

# An alle Freunde der ADIM!

Martin Weissenböck, Leiter der ADIM

Wenn Sie die ADIM noch nicht kennen, wollen wir uns kurz vorstellen:

Die Arbeitsgemeinschaft für Didaktik, Informatik und Mikroelektronik (kurz: ADIM) wurde mit dem Ziel gegründet, **preiswerte und qualitativ gute Medien** für alle Formen des Unterrichts in den Bereichen "Informatik" oder "Elektronische Datenverarbeitung" oder ähnlicher Unterrichtsgegenstände bereit zu stellen. Die meisten Mitarbeiter der ADIM sind Lehrer, die in ihrer Freizeit an Skripten und anderen Hilfsmitteln arbeiten. (Genauere Angaben über die ADIM finden Sie am Bestellblatt auf Seite 11 sowie im Impressum auf Seite 38).

Schon vor mehr als 10 Jahren wurden - vorerst nur für kleinere Gruppen - Unterlagen über den Apple II, über Hardware, über Studienvorschriften und Lehrpläne und ähnliche Themen zusammengestellt.

Einige Skripten - wie etwa eine Bedienungsanleitung für den Apple II - sind nicht mehr interessant. Andere sind seit Jahren immer wieder gefragt: so zum Beispiel die Unterlagen über Turbo-Pascal oder C. Natürlich werden die Skripten immer wieder überarbeitet und an die aktuellen Versionen der Software angepaßt. Aber auch die Erfahrungen beim Einsatz im Unterricht führen dazu, daß immer wieder Beispiele angepaßt oder Teile verändert werden. Deshalb werden die Skripten in der Regel auch einmal pro Jahr neu aufgelegt.

Ein durchgehendes **wichtiges didaktisches Prinzip** ist der einheitliche Aufbau der Programmiersprachenbände. Alle Beispiele werden in den einzelnen Sprachen möglichst detailgetreu so formuliert, daß sie als Beispiele für die Übersetzung von einer Sprache in eine andere verwendet werden können. Dazu zählen nun: Turbo-Pascal, C, C++, Turbo/Power-Basic, Quick-Basic und in Kürze auch Modula. Sie finden nach diesem Artikel Mustertexte aus einzelnen Bänden.

Die **objektorientierte Programmierung** hat vor einiger Zeit die umfangreichsten Änderungen gebracht. Gerade das Beispiel dieser modernen Programmieretechnik zeigte, daß die Umsetzung im Unterricht gar nicht so einfach war und die dabei gewonnenen Erkenntnisse "von Jahr zu Jahr" eingebaut werden konnten.

ADIM-Skripten werden bereits an etlichen Schulen für den EDV-Unterricht werden. Für höhere technische Lehranstalten stehen sogar vier Bände als **Gratis-Fachbuch** zur Verfügung.

Als Übersicht über derzeit verfügbare Bände finden Sie in auf Seite 11 einen Bestellschein.

Eine weitere Aktivität der ADIM ist die Organisation von **Sammelaufkäufen**. In den Anfängen der PC-Zeit waren Sammelkäufe sehr günstig; inzwischen hat der allgemeine Preisverfall und auch die Konkurrenz verschiedener Anbieter zu erfreulich niedrigen Preisen geführt. (Wer weiß schon noch, daß die ersten Disketten vor 10 Jahren 140,- Schilling (!) gekostet haben und damals als Clubaktion um 70,- Schilling zu kaufen waren?) Die Hardwareaktivitäten der ADIM beschränken sich nun auf besondere Komponenten. Derzeit sind **Modems** mit besonders vielen Eigenschaften nach wie vor aktuell.

Seit einiger Zeit wollen wir Sie über ein eigenes **Mitteilungsblatt** über unsere aktuellen und neuesten Aktivitäten informieren. Der große damit verbundene Aufwand hat aber dieses Projekt bisher verhindert. Nun haben wir in den **PC-NEWS** einen guten Partner gefunden. In dieser Zeitschrift wollen wir Sie regelmäßig über aktuelle Neuigkeiten, unsere Ziele und pädagogische Ideen informieren und damit auch ganz allgemein über Fragen des EDV-Unterrichts diskutieren.

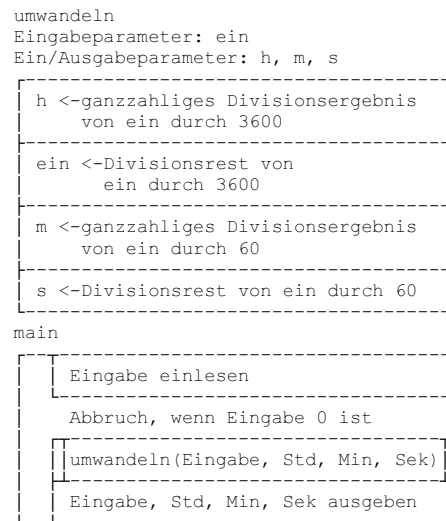
**Wir laden daher alle ADIM-Freunde sehr herzlich ein, die PC-NEWS zu abonnieren!** Auf ein regelmäßiges "Wiederlesen" freut sich

Martin Weissenböck

Ausschnitt aus dem Band 40, Turbo C:

Beispiel 5.1F:

Man verwandle Sekunden in Stunden, Minuten und Sekunden. Die Eingabe wird mit der Zahl 0 abgeschlossen.



```

/* Beispiel 5.1F */
/* Umwandeln einer Sekundenzahl in
  Stunden, Minuten und Sekunden */
#include <stdio.h>
#include <conio.h>
void umwandeln
(long ein, int *h, int *m, int *s);

void umwandeln
(long ein, int *h, int *m, int *s)
{
    *h=(int) (ein/3600);
    ein%=3600;
    *m=(int) ein/60;
    *s=(int) ein%60;
} /* umwandeln */

void main ()
{
    int std, min, sek;
    long eingabe;

    printf("Umwandeln SEK -> STD MIN SEK\n");
    for (;;)
    {
        printf("Zeit in SEK (0... Abbruch):");
        scanf("%ld", &eingabe);
        if (!eingabe) break;
        umwandeln(eingabe, &std, &min, &sek);
        printf
            ("%ld SEK =%d STD%3d MIN%3d SEK\n",
             eingabe, std, min, sek);
    }
} /* b051f */

```

Anmerkung: Da ein Tag bereits 86400 Sekunden hat, wurde für die Eingangsgröße der Typ `long` verwendet. Mit dem Typ `int` wäre 32767 die größte Zahl.

Jeweils ein **aktueller Parameter** und ein **formaler Parameter** müssen paarweise zusammenpassen. Daher ist auf

- den Typ der Parameter,
- die Reihenfolge der Parameter und
- die Anzahl der Parameter

zu achten.

Anmerkung: in C können auch Unterprogramme mit einer variablen Zahl von Parametern vereinbart werden.

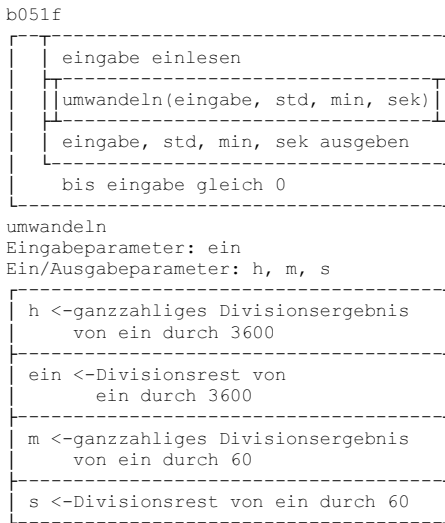
Die Wahl der Namen der formalen Parameter ist nicht von Bedeutung!

Eine **return-Anweisung** kann verwendet werden, um das Unterprogramm sofort zu verlassen.

Ausschnitt aus dem Band 41, Turbo/Power-Basic:

Beispiel 5.1F:

Man verwandle Sekunden in Stunden, Minuten und Sekunden. Die Eingabe wird mit 0 abgeschlossen. Achtung: 32767 ist die größte INTEGER-Zahl, daher werden die Sekunden als LONG-Wert gespeichert.



```

* Beispiel 051F
* Umwandeln einer Sekundenzahl in Stunden,
* Minuten und Sekunden.
PRINT "Umwandeln Sek -> Std Min Sek"
DO
  INPUT "Zeit in Sek (0... Abbruch)", _
    eingabe%
  IF eingabe% = 0 THEN EXIT LOOP
  PRINT
  CALL umwandeln_
    ((eingabe%), std%, min%, sek%)
  PRINT eingabe%; " SEK ="; std%; " STD"; _
    min%; " MIN"; sek%; " SEK"
LOOP
END ' B051F
SUB umwandeln (ein%, h%, m%, s%)
  h% = ein% \ 3600
  ein% = ein% MOD 3600
  m% = ein% \ 60
  s% = ein% MOD 60
END SUB ' umwandeln
    
```

Jeweils ein **aktueller Parameter** und ein **formaler Parameter** müssen paarweise zusammenpassen. Daher ist auf

- den Typ der Parameter
- die Reihenfolge der Parameter und
- die Anzahl der Parameter

zu achten.

Die Wahl der Namen der formalen Parameter ist nicht von Bedeutung!

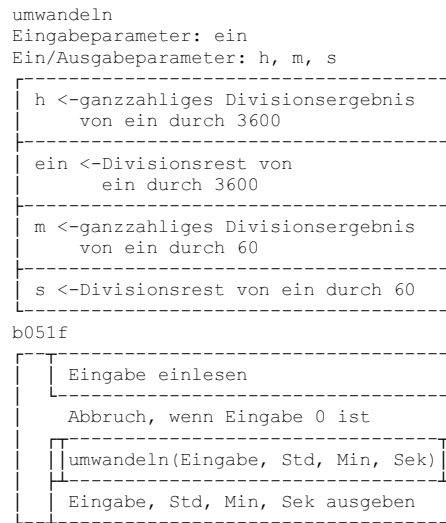
Eine EXIT SUB - Anweisung kann verwendet werden, um das Unterprogramm sofort zu verlassen.

Wenn das gesamte Programm beendet werden soll, ist einer der Befehle STOP, END oder SYSTEM zu verwenden.

Ausschnitt aus dem Band 47, Turbo Pascal:

Beispiel 5.1F:

Man verwandle Sekunden in Stunden, Minuten und Sekunden. Die Eingabe wird mit 0 abgeschlossen. Achtung: 32767 ist die größte integer-Zahl, daher werden die Sekunden als longint-Wert gespeichert.



```

PROGRAM b051f;
(* Umwandeln einer Sekundenzahl in Stunden,
  Minuten und Sekunden *)
PROCEDURE umwandeln(ein: longint;
  VAR h, m, s: integer);
  BEGIN
    h := ein DIV 3600;
    ein := ein MOD 3600;
    m := ein DIV 60;
    s := ein MOD 60
  END; (* umwandeln *)
VAR
  eingabe: longint;
  std, min, sek: integer;
BEGIN
  writeln('Umwandeln s -> std min s');
  REPEAT
    write('Zeit in Sek (0... Abbruch)');
    readln(eingabe);
    umwandeln(eingabe, std, min, sek);
    writeln(eingabe: 5, ' s = ',
      std: 4, ' std', min: 3, ' min', sek: 3, ' s')
  UNTIL eingabe=0
END. (* b051f *)
    
```

Jeweils ein **aktueller Parameter** und ein **formaler Parameter** müssen paarweise zusammenpassen. Daher ist auf

- den Typ der Parameter
- die Reihenfolge der Parameter und
- die Anzahl der Parameter

zu achten.

Die Wahl der Namen der formalen Parameter ist nicht von Bedeutung!

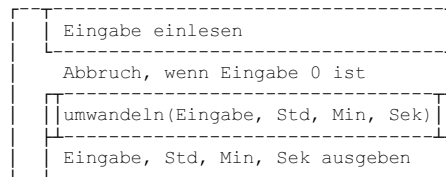
Eine exit-Anweisung kann verwendet werden, um das Unterprogramm sofort zu verlassen. Mit halt wird das gesamte Programm verlassen. Wird halt mit einem Parameter verwendet, wird dessen Wert auf die DOS-Ebene übertragen und kann dort als ERRORLEVEL abgefragt werden.

## Ausschnitt aus dem Band 49, Quick-Basic:

## Beispiel 5.1F:

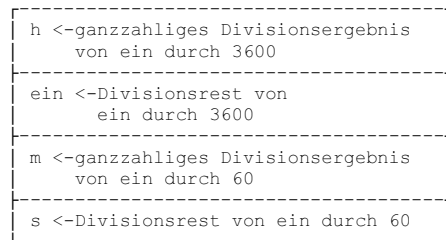
Man verwandle Sekunden in Stunden, Minuten und Sekunden. Die Eingabe wird mit 0 abgeschlossen. Achtung: 32767 ist die größte INTEGER-Zahl, daher werden die Sekunden als LONG-Wert gespeichert.

b051f



```

umwandeln
Eingabeparameter: ein
Ein/Ausgabeparameter: h, m, s
  
```



```

* Beispiel 051F
* Umwandeln einer Sekundenzahl in
* Stunden, Min. und Sek.
DECLARE SUB umwandeln (ein&, h%, m%, s%)
PRINT "Umwandeln Sek -> Std Min Sek"
DO
  INPUT "Zeit in Sek (0...Abbruch)",_
  eingabe&
  IF eingabe& = 0 THEN EXIT DO
  PRINT
  CALL umwandeln_
  ((eingabe&), std%, min%, sek%)
  PRINT eingabe&: " SEK ="; std%; " STD"; _
  min%; " MIN"; sek%; " SEK"
LOOP
END * B051F
SUB umwandeln (ein&, h%, m%, s%)
  h% = ein& \ 3600
  ein& = ein& MOD 3600
  m% = ein& \ 60
  s% = ein& MOD 60
END SUB * Umwandeln
  
```

Jeweils ein **aktueller Parameter** und ein **formaler Parameter** müssen paarweise zusammenpassen. Daher ist auf

- den Typ der Parameter
- die Reihenfolge der Parameter und
- die Anzahl der Parameter

zu achten.

Die Wahl der Namen der formalen Parameter ist nicht von Bedeutung!

Eine **EXIT SUB** Anweisung kann verwendet werden, um das Unterprogramm sofort zu verlassen.

Wenn das gesamte Programm beendet werden soll, ist einer der Befehle STOP, END oder SYSTEM zu verwenden.

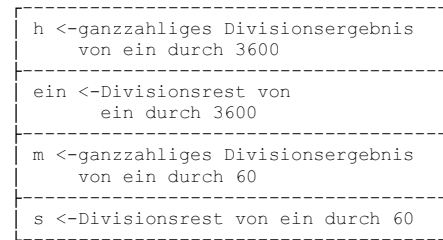
## Ausschnitt aus dem Band 50, Borland C++:

## Beispiel 5.1F:

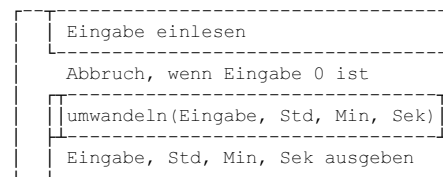
Man verwandle Sekunden in Stunden, Minuten und Sekunden. Die Eingabe wird mit der Zahl 0 abgeschlossen.

```

umwandeln
Eingabeparameter: ein
Ein/Ausgabeparameter: h, m, s
  
```



main



```

// Beispiel 5.1F
/* Umwandeln einer Sekundenzahl in
  Stunden, Minuten und Sekunden */
#include <iostream.h>
#include <iomanip.h>

void umwandeln
(long ein,int &h,int &m,int &s)
{
  h=(int) (ein/3600);
  ein%=3600;
  m=(int) ein/60;
  s=(int) ein%60;
} // umwandeln

void main ()
{
  int std,min,sek;
  long eingabe;

  cout << "Umwandeln SEK -> STD MIN SEK\n";
  for (;;)
  {
    cout << "Zeit in SEK (0...Abbruch)";
    cin >> eingabe;
    if (!eingabe) break;
    umwandeln(eingabe, std, min, sek);
    cout << setw(7) << eingabe << " SEK ="
    << setw(4) << std << " STD" <<
    setw(3) << min << " MIN" <<
    setw(3) << sek << " SEK\n";
  }
} // b051f
  
```

Anmerkung: Da ein Tag bereits 86400 Sekunden hat, wurde für die Eingangsgröße der Typ `long` verwendet. Mit dem Typ `int` wäre 32767 die größte Zahl.

Jeweils ein **aktueller Parameter** und ein **formaler Parameter** müssen paarweise zusammenpassen. Daher ist auf

- den Typ der Parameter,
- die Reihenfolge der Parameter und
- die Anzahl der Parameter

zu achten.

Anmerkung: in C++ können auch Unterprogramme mit einer variablen Zahl von Parametern vereinbart werden.

Die Wahl der Namen der formalen Parameter ist nicht von Bedeutung!

Eine **return**-Anweisung kann verwendet werden, um das Unterprogramm sofort zu verlassen.

Und so soll es auch in Zukunft weitergehen...