

Berechnung der Fakultäten von 1 bis 20

Unter der Fakultät einer (natürlichen) Zahl versteht man das Produkt aller natürlichen Zahlen von 1 bis zu dieser Zahl. Das Fakultätszeichen ist in der Mathematik das Ausrufungszeichen. Der Wert von $4!$ ist also mit $1 \cdot 2 \cdot 3 \cdot 4$, die Zahl 24.

Fakultäten werden ganz schön schnell ganz schön groß. Wie groß? In RTA kann man das wie folgt programmieren:

```

;
; FAKULTÄT
; =====
;
; DEKLARATIONEN
;
;         _name      Fakultät
;         _var       factor          ; Laufender Faktor
;         _var       f              ; Fakultät
;
;
; INITIALISIERUNGEN
;
;         pause      Das~Programm~berechnet~Fakultäten~von~1~bis~20.
;         mov        factor    1          ; laufenden Faktor auf 1
;         mov        f        1          ; die Fakultät auf 1
;         cls        ; Ausgabertext löschen
;
; SCHLEIFE
;
$loop:
;         add        factor    1          ; faktor eins weiter
;         mul        f        factor      ; fakultät aufmultiplizieren
;         cmpgt      factor    20         $ende ; Bei factor 20 abbrechen
;         printn     factor    2         0     ; factor mit 2 Stellen
;         prints     !:~              ; !, : und Leerzeichen
;         printn     f        19         0     ; Fakultät mit 19 St.
;         prints     \              ; Zeilenvorschub ausgeben
;         jump       $loop              ; nächste Schleife
;
; ENDE
;
$ende:
;         save       fakultaeten
;         pause      Das~Ergebnis~wurde~in~die~Datei~fakultaeten.txt~geschrieben.
;         exit
;         _end

```

Außer der eigentlichen Berechnung gibt es hier etwas Ein-/Ausgabe. Die pause-Befehle teilen dem Anwender mit, was das Programm macht und die Befehle cls, printn, prints und save schreiben das Rechenergebnis in den globalen Ausgabertext und in eine Datei „fakultäten.txt“.

Berechnung der Zahl e mit einer Reihenentwicklung

Reihenentwicklungen gehören zu den Aufgaben, vor denen gewöhnliche Programmierer einen Heidenrespekt haben. So kompliziert ist es aber nun auch wieder nicht. Um die Eulersche Zahl e zu berechnen, gibt es die Formel

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

Und so codet man das Ganze in RTA ab:

```
;
; EULERSCHE ZAHL e BERECHNEN
; =====
;
; DEKLARATIONEN
;
;     _name      Euler
;     _var       e                ; e
;     _var       n                ; Zykluszähler
;     _var       f                ; Fakultät
;     _var       restgl           ; Restglied
;
; INITIALISIERUNGEN
;
;     pause      Euler~berechnet~die~Zahl~e~mit~einer~Reihenentwicklung.
;     mov        e            1          ; e ist erst einmal 1
;     clr        n            0          ; Zykluszähler löschen
;     mov        f            1          ; die Fakultät von 0 ist 1
;     clr        restgl       0          ; Restglied löschen
;     cls        0             ; Augabetext löschen
;
; HAUPTSCHLEIFE
;
; $loop:
;     add        n            1          ; Zykluszähler inkrementieren
;     mov        restgl       1          ; Zähler des Bruches: 1
;     mul        f            n          ; Fakultät multiplizieren
;     div        restgl       f          ; Bruch berechnen
;     add        e            restgl     ; Auf e aufsummieren
;     printn     n            2          0 ; n mit 2 Vorkommastellen
;     prints     :~           0         ; Doppelpunkt und Leerzeichen
;     printn     e            1         14 ; e mit 14 Stellen ausgeben
;     prints     \            0         ; Zeilenvorschub ausgeben
;     mov        r0           restgl     ; restgl auf Hilfszelle r0
;     abs        r0           0         ; Absolutbetrag bilden
;     cmpgt     r0            1e-16     $loop ; restgl<1e-16 weitermachen
;
; ENDE
;
; $ende:
;     save       e            0         ; Text in Datei e.txt
;     pause      Jetzt~steht~e~in~der~Datei~e.txt ; Mitteilung
;     exit       0             ; Programmende
;     _end
```

Auch hier erfolgt zusätzlich zur eigentlichen Berechnung wieder eine Ausgabe in den globalen Augabetext und in eine Datei, hier mit Namen „e.txt“.