



1.) Struktogramme - Einsteigerniveau

Aufgabe 1.1: Fakultät

Erstellen Sie ein Struktogramm zur **rekursiven** und zur **iterativen** Berechnung der Fakultät einer natürlichen Zahl n .

Aufgabe 1.2: Fibonacci

Die Fibonacci-Folge f_n mit n Element der natürlichen Zahlen ist definiert über die rekursive Formel:

$$f_n = f_{n-1} + f_{n-2}, \forall n > 1$$

mit den Startwerten $f_0 = 0$ und $f_1 = 1$.

Erstellen Sie ein Struktogramm für einen Algorithmus, der die n -te Fibonaccizahl berechnet und erklären Sie kurz wie der Ablauf im Computer stattfindet.

Aufgabe 1.3: Echte Teiler

Erstellen Sie ein Struktogramm zu einem Programm, dass eine positive ganze Zahl einliest und die Anzahl ihrer echten Teiler ausgibt.

Ist die Anzahl 1, soll zusätzlich der Ausdruck „ - Primzahl!“ ausgegeben werden.

Testdaten:

- a) 10 hat die echten Teiler 1, 2, 5 und führt daher zur Ausgabe: „3“;
- b) 1 hat keine echten Teiler, daher Ausgabe: „0“;
- c) 7 führt zur Ausgabe: „1 - Primzahl!“;

Aufgabe 1.4: Formatierter Text

Erstellen Sie ein Struktogramm zu folgender Aufgabe:

Ein Programm soll einen Text von der Standard-Eingabe lesen. Formatieren und auf die Standard-Ausgabe schreiben.

Die Formatierung soll folgende Aspekte berücksichtigen:

- Blanks am Anfang einer Zeile werden nicht ausgegeben (Ausgabe linksbündig)
- Mehrfach aufeinander folgende Blanks werden in der Ausgabe durch ein einzelnes Blank ersetzt
- Großgeschriebene Vokale werden durch die entsprechenden kleinen Buchstaben ersetzt

Aufgabe 1.5: Klammerpaare

Von einer vorgegebenen Zeichenkette (Satz, Gleichung, Formel, Text, ...) soll die Schachtelungstiefe bestimmt werden. Stimmen Anzahl und Position der Klammerpaare nicht, soll eine Fehlermeldung ausgegeben werden und das Programm wird beendet. Entwerfen Sie ein Struktogramm.

Klammernpaare: (); []; {}

Beispiele:

Eingabe: 3 * (1 - (5 * 2) / 2,5 % (18 / 3 * 5))

Ausgabe: Schachtelungstiefe: 2

Eingabe: 7 / (5 * (2 - 17)

Ausgabe: Fehler

Tipp: Machen Sie sich zunächst Gedanken, was Sie sich merken müssen und was konkret verglichen werden muss. Wie könnte ein sinnvolles Speichermedium aussehen? Bzw. nach welchem Prinzip sollte das funktionieren?

Aufgabe 1.6: Gauß Eliminationsverfahren - über Klausurniveau

Erstellen sie ein Struktogramm für das Gaußsche Eliminationsverfahren (siehe: http://de.wikipedia.org/wiki/Gau%C3%9Fsches_Eliminationsverfahren), inklusive Zeilentausch, aber ohne Pivotisierung.

Beispiel:

Aus der Matrix

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 9 & 8 & 3 \\ 4 & 6 & 2 & 5 \\ 1 & 2 & 3 & 6 \end{pmatrix}$$

wird

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -7 & -17 \\ 0 & 0 & 4 & 23 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$