungstests durch Modulverantwortliche
4. Satz an Einstufungsfragen, sortiert nach Kategorien (Fragekategorien im Sinne von Moodle):
  - a. Themenbereich aus OOP1 Java gem. Modulbeschreibung ([Link](#))
  - b. Zweisprachige Fragen DE, FR; gesamter Code EN
  - c. Schwierigkeitsstufen (gem. Bloom'scher Taxonomie, z. B. [hier](#) erklärt)
    - i. **Remember** – einfache Fakten und Grundkonzepte definieren
    - ii. **Understand** – klassifizieren, selektieren, erklären von Grundkonzepten
    - iii. **Apply** – vorhandene Information in neuen Situationen anwenden
    - iv. **Analyse** – Ideen zwischen Konzepten ziehen: vergleichen, differenzieren
    - v. **Evaluate** – begründen, kritisieren, verteidigen
  - d. **Abstufungen**
    - i. Mindestverständnis für Eintritt in OOP: z. B. etwas höher "Java Programmieren lernen – die Basics" von Chip ([Link](#))
    - ii. Mindestverständnis für Eintritt in Kotlin: A. Laube anfragen
5. Vorbereitungs-Test, mit dem die Studierenden die technischen Voraussetzungen prüfen können (gleichbleibend über die Jahre): typische (einfache) Fragen, in höchstens 5 Minuten durch Studierende zu absolvieren
6. Einen ersten Einstufungstest für das HS 2020-21, enthaltend ausgewählte Einstufungsfragen
7. Dokumentation für Modulverantwortliche: "OOP: Development of Self-Assessment Tests"
  - a. Sprache: Deutsch, mit Deepl auf FR/EN übersetzt
  - b. Format BFH-Vorlage de\_Neutral\_hoch
  - c. Zeigt auf, wie Modulverantwortliche (Dozierende)
    - i. die Ergebnisse von Einstufungen ablesen resp. die Resultate anzeigen können
    - ii. Prüfungsfragen erstellen können, diese testen und in einen neuen Einstufungstest integrieren können
    - iii. Neue Einstufungstests selbst prüfen können

## Bedingungen:

1. Stakeholder und Stories
  - a. Administrator:
    - i. Als Administrator möchte ich das Kurskonzept auf einen Blick übersehen können, Supportbedarf (Nutzereinschreibung, Testaufbau, Resultate etc.) abschätzen können.
  - b. Studierende:
    - i. Als Student möchte ich auf einfache Weise herausfinden, ob meine Java-Vorkenntnisse für den Besuch der Modulvariante "Kotlin" ausreichen, oder ob ich zuerst den Vorkurs belegen sollte.
    - ii. Als Student möchte ich mich in einem Vorbereitungs-Test überzeugen können, dass meine Browsereinstellungen kompatibel mit den Anforderungen des Einstufungstests sind, sodass ich diesen technisch problemlos absolvieren kann.
    - iii. Als Student möchte ich das Ergebnis des Einstufungstests gleich nach dem Abschluss einsehen, meine Punktzahl sowie die Korrektur (richtig/falsch) einsehen können.
  - c. Modulverantwortliche:
    - i. Als Modulverantwortlicher möchte ich auf einfache Weise das Testergebnis aller Kandidat/innen einsehen können.
    - ii. Als Modulverantwortlicher möchte ich auf einfache Weise den Fragepool erweitern können.
    - iii. Als Modulverantwortlicher möchte ich für das Folgejahr eine neue Version des Einstufungstests erstellen und für künftige Studierende bereitstellen können.
2. Rohmaterial für Beispielfragen
  - a. Wird von A. Laube und Kolleg/innen bereitgestellt, s. Fragekategorie "Inspirationen-OOP" ([Link](#))
  - b. Bitte **diese Fragen nicht 1:1 übernehmen**, sondern eigene Beispiele auf dieser/ähnlicher Basis konstruieren!!
3. Kurseinrichtung und Beispielfragen zu allen Fragetypen
  - a. Wird von M. Röthlin bereitgestellt
4. Umsetzung Code-Beispiele - Wünsche
  - a. Der Moodle-Test wird mit Copy-Paste-Blocker ausgeführt (Kandidaten können nicht einfach den Sourcecode übernehmen und ausführen) → evtl. sind die Codebeispiele dann eher als Code formatiert einzufügen (Absatzformat "Vorformatiert" wählen) anstatt als Bild.
  - b. Gute Fragen könnten das Einfügen (drag-and-drop) von 1 aus 4 Codesegmenten in einen übergeordneten Codeblock beinhalten, damit der Gesamtcode eine bestimmte Funktion erfüllt.

## Mitarbeitende und Zeitbudgets:

- Matthias Keller (Zeitbudget 80 Arbeitsstunden); bitte ab 60 geleisteten Stunden M. Röthlin informieren, um Erweiterung Budget oder Kürzung Scope zu besprechen
- Annett Laube (Zeitbudget nach Verfügbarkeit), darf von Matthias bei fachlichen Fragen zu OOP/Java direkt kontaktiert werden
- Michael Röthlin (Zeitbudget 10 Arbeitsstunden), soll von Matthias bei organisatorischen/Moodle-Fragen direkt kontaktiert werden.

Kontierung Arbeitsstunden: **PSP-Element Z.009456-90-SRAK-02** (PgB E-Assessment)

## Meetings und Koordination:

- Grundsätzlich alle 2 Wochen (ausserhalb Ferienzeit), per MS Teams.