



# Chess of Duty

## Advanced Software-Engineering

im Rahmen der Prüfung zum  
**Bachelor of Science (B.Sc.)**

des Studienganges Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

**Clemens Richter & Johannes Peters**

Abgabedatum: 30. April 2023  
Bearbeitungszeitraum: 04.10.2022 - 30.04.2023  
Matrikelnummer, Kurs: xxxxxxx & 5802185, TINF20B1  
Betreuer der Arbeit: Herr Daniel Lindner

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Domain Driven Design</b>	<b>2</b>
2.1	Ubiquitous Language . . . . .	2
2.2	Domainenbausteine . . . . .	2
2.3	Domainenevents . . . . .	3
<b>3</b>	<b>Clean Architecture</b>	<b>4</b>

# 1 Einleitung

Im Rahmen des Moduls „Advanced Software Engineering“ wurde ein Schachspiel als Projektgrundlage ausgewählt. Chess of Duty ist ein Offline-Multiplayer-Schachspiel für zwei Personen. Das Hauptziel des Programmentwurfs besteht darin, Schach gemäß den Standardregeln zu implementieren. Dabei soll ein Leaderboard für die Spieler eingeführt werden, das auf einem Elo-System basiert.

Der Nutzen des Schachspiels für unsere Kunden entspricht dem von anderen Videospielen. Die Anwendung dient ausschließlich der Unterhaltung der Nutzer. Zusätzlich können die Anwender ihre strategischen Fähigkeiten und logisches Denken trainieren.

Das Schachspiel wird objektorientiert konzipiert und in Processing programmiert. Processing ist eine Open-Source-Programmiersprache, die auf Java basiert und einen besonderen Schwerpunkt auf die einfache Erstellung von Grafiken und Animationen setzt. Dadurch eignet sich Processing besonders für die Gestaltung interaktiver Benutzeroberflächen.

## 2 Domain Driven Design

### 2.1 Ubiquitous Language

Der Auftraggeber des Programmentwurfs ist ein deutscher Kunde. Obwohl in der Programmierung Englisch als die Standardsprache angenommen wird, wird aufgrund der Kundenlokalität die Projektsprache als „Deutsch“ festgelegt. Dadurch wird versucht die meisten Ausdrücke aus der Domäne ins Deutsche zu übernehmen. Auch wenn dies für den ungeübten Programmierer, der das Programmieren nur in Englisch ausübt, eine zusätzliche Herausforderung darstellt und teilweise zu Namensfindungsschwierigkeiten oder längeren Funktionsnamen führen kann, wird an der Domänensprache festgehalten, um dem Kunden den Code so übersichtlich wie möglich übergeben zu können.

### 2.2 Domainenbausteine

#### Value Objects

Ein Value Object ist ein unveränderliches Objekt, das einen bestimmten Wert repräsentiert und keine eigene Identität besitzt. Es definiert sich durch seine Eigenschaften und nicht durch eine eindeutige Identität, wodurch es austauschbar und vergleichbar wird.

Ein **Schachzug** setzt sich aus einer ausgewählten Figur, der Startposition, der Endposition und weiteren Informationen über den Einfluss der Bewegung der betreffenden Figur auf umliegende Figuren zusammen. Dazu zählen Informationen darüber, ob eine andere Figur geschlagen wird, ob durch den Zug Schach geboten wird, ob eine Bauernumwandlung stattfindet oder ob andere Figuren ebenfalls bewegt werden müssen, wie es bei der Rochade der Fall ist. Die Implementierung eines Schachzugs dient der Protokollierung einer gespielten Schachpartie. Ein Schachzug wurde als Value Object klassifiziert, da er keinen erkennbaren Lebenszyklus besitzt und seine Eigenschaften nach der Erstellung nicht mehr verändert werden.

- Schachfeld:
- Schachbrett

## **Entities**

- Schachbrett
- Figuren (König, Dame, Turm, Läufer, Springer, Bauer)
- Spieler
- Schachspiel
- Spielzug:

## **Aggregate**

## **Repositories**

## **2.3 Domainenevents**

## **3 Clean Architecture**