

**Praxisprojekt im Rahmen von  
Advanced Software Engineering**

**Technische Dokumentation zum Programmentwurf**

**Weeping Snake**

Duale Hochschule Baden-Württemberg Karlsruhe  
Fakultät Technik

Studiengang Informatik – Informatik

von

Silas Mario Schnurr

18.05.2021



**Kurs:** TINF18B5

**Dozent:**  Herr Maurice Müller

**Erklärung zur Eigenleistung**

Gemäß § 5 (3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 22. September 2011.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

----------------------------------- --------------------------

Ort Datum Unterschrift

**Inhaltsverzeichnis**

[Abkürzungsverzeichnis / Glossar IV](#_Toc72236654)

[Abbildungs- und Tabellenverzeichnis V](#_Toc72236655)

[1 Projektbeschreibung 1](#_Toc72236656)

[2 Unit Tests 1](#_Toc72236657)

[Literaturverzeichnis VIII](#_Toc72236658)

[Anhangsverzeichnis IX](#_Toc72236659)

# Abkürzungsverzeichnis / Glossar

**API** Application Programming Interface, Programmierschnittstelle

# Abbildungs- und Tabellenverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# Projektbeschreibung

Das ist ein Demo-Text

enthält Ergebnisse der Analysen und Begründungen der Entscheidungen • Code-Struktur als UML-Diagramme notiert – Detailgrad an die Aussagen anpassen

Code und schriftliche Dokumentation

Objektorientierte Mainstream Programmiersprache C# .Net 5.0

> 2k Zeilen Code (derzeit 1.4k)

> 20 Klassen

klar definierter (sinnvoller) Nutzen

* Spielen, gegner rammen (5 runden lang) gibt punkte

Desktopanwendung

# Domain Driven Design

* Analyse der Ubiquitous Language mit >= 5 Beispielen
* Analyse und Begründung von
  + Repositories
  + Aggregates
  + Entities
  + Value Objects

# Programming Principles

## SOLID

Analyse und Begründung

## GRASP

Analyse und Begründung

(insbes. Kopplung und Kohäsion)

## DRY

Analyse und Begründung

# Entwurfsmuster

* >= 1 Entwurfsmuster einsetzen und begründen
* UML (vorher/)nachher

# Clean Architecture

* >= 1 Umsetzung einer Klasse der Adapterschicht
* Begründung

# Legacy Code

* >= 2 Abhängigkeiten mit Techniken aus der Vorlesung brechen
* ausgewählte Stellen und Techniken begründen

# Refactoring

* Code Smells identifizieren
* >= 2 Refactorings anwenden und begründen
* für 2 Stelle vorher/nacher UML

# Unit Tests

* >= 10 Unit Tests
* ATRIP-Regeln
* Code Coverage
* Einsatz von Fake-/MockObjekten (ohne Frameworks wie z.B. Mockito)
* Analyse und Begründung für
  + Umfang der Tests
  + Einsatz der Fake/MockObjekte

# Zusatz: API-Design

* Bereitstellung einer sinnvollen API
  + Programmiersprachenebene oder http
  + >= 10 Methoden / HTTP Endpunkte
* Analyse der API anhand der Qualitätsmerkmale und Begründung für das Design

# Literaturverzeichnis

**Im aktuellen Dokument sind keine Quellen vorhanden.**J

# Anhangsverzeichnis

[**Anhang 1: Beispielanhang** X](#_Toc67738685)

**Anhang 1: Beispielanhang**

Die Nummer 1 muss für weitere Anhänge kopiert werden, damit diese korrekt im Anhangsverzeichnis angezeigt werden.