

# WeatherWallpaper

Janis Fix, Leon Gieringer

TINF18B3

Advanced Software Engineering

16. Mai 2021

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Clean Architecture</b>	<b>2</b>
<b>3</b>	<b>Entwurfsmuster</b>	<b>2</b>
<b>4</b>	<b>Programming Principles</b>	<b>2</b>
4.1	SOLID . . . . .	2
4.1.1	Single Responsibility Principle . . . . .	2
4.1.2	Open/Closed Principle . . . . .	2
4.1.3	Liskov Substitution Principle . . . . .	2
4.1.4	Interface Segregation Principle . . . . .	2
4.1.5	Dependency Inversion Principle . . . . .	2
4.2	GRASP . . . . .	2
4.3	DRY . . . . .	2
<b>5</b>	<b>Refactoring</b>	<b>2</b>
<b>6</b>	<b>Unit Tests</b>	<b>2</b>

# **1 Einleitung**

Hier steht meine Einleitung

## 2 Clean Architecture

### 2.1 Vorher

### 2.2 Nachher

- Schichtarchitektur planen und begründen
- $\geq 2$  Schichten umsetzen

## 3 Entwurfsmuster

- $\geq 1$  Entwurfsmuster einsetzen und begründen
- UML-Diagramm vorher und nachher

## 4 Programming Principles

### 4.1 SOLID

#### 4.1.1 Single Responsibility Principle

#### 4.1.2 Open/Closed Principle

#### 4.1.3 Liskov Substitution Principle

Das Liskov Substitution Principle ist erfüllt, da abgesehen von den verwendeten Interfaces keine Vererbung verwendet wird.

#### 4.1.4 Interface Segregation Principle

#### 4.1.5 Dependency Inversion Principle

### 4.2 GRASP

### 4.3 DRY

## 5 Refactoring

- Code Smells identifizieren
- $\geq 2$  Refactoring anwenden und begründen

## 6 Unit Tests

Insgesamt wurden 29 Unit-Test geschrieben.

### 6.1 ATRIP

### 6.2 Mocks

### 6.3 Code Coverage

- $\geq 10$  Unit Tests
- ATRIP-Regeln

- Code Coverage
- Einsatz von Mocks

## Literaturverzeichnis

- [1] Foo (2021). bar.