# Legacy Code

Lukas Panni
TINF18B5

DHBW Karlsruhe
Vorlesung Advanced Software Engineering Semester 5/6

# Inhaltsverzeichnis

# 1 Abhängigkeiten brechen

Im folgenden sollen einige Beispiele aufgeführt werden, bei denen Abhängigkeiten mithilfe der in der Vorlesung behandelten Techniken beseitigt werden. Ohne diese Abhängigkeiten sind Tests leichter zu entwickeln, sodass weitere Änderungen, zum Beispiel zur Verbesserung des Designs, später einfacher und komfortabler möglich sind. Das Ziel beim Brechen der Abhängigkeiten ist es, die Testbarkeit der betroffenen Klassen beziehungsweise Methoden zu erhöhen.

## 1.1 ExtractInterface bei Commit 1e47e7b

### Ausgangszustand

Die Klasse *AuthHandler* ist stark abhängig von der Android-Klasse *Activity* und kann auch nicht erstellt werden ohne eine Instanz dieser Klasse der *getInstance*-Methode zu übergeben. Die Testbarkeit ist schlecht, da eine Android-*Activity*-Instanz nicht ohne weiteres im Test-Kontext erstellt werden kann. Auch ein Fake-Objekt ist schwer zu erstellen, da eine finale Methode genutzt wird, in Ableitungen von *Activity* nicht überschrieben werden kann. Da die Klasse *AuthHandler* für die OAuth-Authentifizierung verantwortlich ist und damit wichtig für das Gesamtsystem, ist es wichtig, Tests für diese Klasse zu ermöglichen.

### Gewählte Technik

Die *getInstance*-Methode benötigt eine Instanz der Klasse *Activity*, die im Test-Kontext nicht leicht erstellbar ist und für die auch Fake-Objekte nur schwer erstellt werden können. Da allerdings nur wenige Methoden der *Activity*-Klasse verwendet werden, und eine Änderung der *AuthHandler* Klasse nur vergleichsweise wenige Änderungen erfordert, kann in diesem Fall **ExtractInterface** angewendet werden. So wird die Abhängigkeit von *AuthHandler* zu *Activity* gelöst, indem das Interface *AuthHandlerActivity* eingeführt und der Parameter von *getInstance* angepasst wird, sodass eine Instanz vom Typ des Interfaces verwendet wird. Dabei werden auch kleinere Anpassungen an Methodenaufrufen vorgenommen, sodass Methoden des Interfaces aufgerufen werden. **ExtractInterface** ist durch IDE-Unterstützung vergleichsweise einfach und ohne große Fehleranfälligkeit durchzuführen. Außerdem wird die Abstraktion durch diese Technik verbessert.

### Endzustand

*AuthHandler* nutzt nach dieser Änderung nur noch die Methoden des Interfaces, wodurch die Testbarkeit erhöht wird. Allerdings gibt es hier das Problem, dass *AuthHandler* die Third-Party-Klasse *AuthorizationService* verwendet, die auf eine *Activity*-Instanz angewiesen ist. Deshalb gibt es im neu eingeführten Interface eine Methode, die eine *Activity* zurückgibt. Die Abhängigkeit konnte also in diesem Fall nicht komplett aufgelöst werden. Trotzdem wurde die Testbarkeit erhöht und die Abstraktion verbessert. Diese Änderung stellt damit eine gute Basis für weitere Refactorings und Änderungen dar, wie zum Beispiel in Commit cbc5318.