



# Datenstrukturen und Algorithmen (SS 2013)

## Übungsblatt 1

Abgabe: Montag, **22.04.2013**, 14:00 Uhr

- Die Übungen sollen in Gruppen von zwei bis drei Personen bearbeitet werden.
- Schreiben Sie die Namen jedes Gruppenmitglieds sowie alle Matrikelnummern auf die abgegebenen Lösungen.
- Schreiben Sie die Namen jedes Gruppenmitglieds sowie alle Matrikelnummern auch in die Quellcode-Dateien.
- Geben Sie Ihre Lösungen am **Anfang** der Globalübung, montags, 14:00 Uhr, ab.
- Schicken Sie den jeweiligen Quellcode bitte per **E-Mail** direkt an Ihre/n Tutor/in.
- Geben Sie außerdem den ausgedruckten Quellcode zusammen mit den schriftlichen Lösungen ab.
- Zu spät abgegebene Lösungen werden nicht bewertet.
- Sofern nicht anders gefordert, müssen alle Lösungen und Zwischenschritte kommentiert werden.



### Aufgabe 1 (Fibonacci-Folge [20 Punkte])

Im Laufe des Übungsbetriebs, der begleitend zur Vorlesung stattfinden wird, werden Sie verschiedene Algorithmen und Datenstrukturen implementieren. Dafür verwenden wir die Programmiersprache *Java*. In der Regel werden wir Ihnen alle notwendigen Quelldateien zu Beginn der Globalübung zur Verfügung stellen. Dies beinhaltet die Datei `Main.java` sowie Dateien, in der die zum Problem gehörigen Klassen implementiert sind. Diese Klassen werden aus einem Grundgerüst inklusive Methodendeklarationen bestehen, welches dann an den dafür vorgesehenen Stellen durch Ihren Quellcode ergänzt werden soll. Bitte editieren Sie **niemals** die Datei `Main.java`, sondern immer nur die von uns bereitgestellten Problemklassen. Ändern Sie des Weiteren bitte **niemals** von uns bereitgestellte Methodensignaturen (also den Namen oder die Parameter und Rückgabewerte einer Methode). Sie dürfen selbstverständlich immer eigene Variablen und Methoden hinzufügen und benutzen, sofern Sie das von uns vorgegebene Grundgerüst nicht verändern.

Für den Einstieg in die Programmiersprache *Java* werden Sie in dieser Übung die *Fibonacci-Folge* implementieren. Die Fibonacci-Folge  $f_i$  lässt sich durch folgende Gleichung rekursiv definieren:

$$f_i = \begin{cases} 0 & \text{falls } i = 0 \\ 1 & \text{falls } i = 1 \\ f_{i-2} + f_{i-1} & \text{falls } i \geq 2 \end{cases}$$

Wir stellen Ihnen für diese Aufgabe die Klassen `Main.java` und `Fibonacci.java` zur Verfügung. Sofern Sie die Teilaufgaben implementiert haben gibt Ihnen das Programm die Fibonacci-Zahlen  $f_0$  bis  $f_{40}$  aus, inklusive der Zeit in Sekunden, welche für die Berechnung der jeweiligen Fibonacci-Zahl benötigt wurde. Für sämtliche Teilaufgaben können Sie davon ausgehen, dass nur valide Werte, also Werte für die die Fibonacci-Folge definiert ist eingegeben werden.

- (a) In der Klasse `Fibonacci.java` finden Sie die Methode `fibonacciRecursive`, welche eine Ganzzahl  $i$  als Parameter erwartet. Ergänzen Sie die Methode so, dass Sie *rekursiv*, also durch Aufrufe von `fibonacciRecursive(i-2)` und `fibonacciRecursive(i-1)`, die  $i$ -te Fibonacci-Zahl berechnet und zurückgibt. Notieren Sie die Zeiten, welche für die Berechnung der einzelnen Fibonacci-Zahlen benötigt wurde. [7 Punkte]
- (b) Neben der rekursiven Berechnung, kann die  $i$ -te Fibonacci-Zahl auch *iterativ* ermittelt werden, indem bei  $j = 2$  angefangen die Werte der Fibonacci-Zahlen  $f_{j-2}$  und  $f_{j-1}$  in Variablen zwischengespeichert und in jeder Iteration aktualisiert werden, solange  $j < i$  ist.

Implementieren Sie die Berechnung der  $i$ -ten Fibonacci-Zahl iterativ, also ohne rekursive Methodenaufrufe, in der Methode `fibonacciIterative`. Notieren Sie erneut die Berechnungszeiten. [7 Punkte]

- (c) Vergleichen Sie die gemessenen Zeiten aus Aufgabe (a) mit denen aus Aufgabe (b). Interpretieren Sie die Resultate. [6 Punkte]

Bitte geben Sie nur die von Ihnen bearbeitete `Fibonacci.java` ab.