

2048

03.06.22

# Mein Wochenplan

2048

V2

Woche Nr.

von \_\_\_\_\_ bis \_\_\_\_\_

Uhr	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
22 08.00							
23 09.00							
24 10.00	Review morgens		Algo: Zufällig Vorstellung		C - Vorlesung		
25 11.00	Review morgens				C - Vorlesung		
26 12.00	Review morgens		Algo: Kooperativ Gegenspieler		C - Vorlesung		
27 13.00	Review morgens				Fertig / Reserve		
28 14.00	Review morgens		Präsentation erstellen		Abgabe		
15.00					(schriftlich)		
16.00							
17.00							
18.00							
19.00							
20.00							
21.00							
22.00							

PRÜFUNGEN

Abgabe

# Agenda

03.06.22  
Beteiligt: Dani  
Emma  
Nina

- ✓ Anforderungen an das Programm (Frontend, Backend)
    - UML-Diagramm **ausstehend**
  - ✓ GitHub
    - Code with me Strategie Änderung
  - ✓ Pattern
  - ✓ Aussuchen Vorlage → YouTube Video siehe Teams
    - Vorlage durchgehen durch Strategie Änderung hinfällig
    - Gegenspieler mit Uhl gesprochen siehe Anforderungen
  - ✓ Aufgabenteilung überdenken
  - ✓ Abstimmung Software-Typ (App, **PC-Software**)
  - ✓ Einheitlicher Code
- 
- Jeden Dienstag Morgen Besprechung / Review!

# Anforderungen an das Programm

03.06.22

## Backend

- Klasse View-Model
- Random Zahlen-Blöcke einfügen
- { - Muss 2er & 4er einfügen
- Zufällig, Kooperativ, Gegenspieler am Anfang abfragen
- Addieren von 2 gleichen Zahlen beim Zusammenfügen
- Erkennung Spielende bei erreichen 2048 oder geht nicht mehr weiter
- Blöcke verschieben durch Pfeiltasten <sup>abbruch</sup> oder „WASD“ Tasten

## Frontend

- Blöcke
- Layout 4x4
- Farben
- Home - Button
- Werbung für Florians Programm

## Unbekannt ob Frontend oder Backend

- Bewegung Pfeiltasten (Backend?)

## Wunschziele

- Punktzahl
- Rekord
- Bei 2048 weiter spielen
- Werbung erstellen

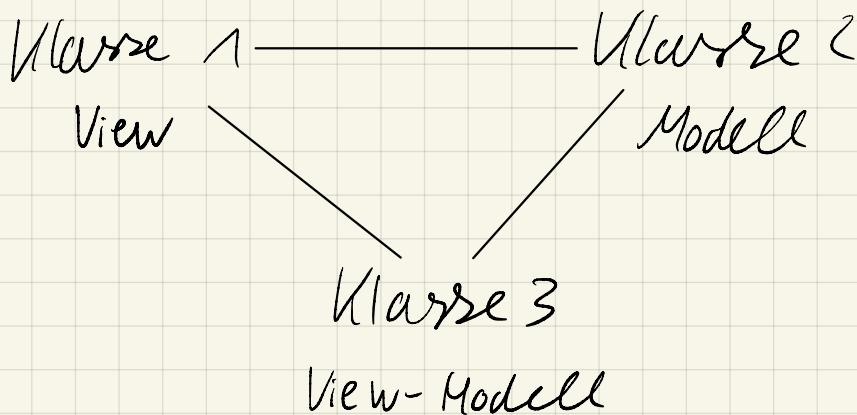
- Programm verstehen bevor es geschrieben wird

### Klasse:

Verallgemeinerung gemeinsamer Objekte aus der realen Welt  
 Auto → Farbe, Marke

Objekt: Konkrete Ausprägung einer Klasse

Blauer BMW



Klasse
Attribute
Methoden

klein  
 attributname: Datentyp  
 klein  
 methodename()

# KLASSEN-DIAGRAMME

## Aufbau:

Sichtbarkeit:

public +

private -

protected # (innerhalb Klasse)

package ~  
(selbes Pachet)

Vererbung

### Klassenname

attributname: Datentyp

→ weil immer gleich!

klassen attributname : Datentyp (unterstriche z.B. anzahlraeder: int)

methodename (attributname: Datentyp) : Datentyp

klassen methode:

Eingabeparameter

Rückgabetyp

▶ 2:52 / 7:31 • Aufbau >

### Auto

marke: String

sitzplaetze: integer

farbe: String

motorStarten()

getFahrer(): String

### Kino

+ anzahlMitarbeiter: integer

+ name: String

+ anzahlSitzplaetze: integer

- umsatz: integer

- filmStarten()

+ popcornVerkaufen()

+ kartenVerkaufen()

# UML - Ausarbeitung

03.06.22

View (Board)

# Aufgabenteilung überdenken

Möglichkeit 1:

Frontend	Backend	Zusammenfassung
1 Person	1 Person	2 Personen

Möglichkeit 2:

Frontend	Backend	Zusammenfassung
2 Personen	2 Personen	2 Personen
		→

Möglichkeit 3:

Grundgerüst (Neutral)	Gegenspieler	Kooperativ
Alle	2 Personen	2 Personen

Möglichkeit : Wind angewendet

Grundgerüst (Neutral)	Gegenspieler	Kooperativ
2 Personen	1 Person	1 Person

# Einheitlicher Code

03.06.22

- Alles auf Englisch schreiben  
(weil get, set...)

Pattern: Weicht evtl. ab, da Strategieänderung

Wie man Code aufteilt

Emma: MVVM

Modell	View	View-Model
Logik	Was man sieht	Was View kontrolliert
nicht View	sieht	
kennt View nicht	kennt	
kennt View nicht	Modell View nicht	

(Dennis)      Emma      Dennis

connect Dani, Nina

2 Klassen

evtl. 3. Klasse Model

## Algorithmen

Zufällig

Fügt Blöcke Random ein

Kooperativ

Fügt Blöcke Nett ein

Vor Spiel kann man auswählen, ob kooperativ oder Gegenspieler

Gegenspieler

Fügt Blöcke „Assi“ ein