



Parcoursprüfungen im Praxistest

Carsten Gips (HSBI)

Ausgangssituation

| W | Einheit 1 | Einheit 2 | Einheit 3 | Einheit 4 | Abgabe/Vorstellung | Monster-Jagd |
|----|-------------------------------|-------------------------------|-------------------------------|------------------------|--|---|
| 1 | Orga | Build3: Gradle | | | | |
| 2 | Generics1: Klassen & Methoden | Generics2: Bounds & Wildcards | Generics3: Type Erasure | Generics4: Polymorphie | B01: Dungeon L0: Demo-Level | (1) Stadtwald-Monster (Generics) |
| 3 | Feiertag | | | | | |
| 4 | Git1: Intro | Git2: Basics | Funktionsinterfaces & Lambdas | Javadoc | B02: Dungeon: L1 Zerbrechende Tiles | (2) Stier-Monster (Git Basics, Lambda, Javadoc) |
| 5 | Git3: Branches | Git4: Brachingstrategien | Methodenreferenzen | Logging | B03: Git Basics, Lambdas | (3) Bahnhofs-Moster (Git Branches, Methodenreferenzen, Logging) |
| 6 | Git5: Remotes | Git6: Workflows | Stream API | Record-Klassen | B04: Dungeon L2: Fackeln im Sturm | (4) Uhren-Monster (Git Remotes, Streams, Record) |
| 7 | JUnit1: Intro | JUnit2: Basics | Optional | Visitor-Pattern | B05: Streams und Records | (5) Pudding-Monster (JUnit Basics, Optional, Visitor) |
| 8 | JUnit3: Testfall-Ermittlung | JUnit4: Mocking (Mockito) | Defaultmethoden | Observer-Pattern | B06: Dungeon L3: Fog of War | (6) Burg-Monster (ÄK/GW, Observer) |
| 9 | Bad Smells | Coding Rules und Metriken | Refactoring | Build4: CI | B07: Testen | (7) Kneipen-Monster (Clean-Code) |
| 10 | RegExp | Template-Method-Pattern | Command-Pattern | Enumerationen | B08: Dungeon L4: Bridge-Guard | (8) Boss-Monster I (RegExp) |
| 11 | Threads1: Intro | Threads2: Synchronisierung | Threads3: High-Level Konzepte | Configuration | B09: Lexer mit Template-Method-Pattern | (10) Boss-Monster II (Threads) |
| 12 | Rückblick | Klausurvorbereitung | | | B10: Hamster mit Multithreading | |
| | Prüfung Phase | | X | | X | |
| | Prüfung Phase | | X | | X | |

■ Veranstaltung 2. Semester:

■ VL + PR

■ Performanzprüfung:
50% PR + 50% Klausur

■ Übungszettel: Teamwork

■ 15' pro Team für Vorstellung der Lösung, Nachfragen, Feedback, Bewertung

6 Module => 6 Klausuren in zwei Wochen

Parcoursprüfungen

"Bei Parcoursprüfungen durchlaufen Studierende [...] eine bestimmte Anzahl an Prüfungsstationen [...]."

Quelle: §18 Parcoursprüfungen, „Studiengangsprüfungsordnung (SPO) für den Bachelorstudiengang Informatik an der Hochschule Bielefeld (Version 23)“, Hochschule Bielefeld, Hrsg., 01. September 2023.

Setup: Freiheitsgrade

- | Anzahl der Stationen
- | Dauer der Stationen
- | Zeitslot für Stationen
- | Einzel- vs. Team-Leistung
- | Design der Stationen:
 - | E-Test
 - | Labor-Aufgaben
 - | Vorträge
 - | Mini-Projekte
 - | ...

Setup

| | | |
|---------------------|------------|------------|
| Übungsaufgaben: 35% | Station 01 | 10 |
| | Station 02 | 13 |
| | Station 03 | 12 |
| E-Assessment: 65% | Station 04 | 30 |
| | Station 05 | 35 |
| Summe | | 100 |

plus Testat für Praktikum

Erfahrungen: Positiv

- Studierende beschäftigen sich regelmäßig und zeitnah mit den Inhalten
- Stationen:
 - Intensiveres Abfragen des Stoffs und der Kompetenzen
 - Prüfungen an sich sind kürzer (= angenehmer)
 - Weniger Klausuren im Prüfungszeitraum
- Übungen: Intensive Diskussionen unter den Studierenden, Fehlerkultur => näher am Berufsleben

... Studierende äußern Wunsch/Bedarf nach dieser Prüfungsform

Erfahrungen: Vorsicht

- Mehr Aufwand für die Lehrenden (Organisation, Vor-/Nachbereitung)
- Man "verschenkt" Vorlesungs- bzw. Übungstermine
- Abstimmung im Kollegium notwendig

Gelingensbedingungen

- Abstimmung im Studiengang: Nicht zu viele Module gleichzeitig mit Parcoursprüfung pro Semester
- Nicht zu viele Prüfungsstationen: 2..3 E-Tests plus max. 4 Übungsstationen
- Dauer Stationen: 45' für E-Test, 60..90' für Live-Coding
- Variation der Formate für die Stationen: E-Tests, Live-Coding, Vorträge, ...
- Erscheint vor allem in erster Studienhälfte sinnvoll

Ausblick

- Kombination mit Standards-based Grading
 - Note über das Erreichen von Lernzielen definiert
 - Jedes Lernziel kann mehrfach nachgewiesen werden
 - Fokus auf inhaltliches Feedback für individuelle Verbesserung

D. Clark & R. Talbert: "Grading for Growth", 2023, Routledge.

License



Unless otherwise noted, this work is licensed under CC BY 4.0. All logos are excluded from this licence.

See github.com/cagix/dlk24 for sources, slides and handout.