



Visual Basic for Applications

Prof. Dr. Ralf Laue



Gliederung

- Grundlegende Begriffe
- Was ist VBA? VBA-Entwicklungsumgebung
- Zahlendarstellung im Computer, Variablentypen
- Steueranweisungen, Struktogramme
- Arbeit mit Dateien
- Funktionen, Unterprogramme
- Felder, eigene Datentypen
- Objektorientierte Programmierung
- Erweiterungen von Excel und Word
- Formulare, graphische Oberflächen



Literatur

- http://de.wikibooks.org/wiki/VBA_in_Excel
- LUIS-Buch: Excel 2016 Automatisierung, Programmierung
- Wolf-Gert Matthäus: Programmierung für Wirtschaftsinformatiker



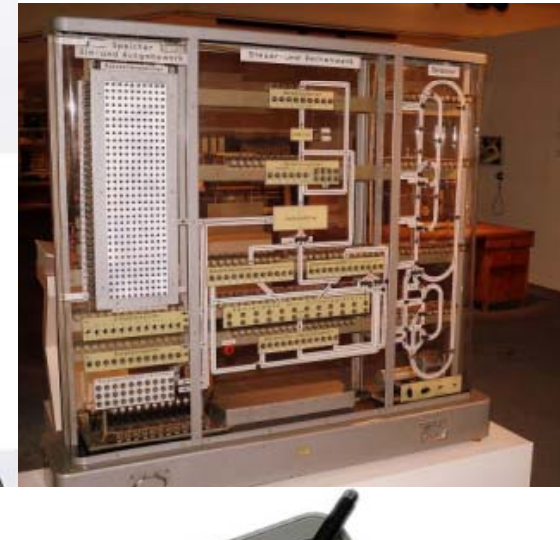


Prüfungsleistung

- Schriftliche Klausur
- 120 Minuten, Verwendung von Unterlagen ist erlaubt
- Voraussetzung für eine Zulassung zur Klausur ist die Lösung der Praktikumsaufgaben.
- In den Praktika behandelte Themen sind ebenso Gegenstand der Prüfung wie der Inhalt der Vorlesung!

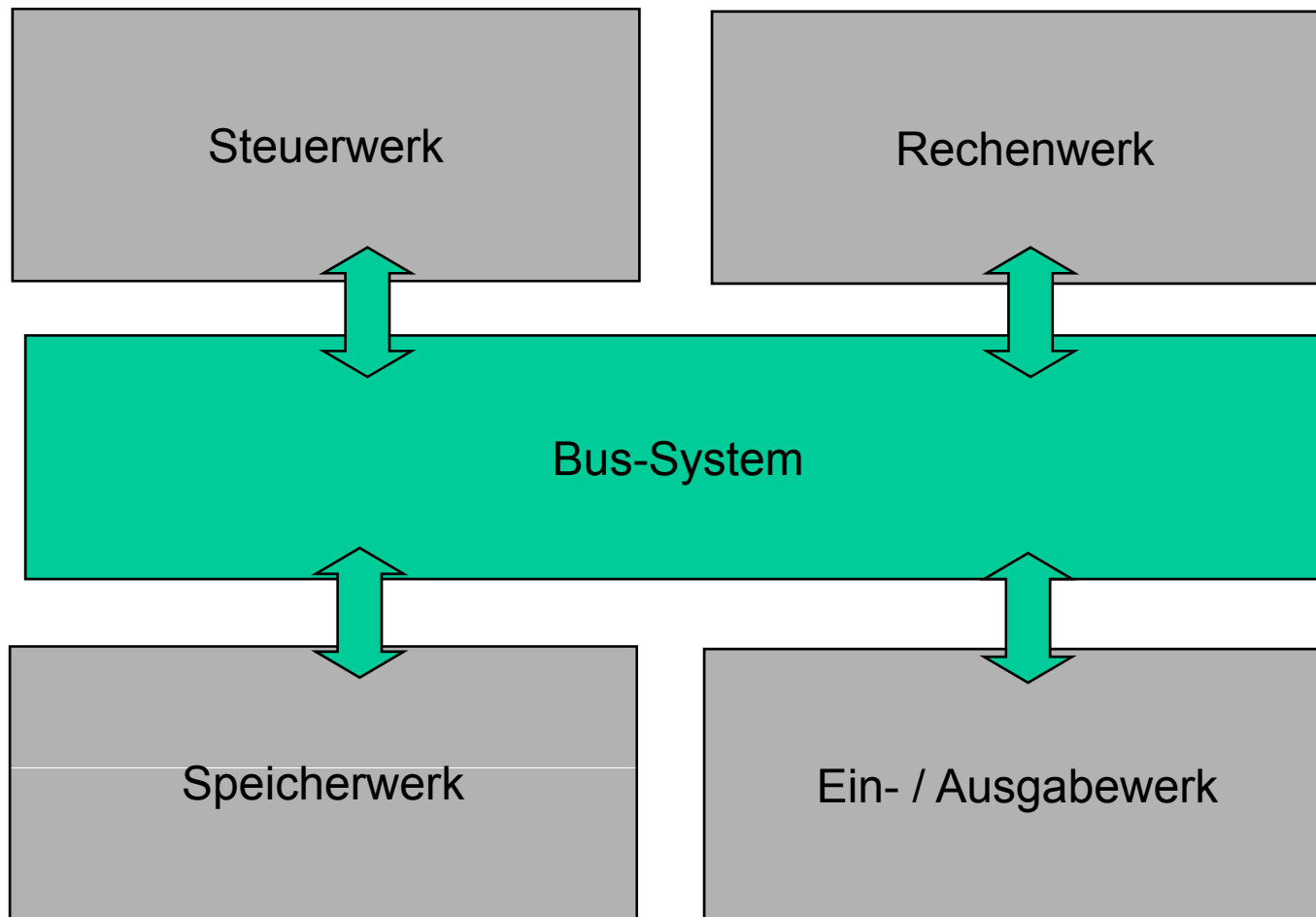


Von-Neumann-Rechner





Von-Neumann-Rechner

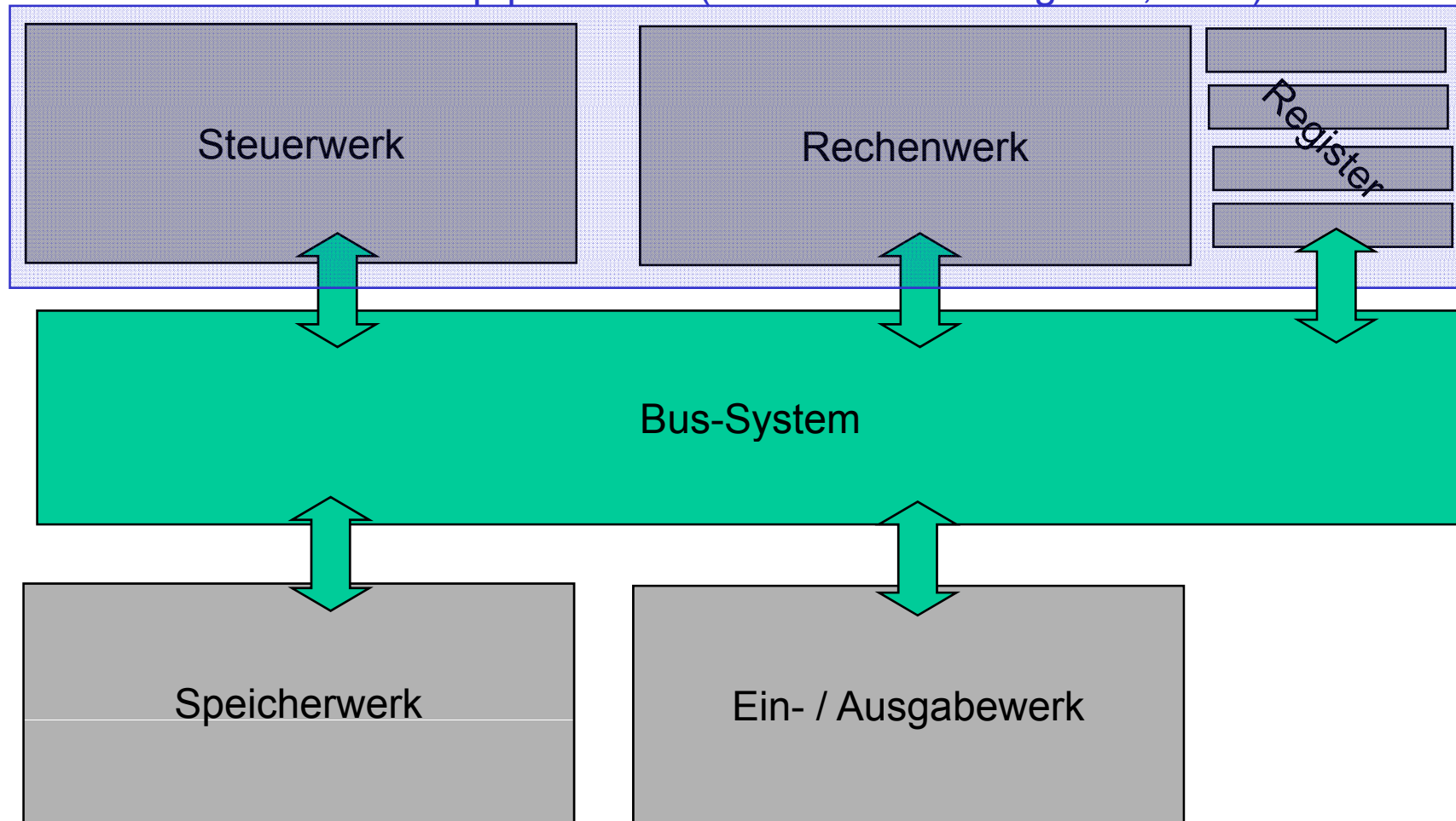


Das Bus-System dient der Kommunikation zwischen den Bestandteilen.



Von-Neumann-Rechner

Hauptprozessor (Central Processing Unit, CPU)



Register – vom Hauptspeicher unabhängiger Speicherort für die an der aktuellen Berechnung beteiligten Daten, sehr schneller Zugriff



Aufbau des Speichers: Speicherzellen

	Nummer (Adresse)	Inhalt
	300	3
	301	"Zwickau"
	...	
	308	
	309	7,81
	...	
	312	



Abarbeitung von Maschinencode

- Wie arbeitet der Rechner bei der Ausführung eines Programms?
- Beispiel (80286-Prozessor)
- Die Aufgabe:
Zwei aufeinanderfolgende Speicherzellen enthalten zwei Werte m und n .
Schreibe den Wert $mn-1$ in die darauffolgende Speicherzelle.



Aufgabe

	Nummer (Adresse)	Inhalt
	300	m
	301	n
	302	mn-1

Ergebnis



Abarbeitung von Maschinencode

Wir sehen in der Abbildung einen Teil des Hauptspeichers sowie die Register.

The screenshot shows a debugger window for the file `D:\SOFTWA~3\Kap02\AFD\AFD.EXE`. The top section displays the state of the CPU registers and the stack. The registers are:

AX	0000	SI	0000	CS	1FA8	IP	0100	Stack	+0 0000	Flags	3204
BX	0000	DI	0000	DS	1FA8				+2 20CD		
CX	0000	BP	0000	ES	1FA8	HS	1FA8		+4 9FFF	OF	DF
DX	0000	SP	FFFE	SS	1FA8	FS	1FA8		+6 9A00	IF	SF
										ZF	AF
										PF	CF

The command line shows the current instruction being executed:

```
CMD > 0100 A00003 MOV AL, [0300]
```

The memory dump shows the contents of memory addresses from `DS:0300` to `DS:0348`:

DS:0300	03	02	00	00	00	00	00	00	00
DS:0308	00	00	00	00	00	00	00	00	00
DS:0310	00	00	00	00	00	00	00	00	00
DS:0318	00	00	00	00	00	00	00	00	00
DS:0320	00	00	00	00	00	00	00	00	00
DS:0328	00	00	00	00	00	00	00	00	00
DS:0330	00	00	00	00	00	00	00	00	00
DS:0338	00	00	00	00	00	00	00	00	00
DS:0340	00	00	00	00	00	00	00	00	00
DS:0348	00	00	00	00	00	00	00	00	00

The bottom section shows a larger memory dump with hexadecimal values and their corresponding ASCII characters:

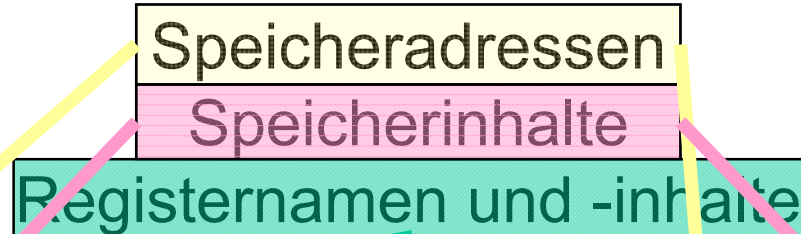
DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01	=	f.ü-■	.-..<.K.
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF	¶.U.¶.§.	
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11			N.ñ.
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00	¿.
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The bottom status bar shows the current step and various debugger controls:

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri



Abarbeitung von Maschinencode



cmd D:\SOFTWA~3\Kap02\APP\APP.EXE

AX 0000	SI 0000	CS 1FA8	IP 0100	Stack +0 0000	Flags 3204
BX 0000	DI 0000	DS 1FA8		+2 20CD	
CX 0000	BP 0000	ES 1FA8	HS 1FA8	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 1FA8	FS 1FA8	+6 9100	0 0 1 0 0 0 1 0

CMD > |

00FF	90	NOP		
0100	A00003	MOV	AL, [0300]	
0103	F72E0103	IMUL	W/[0301]	
0107	48	DEC	AX	
0108	A30203	MOV	[0302], AX	
010B	90	NOP		
010C	90	NOP		
010D	0000	ADD	[BX+SI], AL	
010F	0000	ADD	[BX+SI], AL	

i		0	1	2	3	4	5	6	7
DS:	0300	03	02	00	00	00	00	00	00
DS:	0308	00	00	00	00	00	00	00	00
DS:	0310	00	00	00	00	00	00	00	00
DS:	0318	00	00	00	00	00	00	00	00
DS:	0320	00	00	00	00	00	00	00	00
DS:	0328	00	00	00	00	00	00	00	00
DS:	0330	00	00	00	00	00	00	00	00
DS:	0338	00	00	00	00	00	00	00	00
DS:	0340	00	00	00	00	00	00	00	00
DS:	0348	00	00	00	00	00	00	00	00

2		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:	0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01	= f.ü- .-..<.K.
DS:	0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF	¶.U.¶.§.
DS:	0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11	N.ñ.
DS:	0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00ç.
DS:	0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri



Abarbeitung von Maschinencode

Programm und Daten liegen gemeinsam im Speicher.

Annahme: Programm liegt ab **Speicherzelle 0100**

Das Programm ist nichts anderes als eine Folge von Bytes

The screenshot shows a debugger window with the following assembly code and memory dump:

Address	Disassembly
00FF	NOP
0100	MOV AL, [0300]
0103	MUL W/[0301]
0107	DEC AX
0108	MOV [0302], AX
010B	NOP
010C	NOP
010D	ADD [BX+SI], AL
010F	ADD [BX+SI], AL

Address	0	1	2	3	4	5	6	7
DS:0300	03	02	00	00	00	00	00	00
DS:0308	00	00	00	00	00	00	00	00
DS:0310	00	00	00	00	00	00	00	00
DS:0318	00	00	00	00	00	00	00	00
DS:0320	00	00	00	00	00	00	00	00
DS:0328	00	00	00	00	00	00	00	00
DS:0330	00	00	00	00	00	00	00	00
DS:0338	00	00	00	00	00	00	00	00
DS:0340	00	00	00	00	00	00	00	00
DS:0348	00	00	00	00	00	00	00	00

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Legend: = f.ü- .-..<.K. 9.U.9.5. N.ñ.z.

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri



Abarbeitung von Maschinencode

Ein spezielles Register ist der Programmzähler IP:
Hier steht, welcher Befehl abzuarbeiten ist,
im Beispiel der in Speicherzelle 0100

The screenshot shows a debugger window for the file `D:\SOFTWA~3\Kap02\AFD\AFD.EXE`. The CPU registers are displayed as follows:

AX	0000	SI	0000	CS	1FA8	IP	0100	Stack	+0 0000	Flags	3204
BX	0000	DI	0000	DS	1FA8				+2 20CD		
CX	0000	BP	0000	ES	1FA8	HS	1FA8		+4 9FFF	OF	DF
DX	0000	SP	FFFE	SS	1FA8	FS	1FA8		+6 9A00	IF	SF
										ZF	AF
										PF	CF

The command line shows the current instruction at address 0100:

```

00FF 90          NOP
0100 A00003      MOV     AL, [0300]
0103 F72E0103   IMUL   W/[0301]
0107 48        DEC     AX
0108 A30203      MOV     [0302], AX
010B 90          NOP
010C 90          NOP
010D 0000      ADD     [BX+SI], AL
010F 0000      ADD     [BX+SI], AL
  
```

The memory dump shows the contents of memory addresses 0000 to 0040:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The bottom status bar shows navigation options: 1 Step, 2 ProcStep, 3 Retrieve, 4 Help ON, 5 BRK Menu, 6, 7 up, 8 dn, 9 le, 10 ri.



Abarbeitung von Maschinencode

Der Inhalt der Speicherzellen ab Adresse 0100 wird ins Steuerwerk geholt

Dort wird der gelesene Befehl "A00003" decodiert.

The screenshot shows a debugger window with the following components:

- Registers:** AX 0000, SI 0000, CS 1FA8, IP 0100, Stack +0 0000, Flags 3204; BX 0000, DI 0000, DS 1FA8, +2 20CD; CX 0000, BP 0000, ES 1FA8, HS 1FA8, +4 9FFF; DX 0000, SP FFFE, SS 1FA8, FS 1FA8, +6 9A00. Status flags: OF 0, DF 0, IF 1, SF 0, ZF 0, AF 0, PF 1, CF 0.
- Command Line:** CMD > A00003 (highlighted in orange)
- Assembly List:**
 - 00FF 90 NOP
 - 0100 A00003 MOV AL, [0300] (highlighted in orange)
 - 0103 F72E0103 IMUL W/[0301]
 - 0107 48 DEC AX
 - 0108 A30203 MOV [0302], AX
 - 010B 90 NOP
 - 010C 90 NOP
 - 010D 0000 ADD [BX+SI], AL
 - 010F 0000 ADD [BX+SI], AL
- Memory Dump:**

Address	0	1	2	3	4	5	6	7
DS:0300	03	02	00	00	00	00	00	00
DS:0308	00	00	00	00	00	00	00	00
DS:0310	00	00	00	00	00	00	00	00
DS:0318	00	00	00	00	00	00	00	00
DS:0320	00	00	00	00	00	00	00	00
DS:0328	00	00	00	00	00	00	00	00
DS:0330	00	00	00	00	00	00	00	00
DS:0338	00	00	00	00	00	00	00	00
DS:0340	00	00	00	00	00	00	00	00
DS:0348	00	00	00	00	00	00	00	00
- Hex Dump:**

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
- Navigation:** 1 Step, 2 ProcStep, 3 Retrieve, 4 Help ON, 5 BRK Menu, 6, 7 up, 8 dn, 9 le, 10 ri



Abarbeitung von Maschinencode

"A00003" heißt:

Lade den Inhalt von Speicherzelle 0300 ins Register AX

The screenshot shows a debugger window titled "D:\SOFTWA~3\Kap02\AFD\AFD.EXE". The top section displays register values and stack pointers. The main area shows assembly instructions being executed, with the instruction "MOV AL, [0300]" highlighted. A memory dump on the right shows the contents of memory addresses starting from DS:0300. The bottom status bar shows various debugger controls.

Register	Value	Register	Value	Register	Value	Register	Value	Stack	Flags
AX	0000	SI	0000	CS	1FA8	IP	0100	+0	0000
BX	0000	DI	0000	DS	1FA8			+2	20CD
CX	0000	BP	0000	ES	1FA8	HS	1FA8	+4	9FFF
DX	0000	SP	FFFE	SS	1FA8	FS	1FA8	+6	9A00

Address	Instruction	Comment
00FF	90	NOP
0100	A00003	MOV AL, [0300]
0103	F72E0103	IMUL W/[0301]
0107	48	DEC AX
0108	A30203	MOV [0302], AX
010B	90	NOP
010C	90	NOP
010D	0000	ADD [BX+SI], AL
010F	0000	ADD [BX+SI], AL

Address	0	1	2	3	4	5	6	7
DS:0300	03	02	00	00	00	00	00	00
DS:0308	00	00	00	00	00	00	00	00
DS:0310	00	00	00	00	00	00	00	00
DS:0318	00	00	00	00	00	00	00	00
DS:0320	00	00	00	00	00	00	00	00
DS:0328	00	00	00	00	00	00	00	00
DS:0330	00	00	00	00	00	00	00	00
DS:0338	00	00	00	00	00	00	00	00
DS:0340	00	00	00	00	00	00	00	00
DS:0348	00	00	00	00	00	00	00	00

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri



Abarbeitung von Maschinencode

Der Befehl wurde ausgeführt.

Im **Register AX** steht jetzt dasselbe wie auf **Speicherzelle 300.**

Der **Befehlszähler** enthält die Adresse des nächsten Befehls.

The screenshot shows a debugger window for the file `D:\SOFTWA~3\Kap02\AFD\AFD.EXE`. The CPU registers are displayed as follows:

AX	0003	SI	0000	CS	1FA8	IP	0103	Stack	+0 0000	Flags	3204
BX	0000	DI	0000	DS	1FA8				+2 20CD		
CX	0000	BP	0000	ES	1FA8	HS	1FA8		+4 9FFF	OF	DF
DX	0000	SP	FFFE	SS	1FA8	FS	1FA8		+6 9A00	IF	SF
										ZF	AF
										PF	CF

The command list shows the following instructions:

```

0100 A00003      MOV     AL,[0300]
0103 F72E0103    IMUL   W/[0301]
0107 48         DEC     AX
0108 A30203      MOV     [0302],AX
010B 90         NOP
010C 90         NOP
010D 0000      ADD     [BX+SI],AL
010F 0000      ADD     [BX+SI],AL
0111 0000      ADD     [BX+SI],AL
    
```

The memory dump shows the following data:

DS:0000	CD 20 FF 9F 00 9A F0 FE	1D F0 1B 05 28 0C 4B 01	= f.ü-■	.-..<.K.
DS:0010	F4 03 56 01 F4 03 F5 06	01 01 01 00 02 FF FF FF	¶.U.¶.§.
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF 4E 1F A4 11		N.ñ.
DS:0030	05 07 14 00 18 00 A8 1F	FF FF FF FF 00 00 00 00¿.
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

Navigation keys at the bottom: 1 Step, 2 ProcStep, 3 Retrieve, 4 Help ON, 5 BRK Menu, 6, 7 up, 8 dn, 9 le, 10 ri.



Abarbeitung von Maschinencode

Der nächste Befehl **"F72E0103"** wird also ab Speicherzelle 103 gelesen. Das Steuerwerk decodiert den Befehl:

"Multipliziere den Inhalt des Registers AX mit dem Inhalt der Speicherzelle mit Adresse 0301. Lege das Ergebnis in das Register AX.

The screenshot shows a debugger window titled "D:\SOFTWA~3\Kap02\AFD\AFD.EXE". The register window at the top shows: AX 0003, SI 0000, CS 1FA8, IP 0103, Stack +0 0000, Flags 3204; BX 0000, DI 0000, DS 1FA8, +2 20CD; CX 0000, BP 0000, ES 1FA8, HS 1FA8, +4 9FFF, OF 0, DF 0, IF 1, SF 0, ZF 0, AF 0, PF 1, CF 0; DX 0000, SP FFFE, SS 1FA8, FS 1FA8, +6 9A00.

The command window shows the following assembly code:

```

0100 A00003      MOV     AL,[0300]
0103 F72E0103     IMUL   W/[0301]
0107 48          DEC     AX
0108 A30203      MOV     [0302],AX
010B 90          NOP
010C 90          NOP
010D 0000      ADD     [BX+SI],AL
010F 0000      ADD     [BX+SI],AL
0111 0000      ADD     [BX+SI],AL
  
```

The memory dump window shows the following data:

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The status bar at the bottom shows: 1 Step, 2 ProcStep, 3 Retrieve, 4 Help ON, 5 BRK Menu, 6, 7 up, 8 dn, 9 le, 10 ri.



Abarbeitung von Maschinencode

Der nächste Befehl "48" wird vom Speicher ab Adresse 107 gelesen. Das Steuerwerk decodiert den Befehl:
Verringere den Inhalt des Registers AX um 1

cmd D:\SOFTWA~3\Kap02\AFD\AFD.EXE

AX 0006	SI 0000	CS 1FA8	IP 0107	Stack +0 0000	Flags 3204
BX 0000	DI 0000	DS 1FA8		+2 20CD	
CX 0000	BP 0000	ES 1FA8	HS 1FA8	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 1FA8	FS 1FA8	+6 9A00	0 0 1 0 0 0 1 0

CMD >				i	0	1	2	3	4	5	6	7
0103	F72FA103	IMUL	W/[0301]	DS:0300	03	02	00	00	00	00	00	00
0107	48	DEC	AX	DS:0308	00	00	00	00	00	00	00	00
0108	A30203	MOV	[0302],AX	DS:0310	00	00	00	00	00	00	00	00
010B	90	NOP		DS:0318	00	00	00	00	00	00	00	00
010C	90	NOP		DS:0320	00	00	00	00	00	00	00	00
010D	0000	ADD	[BX+SI],AL	DS:0328	00	00	00	00	00	00	00	00
010F	0000	ADD	[BX+SI],AL	DS:0330	00	00	00	00	00	00	00	00
0111	0000	ADD	[BX+SI],AL	DS:0338	00	00	00	00	00	00	00	00
0113	0000	ADD	[BX+SI],AL	DS:0340	00	00	00	00	00	00	00	00
				DS:0348	00	00	00	00	00	00	00	00

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01	= f.ü-■ .-..<.K.
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF	¶.U.¶.§.
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11	N.ñ.
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00¿.
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri



Abarbeitung von Maschinencode

Der Befehl wurde ausgeführt.

Der Wert im Register **AX** wurde von 6 auf 5 geändert.

Der **Befehlszähler** zeigt nun auf den nächsten Befehl.

The screenshot shows a debugger window for the file `D:\SOFTWARE~3\Kap02\AFD\AFD.EXE`. The CPU registers are displayed as follows:

AX	0005	SI	0000	CS	1FA8	IP	0108	Stack	+0 0000	Flags	3204
BX	0000	DI	0000	DS	1FA8				+2 20CD		
CX	0000	BP	0000	ES	1FA8	HS	1FA8		+4 9FFF	OF	DF
DX	0000	SP	FFFE	SS	1FA8	FS	1FA8		+6 9A00	IF	SF
										ZF	AF
										PF	CF

The command list shows the following instructions:

```

0107 48          DEC     AX
0108 A30203      MOV     [0302],AX
010B 90          NOP
010C 90          NOP
010D 0000      ADD     [BX+SI],AL
010F 0000      ADD     [BX+SI],AL
0111 0000      ADD     [BX+SI],AL
0113 0000      ADD     [BX+SI],AL
0115 0000      ADD     [BX+SI],AL
  
```

The memory dump shows the following data:

DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The status bar at the bottom shows: 1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri



Abarbeitung von Maschinencode

Der nächste Befehl "A30203" wird vom Speicher ab Adresse 108 gelesen. Das Steuerwerk decodiert den Befehl: Speichere den Inhalt des Registers AX in Speicherzelle 302.

The screenshot shows a debugger window titled "D:\SOFTWA~3\Kap02\AFD\AFD.EXE". The register window at the top shows: AX 0005, SI 0000, CS 1FA8, IP 0108, Stack +0 0000, Flags 3204; BX 0000, DI 0000, DS 1FA8; CX 0000, BP 0000, ES 1FA8, HS 1FA8; DX 0000, SP FFFE, SS 1FA8, FS 1FA8. The command window shows the instruction at address 0108: MOV [0302], AX. The memory dump below shows the contents of memory addresses DS:0000 to DS:0040, with a hex-to-ASCII conversion on the right.

Address	Hex	ASCII
DS:0000	CD 20 FF 9F 00 9A F0 FE 1D F0 1B 05 28 0C 4B 01	= f.ü-■ .-..<.K.
DS:0010	F4 03 56 01 F4 03 F5 06 01 01 01 00 02 FF FF FF	¶.U.¶.§.
DS:0020	FF FF FF FF FF FF FF FF FF FF FF FF 4E 1F A4 11 N.ñ.
DS:0030	05 07 14 00 18 00 A8 1F FF FF FF FF 00 00 00 00¿.
DS:0040	05 00 00 00 00 00 00 00 00 00 00 00 00 00 00



Abarbeitung von Maschinencode

Fertig:

Die Speicherzelle mit Adresse 302 enthält jetzt das Ergebnis.

The screenshot shows a DOS debugger window for the file `D:\SOFTWA~3\Kap02\AFD\AFD.EXE`. The register window at the top shows the following values:

AX	0005	SI	0000	CS	1FA8	IP	010B	Stack	+0 0000	Flags	3204
BX	0000	DI	0000	DS	1FA8				+2 20CD		
CX	0000	BP	0000	ES	1FA8	HS	1FA8		+4 91FF	OF	DF
DX	0000	SP	FFFE	SS	1FA8	FS	1FA8		+6 9A00	IF	SF
										ZF	AF
										PF	CF

The command window shows the following assembly code:

```

CMD >
0108 A30203      MOV     [0302],AX
010B 90          NOP
010C 90          NOP
010D 0000      ADD     [BX+SI],AL
010F 0000      ADD     [BX+SI],AL
0111 0000      ADD     [BX+SI],AL
0113 0000      ADD     [BX+SI],AL
0115 0000      ADD     [BX+SI],AL
0117 0000      ADD     [BX+SI],AL
  
```

The memory dump shows the following values:

i	0	1	2	3	4	5	6	7
DS:0300	03	02	05	00	00	00	00	00
DS:0308	00	00	00	00	00	00	00	00
DS:0310	00	00	00	00	00	00	00	00
DS:0318	00	00	00	00	00	00	00	00
DS:0320	00	00	00	00	00	00	00	00
DS:0328	00	00	00	00	00	00	00	00
DS:0330	00	00	00	00	00	00	00	00
DS:0338	00	00	00	00	00	00	00	00
DS:0340	00	00	00	00	00	00	00	00
DS:0348	00	00	00	00	00	00	00	00

The register window at the bottom shows the following values:

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	9A	F0	FE	1D	F0	1B	05	28	0C	4B	01
DS:0010	F4	03	56	01	F4	03	F5	06	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4E	1F	A4	11
DS:0030	05	07	14	00	18	00	A8	1F	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

The status bar at the bottom shows the following values:

1	Step	2	ProcStep	3	Retrieve	4	Help ON	5	BRK Menu	6		7	up	8	dn	9	le	10	ri
---	------	---	----------	---	----------	---	---------	---	----------	---	--	---	----	---	----	---	----	----	----



Assembler

- Im Beispiel war unser Programm eine Folge von Hexadezimalzahlen:
- A0 00 03 F7 2E 01 03 48 A3 02 03
- Für Menschen leichter zu merken: Kurzbezeichnungen (Mnemonics) für die verschiedenen Befehle.
- Ein Assembler übersetzt die Mnemonics in ein Programm.

- Die beiden letzten Zeilen unseres Beispielprogramms sehen dann so aus:
- DEC AX (Verringere den Wert im Register AX um 1)
MOV [0302],AX (Speichere den im Register AX enthaltenen Wert in die Speicherzelle mit der Adresse 0302)



Assembler

Wann benutzt man Assemblersprache?

Wenn das Programm sehr schnell sein muss

Wenn das Programm sehr wenig Platz in Anspruch nehmen darf (z.B. bei eingebetteten Systemen)

Wenn direkt auf die Hardware zugegriffen werden muss

Nachteile

Schwer verständlich, schwer zu warten

Läuft nur mit einem bestimmten Hauptprozessor.



Höhere Programmiersprachen

- Problem: Ein Computerprozessor (Central Processing Unit, CPU) versteht nur sehr einfache Anweisungen wie "erhöhe den Inhalt des Speicherplatzes x um 1".
- Die meisten Menschen möchten nicht mit diesen schwer handhabbaren Anweisungen programmieren.
- Höhere Programmiersprachen formulieren einen Algorithmus in einer für den Computer verständlichen, aber auch für den Menschen noch gut lesbaren Form.
- Geben Anweisungen, was der Computer in welcher Reihenfolge zu tun hat.



Compiler / Interpreter

- Das Programm wird in einer höheren Programmiersprache geschrieben.
- Es erfolgt eine Übersetzung dieses Programms in das von der Maschine ausführbare Maschinenprogramm.
- Unterstützung verschiedener Plattformen (Kombination "Typ des Hauptprozessors" + "Betriebssystem"):
Ein und das selbe Programm kann zu einem Maschinenprogramm für verschiedene Plattformen übersetzt werden.



Compiler / Interpreter

- **Compiler:**
Vor der Programmausführung übersetzt (compiliert) ein besonderes Programm (der Compiler) den in der Programmiersprache geschriebenen Text in maschinell ausführbaren Code.
- **Interpreter:**
Zum Zeitpunkt der Programmausführung übersetzt (interpretiert) ein besonderes Programm (der Interpreter) den in der Programmiersprache geschriebenen Text in maschinell ausführbaren Code.

Interpretierte Programme laufen in der Regel langsamer ab, da die Übersetzung zur Laufzeit Zeit kostet.



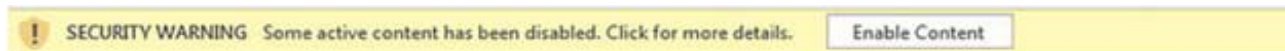
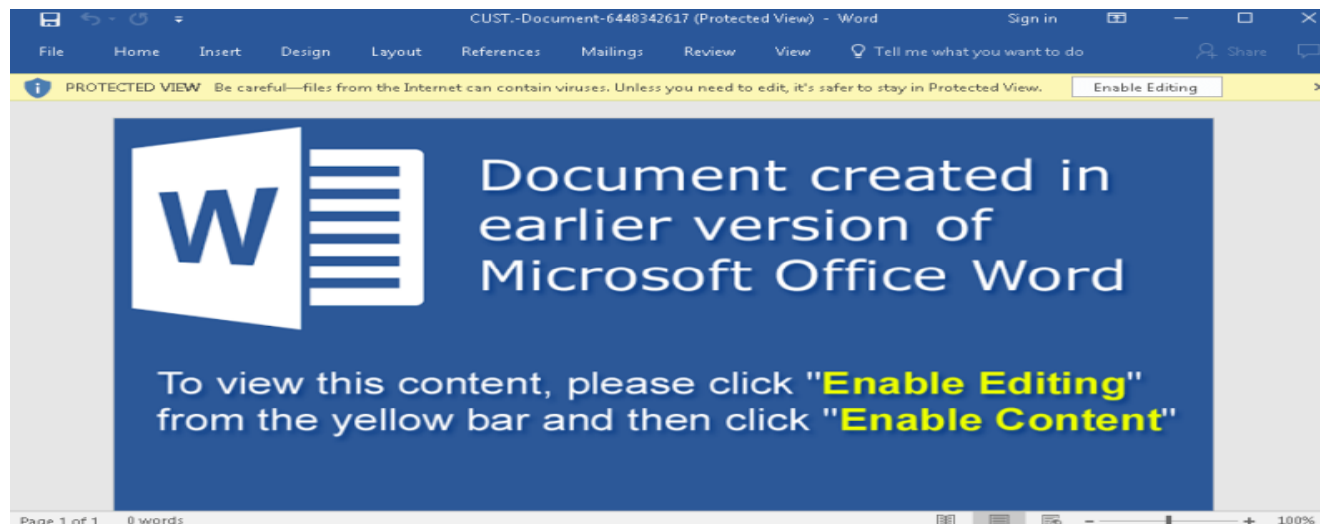
Visual Basic for Applications

- Visual Basic for Applications (VBA) – zum Erstellen von Erweiterungen für MS Office-Programme
- VBA – Programmiersprache, an der wir die prinzipiellen Konzepte der Programmierung lernen können.
- Visual Basic – zum Erstellen eigenständiger Anwendungen (Compiler)



Office-Makros

- Befehle lassen sich aufzeichnen und später in gleicher Reihenfolge wieder abspielen.
- Aufzeichnung beginnen: Entwicklertools – Makro aufzeichnen
- Aufzeichnung beenden: Entwicklertools – Aufzeichnung anhalten
- Aufgezeichnete Befehle ausführen: Entwicklertools - Makros



Accept the license agreement

You can use Microsoft Word until Friday, September 20, 2019.
After that date, most features of Microsoft Word will be disabled.

To accept and start Word click Enable Editing and click Enable Content



Vom Office-Dokument zum Programm

- ALT + F11 - VBA-Entwicklungsumgebung
- In der Entwicklungsumgebung:
STRG + G - Direktfenster
- Projekt = Office-Dokument mitsamt Makros und Formularen für Benutzereingaben.
- Makros werden in Modulen zusammengefasst; ein Modul kann mehrere Makros beinhalten.



Coderahmen

Option Explicit

Sub Programmname()

' Alles hinter einem Hochkomma ist Kommentar

... Anweisungen für den Computer...

End Sub

Jede Anweisung in eine neue Zeile schreiben!

STRG+Leertaste schlägt Vervollständigungen unvollständig
eingegebener Befehlsörter vor.

Befehl markieren + F1 → Hilfe zum Befehl



Die einfachsten Befehle - Ausgabe

- `MsgBox("Text")`
Ausgabe in einem eigenen Fenster



- `Debug.Print("Text")`
Ausgabe im Direktbereich



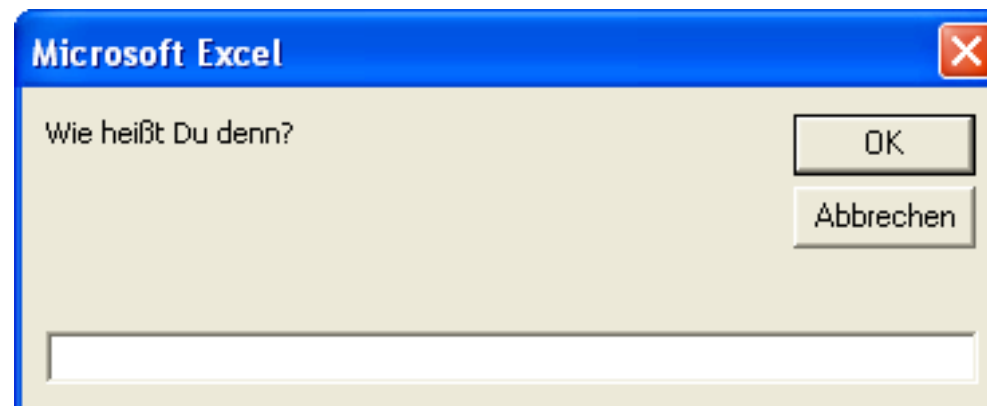


Die einfachsten Befehle - Eingabe

```
name = InputBox("Wie heißt Du denn?")
```

"=" hier gelesen als "wird zu"

name ist eine Variable – unter diesem Namen wird die Eingabe im Computer gespeichert





Name	Nummer (Adresse)	Inhalt
	300	
	301	
name	302	"Anna"
	303	
	304	



Eingabe und Ausgabe zusammen

```
Sub eingabe_und_ausgabe()
```

```
    ' Eingabe:
```

```
    x = InputBox("Wie heißt Du denn?")
```

```
    ' Ausgabe:
```

```
    MsgBox (x)
```

```
End Sub
```



Namen für Variable

- Beginnen mit Buchstaben
- Enthalten nur Buchstaben, Unterstrich (_) und Ziffern

- gültige Namen:

x3

zahl_der_eingeschriebenen_studenten

- ungültige Namen:

3x

zahl der eingeschriebenen studenten

steuer-satz

Wichtig: Namen sollen den Inhalt der Variablen erklären!



Informationen zu Urhebern und Lizenzen der verwendeten Bilder finden Sie unter den angegebenen Links:

Porträt John von Neumann

Urheber: Los Alamos National Laboratory

<https://commons.wikimedia.org/wiki/File:JohnvonNeumann-LosAlamos.gif>

Rechenmaschine Collosus

Britisches Nationalarchiv <https://commons.wikimedia.org/wiki/File:Colossus.jpg>

Spiel "Rush Hour"

Urheber: Welt-der-Form,

https://commons.wikimedia.org/wiki/File:Rush_Hour_sliding_block_puzzle.jpg

Bilder von Rechnern verschiedener Generationen:

Urheber: Erfurth

https://commons.wikimedia.org/wiki/File:Modellrechenautomat_TH_Dresden_1958_TSD.jpg

Urheber: Ellinor Algin

https://commons.wikimedia.org/wiki/File:Facit_DTC_personal_computer.jpg

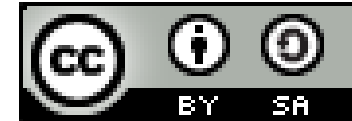
Urheber: Janto Dreijer

[https://commons.wikimedia.org/wiki/File:HP_Tablet_PC_running_Windows_XP_\(Tablet_PC_edition\)_\(2006\).jpg](https://commons.wikimedia.org/wiki/File:HP_Tablet_PC_running_Windows_XP_(Tablet_PC_edition)_(2006).jpg)



Dieses Werk bzw. Inhalt steht unter einer "Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz".

Sie dürfen:



- das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen
- Bearbeitungen des Werkes anfertigen

zu den folgenden Bedingungen:

- Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.
- Weitergabe unter gleichen Bedingungen — Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.
- Weiteres dazu: <http://creativecommons.org/licenses/by-sa/3.0/de/>