

Das Simpsons-Quiz: Finde deinen Zwilling in Springfield!

Programmentwurf

von

Dominik Veith

Abgabedatum: 04. April 2023

Bearbeitungszeitraum: 04.10.2022 - 04.04.2023

Matrikelnummer, Kurs: 3352220, TINF20B2

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Quellcodeverzeichnis	VI
1 Einführung	1
1.1 Übersicht über die Applikation	1
1.2 Start der Applikation	2
1.3 Testen der Applikation	2
2 Clean Architecture	3
2.1 Was ist Clean Architecture?	3
2.2 Analyse der Dependency Rule	4
2.3 Analyse der Schichten	4
3 SOLID	5
3.1 Analyse Single-Responsibility-Principle (SRP)	5
3.2 Analyse Open-Closed-Principle (OCP)	5
3.3 Analyse Liskov-Substitution- (LSP), Interface-Segregation- (ISP), Dependency-Inversion-Principle (DIP)	5
4 Weitere Prinzipien	6
4.1 Analyse GRASP: Geringe Kopplung	6
4.2 Analyse GRASP: Hohe Kohäsion	6
4.3 Don't Repeat Yourself (DRY)	6
5 Unit Tests	7
5.1 10 Unit Tests	7
5.2 ATRIP: Automatic	7
5.3 ATRIP: Thorough	7
5.4 ATRIP: Professional	7
5.5 Code Coverage	7
5.6 Fakes und Mocks	7
6 Domain Driven Design	8
6.1 Entities	8

6.2	Value Objects	8
6.3	Aggregates	8
7	Refactoring	9
7.1	Code Smells	9
7.2	2 Refactorings	9
	Literaturverzeichnis	VII

Abkürzungsverzeichnis

JDK	Java Development Kit
IDE	Integrated Development Environment
ASCII	American Standard Code for Information Interchange

Abbildungsverzeichnis

2.1	Clean Architecture [2]	3
-----	----------------------------------	---

Tabellenverzeichnis

Quellcodeverzeichnis

1 Einführung

1.1 Übersicht über die Applikation

Die Fernsehserie 'Die Simpsons' ist eine US-amerikanische Zeichentrickserie, die seit 1989 ausgestrahlt wird und mittlerweile über 700 Episoden hat. Die Serie handelt von der Familie Simpson, bestehend aus Homer, Marge, Bart, Lisa und Maggie, sowie zahlreichen Nebenfiguren, die in der fiktiven Stadt Springfield leben. Die Simpsons zeichnet sich durch ihren satirischen und humorvollen Stil aus, der sowohl politische als auch soziale Themen behandelt. Die Serie parodiert oft bekannte Filme, Serien, Persönlichkeiten und Institutionen und enthält viele popkulturelle Referenzen. Die Simpsons wurde mit zahlreichen Preisen ausgezeichnet, darunter 34 Emmy Awards, und gilt als eine der erfolgreichsten Fernsehserien aller Zeiten. Die Serie wurde in mehr als 100 Ländern ausgestrahlt und hat eine große Fangemeinde auf der ganzen Welt.[1]

Die Applikation Simpsons-Quiz ist ein Terminal-basierendes Minispiel. Durch gezielte Fragen, welche User:innen mittels Tastatureingaben beantworten, soll bestimmt werden, durch welchen Charakter des Simpsons Universum er oder sie am ehesten repräsentiert wird. Zusätzlich werden Informationen über den Zielcharakter ausgegeben. Dies umfasst den Wohnort, den Arbeitsplatz, die Art der Fortbewegung und das Lieblingsessen des Charakters. Zusätzlich werden individuelle, charakterspezifische Fakten präsentiert.

Die visuelle Ausgabe im Terminal wird durch eine American Standard Code for Information Interchange (ASCII)- Repräsentation des verifizierten Charakterbildes unterstützt. Nachdem alle Fragen beantwortet wurden und der oder die User:in seinen Charakter mit Erläuterungen erhalten hat, werden alle Informationen zusätzlich in einer Textdatei abgelegt um sie später noch einmal nachlesen zu können.

1.2 Start der Applikation

Zum Start der Applikation sind folgende Voraussetzungen notwendig:

- Das Java Development Kit (JDK) um den Code kompilieren und auszuführen zu können.
- Ein Integrated Development Environment (IDE) um die Ausführung des Codes komfortabler zu gestalten.

Um die Applikation zu starten sollte der Code innerhalb einer IDE der Wahl geöffnet werden. Danach muss die Java Klasse 'SimpsonsTerminal' im Ordner 'SimpsonsGame/src/main/java/de/dhbw/ase/simpsons/plugin' ausgeführt werden. Alle Interaktionen der Applikation mit dem User erfolgen anhand einer textbasierten Ausgabe über das Terminal der IDE.

1.3 Testen der Applikation

Um die Applikation zu testen gelten folgende Voraussetzungen:

- Maven ist installiert
- Optional: Die IDE IntelliJ ist installiert

Zum Start des Test ist sicherzustellen, dass der Code in der IDE der Wahl geöffnet ist. Danach können alle Tests-Klassen im Verzeichnis unter: SimpsonsGame/src/test/java/de/dhbw/ase/simpsons gefunden und einzeln ausgeführt werden.

Alternativ kann im Wurzelverzeichnis 'SimpsonsGame' auch die Kommandozeile geöffnet werden und mit dem Befehl 'mvn test' alle Tests ausgeführt werden.

2 Clean Architecture

2.1 Was ist Clean Architecture?

Die Clean Architecture ist ein Architekturmuster, das sich auf die Trennung von Verantwortlichkeiten und die Abhängigkeiten zwischen den Schichten konzentriert. Die Architektur besteht aus mehreren Schichten, die von innen nach außen, wie in Abbildung 2.1 zu sehen, angeordnet sind:

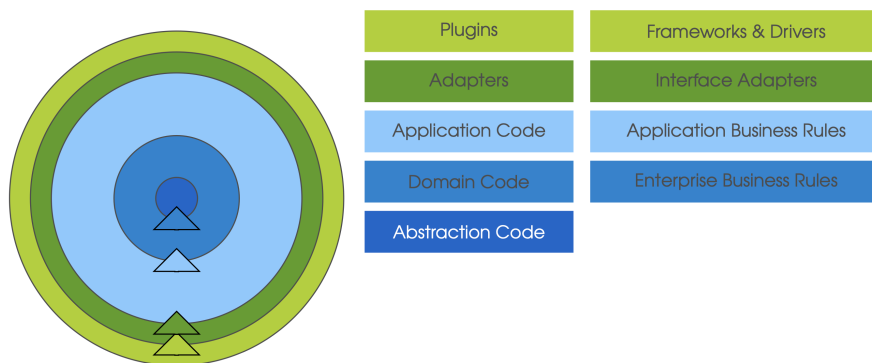


Abbildung 2.1: Clean Architecture [2]

1. **Abstraction Code (Schicht 4):** Code, welcher Konzepte, grundlegende Algorithmen und Datenstrukturen implementiert. Dieser Code enthält Domänenübergreifendes Wissen und wird nur selten berührt.
2. **Domain Code (Schicht 3):** Diese Schicht enthält die Kernlogik der Anwendung und wird am seltensten geändert. Hier werden die Geschäftsregeln und -modelle definiert. Die Entitäten der Domäne werden von der Application Schicht verwendet.
3. **Application Code (Schicht 2):** Diese Schicht enthält die Geschäftslogik der Anwendung (Use Cases welche direkt aus Anforderungen resultieren) und koordiniert den Datenfluss mit Hilfe der Entitäten der Domäne. Dabei sind die Regeln der anwendungsspezifischen Logik nicht projektweit gültig.
4. **Adapters (Schicht 1):**

5. Plugins(Schicht 0):

2.2 Analyse der Dependency Rule

2.2.1 Positiv-Beispiel: Dependency Rule

2.2.2 Negativ-Beispiel: Dependency Rule

2.3 Analyse der Schichten

3 SOLID

3.1 Analyse Single-Responsibility-Principle (SRP)

3.1.1 Positiv-Beispiel

3.1.2 Negativ-Beispiel

3.2 Analyse Open-Closed-Principle (OCP)

3.2.1 Positiv-Beispiel

3.2.2 Negativ-Beispiel

3.3 Analyse Liskov-Substitution- (LSP), Interface-Segregation- (ISP), Dependency-Inversion-Principle (DIP)

3.3.1 Positiv-Beispiel

3.3.2 Negativ-Beispiel

4 Weitere Prinzipien

4.1 Analyse GRASP: Geringe Kopplung

4.1.1 Positiv-Beispiel

4.1.2 Negativ-Beispiel

4.2 Analyse GRASP: Hohe Kohäsion

4.3 Don't Repeat Yourself (DRY)

5 Unit Tests

5.1 10 Unit Tests

5.2 ATRIP: Automatic

5.3 ATRIP: Thorough

5.4 ATRIP: Professional

5.5 Code Coverage

5.6 Fakes und Mocks

6 Domain Driven Design

6.1 Entities

6.2 Value Objects

6.3 Aggregates

7 Refactoring

7.1 Code Smells

7.2 2 Refactorings

Charakter klassen entzerzt mit einführen einer Superklasse und einem Interface

Literaturverzeichnis

- [1] Kelly, S. *Die Simpsons*. Hrsg. von IMDB. <https://www.imdb.com/title/tt0096697/>. o.O., 2023. (Einsichtnahme: 19.03.2023).
- [2] Briem, L. *Clean Architecture*. Hrsg. von DHBW. <https://github.com/briemla/asetinf20/blob/main/briem/clean-architecture.pdf>. o.O., 2023. (Einsichtnahme: 19.03.2023).