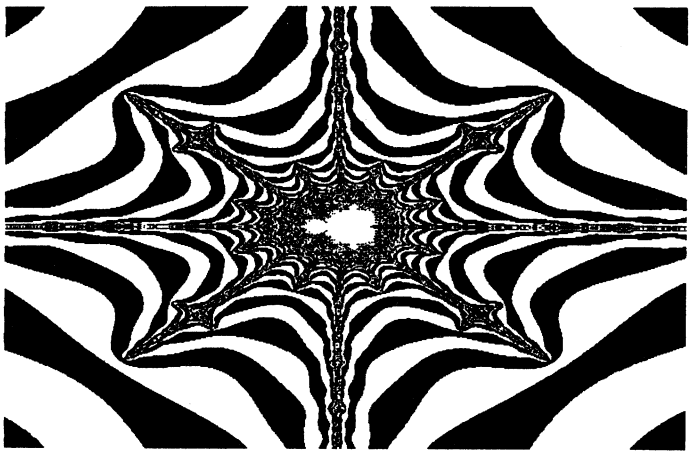
Apfelmännchen mit 256000 Punkten



Normale Hardcopies erreichen beim Vergrößern keine höhere Auflösung. Im Gegensatz dazu können Sie auf dem MPS 802 mit wenig Aufwand und viel Zeit die Grafiken des »Apfelmännchen« in achtfacher Auflösung drucken.

Bevor Sie ans Abtippen gehen, vergewissern Sie sich bitte, daß Ihr Drucker ein MPS 802 der Version 7 ist. Das erreichen Sie, indem Sie beim Einschalten des Druckers die Taste »Paper Advance« gedrückt halten. Nur bei der Version 7 kann das Programm hundertprozentig funktionieren.

Geben Sie die Programme »APFEL/BAS« (Listing 1) und »APFEL/OBJ« (Listing 2) ein und speichern Sie sie auf Diskette oder Kassette. Für die Kassettenversion ändern Sie in Zeile 100 die Gerätenummer »8« zu »1«. Dann laden Sie das Basic-Programm »APFEL/BAS« und starten es mit <RUN>. Dieses Programm lädt dann das Maschinenprogramm »APFEL/OBJ« nach. Nachdem das passiert ist, erscheint das Menü. In diesem können Sie aus vier Menüpunkten auswählen.

Wenn Sie den Menüpunkt »Berechnen« wählen, können Sie die Parameter eingeben, wie Sie es von dem Programm aus Ausgabe 11/85 gewohnt sind. Nachdem Sie diese Eingaben gemacht haben, beginnt der Computer zu rechnen. Sie erkennen das an der wechselnden Umrandungsfarbe, die für jeden berechneten Punkt einmal ihre Farbe wechselt (daran erkennen Sie, daß Ihr Computer arbeitet). Sobald das Bild fertig berechnet ist, erscheint wieder das Menü.

Wenn Sie anschließend den Menüpunkt »Speichern« wählen, brauchen Sie nur den Filenamen einzugeben und das aktuelle Bild wird gespeichert. Anschließend wird der Fehlerkanal der Floppy ausgelesen und ausgegeben. Das Speichern funktioniert übrigens auch nach einem Abbruch.

Das Laden funktioniert genauso. Nur werden nach der Fehlermeldung noch die Parameter des Bildes ausgegeben.

Wenn Sie nun den Menüpunkt »Drucken« wählen, erscheint zunächst die Frage, ob Sie Normal oder Invers drucken wollen. Dabei bedeutet »Normal«, daß eine Fläche, bei der die Folge den gewählten Grenzwert nicht überschreitet, weiß bleibt. Das heißt also, daß das Apfelmännchen weiß bleibt. Wenn Sie »Invers« wählen, wird eben dieses Apfelmännchen schwarz (je nach Farbband auch grau) gedruckt. Da dieses Programm aufgrund der hohen Auflösung sehr viel Rechenzeit benötigt, empfiehlt es sich, dieses Programm für Bilder zu benutzen, die mit dem Programm aus Ausgabe 11/85 erstellt wurden.

Die Rechenzeit des »Apfelpainters« beträgt etwa das Achtfache der Zeit, die das Programm für ein Bild benötigt (wegen der achtfachen Auflösung).

Da das Basic-Programm sehr einfach programmiert ist, soll ausschließlich das Assemblerprogramm (Listing 3) erklärt werden.

Dieses besteht aus den Hauptroutinen »APFEL«, »LOAD«, »SAVE«, und »PRINT«. Die am Anfang des Programms stehende Sprungleiste gewährt einwandfreien Zugang zu allen Programmteilen. Die Routine »START« liest erst die Werte aus dem SYS-Befehl aus (»HOLWERTE«) und springt dann in die Routine zum Berechnen des Bildes »APFEL«. Diese entspricht größtenteils der Rechenroutine von G. Pehlands Programm. Die im folgenden erwähnten Unterschiede erklären sich durch die bedeutend größere Auflösung und der Anordnung der Pixel im Speicher.

Der erste Unterschied besteht in den Zeilen 4080 - 4100. Dort wird der 16-Bit-Wert »SPALTEN« in Fließkommaformat umgerechnet. Die Routine »INTTOFAC« ist aus der Betriebssystemroutine »$BDCD« entnommen. Die gleiche Befehlsfolge steht in den Zeilen 4230 - 4250, nur diesmal für den 16-Bit-Wert »ZEILEN«. Die nächste Änderung steht in den Zeilen 4430 - 4433. Diese Änderung ergibt sich daraus, daß der Einfachheit halber »POS« von Acht auf Null heruntergezählt wird. Der nächste Unterschied besteht darin, daß das Carry-Flag zur Anzeige eines gesetzten Punktes benutzt wird (Zeile 4560 - 4570). Diese Befehlsfolge löscht also den nächsten Punkt. In Zeile 4810 - 4830 wird der Routine »PLOT« mitgeteilt, ob der entsprechende Punkt gesetzt oder gelöscht ist. Dazu wird ausgenutzt, daß das niederwertigste Bit angibt, ob der Byte-Wert gerade oder ungerade ist. (Die Auflösung der Farben beschränkt sich beim Drucker ja logischerweise auf zwei Farben: Weiß und Schwarz entsprechend einem geraden oder ungeraden Wert von »TIEFE«.) Aufgrund dieser Übergabemethode wurde natürlich auch der Anfang der Routine »PLOT« (Zeile 5200 - 5220) geändert. Dort wird dann das Carry-Flag in den Wert »BYTE« geSHIFTet. Und in Zeile 5225 wird für jeden berechneten Punkt die Umrandungsfarbe geändert. In Zeile 5400 wurde die Anzahl der Spalten von 40 (Bildschirm) auf 80 (Drucker) hochgesetzt. In den Zeilen 5520 - 5580 wurde von der ursprünglichen 8-Bit-Operation auf eine 16-Bit-Operation umgestellt. Die Routine »SETBYTE« (Zeile 6000 - 6080) speichert das in der Routine »PLOT« zusammengestellte Byte. Und zwar werden diese Bytes in der Reihenfolge, wie sie errechnet werden, gespeichert. Es entsteht also eine andere Anordnung der Bitmap als in der Bitmap des Grafikbildschirms.

Die Routine »PRINT« transportiert jeweils den Inhalt einer Druckzeile in einen Zwischenspeicher, der dann von der Routine »OUTPUT« ausgegeben wird.

In Zeile 12000 werden die nötigen Files geöffnet. Dann wird in Zeile 12010 die Anzahl der Druckzeilen definiert. Anschließend wird in den Zeilen 12030 - 12060 der Zeiger

auf den Grafikspeicher gesetzt. In den Zeilen 12070 - 12100 wird nun ein zweiter Zeiger auf den Zwischenspeicher gesetzt. In den Zeilen 12120 - 12130 wird dann der Inhalt der betreffenden Druckzeile aus dem Grafikspeicher in den Zwischenspeicher transportiert. In den folgenden Zeilen werden nun die Zeiger erhöht und anschließend wird abgefragt, ob der Zwischenspeicher schon voll ist (Zeile 12210 - 12250). Wenn der Zwischenspeicher voll ist, wird dieser ausgegeben (Zeile 12260). Anschließend wird der Zähler für die Zeilen dekrementiert. Wenn er noch nicht gleich Null ist, wird auf die gleiche Art die nächste Druckzeile ausgegeben. Sonst wird die Ausgabe wieder auf den Bildschirm gelegt (Zeile 12285) und die Files werden geschlossen (Zeile 12290).

Nachdem das Bitmuster einer Zeile in den Puffer »SPEICHER« übertragen wurde, wird diese Zeile von der Routine »OUTPUT« ausgedruckt. Als erstes werden die 8 Byte eines Zeichens in einen Zwischenspeicher gebracht. Diese 8 Byte befinden sich aufgrund der Anordnung der Bitmap in jeweils 80 Byte Abstand. Zuerst wird ein Zwischenspeicher für den Zeiger auf das jeweilige Byte angelegt (Zeile 13000 - 13650). In den Zeilen 13060 - 13070 wird dann der Zähler für die Anzahl der Leerzeichen vor dem nächsten Zeichen gesetzt und in den nächsten beiden Zeilen wird dann noch ein Zähler für die Anzahl der Spalten gesetzt. In der Zeile 13120 wird nun das X-Register als Zähler für die Anzahl der Pixelzeilen definiert. Dann wird in den Zeilen 13140 - 13170 ein Zeiger auf das aktuelle Byte definiert. In den Zeilen 13180 - 13270 werden jetzt die 8 Byte eines Zeichens aus »SPEICHER« nach »PUFFER« kopiert. Die Bytes stehen jeweils in einem Abstand von 80 Byte im »SPEICHER« (Zeile 13220). Anschließend werden nun die 8 Byte eines Zeichens, die ja horizontal orientiert sind, vertikal orientiert. Das heißt, jeweils die Bits der gleichen Wertigkeit aller 8 Byte ergeben ein Byte. Das geht folgendermaßen vonstatten: Es wird ein Maskenbyte definiert (Zeile 13300), in dem jeweils das entsprechende Bit gesetzt ist. Dieses Maskenbyte wird nun mit allen 8 Byte AND-verknüpft (Zeile 13330) und entsprechend dem Ergebnis wird dann das Carry-Flag gesetzt oder gelöscht (Zeile 13340 - 13360). Dieses Carry-Flag wird dann in das Ergebnisbyte geschoben (Zeile 13370). Dieser Vorgang wiederholt sich nun, bis alle 8 Byte so behandelt worden sind (Zeile 13380 - 13400). Nun wird das Ergebnisbyte noch in Abhängigkeit, ob Normal oder Invers gedruckt werden soll, invertiert (Zeile 13420). Das Ergebnis wird dann im »PUFFER« abgelegt (Zeile 13430). Nun wird noch im Maskenbyte das nächste Bit gesetzt und gleichzeitig getestet, ob schon alle 8 Bits durch sind (Zeile 13442 - 13448).

Nun wird als erstes getestet, ob alle 8 Bitmusterbyte leer sind (Zeile 13450 - 13462). Wenn ja, wird dieses Zeichen gar nicht erst gedruckt, sondern es werden nur die Zeiger aktualisiert (Zeile 13630). In den Zeilen 13469 - 13490 werden nun die 8 Bitmusterbyte an den Drucker geschickt. Anschließend wird in Zeile 13500 - 13580 der Druckknopf auf das aktuelle Zeichen positioniert, was nach einem Wagenrücklauf ja nötig ist. Nachdem das geschehen ist, wird in Zeile 13590 - 13600 das eben definierte Zeichen ausgedruckt. In den nächsten beiden Zeilen wird durch einen Wagenrücklauf das Definieren des nächsten Zeichens ermöglicht. In den Zeilen 13630 - 13660 werden dann die Zähler und Zeiger aktualisiert. Wenn nun die Zeile noch nicht fertig gedruckt ist, wird das nächste Zeichen auf die gleiche Art ausgedruckt.

Das Programm wurde extra so entwickelt, daß man es leicht für einen anderen Drucker umschreiben kann. Man muß dafür dann nur die druckerspezifischen Routinen anpassen (Openfls, Closefls). Die Routine »OUTPUT« muß jedoch für den entsprechenden Drucker vollkommen neu geschrieben werden. Diese Routine muß die Bitmuster, die im »SPEICHER« enthalten sind, ausdrucken. Dieses Bitmuster ist folgendermaßen gespeichert: In den ersten 80 Byte sind der Reihe nach alle Bits der nächsten Pixelzeile enthalten. Die nächsten 80 Byte enthalten die Bits der nächsten Pixelzeile und so weiter bis zur achten Pixelzeile.

Für den erfahrenen Programmierer dürfte das Umschreiben dieses Programms auf einen 8-Nadeldrucker keine Probleme bereiten. Bei Druckern mit 9 Nadeln gibt es zwei Möglichkeiten. Einmal besteht die Möglichkeit, nur mit 8 Nadeln zu arbeiten und damit kompatibel zu der vorliegenden Version zu bleiben, oder man arbeitet mit allen 9 Nadeln, wobei man dann die vertikale Auflösung anpassen oder aber die letzte Zeile gesondert verarbeiten muß, denn die jetzige vertikale Auflösung ist ja nicht durch 9 teilbar.

Das Programm funktioniert auch mit dem C128 einwandfrei. Für die Benutzer dieses Computers empfiehlt es sich aber, in Zeile 2070 den Prozessor auf 2 MHz umzuschalten und in Zeile 2095 wieder auf 1 MHz zurückzuschalten.

Ach ja, es gibt ja die leidigen Probleme mit den verschiedenen Betriebssystemen des MPS 802. Dieses Programm wurde auf einem MPS 802 Version 7c entwickelt. Bei anderen Versionen könnte es Probleme geben, da diese zum Teil nicht grafikfähig sind. (P. Siepen/og)

```
10 REM ********************************    <060>
11 REM *                              *    <060>
12 REM *          APFELPRINTER        *    <125>
13 REM *          WRITTEN 1985        *    <115>
14 REM *        BY PETER SIEPEN       *    <247>
15 REM *     VON-STEPHAN-STRASSE 6    *    <255>
16 REM *       4200 OBERHAUSEN 1      *    <102>
17 REM *                              *    <066>
18 REM ********************************    <068>
100 A=A+1:IF A=1 THEN LOAD"APFEL/OBJ",8,1   <245>
110 POKE 55,0:POKE 56,32:CLR                <005>
120 REM *** SPEICHERGRENZE SETZEN ***       <183>
130 :                                       <106>
140 :                                       <116>
1000 REM ********** MENUE ***********       <180>
1003 POKE 53280,11:POKE 53281,0:PRINT CHR$
     (5);                                   <215>
1005 PRINT CHR$(8)CHR$(14)                  <240>
1010 PRINT"{CLR,DOWN}"TAB(15)"M E N U E"    <189>
1020 PRINT"{3DOWN,SPACE,RVSON,SPACE}1{SPAC
     E,RVOFF,SPACE}BERECHNEN"               <103>
1030 PRINT"{DOWN,SPACE,RVSON,SPACE}2{SPACE
     ,RVOFF,SPACE}DRUCKEN"                  <132>
1040 PRINT"{DOWN,SPACE,RVSON,SPACE}3{SPACE
     ,RVOFF,SPACE}SPEICHERN"                <018>
1050 PRINT"{DOWN,SPACE,RVSON,SPACE}4{SPACE
     ,RVOFF,SPACE}LADEN"                    <243>
1100 GET A$:IF A$=""THEN 1100               <069>
1110 IF VAL(A$)<1 OR VAL(A$)>4 THEN 1100    <178>
1120 ON VAL(A$)GOTO 2000,3000,5000,6000     <168>
1130 GOTO 1000                              <054>
1140 :                                      <100>
1150 :                                      <110>
1900 REM ****** WERTE EINGEBEN *******      <200>
2000 PRINT"{CLR,2DOWN,SPACE}W E R T E :"    <042>
2010 INPUT "{2DOWN,SPACE}LINKER RAND{2SPAC
     E}";LI                                 <193>
2020 INPUT "{DOWN,SPACE}RECHTER RAND ";RE   <034>
2030 INPUT "{DOWN,SPACE}UNTERER RAND ";UN   <164>
2040 INPUT "{DOWN,SPACE}OBERER RAND{2SPACE
     }";OB                                  <005>
2050 INPUT "{2DOWN,SPACE}TIEFE{8SPACE}";T   <121>
2060 IF T<5 OR T>254 THEN 2050              <067>
2065 REM MASCHINENPRG AUFRUFEN              <218>
2070 POKE 821,0                             <177>
2080 POKE 52999,T                           <014>
2090 SYS 49152,LI,RE,UN,OB                  <053>
2100 POKE 53280,0                           <118>
2110 GOTO 1000 : REM MENUE                  <220>
2120 :                                      <064>
2130 :                                      <074>
3000 REM ********** DRUCKEN **********     <006>
3010 PRINT"{CLR,2DOWN,SPACE}D R U C K E N
     "                                      <088>
```

Listing 1. »APFEL/BAS« geben Sie bitte mit dem Checksummer V3 ein

```
3020 PRINT:PRINT:PRINT" WOLLEN SIE NORMAL
     ODER INVERS DRUCKEN ?"                <131>
3030 GET A$:IF A$<>"N"AND A$<>"I"THEN 3030 <174>
3040 IF A$="N"THEN POKE 820,0   :REM MASKE <218>
3050 IF A$="I"THEN POKE 820,255            <120>
3060 PRINT:PRINT:PRINT" FERTIG ?"          <047>
3070 GET A$:IF A$=""THEN 3070              <136>
3080 POKE 821,255 : REM AUFRUF MC-PRG      <106>
3090 SYS 49161                             <035>
3100 PRINT"{CLR,DOWN,SPACE}FERTIG"         <212>
3110 PRINT:PRINT" WOLLEN SIE DIE WERTE AUS
     DRUCKEN LASSEN"                       <003>
3120 GET A$:IF A$<>"J"AND A$<>"N"THEN 3120 <025>
3130 IF A$="N"THEN 1000  :REM MENUE        <243>
3140 OPEN 4,4,7 :REM WERTE DRUCKEN         <235>
3150 FOR I=1 TO 3:PRINT#4:NEXT             <253>
3160 PRINT#4,"LINKER RAND{2SPACE}: ";LI;   <172>
3170 PRINT#4,CHR$(16)"40";                 <159>
3175 PRINT#4,"RECHTER RAND : ";RE          <008>
3178 PRINT#4,CHR$(13)                      <102>
3180 PRINT#4,"UNTERER RAND : ";UN;         <117>
3185 PRINT#4,CHR$(16)"40";                 <174>
3190 PRINT#4,"OBERER RAND{2SPACE}: ";OB    <077>
3200 PRINT#4,CHR$(13)                      <124>
3210 PRINT#4,"TIEFE : ";PEEK(52999)        <134>
3220 CLOSE 4                               <207>
3230 GOTO 1000 :REM MENUE                  <068>
3240 :                                     <168>
3250 :                                     <178>
5000 REM ******** SPEICHERN *********      <079>
5005 PRINT"{CLR,2DOWN,SPACE}S P E I C H E
     R N"                                  <212>
5010 PRINT:PRINT:INPUT " NAME ";NA$        <208>
5015 REM NAME UEBERGEBEN                   <049>
5020 IF LEN(NA$)>16 THEN NA$=LEFT$(NA$,16) <121>
5030 FOR I=1 TO LEN(NA$)                   <243>
5040 POKE 828+I-1,ASC(MID$(NA$,I,1))       <181>
5050 NEXT                                  <234>
5055 POKE 822,LEN(NA$)                     <213>
5060 SYS 49155 : REM AUFRUF MC-PRG         <233>
5070 OPEN 15,8,15 : REM FEHLERKANAL        <121>
5080 INPUT#15,EN,ER$,A,A                   <158>
5090 PRINT"{DOWN}"EN,ER$                   <108>
5100 CLOSE 15                              <112>
5110 GET A$:IF A$=""THEN 5110              <013>
5120 GOTO 1000: REM MENUE                  <182>
5130 :                                     <026>
5140 :                                     <036>
6000 REM ********* LADEN ************     <003>
6005 PRINT"{CLR,2DOWN,SPACE}L A D E N"     <145>
6010 PRINT:PRINT:INPUT " NAME ";NA$        <192>
6015 REM NAME UEBERGEBEN                   <033>
6020 IF LEN(NA$)>16 THEN NA$=LEFT$(NA$,16) <105>
6030 FOR I=1 TO LEN(NA$)                   <227>
6040 POKE 828+I-1,ASC(MID$(NA$,I,1))       <165>
6050 NEXT                                  <218>
6055 POKE 822,LEN(NA$)                     <197>
6060 SYS 49158 : REM AUFRUF MC-PRG         <091>
6070 OPEN 15,8,15 :REM FEHLERKANAL         <105>
6080 INPUT#15,EN,ER$,A,A                   <142>
6090 PRINT"{DOWN}"EN,ER$                   <092>
6100 CLOSE 15                              <096>
6110 GET A$:IF A$=""THEN 6110              <029>
6115 REM WERTE AUSGEBEN                    <228>
6120 PRINT"{DOWN}RECHTER RAND : ";RE       <219>
6130 PRINT"{DOWN}LINKER RAND{2SPACE}: ";LI <140>
6140 PRINT"{DOWN}OBERER RAND{2SPACE}: ";OB <138>
6150 PRINT"{DOWN}UNTERER RAND : ";UN       <199>
6160 PRINT"{DOWN}TIEFE{8SPACE}: ";PEEK(529
     99)                                   <177>
6170 GET A$:IF A$=""THEN 6170              <094>
6180 GOTO 1000:REM MENUE                   <226>
6190 REM ENDE                              <131>
```

© 64'er

Listing 1. »APFEL/BAS« (Schluß)

```
programm : apfel/obj      c000 c465
-----------------------------------

c000 : 4c 0c c0 4c 5a c2 4c bf  78
c008 : c2 4c 5d c3 20 12 c0 4c  ee
c010 : b8 c0 20 fd ae 20 8a ad  62
c018 : a2 b0 a0 cf 20 d4 bb 20  0c
c020 : fd ae 20 8a ad a2 a0 a0  81
c028 : cf 20 d4 bb 20 fd ae 20  a1
c030 : 8a ad a2 d0 a0 cf 20 d4  06
c038 : bb 20 fd ae 20 8a ad a2  ab
c040 : c0 a0 cf 20 d4 bb a9 00  1a
c048 : 85 b5 a9 20 85 b6 60 a9  f9
c050 : 0f a2 04 a0 05 20 ba ff  02
c058 : a9 00 20 bd ff 20 c0 ff  c5
c060 : a9 0a a2 04 a0 00 20 ba  37
c068 : ff a9 00 20 bd ff 20 c0  1e
c070 : ff a9 1a a2 04 a0 06 20  bc
c078 : ba ff a9 00 20 bd ff 20  cc
c080 : c0 ff a2 1a 20 c9 ff a9  cf
c088 : 15 20 a8 ff a2 0a 20 c9  66
c090 : ff a9 0d 4c a8 ff a9 0f  80
c098 : 20 c3 ff a9 0a 20 c3 ff  80
c0a0 : a9 1a 4c c3 ff 85 63 86  a9
c0a8 : 62 a2 90 38 4c 49 bc a5  d4
c0b0 : 6e 45 66 85 6f a5 61 60  75
c0b8 : 78 a9 b0 a0 cf 20 a2 bb  45
c0c0 : a9 a0 a0 cf 20 50 b8 20  83
c0c8 : 0c bc a9 80 a2 02 20 a5  b3
c0d0 : c0 20 af c0 20 12 bb a2  6b
c0d8 : 90 a0 cf 20 d4 bb a9 d0  24
c0e0 : a0 cf 20 a2 bb a9 c0 a0  11
c0e8 : cf 20 50 b8 20 0c bc a9  9b
c0f0 : 90 a2 01 20 a5 c0 20 af  56
c0f8 : c0 20 12 bb a2 30 a0 cf  92
c100 : 20 d4 bb a0 04 b9 b0 cf  fe
c108 : 99 40 cf b9 c0 cf 99 60  9e
c110 : cf 88 10 f1 a9 00 8d 00  36
c118 : cf 8d 01 cf 8d 04 cf a9  74
c120 : 08 8d 05 cf a9 00 8d 03  01
c128 : cf a9 00 8d 06 cf a0 04  e7
c130 : 99 70 cf 99 80 cf 88 10  f1
c138 : f7 ee 06 cf ad 07 cf cd  10
c140 : 06 cf b0 04 18 4c cd c1  79
c148 : a9 70 a0 cf 20 a2 bb a9  a5
c150 : 70 a0 cf 20 28 ba a2 57  9a
c158 : a0 00 20 d4 bb a9 80 a0  e7
c160 : cf 20 a2 bb a9 80 a0 cf  20
c168 : 20 28 ba a2 f7 a0 00 20  64
c170 : d4 bb a9 57 a0 00 20 67  d0
c178 : b8 a5 61 c9 84 90 07 ad  d9
c180 : 06 cf 6a 4c cd c1 a9 80  24
c188 : a0 cf 20 a2 bb a9 70 a0  78
c190 : cf 20 28 ba a5 61 f0 02  fe
c198 : e6 61 a9 60 a0 cf 20 50  4f
c1a0 : b8 20 b4 bf a2 80 a0 cf  de
c1a8 : 20 d4 bb a0 00 a9 f7 20  a3
c1b0 : a2 bb a0 00 a9 57 20 50  ce
c1b8 : b8 a9 40 a0 cf 20 50 b8  1a
c1c0 : 20 b4 bf a2 70 a0 cf 20  0a
c1c8 : d4 bb 4c 39 c1 ad 03 cf  e9
c1d0 : 2a 8d 03 cf ee 20 d0 a9  02
c1d8 : 40 a0 cf 20 a2 bb a9 90  30
c1e0 : a0 cf 20 67 b8 a2 40 a0  40
c1e8 : cf 20 d4 bb ce 05 cf f0  aa
c1f0 : 03 4c 29 c1 a9 08 8d 05  b7
c1f8 : cf 20 4c c2 ee 04 cf ad  ec
c200 : 04 cf c9 50 f0 03 4c 24  09
c208 : c1 a9 00 8d 04 cf a0 05  9b
c210 : b9 b0 cf 99 40 cf 88 10  0d
c218 : f7 ee 00 cf d0 03 ee 01  63
c220 : cf ad 01 cf c9 01 90 0a  fb
c228 : ad 00 cf c9 90 90 03 4c  35
c230 : 4a c2 a9 30 a0 cf 20 a2  9a
c238 : bb a9 60 a0 cf 20 50 b8  a5
c240 : a2 60 a0 cf 20 d4 bb 4c  65
c248 : 24 c1 58 60 a0 00 ad 03  36
c250 : cf 91 b5 e6 b5 d0 02 e6  ea
c258 : b6 60 a9 08 a2 08 a0 01  99
c260 : 20 ba ff ad 36 03 a2 3c  11
c268 : a0 03 20 bd ff 20 c0 ff  4e
c270 : a2 08 20 c9 ff a9 00 20  e5
c278 : a8 ff a5 90 30 39 a9 20  4f
c280 : 20 a8 ff a5 90 30 30 a9  47
c288 : 00 85 b5 a9 20 85 b6 a0  38
c290 : 00 b1 b5 20 a8 ff a5 90  1c
c298 : 30 1d e6 b5 d0 02 e6 b6  ed
c2a0 : a5 b6 c9 9d d0 eb a5 b5  35
c2a8 : c9 01 d0 e5 a2 00 bd 00  04
c2b0 : cf 20 a8 ff e8 d0 f7 a9  02
c2b8 : 08 20 c3 ff 4c cc ff a9  40
c2c0 : 08 a2 08 a0 00 20 ba ff  1b
c2c8 : ad 36 03 a2 3c a0 03 20  bb
c2d0 : bd ff 20 c0 ff a2 08 20  22
c2d8 : c6 ff 20 a5 ff a5 90 4a  5f
c2e0 : 4a b0 2f 20 a5 ff a9 00  53
c2e8 : 85 b5 a9 20 85 b6 a0 00  47
c2f0 : 20 a5 ff 91 b5 e6 b5 d0  20
c2f8 : 02 e6 b6 a5 b6 c9 9d d0  a1
c300 : ef a5 b5 c9 01 d0 e9 a2  ec
c308 : 00 20 a5 ff 9d 00 cf e8  6c
c310 : d0 f7 20 cc ff a9 08 20  2b
c318 : c3 ff a9 b0 a0 cf 20 a2  a9
c320 : bb a9 4c a0 49 20 52 c3  3d
c328 : a9 a0 a0 cf 20 a2 bb a9  9d
c330 : 52 a0 45 20 52 c3 a9 d0  b3
c338 : a0 cf 20 a2 bb a9 4f a0  a4
c340 : 42 20 52 c3 a9 c0 a0 cf  62
c348 : 20 a2 bb a9 55 a0 4e 4c  0a
c350 : 52 c3 85 45 84 46 20 e7  59
c358 : b0 aa 4c d4 bb 20 4f c0  86
c360 : a9 32 8d 00 cf a9 00 85  db
c368 : 57 a9 20 85 58 a9 00 85  2a
c370 : 59 a9 c5 85 5a a0 00 b1  ce
c378 : 57 91 59 e6 57 d0 02 e6  9d
c380 : 58 e6 59 d0 02 e6 5a a5  c8
c388 : 5a c9 c7 d0 ea a5 59 c9  a7
c390 : 80 d0 e4 20 a1 c3 ce 00  29
c398 : cf d0 d2 20 cc ff 4c 96  b3
c3a0 : c0 a9 00 8d 08 cf a9 c5  18
c3a8 : 8d 09 cf a9 08 8d 02 cf  77
c3b0 : a9 00 8d 0e cf a9 50 8d  25
c3b8 : 0a cf a2 08 a0 00 ad 08  24
c3c0 : cf 85 a8 ad 09 cf 85 a9  aa
c3c8 : b1 a8 9d 3c 03 18 a5 a8  95
c3d0 : 69 50 85 a8 90 02 e6 a9  e0
c3d8 : ca d0 ed a9 80 a0 08 a2  2d
c3e0 : 08 8d 0b cf 3d 3c 03 18  5e
c3e8 : f0 01 38 2e 0c cf ad 0b  39
c3f0 : cf ca d0 f0 ad 0c cf 4d  8c
c3f8 : 34 03 99 a1 02 88 ad 0b  7a
c400 : cf 4a 8d 0b cf 90 d8 a2  e3
c408 : 08 bd a1 02 c9 00 d0 09  89
c410 : ca d0 f6 ee 0e cf 4c 4e  0b
c418 : c4 a2 0f 20 c9 ff a2 08  2c
c420 : bd a1 02 20 a8 ff ca d0  8a
c428 : f7 a9 0d 20 a8 ff a2 0a  64
c430 : 20 c9 ff ee 0e cf ae 0e  49
c438 : cf a9 20 ca f0 06 20 a8  4e
c440 : ff 4c 3b c4 a9 fe 20 a8  31
c448 : ff a9 8d 20 a8 ff ee 08  da
c450 : cf d0 03 ee 09 cf ce 0a  84
c458 : cf f0 03 4c ba c3 a9 0d  74
c460 : 20 a8 ff 60 03 ff 88 ff  33
```

Listing 2. »APFEL/OBJ« geben Sie bitte mit dem MSE ein

```
0 REM ********************************
1 REM *          APFELPRINTER        *
2 REM *          WRITTEN 1985        *
3 REM *  FOR C64 + VC 1541 + MPS 802 *
4 REM *       BY PETER SIEPEN        *
5 REM *                              *
6 REM *      RECHENROUTINEN VON      *
7 REM *       GERHARD PEHLAND        *
8 REM *   (AUS 64'ER 11.85 SEITE 80) *
9 REM ********************************
10 OPEN4,4
11 SYS9*4096
20 .OPT P4
30 *= $C000
90 ;
92 ; ********** QUELLCODE ***********
95 ;
98 ; SYSTEMADRESSEN
100 FAC         = 97
110 ARG         = 105
120 CHROUT      = $FFD2
130 DIV         = $BB12 ;FAC=FAC/ARG
140 SUB         = $B853 ;FAC=FAC-ARG
160 FACNARG     = $BC0C ;ARG = FAC
170 ZEIGSUB     = $B850 ;FAC=(A/Y)-FAC
180 ZEIGADD     = $B867 ;FAC=(A/Y)+FAC
190 ZEIGMULT    = $BA28 ;FAC=(A/Y)*FAC
200 KONNARG     = $BA8C ;ARG=(A/Y)
210 KONNFAC     = $BBA2 ;FAC=(A/Y)
220 FACNKON     = $BBD4 ;(X/Y)=FAC
230 CHSIGN      = $BFB4 ;FAC=-FAC
240 FILEPAR     = $FFBA ;FILEPARAMETER
250 FILENAM     = $FFBD ;FILENAME SETZEN
252 FRMNUM      = $AD8A ;AUSDRUCK AUSWERTEN
254 CHKCOM      = $AEFD ;TEST AUF ","
256 IECOUT      = $FFA8
257 IECIN       = $FFA5
258 SUCHVAR     = $B0E7 ;SUCHT VARNAM
260 ;
265 ; VARIABLEN
270 ZEIG1       = $57   ;UNIVERSAL
275 ZEIG2       = $59   ;ZEIGER
280 ZEIGER      = $A8   ;ZEIGER BEI DRUCK
295 SPZEIG      = $B5   ;ZEIGER $2000
300 ZEILE       = $CF00 ;AKT.ZEILE
305 SPALTE      = $CF04 ;AKT.SPALTE
310 ZEILZAHL    = $CF02 ;ZEILENZAEHLER
315 BYTE        = $CF03 ;ENTHAELT 8 PKT
320 POS         = $CF05 ;ZAEHLER 8 PKTE
325 TIEFE       = $CF06 ;AKT. TIEFE
330 TIEFEMAX    = $CF07 ;MAXIMALE TIEFE
335 ZEIGSP      = $CF08 ;ZWISCHENSPEICHER
337 :                   ;FUER ZEIGER
340 SPALTZAHL   = $CF0A ;BEI DRUCK
345 ZS          = $CF0B ;UNIVERSAL-
350 ZS2         = $CF0C ;SPEICHER
355 SPACES      = $CF0E ;ZAEHLER BEI
357 :                   ;DRUCK
360 MASKE       = $0334 ;ENTHAELT MASKE
362 :                   ;BEI DRUCK
365 NAMLEN      = $0336 ;LAENGE FILENAM
370 DX          = $CF90 ;RECHENVARIABLEN
375 DY          = $CF30
380 CX          = $CF40
385 CY          = $CF60
390 XQUAD       = 87
395 YQUAD       = 247
400 XWERT       = $CF70
405 YWERT       = $CF80
410 XMAX        = $CFA0
415 XMIN        = $CFB0
420 YMAX        = $CFC0
425 YMIN        = $CFD0
490 ; KONSTANTEN
500 SPALTEN     = 640   ;ANZAHL SPALTEN
510 ZEILEN      = 400   ;ANZAHL ZEILEN
520 PUFFER      = $033C ;SPEICHER 8 BYTES
530 SPEICHER    = $C500 ;SPEICHER FUER
532 PUFFER1     = $02A1
535 :                   ;EINE DRUCKZEILE
690 ;
800 :           JMP START
810 :           JMP SAVE
820 :           JMP LOAD
830 :           JMP PRINT
890 ;
895 ;
900 START       JSR HOLWERTE
910 :           JMP APFEL
990 ;
995 ;
1000 HOLWERTE   JSR CHKCOM ;LIEST WERTE
1010 :          JSR FRMNUM ;AUS SYS
1020 :          LDX #<XMIN
1030 :          LDY #>XMIN
1040 :          JSR FACNKON
1050 :          JSR CHKCOM
1060 :          JSR FRMNUM
1070 :          LDX #<XMAX
1080 :          LDY #>XMAX
1090 :          JSR FACNKON
1100 :          JSR CHKCOM
1110 :          JSR FRMNUM
1120 :          LDX #<YMIN
1130 :          LDY #>YMIN
1140 :          JSR FACNKON
1150 :          JSR CHKCOM
1160 :          JSR FRMNUM
1170 :          LDX #<YMAX
1180 :          LDY #>YMAX
1190 :          JSR FACNKON
1240 :          LDA #<$2000 ;UND SETZT
1250 :          STA SPZEIG  ;ZEIGER
1260 :          LDA #>$2000
1270 :          STA SPZEIG+1
1280 :          RTS
1290 ;
1295 ;
2000 OPENFLS    LDA #15    ;OEFFNET FILES
2010 :          LDX #04
2020 :          LDY #05
2030 :          JSR $FFBA
2040 :          LDA #$00
2050 :          JSR $FFBD
2060 :          JSR $FFC0
2070 :          LDA #10
2080 :          LDX #04
2090 :          LDY #00
2100 :          JSR $FFBA
2110 :          LDA #$00
2120 :          JSR $FFBD
2130 :          JSR $FFC0
2180 :          LDA #26
2190 :          LDX #04
2200 :          LDY #06
2210 :          JSR $FFBA
2220 :          LDA #$00
2230 :          JSR $FFBD
2240 :          JSR $FFC0
2250 :          LDX #26    ;UND BEREITET
2260 :          JSR $FFC9 ;DRUCKER VOR
2270 :          LDA #21
2280 :          JSR IECOUT
2290 :          LDX #10
2295 :          JSR $FFC9
2300 :          LDA #$0D
```

```
2305 :          JMP IECOUT
2310 ;
2320 ;
3000 CLOSEFLS   LDA #15
3010 :          JSR $FFC3
3020 :          LDA #10
3030 :          JSR $FFC3
3040 :          LDA #26
3050 :          JMP $FFC3
3060 ;
3500 INTTOFAC   STA $63 ;POS. INT-ZAHL
3510 :          STX $62 ;IN A/X IN FAC
3520 :          LDX #$90
3530 :          SEC
3540 :          JMP $BC49
3590 ;
3700 FACSARG    LDA ARG+5 ;VORZEICHEN-
3710 :          EOR FAC+5 ;WECHSEL
3720 :          STA ARG+6 ;FAC + ARG
3730 :          LDA FAC
3740 :          RTS
3750 ;
4000 APFEL      SEI         ;RECHENROUT.
4010 :          LDA #<XMIN ;DX BERECHNEN
4020 :          LDY #>XMIN
4030 :          JSR KONNFAC
4040 :          LDA #<XMAX
4050 :          LDY #>XMAX
4060 :          JSR ZEIGSUB
4070 :          JSR FACNARG
4080 :          LDA #<SPALTEN
4090 :          LDX #>SPALTEN
4100 :          JSR INTTOFAC
4110 :          JSR FACSARG
4120 :          JSR DIV
4130 :          LDX #<DX
4140 :          LDY #>DX
4150 :          JSR FACNKON
4160 :          LDA #<YMIN ;DY BERECHNEN
4170 :          LDY #>YMIN
4180 :          JSR KONNFAC
4190 :          LDA #<YMAX
4200 :          LDY #>YMAX
4210 :          JSR ZEIGSUB
4220 :          JSR FACNARG
4230 :          LDA #<ZEILEN
4240 :          LDX #>ZEILEN
4250 :          JSR INTTOFAC
4260 :          JSR FACSARG
4270 :          JSR DIV
4280 :          LDX #<DY
4290 :          LDY #>DY
4300 :          JSR FACNKON
4310 :          LDY #4      ;CX=XMIN
4320 L13        LDA XMIN,Y ;CY=YMAX
4330 :          STA CX,Y
4340 :          LDA YMAX,Y
4350 :          STA CY,Y
4360 :          DEY
4370 :          BPL L13
4380 :          LDA #0
4390 :          STA ZEILE
4400 :          STA ZEILE+1
4410 :          STA SPALTE
4430 :          LDA #$08
4433 :          STA POS
4440 ITERAT     LDA #0
4450 :          STA BYTE
4460 ITERAT1    LDA #0
4470 :          STA TIEFE
4480 :          LDY #4
4490 L15        STA XWERT,Y
4500 :          STA YWERT,Y
4510 :          DEY
4520 :          BPL L15
4525 ITER1      INC TIEFE
4530 :          LDA TIEFEMAX
4540 :          CMP TIEFE ;"MAX. TIEFE ?
4550 :          BCS ITER4
4560 :          CLC         ;PUNKT CLEAR
4570 :          JMP PLOT
4580 ITER4      LDA #<XWERT ;XWERT↑2
4590 :          LDY #>XWERT ;IN XQUAD
4600 :          JSR KONNFAC
4610 :          LDA #<XWERT
4620 :          LDY #>XWERT
4630 :          JSR ZEIGMULT
4640 :          LDX #XQUAD
4650 :          LDY #0
4660 :          JSR FACNKON
4670 :          LDA #<YWERT ;YWERT↑2
4680 :          LDY #>YWERT ;IN YQUAD
4690 :          JSR KONNFAC
4700 :          LDA #<YWERT
4710 :          LDY #>YWERT
4720 :          JSR ZEIGMULT
4730 :          LDX #YQUAD
4740 :          LDY #0      ;XQUAD + YQUAD
4750 :          JSR FACNKON
4760 :          LDA #XQUAD
4770 :          LDY #0
4780 :          JSR ZEIGADD
4790 :          LDA FAC
4800 :          CMP #$84   ;"SUMME > 8 ?
4810 :          BCC ITER3
4820 :          LDA TIEFE
4830 :          ROR         ;CARRY SETZEN
4840 :          JMP PLOT
4850 ITER3      LDA #<YWERT ;YWERT =
4860 :          LDY #>YWERT ;XWERT*YWERT
4870 :          JSR KONNFAC ;*2-CY
4880 :          LDA #<XWERT
4890 :          LDY #>XWERT
4900 :          JSR ZEIGMULT
4910 :          LDA FAC
4920 :          BEQ ITER2
4930 :          INC FAC
4940 ITER2      LDA #<CY
4950 :          LDY #>CY
4960 :          JSR ZEIGSUB
4970 :          JSR CHSIGN
4980 :          LDX #<YWERT
4990 :          LDY #>YWERT
5000 :          JSR FACNKON
5010 :          LDY #0        ;XWERT =
5020 :          LDA #YQUAD   ;XQUAD-YQUAD
5030 :          JSR KONNFAC ;-CX
5040 :          LDY #0
5050 :          LDA #XQUAD
5060 :          JSR ZEIGSUB
5070 :          LDA #<CX
5080 :          LDY #>CX
5090 :          JSR ZEIGSUB
5100 :          JSR CHSIGN
5110 :          LDX #<XWERT
5120 :          LDY #>XWERT
5130 :          JSR FACNKON
5140 :          JMP ITER1
5200 PLOT       LDA BYTE ;SETZE BIT MIT
5210 :          ROL         ;CARRY
5220 :          STA BYTE
5225 :          INC 53280 ;FARBE
```

Listing 3. »APFEL/SRC«, der Quell-Code für eventuelle Veränderungen

```
5230 :          LDA #<CX ;CX=CX+DX
5240 :          LDY #>CX
5250 :          JSR KONNFAC
5260 :          LDA #<DX
5270 :          LDY #>DX
5280 :          JSR ZEIGADD
5290 :          LDX #<CX
5300 :          LDY #>CX
5310 :          JSR FACNKON
5320 :          DEC POS
5330 :          BEQ APFEL1
5340 :          JMP ITERAT1
5350 APFEL1     LDA #08
5360 :          STA POS
5370 :          JSR SETBYTE
5380 :          INC SPALTE
5390 :          LDA SPALTE
5400 :          CMP #80 ;DRUCKER
5410 :          BEQ APFEL2
5420 :          JMP ITERAT
5430 APFEL2     LDA #0
5440 :          STA SPALTE
5450 :          LDY #5      ;CX=XMIN
5460 L30        LDA XMIN,Y
5470 :          STA CX,Y
5480 :          DEY
5490 :          BPL L30
5500 :          INC ZEILE
5510 :          BNE W1
5520 :          INC ZEILE+1
5530 W1         LDA ZEILE+1
5540 :          CMP #>ZEILEN
5550 :          BCC W2
5560 :          LDA ZEILE
5570 :          CMP #<ZEILEN
5580 :          BCC W2
5590 :          JMP ENDAPFEL
5600 W2         LDA #<DY     ;CY=CY-DY
5610 :          LDY #>DY
5620 :          JSR KONNFAC
5630 :          LDA #<CY
5640 :          LDY #>CY
5650 :          JSR ZEIGSUB
5660 :          LDX #<CY
5670 :          LDY #>CY
5680 :          JSR FACNKON
5690 :          JMP ITERAT
5700 ENDAPFEL   CLI
5710 :          RTS
5990 ;
5995 ;
6000 SETBYTE    LDY #$00 ;BYTE SPEICHERN
6005 :          LDA BYTE
6015 :          STA (SPZEIG),Y
6050 :          INC SPZEIG
6060 :          BNE SPW1
6070 :          INC SPZEIG+1
6080 SPW1       RTS
6090 ;
10000 SAVE      LDA #8
10010 :         LDX #08
10020 :         LDY #1
10030 :         JSR $FFBA
10040 :         LDA NAMLEN
10050 :         LDX #<$033C
10060 :         LDY #>$033C
10070 :         JSR $FFBD
10080 :         JSR $FFC0
10090 :         LDX #8
10100 :         JSR $FFC9
10110 :         LDA #<$2000
10112 :         JSR IECOUT
10114 :         LDA $90
10115 :         BMI SAVEENDE
10116 :         LDA #>$2000
10117 :         JSR IECOUT
10118 :         LDA $90
10119 :         BMI SAVEENDE
10120 :         LDA #<$2000  ;$2000-
10125 :         STA SPZEIG
10130 :         LDA #>$2000
10140 :         STA SPZEIG+1
10150 :         LDY #$00
10160 SAVELOOP  LDA (SPZEIG),Y
10170 :         JSR IECOUT
10174 :         LDA $90
10178 :         BMI SAVEENDE
10180 :         INC SPZEIG
10190 :         BNE SAVEW1
10200 :         INC SPZEIG+1 ;-40192
10210 SAVEW1    LDA SPZEIG+1
10220 :         CMP #>32001+$2000
10230 :         BNE SAVELOOP
10240 :         LDA SPZEIG
10250 :         CMP #<32001+$2000
10260 :         BNE SAVELOOP
10270 :         LDX #$00     ;VARIABLEN
10280 SAVEL1    LDA $CF00,X
10290 :         JSR IECOUT
10300 :         INX
10310 :         BNE SAVEL1
10320 SAVEENDE  LDA #08
10330 :         JSR $FFC3
10340 :         JMP $FFCC
10350 ;
11000 LOAD      LDA #8
11010 :         LDX #08
11020 :         LDY #0
11030 :         JSR $FFBA
11040 :         LDA NAMLEN
11050 :         LDX #<$033C
11060 :         LDY #>$033C
11070 :         JSR $FFBD
11080 :         JSR $FFC0
11090 :         LDX #8
11100 :         JSR $FFC6
11101 :         JSR IECIN
11102 :         LDA $90
11103 :         LSR
11104 :         LSR
11105 :         BCS LOADENDE
11106 :         JSR IECIN
11110 :         LDA #<$2000  ;$2000
11120 :         STA SPZEIG
11130 :         LDA #>$2000
11140 :         STA SPZEIG+1
11150 :         LDY #$00
11160 LOADLOOP  JSR IECIN
11170 :         STA (SPZEIG),Y
11180 :         INC SPZEIG
11190 :         BNE LOADW1
11200 :         INC SPZEIG+1
11210 LOADW1    LDA SPZEIG+1 ;-40192
11220 :         CMP #>(32001+$2000)
11230 :         BNE LOADLOOP
11240 :         LDA SPZEIG
11250 :         CMP #<(32001+$2000)
11260 :         BNE LOADLOOP
11270 :         LDX #$00
11280 LOADL1    JSR IECIN    ;VARIABLEN
11290 :         STA $CF00,X
11300 :         INX
11310 :         BNE LOADL1
11320 LOADENDE  JSR $FFCC
11330 :         LDA #08
11340 :         JSR $FFC3
11350 :         LDA #<XMIN   ;WERTE IN
11360 :         LDY #>XMIN   ;BASIC-
```

```
11370 :          JSR KONNFAC ;VARIABLEN
11380 :          LDA #"L"    ;SCHREIBEN
11390 :          LDY #"I"
11400 :          JSR FACVAR
11430 :          LDA #<XMAX
11440 :          LDY #>XMAX
11450 :          JSR KONNFAC
11460 :          LDA #"R"
11470 :          LDY #"E"
11480 :          JSR FACVAR
11490 :          LDA #<YMIN
11500 :          LDY #>YMIN
11510 :          JSR KONNFAC
11520 :          LDA #"O"
11530 :          LDY #"B"
11540 :          JSR FACVAR
11550 :          LDA #<YMAX
11560 :          LDY #>YMAX
11570 :          JSR KONNFAC
11580 :          LDA #"U"
11590 :          LDY #"N"
11600 :          JMP FACVAR
11610 :
11620 FACVAR     STA $45     ;SCHREIBT
11630 :          STY $46     ;FAC IN VAR
11640 :          JSR SUCHVAR ;MIT NAME IN
11650 :          TAX         ;A/Y
11660 :          JMP FACNKON
11670 :
11680 :
12000 PRINT      JSR OPENFLS ;OEFFNET FILE
12010 :          LDA #50     ;ANZAHL DRUCK
12020 :          STA ZEILE   ;ZEILEN
12030 :          LDA #<$2000 ;ZEIGER AUF
12040 :          STA ZEIG1   ;GRAFIK-
12050 :          LDA #>$2000 ;SPEICHER
12060 :          STA ZEIG1+1
12070 PLOOP1     LDA #<SPEICHER ;ZEIGER
12080 :          STA ZEIG2      ;AUF PUFFER
12090 :          LDA #>SPEICHER ;FUER EINE
12100 :          STA ZEIG2+1    ;ZEILE
12110 :          LDY #$00
12120 PLOOP      LDA (ZEIG1),Y
12130 :          STA (ZEIG2),Y
12140 :          INC ZEIG1
12150 :          BNE PW1
12160 :          INC ZEIG1+1
12170 PW1        INC ZEIG2
12180 :          BNE PW2
12190 :          INC ZEIG2+1
12200 PW2        LDA ZEIG2+1
12210 :          CMP #>(640+SPEICHER)
12220 :          BNE PLOOP     ;"FERTIG ?
12230 :          LDA ZEIG2
12240 :          CMP #<(640+SPEICHER)
12250 :          BNE PLOOP
12260 :          JSR OUTPUT ;ZEILE AUSGB.
12270 :          DEC ZEILE
12280 :          BNE PLOOP1
12285 :          JSR $FFCC  ;ENDE
12290 :          JMP CLOSEFLS
12390 ;
12395 ;
13000 OUTPUT     LDA #<SPEICHER
13010 :          STA ZEIGSP
13020 :          LDA #>SPEICHER
13030 :          STA ZEIGSP+1
13060 :          LDA #$00    ;ANZAHL SPACES
13070 :          STA SPACES ;VOR ZEICHEN
13100 :          LDA #80
13110 :          STA SPALTZAHL
13120 OUTLOOP1   LDX #$08
13130 :          LDY #$00
13140 :          LDA ZEIGSP
13150 :          STA ZEIGER
13160 :          LDA ZEIGSP+1
13170 :          STA ZEIGER+1
13180 OUTLOOP2   LDA (ZEIGER),Y ;8 BYTES
13190 :          STA PUFFER,X   ;EINES
13200 :          CLC            ;ZEICHENS
13210 :          LDA ZEIGER     ;IN PUFFER
13220 :          ADC #80
13230 :          STA ZEIGER
13240 :          BCC W4
13250 :          INC ZEIGER+1
13260 W4         DEX
13270 :          BNE OUTLOOP2
13300 :          LDA #$80
13305 :          LDY #8
13310 LOOP2      LDX #08
13320 :          STA ZS
13330 LOOP1      AND PUFFER,X ;8 BYTES
13340 :          CLC          ;WERDEN
13350 :          BEQ W5       ;VON HORIZ.
13360 :          SEC          ;NACH
13370 W5         ROL ZS2      ;VERTIKAL
13380 :          LDA ZS       ;ORIENTIERT
13390 :          DEX
13400 :          BNE LOOP1
13410 :          LDA ZS2
13420 :          EOR MASKE    ;INV ODER
13430 :          STA PUFFER1,Y ;NORMAL
13440 :          DEY
13442 :          LDA ZS       ;NAECHST.
13444 :          LSR          ;BIT
13446 :          STA ZS
13448 :          BCC LOOP2
13450 :          LDX #08      ;8 LEERBYTE
13451 TLOOP      LDA PUFFER1,X ;WENN JA
13452 :          CMP #$00
13453 :          BNE NOSPACE
13454 :          DEX
13455 :          BNE TLOOP
13461 :          INC SPACES   ;AUSGABE
13462 :          JMP CONT     ;UEBERSPR.
13469 NOSPACE    LDX #15
13470 :          JSR $FFC9
13471 :          LDX #8
13472 TLOOP1     LDA PUFFER1,X ;BITMUSTER
13473 :          JSR IECOUT    ;FUER
13474 :          DEX           ;1 ZEICHEN
13475 :          BNE TLOOP1    ;AN DRUCKER
13480 :          LDA #$0D
13490 :          JSR IECOUT
13500 :          LDX #10
13510 :          JSR $FFC9
13520 :          INC SPACES
13530 :          LDX SPACES    ;DRUCKKOPF
13540 :          LDA #$20      ;POSITION
13550 SLOOP      DEX
13560 :          BEQ SPACEEND
13570 :          JSR IECOUT
13580 :          JMP SLOOP
13590 SPACEEND   LDA #$FE    ;DEF. ZEICHEN
13600 :          JSR IECOUT  ;AUSDRUCKEN
13610 :          LDA #141
13620 :          JSR IECOUT
13630 CONT       INC ZEIGSP ;ZEIGER
13640 :          BNE W6     ;AKTUALISIEREN
13650 :          INC ZEIGSP+1
13660 W6         DEC SPALTZAHL
13670 :          BEQ W7
13680 :          JMP OUTLOOP1
13690 W7         LDA #$0D   ;ENDE
13700 :          JSR IECOUT
13710 OUTEND     RTS
```

Listing 3. Source-Code zu »Apfelpainter« (Schluß)