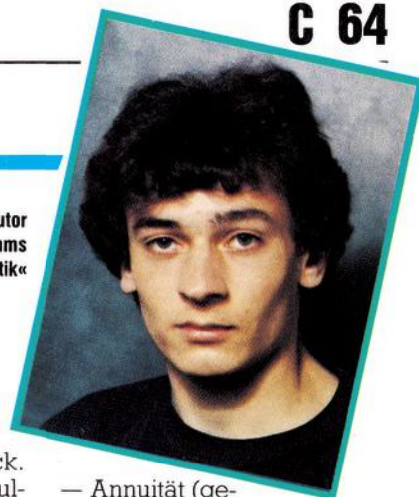


Menügesteuerte Finanzmathematik



Klaus Klöker der Autor
des Programms
»Finanzmathematik«

Zwei herausragende Eigenschaften machen dieses Programm für alle C 64-Besitzer interessant: Die Menütechnik und der modulare Bildschirmmaskenaufbau.

Wer wünscht sich nicht, daß Programme von Diskette auf Tastendruck geladen, beziehungsweise nachgeladen werden? Wenn Sie das Menüprogramm dieses Listings übernehmen und für Ihren Bedarf abändern, ist das Laden von Programmen kein Problem mehr.

Das Programm wird geladen mit dem Befehl: LOAD "Menü",8
Gestartet wird dann mit dem Befehl: RUN

Wählen Sie nun mit den Funktionstasten F3 und F5 die Nummer des Programmes, mit welchem Sie als nächstes arbeiten wollen. Sie erkennen Ihre augenblickliche Position an der blinkenden Zahl.

Nach beendeter Wahl drücken Sie »RETURN«. Dann wird das Programm geladen, dessen Zahl gerade blinkt und an der sich nach »RETURN« ein - (Pfeil links) befindet.

**Bedienungsanleitung für das Programm:
Erstellen eines Tilgungsplanes.**

Nach erfolgter Auswahl im Hauptmenü wird das Programm automatisch gestartet.

Durch entsprechende Informationen in der Fußzeile erklärt sich das Programm im wesentlichen selbst. Allerdings mußte manchmal wegen Platzmangel darauf verzichtet werden.

Grundsätzlich gilt:
— Bis auf die Funktionstasten erfolgt bei jeder Eingabe und Auswahl ein Abschluß mit »RETURN«.

— Bei der Auswahl im Programm, wird immer mit den Tasten F3 und F5 gewählt, nach Drücken von »RETURN« wird immer das als nächstes ausgeführt, was zuletzt geblinkt hat.

— Zum Abschluß der Dateneingabe gilt für die Abfrage, ob die Eingabe korrekt ist, folgendes:

a) Zugelassen sind die Buchstaben j (ja) und n (nein).

b) Die Antwort kann bis zum Abschluß mit »RETURN« durch die Taste »DEL« korrigiert werden.

— Während der Dateneingabe, kann man durch Drücken der Taste F1 zum ersten Auswahlmenü zurückkehren.

— Wird die Tabelle auf dem Bildschirm ausgedruckt, kann man den Bildschirmausschnitt, innerhalb der festgesetzten Grenzen verändern.

Mit F1 kehrt man zur alten Bildschirmmaske zurück.

Das Ausgabefeld »Resultat« enthält folgende Informationen:

— Zinsaufwand (gesamt): Dieser Summe entsprechen sämtliche in den einzelnen Jahren zu zahlende Zinsen.

— Annuität (gesamt): Dies ist die Summe, die man insgesamt zurückzahlen muß (Zins- und Tilgungsanteil zusammen).

Im linken unteren Feld wird noch einmal angege-

200 - 260:

300 - 390:

400 - 410:

900 - 960:

na\$ (...) =

nb\$ (...) =

Z =

S =

Y =

g\$ =

10 - 30

70 - 75

80

90

95

100 - 195

200 - 290

300 - 480

1000 - 1180

2000 - 2140

3000 - 3040

3500 - 3610

Programmdokumentation: Menü

Aufbau der Bildschirmmaske

Auswahlroutine durch Betätigen der Funktionstasten

Sprung in die Zeile, in der das entsprechende Programm geladen, oder in der das Programm beendet wird

Festlegung der Variablen

Variablenliste: Menü

Nummern der Programme

Namen der Programme

Zeile

Spalte

Index der indizierten Variablen

Eingabevariable der get-Anweisung

Programmdokumentation: Zinsrechnung

Unterprogramm zum Löschen des rechten oberen und linken unteren Bildschirmfensters

Unterprogramm zur Rundung der Ergebnisse und zur Umwandlung in einen String

Unterprogramm zum Laden des Menüs

Unterprogramm zum Setzen des Cursors, an die Stelle, welche mit Z = Zeile und S = Spalte vor dem Sprungbefehl festgelegt wurde

Unterprogramm zum Zeichnen einer Linie

Auswahlroutine mit den Funktionstasten der entsprechenden reversen Felder oben links und unten links.

Auswahlroutine mit den Funktionstasten der Fußzeile.

Eingaberoutine

Beginn des Hauptprogramms:

— Aufbau der Bildschirmmaske

— Sprung in die Auswahlroutine, um festzulegen, was berechnet werden soll.

Aufbau der Eingabemaske und Eingabe der Daten mittels der Eingaberoutine. Anschließend erfolgt Abfrage, ob die Eingabe korrekt ist.

Berechnung des vorher ausgewählten Ergebnisses

Ausdruck des Ergebnisses. Es werden ausgedruckt:

— die am Anfang festgelegten Kriterien

— die eingegebenen Daten

— das Endergebnis

ben, wie hoch der Zinsanteil von der Annuität in Prozent ist. Dieses Programm stellt Tilgungspläne bis zu einer Laufzeit von 50 Jahren auf. Falls dies nicht ausreicht, braucht in Zeile neun nur die Dimensionierung der Variablen e und e\$ erhöht werden. (Klaus Klöcker/rg)

mein Abitur bestanden und im August eine Ausbildung als Bankkaufmann begonnen.

In meiner Freizeit beschäftige ich mich gerne mit Geschichte und der Computerprogrammierung.

Entstanden ist das Programm aufgrund eines Kurses über Finanzmathematik.

Es schien mir damals zweckmäßig, Tilgungspläne mit dem Computer aufzustellen. Dieser absolviert seine Arbeit sekundenschnell und fehlerfrei. Wer selbst einmal

Tilgung »zu Fuß« geplant hat, weiß, daß gerade letzteres keine Selbstverständlichkeit

Entstehungsgeschichte und Anwendung

ist (bei 20 Jahren Laufzeit und 6 Beträgen pro Jahr, müssen immerhin 120 Beträge einzeln berechnet werden).

Einen großen Teil der Programmierzeit beanspruchte

der Aufbau der Bildschirmmasken und die Menüsteuerung. Ich hielt diesen Aufwand für angebracht, weil die Anwendung bequemer ist und außerdem mehr Spaß macht. Wer sich das Programm »Tilgungspläne« genauer anschaut, wird feststellen, daß man durch einfaches Austauschen der Formeln verschiedene Tabellen aufstellen kann, obwohl nur eine begrenzte Zahl von Eingabekriterien notwendig ist.

(Klaus Klöcker)

Lebenslauf

Ich bin 20 Jahre alt, habe im Frühjahr dieses Jahres

Es folgt: Aufbau der Fußzeile mit der Möglichkeit durch Wahl mit den Funktionstasten entsprechend zu verzweigen.

Variablenliste: Zinsrechnung

ma\$:	Kopfzeile	
mb\$, mc\$, md\$:	Fußzeilen	3000 - 3720
a\$ (...):	Eingabekriterien	
ka\$; p\$; zi\$		
ja\$; mo\$; ta\$	Ein- und Ausgabekriterien	3800 - 3900
pm\$; pu\$:	Punkte in der Eingabemaske	
Z:	Zeile der Cursorposition	3910 - 3950
S:	Spalte der Cursorposition	
Zl:	niedrigste Zeile, die gelöscht werden soll	4000 - 4590
Z2:	höchste Zeile, die gelöscht werden soll	
Y:	Variablenindex	5000 - 5090
Y1:	kleinster Index	5500 - 5660
Y2:	größter Index	
SW:	Schrittweite	6000 - 6500
k; c:	Konstanten, die, je nach ihrer Wertigkeit, bestimmte Vorgaben ausführen.	
f:	gibt an, welche Berechnung erfolgen soll	ma\$
e:	gibt an, die Art der Laufzeit	mb\$
h:	legt die Art der Laufzeit in der Berechnung fest	mc\$
in\$:	Variable in der Eingaberoutine	me\$
ab\$:	Endergebnis	mf\$
		mh\$
		md\$

Programmdokumentation: Tilgungsrechnung

10 - 20	Unterroutine zum Löschen der rechten Bildschirmhälfte	Z:
30 - 55	Unterroutine zum Runden des Ergebnisses und zum Umwandeln in einen String	S:
70	Unterroutine zum Laden des Menüs	Y:
80	Unterroutine zum Zeichnen einer Linie	Y1:
90	Unterroutine zur Festlegung der Cursorposition	Y2:
95	Unterroutine zum Löschen der Fußzeile	Sw:
100 - 195	Unterroutine zur Auswahl der vorgegebenen Möglichkeiten mit den Tasten F3 und F5, der Abschluß erfolgt mit »RETURN«.	g\$:
200 - 290	Unterroutine zur Auswahl bei der Fußzeile mit den Tasten F3 und F5, der Abschluß erfolgt mit »RETURN«.	g:
300 - 480	Eingaberoutine. Durch Drücken der Taste F1 kann man von vorne beginnen	i,j:
900 - 960	Aufbau der Bildschirmmaske	c:
1010 - 1080	Ausdruck der Kopfzeile und Bildschirmmaske, Sprung in die Auswahlroutine (100-195)	f:
2000 - 2140	Aufbau der Eingabemaske und Sprung	h:
		in\$:
		pns:
		is:
		tb\$:
		ta\$:
		X1:
		X2:

in die Eingaberoutine. Zum Schluß wird abgefragt, ob die Eingabe korrekt ist.

Die Antwort wird mit »RETURN« eingegeben und kann vorher mit »DEL« korrigiert werden.

Ausführung der Rechnung mit Sprung in die Rundungs- und Stringumwandlungsroutine

Ausdruck des Resultates auf dem Bildschirm

Aufbau der Wahlfußzeile und Sprung in die Auswahlroutine (200-290)

Ausdruck der Tabelle und Verändern des Bildschirmausdrucks mit den Cursorstasten: Mit F1 erfolgt ein Rücksprung

Änderung der eingegebenen Daten

Papierausdruck der eingegebenen Daten und der Tabelle

Festlegung der Variablen

Variablenliste: Tilgungsrechnung

	Kopfzeile
	Fußzeilen
	Löschzeile zum Löschen des Bildschirms
	Zeile der Cursorposition
	Spalte der Cursorposition
	Variablenindex, der einfach dimensionierten Variablen
	niedrigster Variablenindex
	höchster Variablenindex
	Schriftweite
	Variable der GET-Anweisung
	ASC (g\$)
	Variablenindizes der zweifach dimensionierten Variablen
	Konstante, von der bestimmte Bedingungen abhängen
	gibt das Ergebnis der Auswahlroutine an
	beschränkt die maximale Eingabelänge
	Variablenindex
	füllt den Ergebnisstring mit Punkten auf
	gibt das Jahr in der Tabelle an
	chr\$(16): Tabulator
	chr\$(10): Leerzeile
	Variable die den kleinsten Index der Tabelle angibt
	Variable die den größten Index der Tabelle angibt
	Variablen der Rechnung t1,t2,J2,r1,a1,Z1,Z2

```

70 REM BY KLAUS KLOEKER
80 GOTO100
90 POKE214,Z:POKE211,S:SYS58732:RETURN
100 REM * MENUE *
110 PRINT"┌───┐:POKE53280,0:POKE53281,0:PO
KE53272,23:POKE788,52:GOSUB900
200 PRINT"┌───┐ ─ \ / * / * \ * | | - \ *
┌───┐
210 Z=4:S=6:GOSUB90:PRINT"\ENUE :":Z=5:G
OSUB90:PRINT"-----"
220 Z=7:S=0:Y=1:FORI=1TO5:GOSUB90:PRINTN
A$(I);NB$(I)
230 Z=Z+2:Y=Y+1:NEXT
240 Z=18:GOSUB90:PRINT"┌───┐ ┌───┐ OAEHLEN
*IE MIT ┌───┐ 3 ┌───┐ UND ┌───┐ 5 ┌───┐ !"
250 Z=20:S=7:GOSUB90:PRINT"┌───┐RUECKEN *IE
DANN '┌───┐ / ' !"
260 Z=23:S=0:GOSUB90:PRINT"┌───┐
"
300 REM * AUSWAHLROUTINE *
310 Z=7:Y=1
320 GOSUB90:PRINT"┌───┐ ┌───┐":FORI=1TO75:NE
XT
330 PRINT"┌───┐";NA$(Y):FORI=1TO125:NEXT
340 GETG$:IFG$<>CHR$(134)ANDG$<>CHR$(135
)ANDG$<>CHR$(13)THEN320
350 IFG$=CHR$(13)THENGOSUB90:PRINT"┌───┐>┌───┐";
GOTO400
355 IFG$=CHR$(135)THEN380
360 Z=Z+2:Y=Y+1:IFZ>15THENZ=7:Y=1
370 GOTO320
380 Z=Z-2:Y=Y-1:IFZ<7THENZ=15:Y=5
390 GOTO320
400 ONYGOTO500,510,520,530,540
410 END
500 LOADNB$(1),8
510 LOADNB$(2),8
520 LOADNB$(3),8
530 LOADNB$(4),8
540 PRINT"┌───┐":POKE53272,21:POKE788,49:END
900 REM * VARIABLEN *
910 NA$(1)="┌───┐ 1 ┌───┐":NA$(2)="┌───┐ 2 ┌───┐":
NA$(3)="┌───┐ 3 ┌───┐"
920 NA$(4)="┌───┐ 4 ┌───┐":NA$(5)="┌───┐ 5 ┌───┐"
930 NB$(1)="┌───┐INSRECHNUNG":NB$(2)="┌───┐INSES
ZINSRECHNUNG"
940 NB$(3)="┌───┐ISKONTIEREN EINES *APITALS"
950 NB$(4)="┌───┐RSTELLEN EINES |ILGUNGSPLAE
NES":NB$(5)="┌───┐ROGRAMMENDE"
960 RETURN
READY.

```

Listing »Menüprogramm«

```

7 POKE45,PEEK(174):POKE46,PEEK(175):CLR
8 REM BY KLAUS KLOEKER
9 CLR:DIMA$(15),A(15):GOTO1000
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *
20 FORZ=Z1TOZ2:GOSUB90:PRINT"
":NEXT:IFC=0THENRETURN
30 Z1=12:Z2=20:C=0:GOTO20
50 RETURN
70 A$(12)=STR$(INT(A(12)*100+.5)):REM *
RUNDEN
75 AB$=LEFT$(A$(12),LEN(A$(12))-2)+". "+R
IGHT$(A$(12),2):RETURN
80 LOAD"MENU",8:REM *LOAD MENU
90 POKE214,Z:POKE211,S:SYS58732:RETURN:R
EM * CURSORPOSITION
95 A=55869:FORI=1597TO1622:POKEI,100:POK

```

```

EA,1:A=A+1:NEXT:RETURN
100 REM * AUSWAHLROUTINE *
110 GOSUB90:PRINT"┌───┐ ┌───┐":FORI=1T
075:NEXTI
120 GOSUB90:PRINT"┌───┐"A$(Y)"┌───┐":FORI=1TO125
:NEXT
130 GETG$:IFG$<>CHR$(133)ANDG$<>CHR$(134
)ANDG$<>CHR$(135)ANDG$<>CHR$(13)THEN110
140 IFG$=CHR$(133)ANDCTHENC=0:RETURN
150 IFG$=CHR$(13)THENGOSUB90:PRINT"┌───┐>┌───┐":
RETURN
160 IFG$=CHR$(135)THEN190
170 Z=Z+SW:Y=Y+1:IFZ>Z2THENZ=Z1:Y=Y1
180 GOTO110
190 Z=Z-SW:Y=Y-1:IFZ<Z1THENZ=Z2:Y=Y2
195 GOTO110
200 REM * WAHL-FUSSZEILE *:S=0
210 Z=23:GOSUB90:PRINT"┌───┐ ┌───┐":FO
RI=1TO75:NEXT
220 GOSUB90:PRINT"┌───┐"BA$(Y)"┌───┐":FORI=1TO1
75:NEXT
230 GETG$:IFG$<>CHR$(134)ANDG$<>CHR$(135
)ANDG$<>CHR$(13)THEN210
240 IFG$=CHR$(13)THENGOSUB90:PRINT"┌───┐>┌───┐";
:RETURN
250 IFG$=CHR$(135)THEN280
260 S=S+10:Y=Y+1:IFS>31THENS=0:Y=1
270 GOTO210
280 S=S-10:Y=Y-1:IFS<0THENS=30:Y=4
290 GOTO210
300 REM * EINGABEROUTINE *
310 GOSUB90
320 IN$=""
330 PRINT"┌───┐";
340 GETG$:IFG$=""THEN340
350 G=ASC(G$):IFG=13THENGOTO460
370 IFG$<"0"ORG$>"9"THEN420
380 IN$=IN$+G$
390 PRINTG$;
400 IFLEN(IN$)>NTHENFORI=1TOLEN(IN$):PRI
NTCHR$(20);:NEXT:GOTO320
410 GOTO330
420 IFG=46THEN380
425 IFG$=CHR$(133)THEN9
430 IFG<>20THEN330
440 IFLEN(IN$)<1THEN330
450 IN$=LEFT$(IN$,LEN(IN$)-1):GOTO390
460 PN$="":FORI=1TO(10-LEN(IN$)):PN$=PN$
+ ". ":NEXT:PRINTPN$
470 IFLEN(IN$)<1THEN310
480 RETURN
999 :
1000 REM ** ZINSRECHNUNG **
1010 GOSUB5000:PRINTMA$
1020 A=55388:FORI=1076TO1906STEP40:POKEI
,101:POKEA,1:A=A+40:NEXT
1030 A=55736:FORI=1464TO1474:POKEI,100:P
OKEA,1:A=A+1:NEXT
1040 A=55749:FORI=1477TO1502:POKEI,100:P
OKEA,1:A=A+1:NEXT
1050 Z=23:S=0:GOSUB90:PRINTMBS$
1060 Z=3:GOSUB90:PRINT"┌───┐AUSWAHL :":PRINT"
┌───┐"
1070 Z=5:FORI=1TO5:GOSUB90:PRINT"┌───┐"A$(I)
"┌───┐":Z=Z+1:NEXT
1080 Z=5:Z1=Z:Z2=9:S=0:Y=1:Y1=Y:Y2=5:SW=
1:C=0:GOSUB100:F=Y:IFF=5THEN80
1090 Z=13:S=0:GOSUB90:PRINT"┌───┐LAUFZEIT IN"
:PRINT"┌───┐"
1100 Z=15:FORI=6TO8:S=0:GOSUB90:PRINT"┌───┐"

```



```

4010 C=1:GOSUB10:IFDTHENGOSUB40
4020 Z=2:S=13:GOSUB80:PRINT"ANWENDUNG :
":Z=3:GOSUB80:PRINT"-----"
4030 Z=4:S=13:FORI=9TO12:GOSUB80:PRINT"
" A$(I) " " :";A(I):Z=Z+2:NEXT
4040 Z=23:S=0:GOSUB80:PRINTMD$;
4050 Z=4:Z1=Z:Z2=10:S=13:Y=9:Y1=Y:Y2=12:
SW=2:C=1:GOSUB100
4060 IFC=0THENGOSUB10:GOTO3000
4070 Z=13:S=13:GOSUB80:PRINT" A$(Y) " " :";
PM$:GOSUB90
4080 Z=23:S=0:GOSUB80:PRINTMC$;:Z=13:S=2
7:GOSUB300:A(Y)=VAL(IN$):GOTO4010
4500 REM * PRINTER
4505 TA$=CHR$(10):TB$=CHR$(16)
4510 OPEN4,4,7:PRINT#4,CHR$(14);TB$"25$
"
4520 PRINT#4,TB$"25-----"TA$
4530 FORI=9TO12:PRINT#4,CHR$(15)TB$"10**
* "A$(I) " : "A(I):NEXT
4540 A=16:GOSUB50:PRINT#4,TA$;TB$"14"A$(
F) " : "A$
4550 PRINT#4,TB$"15";:FORI=1TO2+LEN(A$(F
)+AB$):PRINT#4,"=";:NEXT
4560 PRINT#4,CHR$(13):IFDTHENGOSUB4620
4565 PRINT#4,TA$;TA$;TA$
4570 PRINT#4,CHR$(13);TB$"10***  KAPITAL
UND  INSSATZ WERDEN";
4580 PRINT#4," IN '  '  AUSGEDRUCKT,"
4590 PRINT#4,TB$"16INSSATZ UND  AHRESZI
NS (FALLS VORHANDEN) IN ' % ' !"
4600 PRINT#4:FORI=1TO40:PRINT#4,"=";:NE
XT:PRINT#4,TA$;TA$;TA$
4610 CLOSE4:FORI=56176TO56215:POKEI,0:NE
XT:GOTO3620
4620 IFDTHENPRINT#4,TA$;TB$"15"A$(17) "
A$(18) " : ";
4630 A=19:GOSUB50:PRINT#4,AB$
4640 PRINT#4,TB$"15";:FORI=1TOLEN(A$(18)
+A$(19)+AB$)+10:PRINT#4,"=";:NEXT
4650 RETURN
5000 REM * VARIABLEN
5010 MA$="  /  -  /  -  -
"
5020 MB$="  _3 U.  _5 -  AHL ! ' _ / ' -
-INGABE !"
5030 MC$="  _1 - /EU !  EBEN  IE E
IN !  "
5040 MD$="  _1 -  RESULTAT !  _3 U.  _5
-  AHL ! "
5050 KD$="  N.KAPITAL":P$="  INSSATZ ":
KN$="  NDKAPITAL"
5060 N$="  AHRE " :A$(12)="  INSPER./
A":PM$="....."
5100 BA$(1)="  ENDERN "
5110 BA$(2)="  OPIE "
5120 BA$(3)="  /EU "
5130 BA$(4)="  ENUE "
5500 A$(1)=KN$
5510 A$(2)=KD$
5520 A$(3)=P$
5530 A$(4)="  LAUFZEIT "
5540 A$(5)="  ENUE "
5600 A$(13)="  ENDERN "
5610 A$(14)="  RESULTAT "
5620 A$(15)="  ENUE "
5630 A$(17)="  FEKTIVER"
5640 A$(18)="  AHRESZINS":RETURN

```

READY.

```

8 POKE45,PEEK(174):POKE46,PEEK(175):CLR
9 CLR:DIMA$(15),A(15):GOTO1000
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *
20 FORZ=Z1TOZ2:GOSUB90:PRINT"
":NEXT:IFC=0THENRETURN
30 Z1=12:Z2=20:C=0:GOTO20
50 RETURN
70 A$(12)=STR$(INT(A(12)*100+.5)):REM *
RUNDEN
75 AB$=LEFT$(A$(12),LEN(A$(12))-2)+". "+R
IGHT$(A$(12),2):RETURN
80 LOAD"MENU",8:REM *LOAD MENU
90 POKE214,Z:POKE211,S:SYS58732:RETURN:R
EM * CURSORPOSITION
95 A=55869:FORI=1597TO1622:POKEI,100:POK
EA,1:A=A+1:NEXT:RETURN
100 REM * AUSWAHLROUTINE *
110 GOSUB90:PRINT" " " :";FORI=1T
075:NEXTI
120 GOSUB90:PRINT" A$(Y) " " :";FORI=1TO125
:NEXT
130 GETG$:IFG$<>CHR$(133)ANDG$<>CHR$(134
)ANDG$<>CHR$(135)ANDG$<>CHR$(13)THEN110
140 IFG$=CHR$(133)ANDC=0:RETURN
150 IFG$=CHR$(13)THENGOSUB90:PRINT" > " :";
RETURN
160 IFG$=CHR$(135)THEN190
170 Z=Z+SW:Y=Y+1:IFZ>Z2THENZ=Z1:Y=Y1
180 GOTO110
190 Z=Z-SW:Y=Y-1:IFZ<Z1THENZ=Z2:Y=Y2
195 GOTO110
200 REM * WAHL-FUSSZEILE *:S=0
210 Z=23:GOSUB90:PRINT" " " :";:FO
RI=1TO75:NEXT
220 GOSUB90:PRINT" BA$(Y) " " :";:FORI=1TO1
75:NEXT
230 GETG$:IFG$<>CHR$(134)ANDG$<>CHR$(135
)ANDG$<>CHR$(13)THEN210
240 IFG$=CHR$(13)THENGOSUB90:PRINT" > " :";
:RETURN
250 IFG$=CHR$(135)THEN280
260 S=S+10:Y=Y+1:IFS>31THENS=0:Y=1
270 GOTO210
280 S=S-10:Y=Y-1:IFS<0THENS=30:Y=4
290 GOTO210
300 REM * EINGABEROUTINE *
310 GOSUB90
320 IN$=""
330 PRINT" ";
340 GETG$:IFG$=""THEN340
350 G=ASC(G$):IFG=13THEN460
360 IFG=13THENRETURN
370 IFG$<"0"ORG$>"9"THEN420
380 IN$=IN$+G$
390 PRINTG$;
400 IFLEN(IN$)>NTHENFORI=1TOLEN(IN$):PRI
NTCHR$(20);:NEXT:GOTO320
410 GOTO330
420 IFG=46THEN380
425 IFG$=CHR$(133)THEN9
430 IFG$<>20THEN330
440 IFLEN(IN$)<1THEN330
450 IN$=LEFT$(IN$,LEN(IN$)-1):GOTO390
460 PN$="":FORI=1TO(10-LEN(IN$)):PN$=PN$
+ ". " :NEXT:PRINTPN$
470 IFLEN(IN$)<1THEN310
480 RETURN
1000 REM ** DIKONTIEREN **
1010 GOSUB5000:PRINTMA$
Listing »Zinseszinsrechnung«

```

```

1020 A=55388:FORI=1076T01906STEP40:POKEI
,101:POKEA,1:A=A+40:NEXT
1030 A=55736:FORI=1464T01474:POKEI,100:P
OKEA,1:A=A+1:NEXT
1040 A=55749:FORI=1477T01502:POKEI,100:P
OKEA,1:A=A+1:NEXT
1050 Z=23:S=0:GOSUB90:PRINTM$#
1060 Z=3:GOSUB90:PRINT"USWAHL ":"PRIN!"

```

```

1070 Z=5:FORI=1T05:GOSUB90:PRINT" A$(I)
" : Z=Z+1:NEXT
1080 Z=5:Z1=Z:Z2=9:S=0:Y=1:Y1=Y:Y2=5:SW=
1:C=0:GOSUB100:F=Y:IFF=5THEN80
1150 IFF=1THENA$(9)=KN$:A$(10)=P$:A$(11)
=JA$:K=1
1160 IFF=2THENA$(9)=KB$:A$(10)=P$:A$(11)
=JA$:K=1
1170 IFF=3THENA$(9)=KN$:A$(10)=KB$:A$(11)
=JA$:K=0
1180 IFF=4THENA$(9)=KN$:A$(10)=KB$:A$(11)
=P$:K=2
2000 REM * EINGABEMASKE *
2010 Z=3:S=13:GOSUB90:PRINT"-INGABE ":"Z
=4:GOSUB90:PRINT"
2020 Z=5:FORI=9T011:S=13:GOSUB90:PRINT"
A$(I)" : " : PM$
2030 Z=Z+2:NEXT:Z=23:S=0:GOSUB90:PRINTMC
$
2040 Z=5:S=27:N=10:GOSUB300:A(9)=VAL(IN$
)
2050 Z=7:GOSUB300:A(10)=VAL(IN$)
2060 Z=9:GOSUB300:A(11)=VAL(IN$)
2070 Z=13:S=13:GOSUB90:PRINT"-INGABE KOR
REKT ? (J/N) ":GOSUB95
2080 Z=13:S=38:GOSUB90:PRINT"
";
2090 GETG$:IFG$<>"J"ANDG$<>"N"THEN2090
2100 PRINTG$:IFG$="J"THENY=1
2110 IFG$="N"THENY=0
2120 GETI$:IFI$<>CHR$(13)ANDI$<>CHR$(20)
THEN2120
2130 IFI$=CHR$(20)THEN2080
2140 C=1:GOSUB10:IFY=0THEN2010
3000 REM * RECHNUNG *
3010 IFF=1THENA(12)=A(9)*(1/(1+(A(10)/10
0))^A(11))
3020 IFF=2THENA(12)=A(9)/(1/(1+(A(10)/10
0))^A(11))
3030 IFF=3THENA(12)=((A(9)/A(10))^(1/A(1
1))-1)*100
3040 IFF=4THENA(12)=LOG(A(9)/A(10))/LOG(
1+(A(11)/100))
3500 REM * BILDSCHIRM
3510 Z=3:S=13:GOSUB90:PRINT"-INGABE ":"Z
=4:GOSUB90:PRINT"
3520 Z=5:FORI=9T011:GOSUB90:PRINTA$(I)
:"A(I):Z=Z+2:NEXT
3530 Z=13:GOSUB90:PRINT"_ESULTAT ":"Z=14
:GOSUB90:PRINT"
":GOSUB70
3540 Z=16:GOSUB90:PRINT" A$(F)" : " : ;PR
INTTAB(38-LEN(AB$)):AB$
3550 PRINTTAB(39-LEN(AB$)):FORI=1TOLEN(
AB$)-1:PRINT"=" : ;NEXT
3560 Z=20:GOSUB90:IFK=1THENPRINT" USDRUC
K IN !" :GOTO3580
3570 IFK=0THENPRINT" USDRUCK IN TROZENT
!"
3580 Z=23:S=0:GOSUB90:FORI=1T04:PRINT"
BA$(I)" : ;NEXT
3590 Y=1:GOSUB200:IFY=4THEN80
3600 IFY=3THEN1010

```

```

3610 IFY=2THEN4500
4000 REM * AENDERUNG *
4010 C=1:GOSUB10
4020 Z=3:S=13:GOSUB90:PRINT" AENDERUNG :
" : Z=4:GOSUB90:PRINT"
"
4030 Z=5:S=13:FORI=9T011:GOSUB90:PRINT"
A$(I)" : " : ;A(I):Z=Z+2:NEXT
4040 Z=23:S=0:GOSUB90:PRINTMD$;
4050 Z=5:Z1=Z:Z2=9:S=13:Y=9:Y1=Y:Y2=11:S
W=2:C=1:GOSUB100
4060 IFC=0THENGOSUB10:GOTO3000
4070 Z=13:S=13:GOSUB90:PRINT" A$(Y)" :
" : PM$:GOSUB95
4080 Z=23:S=0:GOSUB90:PRINTMC$;Z=13:S=2
7:GOSUB300:A(Y)=VAL(IN$):GOTO4010
4500 REM * AUSDRUCK *
4505 TA$=CHR$(10):TB$=CHR$(16)
4510 OPEN4,4,7:PRINT#4,CHR$(14);TB$"10-
- - - - -"
4520 PRINT#4,TB$"10- - - - -"
-----"TA$
4530 FORI=9T011:PRINT#4,CHR$(15);TB$"10*
** A$(I)" : " : ;A(I):NEXT
4540 GOSUB70:PRINT#4,TA$:TB$"14" A$(F) " :
"AB$
4550 PRINT#4,TB$"15" : ;FORI=1T02+LEN(A$(F)
)+AB$):PRINT#4,"=" : ;NEXT
4560 PRINT#4,TA$:TA$:TA$:TA$
4570 PRINT#4,CHR$(13);TB$"10*** "NDKAP
ITAL UND ARWERT WERDEN";
4580 PRINT#4," IN " AUSGEDRUCKT,"
4590 PRINT#4,TB$"16DER INSSATZ IN " %
!"
4600 PRINT#4:FORI=1T040:PRINT#4,"=" : ;NE
XT:PRINT#4,TA$:TA$:TA$:TA$
4610 CLOSE4:FORI=56176T056215:POKEI,0:NE
XT:GOTO3580
5000 REM * VARIABLENLISTE *
5010 MA$="
"
5020 MB$=" _3 U. _5 - OAH! ' -
-INGABE !"
5030 MC$=" _1 - /EU ! !EBEN VIE E
IN !"
5040 MD$=" _1 - _ESULTAT ! _3 U. _5
- OAH! !"
5050 KN$=" "NDKAPITAL":P$=" "INSSATZ ":
JA$=" AHRE " : PM$="....."
5060 KB$=" ARWERT "
5110 BA$(1)=" AENDERN "
5120 BA$(2)=" OPIE "
5130 BA$(3)=" /EU "
5140 BA$(4)=" \ENUE "
5510 A$(1)=" ARWERT "
5520 A$(2)=" "NDKAPITAL"
5530 A$(3)=" "INSSATZ "
5540 A$(4)=" LAUFZEIT "
5550 A$(5)=" \ENUE "
5590 A$(13)=" AENDERN "
5600 A$(14)=" _ESULTAT "
5610 A$(15)=" \ENUE " : RETURN
READY.

```

Listing
»Diskontieren eines
Kapitals«

```

7 POKE45,PEEK(174):POKE46,PEEK(175):CLR
8 REM (C) BY KLAUS KLOEKER
9 CLR:DIMA$(15),E(50,6),E$(50,6):GOTO100
0
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *
15 FORZ=Z1T0Z2:GOSUB90:PRINT"
" : NEXT:IFC=0THENRETURN

```



```

3800 REM * SCREEN *
3805 GOSUB900
3810 Z=3:S=13:GOSUB90:PRINT"-INGABE :":Z
=4:GOSUB90:PRINT"-----"
3820 Z=5:S=13:FORI=4TO6:GOSUB90:PRINTA$(
I) : "A(I):Z=Z+2:NEXT
3830 Z=14:S=13:GOSUB90:PRINT"_ESULTAT : "
:Z=15:GOSUB90:PRINT"-----"
3840 Z=17:FORJ=3TO5STEP2:GOSUB90:PRINT"
"EC$(J) "E":EE$(J):Z=Z+2:NEXT
3850 Z=21:GOSUB90:PRINT"USDROCK IN
!"
3860 ZA=VAL(EE$(3))/(VAL(EE$(5))/100)
3870 Z=14:S=0:GOSUB90:PRINT"INS-
"Z=15:GOSUB90:PRINT" ANTEIL "
3880 ZA$=STR$(ZA):IFZA<10THENZA$=" "+ZA$
3890 ZA$=LEFT$(ZA$,5):Z=17:S=2:GOSUB90:P
RINTZA$;" X"
3900 Z=18:S=3:GOSUB90:PRINT"====="
3910 Z=23:S=0:GOSUB90:FORI=1TO5:PRINT"
BA$(I) "":NEXT
3920 Y=1:GOSUB200:IFY=5THEN70
3930 IFY=4THEN9
3940 IFY=3THEN5500
3950 IFY=2THEN5000
4000 REM * TABELLE *
4010 GOSUB95:X1=2:X2=4:Y1=1:Y2=A(6)
4020 Z=1:S=0:GOSUB90:FORI=1TO22:PRINTMD$
:NEXT:Z=22:GOSUB90:PRINTMH$
4030 Z=3:S=0:GOSUB90:PRINTM$(1)
4040 Z=3:S=6:FORI=X1TOX2:GOSUB90:PRINTM$(
I):S=S+11:NEXT
4050 Z=4:S=1:GOSUB90:FORI=1TO37:PRINT"="
;:NEXT
4100 Z=6:S=1:FORI=Y1TOY2
4105 I$=STR$(I):IFI<10THENI$="."+I$
4107 GOSUB90:PRINT"."+I$:S=6
4110 FORJ=X1TOX2
4120 GOSUB90:PRINTE$(I,J)
4130 S=S+11:NEXTJ
4140 IFPEEK(214)>15THENY2=I:Y1=Y2-9:GOTO
4200
4150 Z=Z+1:S=1:NEXTI
4200 Z=17:S=1:GOSUB90:FORJ=1TO38:PRINT"-
";:NEXT
4500 WAIT203,63:GETG$
4510 IFG$=CHR$(29)THEN4550
4520 IFG$=CHR$(157)THEN4580
4530 IFG$=CHR$(17)THEN4610
4540 IFG$=CHR$(145)THEN4640
4543 IFG$=CHR$(133)THEN3800
4545 GOTO4500
4550 X1=X1+1:X2=X2+1:REM * ->
4560 IFX2>6THENX2=6:X1=4:GOTO4500
4570 GOTO4030
4580 X1=X1-1:X2=X2-1:REM * <-
4590 IFX1<2THENX1=2:X2=4:GOTO4500
4600 GOTO4030
4610 IFA(6)<10THENGOTO4500
4615 Y1=Y1+10:Y2=Y2+10:REM * DOWN
4620 IFY2>A(6)THENY2=A(6):Y1=Y2-9:IFI=A(
6)THEN4500
4630 GOTO4030
4640 IFA(6)<10THENGOTO4500
4645 Y1=Y1-10:Y2=Y2-10:REM * UP
4650 IFY1<1THENY1=1:Y2=Y1+9:IFI<11THEN45
00
4660 GOTO4030
5000 REM * AENDERUNG *
5010 C=1:GOSUB10:S=0:FORZ=13TO20:GOSUB90
:PRINT" " :NEXT
5020 Z=3:S=13:GOSUB90:PRINT" AENDERUNG : "
:Z=4:GOSUB90:PRINT"-----"
5030 Z=5:S=13:FORI=4TO6:GOSUB90:PRINT"
" A$(I) "E":A(I):Z=Z+2:NEXT
5040 Z=23:S=0:GOSUB90:PRINTMF$;
5050 Z=5:Z1=Z:Z2=9:S=13:Y=4:Y1=Y:Y2=6:SW
=2:C=1:GOSUB100
5060 IFC=0GOTO3000
5070 Z=14:S=13:GOSUB90:PRINT" A$(Y) "E":
";PM$:GOSUB80
5080 Z=23:S=0:GOSUB90:PRINTMD$;:Z=14:S=2
7:GOSUB300:A(Y)=VAL(IN$)
5090 A$(Y)=" "+MID$(A$(Y),2,10):GOTO5010
5500 REM * PRINTER
5505 TB$=CHR$(16):TA$=CHR$(10)
5510 OPEN4,4,7:PRINT#4,CHR$(14)TB$"10
US
DRUCK EINES | ILGUNGSPLANES"
5520 PRINT#4,TB$;"10-----
-----"TA$
5530 FORI=4TO6:PRINT#4,CHR$(15)TB$"10***
"A$(I) : "A(I):NEXT
5540 PRINT#4,TA$;TA$;TA$
5550 FORI=1TO6:PRINT#4," "M$(I);:NEXT:
PRINT#4," "
5560 FORI=1TO40:PRINT#4,"--";:NEXT:PRINT
#4
5570 FORI=1TOA(6):I$=STR$(I):IFI<10THENI
$="."+I$
5575 PRINT#4," ."+I$ ";
5580 FORJ=2TO6:PRINT#4," .."E$(I,J);:NEX
TJ:PRINT#4," ":NEXTI
5590 FORI=1TO40:PRINT#4,"--";:NEXT:PRINT
#4
5595 PRINT#4," |ESAMT:";
5600 PO=27:FORJ=3TO5STEP2:PRINT#4,SPC(PO
-LEN(EE$(J)))EE$(J);:PO=26:NEXT
5610 PRINT#4,TA$;TA$;TB$"10*** USDROCK
DER |ABELLE ERFOLGT IN " !
5620 PRINT#4,TB$"10*** |ERECHNUNG DER
NSEN ERFOLGT AM NDE DES AHRES !"
5630 PRINT#4,TA$;TB$"10*** |ILGUNG KONST
ANT = _ATENTILGUNG !!"
5640 PRINT#4,TB$"10*** ANNUITAET KONSTAN
T = ANNUITAETENTILGUNG !!"
5650 PRINT#4:FORI=1TO40:PRINT#4,"=";:NE
XTI:PRINT#4,TA$;TA$;TA$;TA$
5660 CLOSE4:GOTO3860
6000 REM * VARIABLEN *
6010 MA$="
"
6020 MB$=" _3 U. _5 - OAH ! _ / -
-INGABE !"
6030 MC$=" _1 - USWAHL ! |EBEN #IE E
IN ! "
6040 MD$="
"
6050 ME$="
"
6060 MF$=" _1 - _ESULTAT ! _3 U. _5
- OAH ! "
6070 MG$=" |ITTE OARTEN !!"
6080 MH$=" _1 - _ESULTAT ! -URSOR -
#TEUERN ! "
6100 A$(1)=" _ATENTILG."
6110 A$(2)=" ANNU.TILG."
6120 A$(3)=" \ENUE "
6130 A$(4)=" -ARLEHEN "
6140 A$(5)=" #INSSATZ "
6150 A$(6)=" AHRE "

```

Listing »Tilgungsplan«

```

6160 PM$="....."
6200 M$(1)="  AHR  "
6210 M$(2)=" _ESTSCHULD"
6220 M$(3)="  #INSEN"
6230 M$(4)="  |ILGUNG"
6240 M$(5)=" #NNUITAET"
6250 M$(6)=" #NNU. / \ON"
6300 EC$(3)=" #INSEN  GES."

```

```

6310 EC$(5)=" #NNUIT. GES."
6400 BA$(1)="  LABELLE"
6410 BA$(2)=" #ENDERN"
6420 BA$(3)="  OPIE  "
6430 BA$(4)="  /EU   "
6440 BA$(5)="  \ENUE  "
6500 RETURN
READY.

```

Listing »Tilgungsplan« (Schluß)

Fortsetzung von Seite 67

HYPRA-LOAD: Schnelles Laden von Diskette

Warum Zeit verschwenden? Laden Sie Ihre Programme in Zukunft fünfmal schneller.

Schon ganze Scharen von Computerbesitzern werden hilflos gestöhnt haben, als sie wieder einmal eine Kaffeepause einlegen mußten, um das Laden des nächsten Programms von Diskette abzuwarten. War das Programm dann auch noch länger als 150 Blocks, so konnte diese »Zwangsendepause« durchaus länger als 2,5 Minuten werden.

Auch Hardware-Hersteller haben diesen Mangel erkannt und bieten neuerdings immer mehr schnelle Laufwerke als Ersatz für die VC 1541 an.

Halt! Jetzt aber Schluß mit dem dauernden Herziehen über die VC 1541, die nämlich eigentlich gar keine Schuld an den langen Wartezeiten hat; vielmehr kann sie es, was die Geschwindigkeit betrifft, ohne weiteres mit den großen CBM-Floppys aufnehmen, oder anders formuliert, die VC 1541 ist genauso schnell wie zum Beispiel die CBM 2031.

Aber wo liegt denn dann nun der Haken? Das Stichwort lautet: Der serielle Bus, also die Schnittstelle zwischen Floppy und Computer. Diese Verbindung erfordert zwar weniger Materialaufwand als der üblicherweise bei Commodore benutzte IEEE-488-Bus, (der 8-Bit-parallel arbeitet). Der serielle Bus hat aber den Nachteil, daß die Geschwindigkeit bei der Übertragung im Normalbetrieb zirka 5 bis 6mal langsamer ist als beim IEC-Bus, was sich dann auch auf die Ladezeit auswirkt.

Die Hardwarelösung habe ich ja schon erwähnt: Sie erfordert außer einer neuen Floppy, auch noch das IEEE-Bus-Modul, da der C 64 in der Grundversion keine derartige Schnittstelle zu bieten hat. Mitunter ist also diese Lösung sehr teuer und übersteigt die Kaufkraft vieler C 64-Anwender.

Aber zum Glück gibt es noch eine Softwaremöglichkeit, hier effektiv einzugreifen. Das Prinzip ist denkbar einfach. Es beruht auf der Möglichkeit, in die Floppy Maschinenprogramme einzuspeichern und diese dann dort auszuführen.

In unserem Fall werden also modifizierte Busroutinen in der Floppy und im Computer abgelegt, die dann die sonst so langsame Übertragung auf ein Vielfaches steigern.

In Zahlen: Der serielle Bus wird durch diese Routinen weit über 10mal so schnell und das Laden geschieht nun mit der maximalen Floppygeschwindigkeit, die etwa 5 bis 6mal höher ist.

Nun aber zum Programm, damit Sie so schnell wie möglich in den Genuß von HYPRA-LOAD kommen: Wenn Sie den Basic-Lader eintippen, achten Sie bitte besonders sorgfältig auf richtiges Eingeben der Checksummen, da diese zur Fehlerkontrolle bei der Eingabe der DATAs dienen!

Vor dem ersten Startversuch das Programm bitte unbedingt auf Diskette speichern, da es sich bei der Ausführung automatisch überschreibt. Anschließend starten Sie mit RUN. Wurde alles korrekt eingegeben, wird am Basic-Anfang ein Maschinenprogramm generiert und das Basic-Programm gelöscht.

Dieses Maschinenprogramm ist das eigentliche HYPRA-LOAD und kann nun mit »SAVE« abgespeichert werden.

Um den Geschwindigkeitsvorteil auch wirklich nutzen zu können, wird empfohlen, HYPRA-LOAD möglichst auf alle Programm-Disketten zu überspielen. Es wurde dazu extra sehr kurz (6 Blocks) gehalten. Müßten Sie nämlich nach jedem Laden zuerst die Diskette wechseln, so wäre die Effektivität von HYPRA-LOAD erheblich vermindert.

Übrigens: Die Einschaltmeldung von HYPRA-LOAD umfaßt auch eine Bytes-free-Meldung, welche jedoch ignoriert werden sollte, da die Zahlenangabe aus programmtechnischen Gründen falsch ist. Dies hat jedoch keinen Einfluß auf die einwandfreie Funktion des Computers!!!

Nun aber zu den besonderen Eigenschaften — HYPRA-LOAD kopiert nach dem Kaltstart das gesamte Betriebssystem vom ROM ins darunterliegende RAM, jedoch ohne dabei Basic-Speicherplatz zu verbrauchen. Danach werden die einzelnen Programmteile im Betriebssystem verteilt und das Betriebssystem im RAM gestartet.

— Abzulesen ist die Aktivierung von HYPRA-LOAD an der Speicherstelle 1 der Zeropage. Diese enthält normalerweise den Wert 55, was besagt, daß ab \$A000 ROM aktiviert ist, welches Basic und das Betriebssystem enthält.

HYPRA-LOAD schaltet den Wert auf 53 um und aktiviert dadurch anstelle von ROM-Bausteinen freien RAM Bereich, in dem nun das Betriebssystem und Basic stehen.

Man kann also ganz einfach HYPRA-LOAD abschalten, durch POKE 1,55

und, sofern nicht zerstört, mit POKE 1,53

wieder einschalten.

— Da HYPRA-LOAD im RAM steht, ergeben sich natürlich einige Probleme. So gibt es viele Programme, die beim Laden den Bereich ab \$A000 überschreiben. Würde nach der Rückkehr aus der Laderoutine nun ins RAM ab \$A000 gesprungen, so wäre ein »Aufhängen« des Computers fast unvermeidlich.

Gegen solche Programme ist HYPRA-LOAD abgesichert, und es wird nach dem Ladevorgang automatisch ins ROM-Basic zurückgesprungen. Bei mehrteiligen Programmen erkennt man dies daran, daß plötzlich wieder »normal« geladen wird.