

MVVM: Model-View-ViewModel

Advanced Softwareengineering - DHBW Stuttgart

Nico Vogel, Lukas Sopora

31.12.2019

Inhaltsverzeichnis

1	Was ist das Problem das MVVM angeht?	2
2	Beschreibung der Komponenten von MVVM	3
2.1	Model	4
2.2	ViewModel	4
2.3	View	4
3	Beispiel des MVVM Patterns in C# WPF	5
4	Vergleich zu dem MVC Pattern	6
5	Vergleich zu dem MVP Pattern	7
6	Kritische betrachtung des MVVM Patterns	8

1 Was ist das Problem das MVVM angeht?

- ui fix coupled mit logik
- schwierig einzelne komponenten oder ganze ui auszutauschen
- cross platform nicht möglich
- wiederverwendung der logik nicht möglich

2 Beschreibung der Komponenten von MVVM

- Komponenten
 - Model
 - * Entität
 - * enthält nur daten oder logik für sich selbst
 - * Programmlogik beschränkt sich auf Validierung
 - ViewModel
 - * business logik
 - * verwendet beliebig viele models
 - * stellt werte bereit für die ui
 - * läßt werte ein aus der ui
 - * leitet events weiter an die logik
 - View
 - * nur design
 - * eigenständig
 - * kennt ViewModel
 - Übersicht komponenten
 - Übersicht einteilung in das Application Layered model
- Kommunikation
 - OneWay Binding: von View zu ViewModel oder von ViewModel zu View
 - TwoWay Binding: Von der View zum ViewModel und gleichzeitig auch andersrum
 - Events: die view reicht ein event an das ViewModel weiter

MVVM besteht aus mehreren Komponenten die ineinandergreifen. Die drei größten sind Model, View und ViewModel. Dabei wird das Zusammenspiel der Komponenten in der Abbildung X dargestellt. Die View und das ViewModel können Bidirektional kommunizieren, während das ViewModel nur das Model aufruft beziehungsweise verwendet.

ABILDUNG X

2.1 Model

Das Model umfasst nur Daten und maximal noch Logik für die Daten. Im C# Umfeld spricht man auch von POCO (Plain old Control Object) oder in Java POJO (Plain old Java Object).

Das folgende Beispiel beschreib eine Schülerklasse die der Definition eines Models entspricht.

CODE BEISPIEL SCHÜLER

2.2 ViewModel

Mithilfe des ViewModels werden Informationen und Funktionen zusammengefasst in eine Klasse, damit eine View diese verwenden kann. Das ViewModel verwenden zur Informationsbereitstellung die Models. Die bereitgestellten Funktionen sind entweder in dem ViewModel selbst oder in separaten Klassen implementiert.

2.3 View

Die View ist ausschließlich zur Darstellung der Inhalte und bereitstellen der Funktionen über Schaltflächen oder ähnliches gedacht. Dabei wird in der View Binding verwendet um Inhalte und Funktionen von dem ViewModel zu verwenden.

3 Beispiel des MVVM Patterns in C# WPF

4 Vergleich zu dem MVC Pattern

5 Vergleich zu dem MVP Pattern

6 Kritische betrachtung des MVVM Patterns