

**Programmiermethoden in der Mathematik WS 02/03**  
**Woche 12 – Tutorium 1: Rekursion**

- Schreibt eine rekursive Funktion, die  $n!$  berechnet. Vergleicht mit einer iterativen Berechnung.
- Die Fibonacci-Zahlen sind definiert durch

$$a_0 = 0, \quad a_1 = 1, \tag{1}$$

$$a_{n+2} = a_n + a_{n+1}, \quad n = 0, 1, \dots \tag{2}$$

- Schreibt eine rekursive Funktion, die die  $n$ -te Fibonacci-Zahl berechnet.
  - Mit dem Ansatz  $a_n := x^n$  und durch Kürzen kann man (2) in eine quadratische Gleichung überführen.  
Jede Folge der Form  $a_n = Ax_1^n + Bx_2^n$  mit beliebigem  $A, B$  erfüllt dann (2).  
 $A, B$  werden aus (1) berechnet.  
Wie lautet damit die explizite Form der Fibonacci-Folge?  
Berechnet damit die  $n$ -te Fibonacci-Zahl.
  - Probiert aus, welche der beiden Methoden mehr Rechenzeit benötigt? Warum ist das wohl so?
- Schreibt eine rekursive Funktion **ggt**, die den größten gemeinsamen Teiler zweier nicht-negativer ganzer Zahlen nach dem Euklidischen Algorithmus berechnet. Benutzt den *modulo*-Operator (vgl. Operatorenliste).