



## Übung zur Vorlesung *Einführung in die Informatik 2 für Ingenieure (MSE)*

Christoph Anneser (anneser@in.tum.de)

<http://db.in.tum.de/teaching/ss23/ei2/>

### Blatt Nr. 1

Dies ist ein **optionales Zusatzblatt** mit extra Aufgaben, um etwas Inspiration zu geben falls Sie Java üben möchten oder einfach nur etwas spaßige Unterhaltung suchen.

#### Aufgabe 1: Ein Würfelspiel (leicht)

Wir wollen ein simples Würfelspiel simulieren. Es gibt 2 Spieler. Jeder Spieler hat einen dreiseitigen Würfel mit den Augenzahlen 2, 3 und 5. Die Spieler werfen abwechselnd ihren Würfel. Der Spieler der in Summe zuerst mehr als 12 Augenzahlen geworfen hat, gewinnt.

1. Erstellen Sie ein UML Klassendiagramm welches mindestens die folgenden Klassen beinhaltet: Würfel und Spieler.
2. Implementieren Sie das Beschriebene Spiel in Java.
3. Es ist klar, dass der Spieler der anfängt einen Vorteil hat, da dieser immer zuerst eine zusätzliche Zahl auf seine Punkte aufaddieren kann. Lassen Sie ihr Programm das Spiel einige hundert oder tausend male simulieren und berechnen Sie damit die Wahrscheinlichkeit dass der Spieler der anfängt gewinnt.

#### Aufgabe 2: Ein Kartenspiel (mittelschwer)

Wir wollen ein simples Kartenspiel simulieren. Das Kartendeck besteht aus 32 Karten. Angelehnt an das Deutsches Blatt gibt es 4 Farben: Eichel, Grass, Herz, Schellen. Der Einfachheit halber gibt es aber nur Zahlenkarten von 1 bis 8. Unser simples Kartenspiel wird von zwei Spielern gespielt. Dabei wird der Stapel zu Beginn gemischt. Danach werden von den beiden Spieler abwechselnd Karten gezogen. Ein Spieler gewinnt sobald er 3 aufeinanderfolgende Karten der gleichen Farbe (z.B.: Schellen 3,4 und 5) auf der Hand hat. Wenn es keine Karten mehr auf dem Stapel gibt und keiner gewonnen hat ist es ein unentschieden.

1. Erstellen Sie ein UML Klassendiagramm welches mindestens die folgenden Klassen beinhaltet: Karte, Deck und Spieler.
2. Implementieren Sie das beschriebene Spiel in Java.
3. Es ist klar, dass der Spieler der anfängt zu ziehen einen Vorteil hat, da dieser immer zuerst eine zusätzliche Karte auf der Hand hat. Lassen Sie ihr Programm das Spiel einige hundert oder tausend male simulieren und berechnen Sie damit die Wahrscheinlichkeit dass der Spieler der anfängt gewinnt.