

Übungen zu Einführung in die Informatik I

Aufgabe 10 **Fibonacci-Folge**

Die Folge der Fibonacci-Zahlen ist definiert als:

$$\begin{aligned}n_0 &= 0 \\n_1 &= 1 \\n_i &= n_{i-1} + n_{i-2} \text{ für } i > 1\end{aligned}$$

Schreiben Sie ein Programm, das nach Eingabe des Index i die Fibonaccizahl n_i berechnet und ausgibt. Überlegen Sie sich sowohl eine rekursive, als auch eine iterative Lösung.

Vergleichen Sie die beiden Lösungen hinsichtlich ihrer Laufzeit.

Aufgabe 11 **Erweiterter Euklid**

Erweitern und implementieren Sie den Euklid-Algorithmus zur Berechnung des ggTs, so dass er für alle $m, n \in \mathbb{N}^+$, $m \leq n$ die eindeutigen ganzen Zahlen $x, y \in \mathbb{Z}$ mit $\text{ggT}(m, n) = mx + ny$ berechnet. Zum Beispiel, für $m = 13$ und $n = 17$ sind $x = 4$ und $y = -3$.

Aufgabe 12 **Berechnung des Pascalschen Dreiecks**

Das Pascalsche Dreieck wurde etwa 1650 von Blaise Pascal angegeben und mathematisch untersucht. Aus ihm lassen sich direkt die Binomialkoeffizienten ablesen. In dieser Aufgabe soll ein Programm erstellt werden, das das Pascalsche Dreieck am Bildschirm ausgibt.

Aufbau des Pascalschen Dreiecks:

Zeilenindex n					
0				1	
1			1	1	
2		1	2	1	
3	1	3	3	1	
4	1	4	6	4	1

Zu einer gegebenen Zeile wird die nächste Zeile nach folgenden Regeln ermittelt:

- Die Anzahl der Elemente erhöht sich pro Zeile um 1.
- Die erste und letzte Zahl einer Zeile ist stets gleich 1.
- Kennzeichnet man die übrigen Elemente einer Zeile von links nach rechts mit Indizes $1 \leq i \leq n - 1$, so ergibt sich das i -te Element einer Zeile, indem man von der vorhergehenden Zeile das $(i-1)$ -te Element und das i -te Element addiert.

Aufgabenstellung: Geben Sie eine Implementierung an, welche das Pascalsche Dreieck in einer Funktion namens `int[][] pascalDreieck(int n)` berechnet. Der Parameter `n` gibt hierbei die Anzahl der zu berechnenden Zeilen an; zurückgegeben wird eine Reihung, die das Pascalsche Dreieck enthält. Implementieren Sie auch die `main`-Funktion;

Hinweise zur Programmierung:

Um das Pascalsche Dreieck zu berechnen, benötigt man eine zweidimensionale Reihung, deren Zeilen jeweils eine unterschiedliche Anzahl von Elementen haben. Dies ist in Java durch partielle Initialisierung einer mehrdimensionalen Reihung möglich. Dabei wird zunächst durch

```
int[][] zweiDimReihung = new int[n][];
```

eine zweidimensionale Reihung mit `n` Zeilen definiert; die Mächtigkeit jeder Zeile bleibt vorerst unbestimmt. An geeigneter Stelle lässt sich die Anzahl `m` der Elemente der `i`-ten Zeile dann durch

```
zweiDimReihung[i] = new int[m];
```

festlegen.