

DUALE HOCHSCHULE BADEN-WÜRTTEMBERG

ADVANCED SOFTWARE ENGINEERING 2

—

DOKUMENTATION

# PIC-Simulator

*David Eymann, Tom Wagner*

Dozent  
Daniel LINDNER

23. September 2022

# Inhaltsverzeichnis

<b>1</b>	<b>Projektbeschreibung</b>	<b>2</b>
1.1	Einrichtung . . . . .	2
<b>2</b>	<b>Entwicklung</b>	<b>3</b>
2.1	Clean Architecture . . . . .	4
2.2	Refactoring . . . . .	4
2.3	Unit Tests . . . . .	4
2.3.1	Einsatz von Mocks . . . . .	4
2.3.2	ATRIP-Regeln . . . . .	4
2.4	Programming Principles . . . . .	4
2.4.1	SOLID . . . . .	4
2.4.2	GRASP . . . . .	4
2.4.3	DRY – Don’t Repeat Yourself . . . . .	4

# Abbildungsverzeichnis

# Listings

# Kapitel 1

## Projektbeschreibung

Als Basis für diese Arbeit dient ein PIC-Simulator, PIC steht hierbei für ein Mikrocontroller von Microchip Technology<sup>1</sup>. Der Simulator ist in C-# geschrieben und mit Windows Forms erhält er seine grafische Oberfläche. **Das für diese Abgabe relevante Repository befindet sich unter:**

[https://github.com/tomwgnr/ASE-PIC\\_Simulator](https://github.com/tomwgnr/ASE-PIC_Simulator)

### 1.1 Einrichtung

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Microchip\\_Technology](https://en.wikipedia.org/wiki/Microchip_Technology)



# Kapitel 2

## Entwicklung

### 2.1 Clean Architecture

### 2.2 Refactoring

### 2.3 Unit Tests

#### 2.3.1 Einsatz von Mocks

#### 2.3.2 ATRIP-Regeln

Automatic

Thorough

Repeatable

Independent

Professional

### 2.4 Programming Principles

#### 2.4.1 SOLID

Single responsibility principle

Open/Closed principle

Liskov substitution principle

Interface segregation principle

Dependency inversion principle

#### 2.4.2 GRASP

High Cohesion

Low Coupling