

## SoSe 2025 – Softwaretechnologie I

# U08 – Buildsysteme und Testen von Objektnetzen

**Kursverantwortlicher:** Prof. Uwe Aßmann

**Übungsleiter:** Dr. Sebastian Götz

### Wichtiger Hinweis zur Vorbereitung

Bitte starten Sie **rechtzeitig vor Beginn der Übung** den Maven-Build-Prozess einmalig mit `./mvnw test` (Linux/Mac/Windows-PowerShell) bzw. `mvnw test` (Windows-Cmd) im entpackten Übungsordner (aus *Aktien\_und\_Investoren.zip*). Beim allerersten Aufruf dieser Kommandos werden zunächst alle Abhängigkeiten des Maven-Werkzeuges aus dem zentralen Maven-Repository heruntergeladen. Dies erfordert viel Zeit und muss deshalb in Vorbereitung von **Aufgabe 2** durchgeführt werden. Während der Übung kann hierfür keine Zeit aufgewendet werden.

### Inhalt der Übung

- Arbeit mit dem Buildsystem Maven
- Testen von Objektnetzen mit JUnit 5

## Aufgabe 1 (Aktien und Investoren)

Für ein Finanzunternehmen soll eine Software bereitgestellt werden, die es Investoren erlaubt, die Entwicklung von Aktienkursen zu überwachen und auf Änderungen dieser schnell zu reagieren. Weiterhin soll die Software leicht zu entwickeln sein und gut getestet vorliegen.

- Laden Sie sich das bereitgestellte Projekt herunter und machen Sie sich die Codestruktur (zu finden unter `src/main/java/`) mittels eines dUML Klassendiagramms klar.
- Welches Entwurfsmuster erkennen Sie? Zeichnen Sie es in das Diagramm ein.

## Aufgabe 2 (Arbeiten mit Maven)

Um die Softwareentwicklung und Einbettung eines aktuellen Testframeworks zu vereinfachen, soll das in der Java-Community verbreitete Buildsystem *Maven* genutzt werden.

- Machen Sie sich mit der Arbeitsweise und Vorteilen von Maven vertraut. Weiterführende Links finden Sie am Ende des Übungsblattes.
- Bauen und testen Sie das gegebene Maven-Projekt auf der Kommandozeile (`./mvnw compile` und `./mvnw test`).
- Importieren Sie das Projekt in eine IDE (zum Beispiel Eclipse).
- Kompletieren Sie die Implementierung des von Ihnen gefundenen Entwurfsmusters. Dabei sollen interessierte Parteien nur bei negativen Preisänderungen informiert werden.

## Aufgabe 3 (Testen von Objektnetzen)

Da in dem Projekt viele Klassen miteinander kommunizieren müssen, reichen einfache Unittests von Klassen und Methoden nicht aus, um eine gute Qualität zu garantieren. Hierfür muss auch das Objektnetz, also die Kommunikation mehrerer Klassen, getestet werden. Nachdem wir in Übung U03 schon JUnit 3 und 4 kennengelernt haben, nutzen wir in dieser Übung die aktuelle Version 5 des JUnit-Frameworks für die Implementierung der Tests.

- Überlegen Sie sich sinnvolle abhängige Klassen/Methoden, welche getestet werden müssten.
- Überlegen Sie sich Testfälle für diese Teilnetze und notieren Sie diese in einer Testfall-tabelle.
- Implementieren Sie die Testfälle als JUnit-5-Tests.

## Hausaufgaben

Erweitern Sie das Projekt mit einer weiteren externen Bibliothek, die nicht den Dependency-Scope `test` nutzt.

- **Möglichkeit 1:** Nutzen Sie die Klasse `Money` der Bibliothek [Moneta](#) für die Geldwerte des Projektes.
- **Möglichkeit 2:** Nutzen Sie statt `Map` die Datenstruktur `Bag` aus dem [Apache Commons Collections](#) Framework.

## Empfehlungen zum Selbststudium

- JUnit 5: [Homepage](#) / [Getting Started](#)
- Maven: [Homepage](#) / [Getting Started](#)