

# 252-0027-00: Einführung in die Programmierung

## Übungsblatt 3

Abgabe: 15. Oktober 2019, 10:00

Checken Sie mit Eclipse die neue Vorlage aus, indem Sie wie in Übung 1 und 2 im Menü zuerst auf ihr Repository in der Repository-Ansicht rechtsklicken und *Pull* wählen. Danach Importieren Sie das neue Projekt durch Rechtsklick auf *u03* und Auswahl von *Import Projects...*

Beachten Sie, dass Sie in dieser Übung mehrere unabhängige Programme im selben Eclipse-Projekt haben werden. Bevor Sie ein Programm starten, achten Sie deshalb darauf, dass Sie die richtige Datei im Package Explorer ausgewählt oder im Editor geöffnet haben. *Vergessen Sie nicht, Ihren Programmcode zu kommentieren!*

### Aufgabe 1: Folgen und Reihen

Schreiben Sie ein Programm "Reihe.java", das eine natürliche Zahl  $N$  (inklusive Null) von der Konsole einliest und dann die folgende Summe  $S$  berechnet:

$$S = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{N^2}$$

**Beispiel:** Für  $N = 4$  sollte Ihr Programm ca. 1.42 ausgeben. Wie verhält sich diese Summe für grosse  $N$ ? (Für  $N = 0$  soll das Programm 0.0 ausgeben.)

### Aufgabe 2: Binärdarstellung

Schreiben Sie ein Programm "Binaer.java", das eine positive Zahl  $Z$  einliest und dann die Binärdarstellung druckt. (Hinweis: Finden Sie zuerst die grösste Zahl  $k$ , so dass die Zweierpotenz  $K = 2^k$  kleiner als  $Z$  ist.)

**Beispiel:** Für  $Z = 14$  sollte Ihr Programm 1110 ausgeben.



[xkcd: Su Doku](#) by Randall Munroe  
(CC BY-NC 2.5)

### Aufgabe 3: Grösster gemeinsamer Teiler

Schreiben Sie ein Programm "GGT.java", das den grössten gemeinsamen Teiler (ggT) zweier ganzer Zahlen mithilfe des Euklidischen Algorithmus berechnet. Hierbei handelt es sich um eine iterative Berechnung, die auf folgender Beobachtung basiert:

1. Wenn  $x$  grösser oder gleich  $y$  ist und sich  $x$  durch  $y$  teilen lässt, dann ist der ggT von  $x$  und  $y$  gleich  $y$ ;
2. andernfalls ist der ggT von  $x$  und  $y$  der gleiche wie der ggT von  $y$  und  $x \% y$ .

Üben Sie diesen Algorithmus zuerst von Hand an ein paar Beispielen und schreiben Sie dann das Java Programm.

**Beispiel:** Für  $x = 36$  und  $y = 44$ :

$$\text{ggT}(36, 44) \stackrel{2.}{=} \text{ggT}(44, \underbrace{36 \% 44}_{36}) \stackrel{2.}{=} \text{ggT}(36, \underbrace{44 \% 36}_{8}) \stackrel{2.}{=} \text{ggT}(8, \underbrace{36 \% 8}_{4}) \stackrel{1.}{=} 4$$

### Aufgabe 4: Zahlenerkennung

Für diese Aufgabe verwenden wir einen String um die erleuchtenden Segmente einer [Sieben-segmentanzeige](#) zu kodieren. Die Segmente sind, wie im Bild gezeigt, von a bis g nummeriert. Die Kodierung einer möglichen Anzeige ist ein String, in welchem der Buchstabe 'x' genau dann vorkommt, wenn das 'x'te Segment der Anzeige erleuchtet ist. Zum Beispiel wird die Zahl 2 kodiert durch 'abged'. Zur Einfachheit darf angenommen werden, dass kein Buchstabe mehr als einmal in der Kodierung vorkommt und dass nur die Zahlen 0 bis 9 kodiert werden.

Schreiben Sie ein Programm "Zahlen.java", das einen String, der eine Anzeige kodiert, einliest und die kodierte Zahl als Integer ausgibt. Überlegen Sie wie viele IF Blöcke benötigt werden um jede Zahl zu erkennen.

**Tipp:** Sie können `str.contains("a")` verwenden, um zu überprüfen, ob ein String `str` den Buchstaben 'a' enthält.

