

# Prüfung berufsrelevanter Kompetenzen in der Programmierausbildung

Prof. Dr.-Ing. Marc Hensel

Hochschule für Angewandte Wissenschaften Hamburg  
Fakultät Technik und Informatik · Department Informations- und Elektrotechnik

## Ausgangssituation

Das Modul *Objektorientierte Programmierung* des Bachelorstudiengangs Elektro- und Informationstechnik an der HAW Hamburg beinhaltete ursprünglich eine benotete schriftliche Klausur als Leistungsüberprüfung. Diese Prüfungsform ist zur Bewertung von Kompetenzen in der Softwareentwicklung wenig geeignet und bietet auf Noten fokussierten Studierenden kaum Anreiz, im Berufsleben benötigte Kompetenzen zu erlernen. Durch eine Änderung der Prüfungsform in eine praktische Programmierprüfung mit optionalem theoretischen Teil ergaben sich jedoch neue Möglichkeiten der Prüfungsgestaltung, mit denen die Prüfung und Vorbereitung der Studierenden auf typische in der beruflichen Praxis benötigte Kompetenzen ausgerichtet wurde.

## Ansatz und Umsetzung

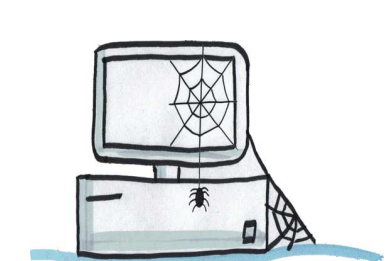
Für den Aufbau der Prüfungen sollten berücksichtigt werden:

- ▶ Typische Tätigkeiten im Beruf (d. h. Anforderungen der Industrie an Absolventen)
- ▶ Prüfungsumfang trotz Abdeckung unterschiedlicher Kompetenzen vertretbar
- ▶ Beschränkte Ressourcen des Prüfers (d. h. Bewertung mit geringem zeitlichen Aufwand)
- ▶ Objektive und für Studierende transparente Bewertung

Aus langjähriger Industrieerfahrung wurden drei Kompetenzen identifiziert, die in der industriellen Softwareentwicklung von fundamentaler Bedeutung sind. Es wurden Lernziele sowie der unten dargestellte dreiteilige Prüfungsansatz formuliert, welche die zugehörigen Kompetenzen abbilden. Die Lehrmaterialien sowie die Aufgabenstellungen der praktischen Labore wurden auf die Lernziele und die Prüfung ausgerichtet (siehe [1]).

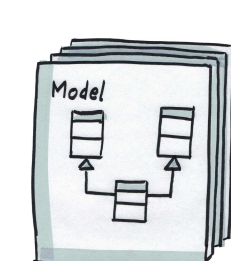


Prüfungen bestehen stets aus drei Aufgaben, die jeweils eine der nachfolgenden Kompetenzen abdecken. Dies ist den Studierenden bekannt.



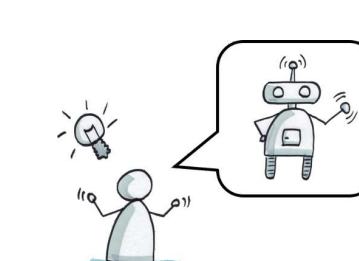
### Analyse von Quelltexten:

- ▶ Taxonomiestufe 4 (*Analyse*): Studierende müssen gegebene Quelltexte verstehen und deren Strukturen erkennen, um z. B. vorhandene Software modifizieren und erweitern zu können.
- ▶ Auswertung von Freitexten (aufwändig) oder Multiple Choice (schnell, objektiv und transparent)



### Umsetzung von Vorgaben:

- ▶ Taxonomiestufe 3 (*Anwendung*): Studierende müssen erlernte Konzepte anwenden, indem sie Software nach detaillierten technischen Vorgaben implementieren.
- ▶ Auswertung über automatische Tests (schnell, objektiv und transparent)

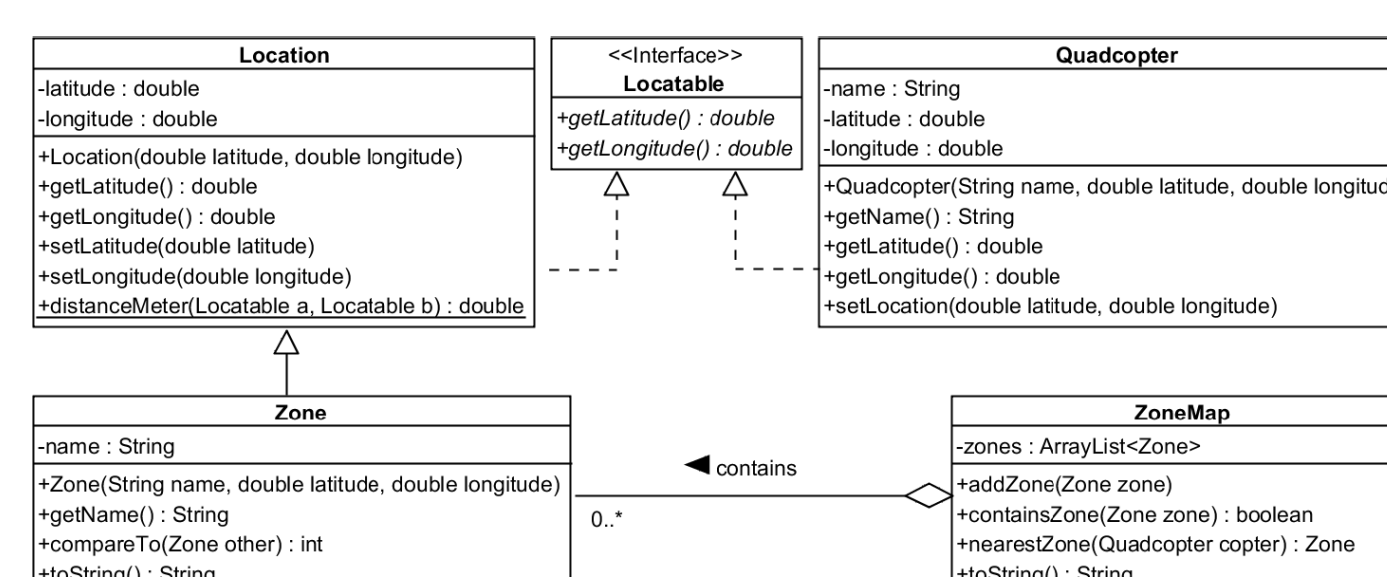


### Kundenanforderungen:

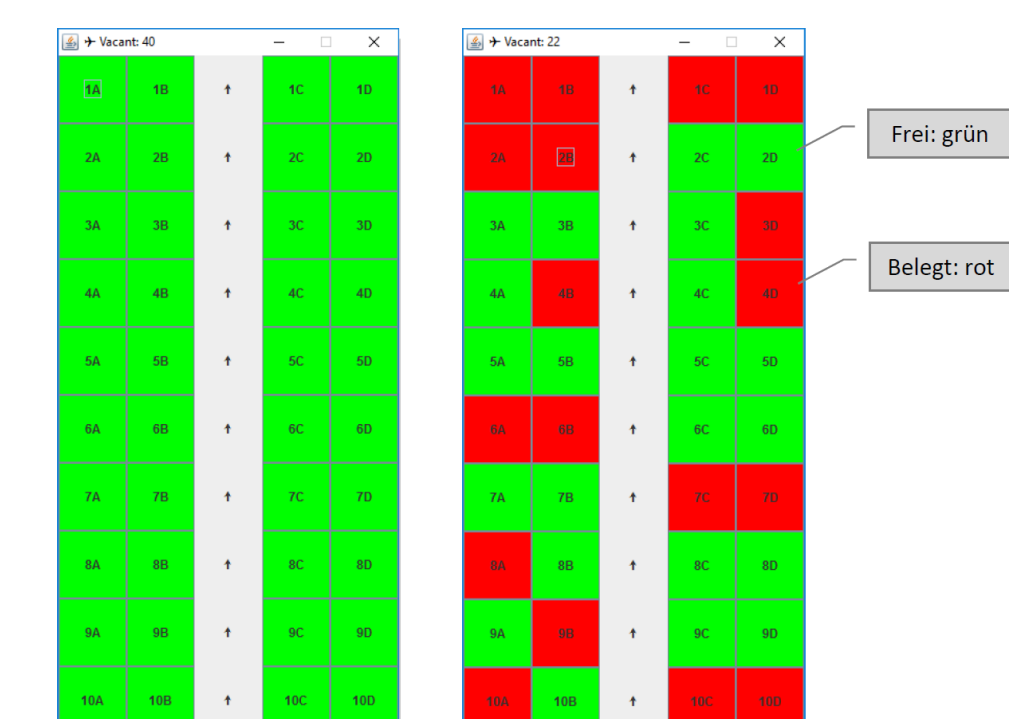
- ▶ Taxonomiestufe 5 (*Synthese*): Studierende müssen eigene technische Lösungen entwickeln, um ein Programm zu erstellen, das geforderte Eigenschaften aus Benutzersicht erfüllt.
- ▶ Auswertung über beobachtete Eigenschaften des erstellten Programms (schnell, objektiv und transparent)

```
public class Singleton {
    private static Singleton instance = new Singleton();
    private Singleton() {
    }
    public static synchronized Singleton getInstance() {
        return instance;
    }
}
```

Beispiel: Analyse eines nicht besprochenen Entwurfsmusters



Beispiel: Implementierung eines Klassendiagramms



Beispiel: Implementierung einer Sitzplatz-Reservierung eines Flugzeuges

## Anmerkungen und Erfahrungen

- ▶ **Kompetenzerwerb:** Den Studierenden ist bekannt, welche Kompetenzen sie in der Prüfung nachweisen müssen. Sie bereiten sich dementsprechend vor und erwerben gezielt für ihr Berufsleben relevante Fähigkeiten.
- ▶ **Motivation:** Der Praxisbezug der geprüften Kompetenzen wird deutlich. Dies führt zu hoher Motivation der Studierenden (→ „für das Leben lernen“).
- ▶ **Objektivität:** Die teilweise Auswertung über automatische Tests sowie beobachtbare Eigenschaften von Programmen wird als objektiv wahrgenommen (→ „einheitliche Bewertung aller Studierenden“).

- ▶ **Transparenz:** Die in der Prüfung erwarteten Kompetenzen sind den Studierenden bewusst (→ „keine bösen Überraschungen“).
- ▶ **Korrektur:** Der Aufwand einer Korrektur über automatische Tests sowie beobachtbare Eigenschaften von Programmen ist für Lehrende vergleichsweise gering. Studierende erhalten die Ergebnisse zeitnah.
- ▶ **Taxonomiestufen:** Die zu den Kompetenzen gewählten Prüfungsansätze befinden sich auf unterschiedlichen Stufen. Dies erscheint durch die gute Abbildung der Aufgabenstellungen auf berufliche Tätigkeiten nicht nachteilig, sondern angemessen.

[1] M. Hensel: *Einführung kompetenzorientierter Lehre in der Programmierausbildung*, nexus-Tagung, 26. Februar 2019, Technische Hochschule Köln