5

5. Programmbeispiele

Hello World

Mit Hello World fängt jede Programmiersprache an ...

Das Programm besteht aus lediglich 2 Befehlen: Der Befehl pause gibt "Hello World!" aus und der Befehl exit beendet das Programm. Hinzu kommt der Pseudobefehl _name, der dem Programm den Namen "Hello World" gibt und _end, das die letzte Zeile kennzeichnet. Der Rest ist Kommentar.

Mittelwert zweier Zahlen

Es werden zwei Zahlenwerte im Dialog (input-Befehl) abgefragt, deren Mittelwert berechnet und das Ergebnis angezeigt (output-Befehl).

```
MITTELWERTBERECHNUNG
       _name Mittelwertberechnung
                                              ; Programmname
                                               ; Variable z1 deklarieren
        var
                 z1
        var
                                               ; Variable z2 deklarieren
                 z2
                                               ; Variable für Mittelwert
        var
       input
input
mov
                 z1
z2
                         Erste~Zahl
Zweite~Zahl
                                               ; Erste Zahl abfragen
                                               ; Zweie Zahl abfragen
                                               ; z1 nach m transportieren
                         z2
2
                                                 z2 zu m addieren
                                               ; Division durch 2
                 m Mittelwert
       output
                                               ; Mittelwert ausgeben
                                               ; Abarbeitung beenden
       exit
                                               ; Das ist die letzte Zeile
        end
```

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 1·2·3·4, 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
                                                ; Fakultät
          var
 INITIALISIERUNGEN
                 Das~Programm~berechnet~Fakultäten~von~1~bis~20.
         pause
                                            ; laufenden Faktor auf 1
                  factor 1
         mov
                                               ; die Fakultät auf 1
         mov
                                               ; Ausgabetext löschen
         cls
 SCHLEIFE
$loop:
         add
                  factor
                                              ; faktor eins weiter
         mul
                           factor
                                               ; fakultät aufmultiplizieren
                 factor 20
factor 2
                                      $ende
         cmpgt
                                              ; Bei factor 20 abbrechen
         printn
                                               ; factor mit 2 Stellen
                  !:~
                                               ; !, : und Leerzeichen
         prints
                                               ; Fakultät mit 19 St.
         printn
         prints
                                               ; Zeilenvorschub ausgeben
         jump
                  $loop
                                               ; nächste Schleife
 ENDE
$ende:
                  fakultaeten
         save
                  Das~Ergebnis~wurde~in~die~Datei~fakultaeten.txt~geschrieben.
         pause
         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 Ausgabetext 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
         var
                 е
                                              ; Zykluszähler
         var
         var
                                              ; Fakultät
        _var
                 restgl
                                              ; Restglied
 INITIALISIERUNGEN
        pause
                Euler~berechnet~die~Zahl~e~mit~einer~Reihenentwicklung.
                                           ; e ist erst einmal 1
        clr
                                             ; Zykluszähler löschen
                                              ; die Fakultät von 0 ist 1
                                              ; Restglied löschen
        clr
                 restgl
                                              ; Augabetext löschen
        cls
 HAUPTSCHLEIFE
$loop:
                                              ; Zykluszähler inkrementieren
        add
                 restgl
f
                                             ; Zähler des Bruches: 1
                                             ; Fakultät multiplizieren
        div
                 restgl
                           f
                                              ; Bruch berechnen
                 e re
2
                                              ; Auf e aufsummieren
        printn
                                              ; n mit 2 Vorkommastellen
        prints
                                             ; Doppelpunkt und Leerzeichen
                                             ; e mit 14 Stellen ausgeben
        printn
        prints
                                             ; Zeilenvorschub ausgeben
                 r0
                           restgl
                                             ; restgl auf Hilfszelle r0
        mov
abs
cmpgt
        mov
                                             ; Absolutbetrag bilden
                 r0
                                    r0
                           1e-16
 ENDE
$ende:
                                             ; Text in Datei e.txt
        save
                 Jetzt~steht~e~in~der~Datei~e.txt ; Mitteilung
        pause
        exit
                                              ; Programmende
         end
```

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