# 1. Was soll ich machen? - Grundaufgabenstellung

## Stufe 1. Basic:

* Erstellung einer spielfähigen Runde 4 gewinnt (7x6-Feld)
  + **1 vs. 1** oder besser noch **1 vs. COM**

## Stufe 2.

* Über mehrere Runden (optional)

## Stufe 3:

* Mit GUI (optional)

# 2. Offene Fragen/Probleme

Was könnte bei der Umsetzung zu Problemen führen? Gibt es noch implizite oder unklare Anforderungen?

* Soll der Computer zufällig spielen? (Dürfte leichter sein als strategisch)
* Sollen wir uns den Jux erlauben und zulassen, dass 2 COMs spielen können?

# 3. Welche Klassen, Methoden, Variablen, Exceptions?

## Klassen:

**Player**

2 Spieler, davon der zweite menschlich oder Computer

**Board**

Dort werden die Steine gesetzt. Jedes Board entspricht einer Runde.

**Verifier**

statische Hilfssammelklasse für alle „Wurde-schon-gewonnen?-Methoden“

**Gameplay** (Main) (static final)

Hier wird der Spielablauf implementiert, später mehrere Runden

## Variablen:

**Player**:

private final String NAME;

private int Counters = 21;

private final char COUNTER\_STYLE = '●' / '○' (oder final String für je ein Emoji (Länge=2)

private int gamesWon = 0

private boolean isCOM

**Board**:

final char[][] = new char[6][7];

Verifier verifier;

**Gameplay** (Main):

Player player1

Player player2

Board board

**Verifier**:

int row

int col

## Methoden:

**Player**:

setCounter(Board board) //random, wenn isCOM = true

setCounter(Board board, Scanner scanner) //manuell, wenn isCOM = false

**Board**:

void displayBoard()

boolean isAnyCounterLeft()

boolean isPileFull(int col)

**Verifier**:

boolean isAnyMatchFour(int row, int col) //invokes following submethods

boolean isUp(), isDown(), isLeft(), isRight()

boolean isUpLeft (), isDownLeft(), isUpRight(), isDownRight()

stattdessen isUpDown(), is isLeftRight(), isDiagonal() (oder in zwei)

**Gameplay** (Main):

boolean placeCounter(Board board, Player player) //achte auf Fallen der Steine

Player createPlayer() //Name, Counterstyle, CPU

int howManyGames() //1,3,5,…

void game()

## Exceptions:

**Player**:

InvalidNameException

InvalidCounterStyleException

**Board**:

InvalidPileException

**Verifier**:

InvalidRowOrColException

**Gameplay** (Main):

InvalidColException