

Inhaltsverzeichnis

I Grundlagen	1
1 Entwicklung von Software allgemein	1
1.1 Software = Daten + Programme	1
1.1.1 Begriffsbildungen für Daten	5
1.1.2 Begriffsbildungen für Programme	7
1.1.2.1 Anwenderprogramme und Systemprogramme	7
1.1.2.2 Vier Standard-Programmpakete	9
1.2 Datentypen und Datenstrukturen	13
1.2.1 Einfache Datentypen als „Moleküle“	14
1.2.2 Datenstrukturen als strukturierte Datentypen	15
1.2.3 Statische und dynamische Datentypen	16
1.2.4 Vordefinierte und benutzerdefinierte Datentypen	17
1.2.5 Datentypen bei den verschiedenen Programmiersprachen	18
1.3 Programmstrukturen	19
1.3.1 Folgestrukturen	20
1.3.2 Auswahlstrukturen	22
1.3.2.1 Einseitige Auswahl	22
1.3.2.2 Einseitige Auswahl	23
1.3.2.3 Mehrseitige Auswahl	23
1.3.3 Wiederholungsstrukturen	24
1.3.4 Unterprogrammstrukturen	26
1.3.5 Mehrere Strukturen in einem Programm	27
1.4 Daten- und Programmstrukturen als Software-Bausteine	29
1.4.1 Modell des Hauptspeichers RAM als Regalschrank	30
1.4.2 Daten als Variablen und Konstanten vereinbaren	32
1.4.3 Programm mit Vereinbarungsteil und Anweisungsteil	33
1.5 Datei als umfangreiche Datenstruktur	35
1.5.1 Zugriffsart, Speicherungsform und Verarbeitungsweise	37
1.5.1.1 Zwei Zugriffsarten	38
1.5.1.2 Vier Speicherungsformen	38
1.5.1.3 Zwei Verarbeitungsweisen	42
1.5.2 Vier Verarbeitungsformen von Dateien	42
1.5.3 Grundlegende Abläufe auf Dateien	43
1.5.4 Datei öffnen, verarbeiten und schließen	45
1.5.5 Eine oder mehrere Dateien verarbeiten	46
1.5.6 Datenbank	47
1.5.6.1 Datenbank-Management-System	47
1.5.6.2 Strukturiertes und unstrukturiertes Datenbank-System	48

1.6	Programmentwicklung in Schritten	51
1.6.1	Aufgabenbeschreibung mit Problemstellung und Strukturbaum	53
1.6.2	Problemanalyse	53
1.6.3	Entwicklung und Darstellung des Algorithmus	54
1.6.3.1	Datenflußplan	54
1.6.3.2	Programmablaufplan (PAP)	56
1.6.3.3	Struktogramm	57
1.6.3.4	EntwurfsSprache bzw. Pseudocode	59
1.6.4	Programmierung im engeren Sinne	60
1.6.5	Programmiertechniken und Entwurfsprinzipien	61
1.6.6	Kleine Mauskunde	63
II	Turbo C-System	67
2	Bedienung und Definitionen des Turbo C-Systems	67
2.1	Turbo C als integrierte Entwicklungsumgebung	69
2.1.1	Zwei Nutzungsformen	70
2.1.2	Turbo C installieren	70
2.1.3	Konfigurationsdatei abspeichern	71
2.1.3.1	Schritt 1: Turbo C starten	72
2.1.3.2	Schritt 2: Suchwege für die Programmierumgebung einstellen	73
2.1.3.3	Schritt 3: Konfigurationsdatei TCCONFIG.TC speichern	74
2.1.3.4	Schritt 4: Turbo C beenden	74
2.2	Das erste Programm	75
2.2.1	Schritt 1: Programmtext editieren	76
2.2.2	Schritt 2: Quelltext auf Diskette sichern	77
2.2.3	Schritt 3: Programm ausführen lassen	78
2.2.4	Schritt 4: Programm als C-, OBJ- und EXE-Datei anzeigen	78
2.2.5	Schritt 5: Programm erweitern	79
2.3	Befehle der Entwicklungsumgebung	81
2.3.1	Menübefehl File	82
2.3.2	Befehl Edit	83
2.3.3	Befehl Run	84
2.3.4	Menübefehl Compile	84
2.3.5	Menübefehl Project	85
2.3.6	Menübefehl Options	85
2.3.7	Menübefehl Debug	86
2.3.8	Heiße Tasten	86
2.4	Sprachreferenz	89
2.4.1	Bezeichner	90
2.4.2	Reservierte Wörter	90

2.4.3	Datentypen	91
2.4.3.1	Integer-Konstanten	91
2.4.3.2	Gleitkommakonstanten	92
2.4.3.3	Zeichenkonstanten	92
2.4.3.4	String-Konstanten	93
2.4.4	Operatoren	94
2.4.5	Konvertierung von Datentypen	95
2.4.5.1	Konvertierung von Zeichen, Integer und Zeiger	95
2.4.5.2	Konvertierung bei Gleitkomma- und Integer-Zahlen	96
2.4.6	Grundlegende Anweisungen und Funktionen	97
2.4.7	Bibliotheksfunktionen	104

III	Programmierkurs mit Turbo C – Grundkurs	113
------------	--	------------

3	Einfache Datentypen	115
3.1	Zahlen	116
3.2	Ausdrücke mit Variablen und Konstanten	117
3.3	Funktion printf() zur Ausgabe	118
3.3.1	Ausgabe von Text	118
3.3.1.1	printf() mit Steuerzeichen \n und \	118
3.3.1.2	Bedingtes Compilieren mit Compilerkommando #if	120
3.3.2	Ausgabe von Zahlen und Variablen	121
3.3.2.1	Formatierte Ausgabe	121
3.3.2.2	Format-Elemente	122
3.3.3	Kommentar	124
3.4	Initialisierung, sizeof- und cast-Operator	125
3.5	Zeiger bzw. Pointer	127
3.5.1	Adreßoperator &	127
3.5.2	Zeigervariablen zur Adreßspeicherung	128
3.5.3	Zeiger auf einen Zeiger	129
3.5.4	Zeiger austauschen	130
3.5.5	Speicherplatz reservieren mit malloc() und calloc()	131
3.6	Gleitpunktzahlen	132
3.7	Zusammengesetzte Zuweisungsoperatoren	133
3.8	Ganze Zahlen	134
3.8.1	Typanpassung	134
3.8.2	Bereichsüberschreitung	135
3.9	Inkrementierung	135
3.10	Ganzzahlige Aufzählungstypen	136
3.11	Vergleiche	137
3.11.1	Vergleichsoperatoren	137
3.11.2	Bitmanipulationen	138

3.12 Zahleneingabe mit der Funktion <code>scanf()</code>	139
3.13 Symbolische Konstanten mit <code>#define</code>	141
4 Programmstrukturen	143
4.1 Auswahlstrukturen mit <code>if ... else</code>	144
4.4.1 Einseitige und zweiseitige Auswahlstruktur	144
4.4.2 Schachtelung von <code>if</code> -Anweisungen	146
4.4.3 Kettenbildung	146
4.2 Bedingungsausdrücke mit dem Operator <code>?:</code>	148
4.3 Logische Operatoren	149
4.4 Auswahlstrukturen mit <code>switch ... case</code>	150
4.5 Wiederholungsstrukturen mit <code>while</code>	152
4.5.1 Abweisende Schleife	152
4.5.2 Inkrementieren in der Schleifenbedingung	153
4.6 Wiederholungsstrukturen mit <code>for</code>	156
4.6.1 Zählerschleife	156
4.6.2 Komma-Operator bei <code>for</code>	157
4.7 Wiederholungsstrukturen mit <code>do ... while</code>	159
4.7.1 Nicht-abweisende Schleife	159
4.7.2 <code>while</code> bzw. <code>for</code> durch <code>do ... while</code> ersetzen	160
4.8 Unstrukturierte Kontrollanweisungen	161
4.8.1 Wiederholungsstrukturen mit <code>goto</code>	161
4.8.2 Verzweigungsanweisungen <code>break</code> und <code>continue</code>	162
4.9 Funktionen	164
4.9.1 Vordefinierte und benutzerdefinierte Funktionen	164
4.9.2 Funktionen ohne bzw. mit Eingabe von Argumenten	164
4.9.3 Funktionen ohne bzw. mit Ausgabe eines Funktionswertes	165
4.9.4 Funktionsdefinition im modernen und im klassischen Format	167
4.9.5 Funktionen mit <code>void</code> -Typ deklarieren	171
4.9.6 Parameter an die Funktion <code>main()</code> übergeben	171
4.10 Variablen und Speicherklassen	173
4.10.1 Globale und lokale Variablen	173
4.10.2 Lokalisierung mit Speicherklasse <code>static</code>	175
4.10.3 Zeiger als Funktionsparameter	176
4.11 Zeiger- bzw. Pointerarithmetik	178
4.11.1 Zeiger inkrementieren	178
4.11.2 Zeigertyp <code>far</code>	179
4.12 Modularer Aufbau von C-Programmen	181
4.12.1 Ein Programm mit <code>#include</code> in den Quelltext einfügen	181
4.12.2 Mehrere Programme über eine Projektdatei verbinden	181
4.13 Makros	183
4.13.1 Makrodefinition	183
4.13.2 Makro mit Escape-Sequenz	185

4.14	Rekursiver Aufruf von Funktionen	188
4.14.1	Rekursion mit Stacküberlauf	188
4.14.2	Rekursion mit Endebedingung	190
4.13.3	Indirekte Rekursion	190
4.14.4	Rekursiver Aufruf der Funktion main()	191
4.15	Zeiger auf Funktionen	192
5	Strukturierte Datentypen	195
5.1	Vektoren als eindimensionale Arrays	196
5.1.1	Lesen und Schreiben über einen Index	196
5.1.2	Vektor bei der Deklaration initialisieren	198
5.1.3	Funktionen malloc() und calloc()	198
5.1.4	Array über geschachtelte Schleifen sortieren	199
5.2	Matrizen als zweidimensionale Arrays	202
5.2.1	Zugriff über Zeilenindex und Spaltenindex	202
5.2.2	Speicherplatz durch Initialisierung bereitstellen	202
5.2.3	Benutzerdefinierter Datentyp für Matrizen	203
5.3	Dreidimensionale Arrays	204
5.4	Stringverarbeitung	205
5.4.1	Strings als Vektoren vom char-Typ	205
5.4.2	Einen String überschreiben	206
5.4.3	Strings übertragen	207
5.5	Eingabe von Strings über die Tastatur	208
5.5.1	Eingabe mit scanf() und gets()	208
5.5.2	Eingabe mit cgets() und fgets()	210
5.5.3	Eingabe von Strings in eine Matrix	211
5.5.4	Eingabestring sortieren	213
5.6	Strings und Zahlen	214
5.6.1	Zahleneingabe über gets()	214
5.6.2	Umwandlung von int-Zahl in String	215
5.6.3	Umwandlung von long-Zahl in String	216
5.6.4	Umwandlung von double-Zahl in String	217
5.6.5	Umwandlung von String in Zahl	218
5.7	Zeichenweise Behandlung von Strings	220
5.7.1	Funktion zur Stringeingabe	220
5.7.2	Eingabe eines Kennworts ohne Echo	221
5.8	Strukturen	223
5.8.1	Zugriff über den Namen mit dem Punkt-Operator	223
5.8.2	Zugriff über Zeiger mit dem Pfeil-Operator	224
5.8.3	Möglichkeiten zur strukt-Vereinbarung	226
5.8.4	Eine vordefinierte Struktur nutzen	227
5.9	Vektoren von Strukturen	229

5.10 Varianten	230
5.10.1 Variante als Komponente einer Struktur	230
5.10.2 Nutzung eines vordefinierten union-Typs	232
5.11 Bitfelder	234
6 Dateiverarbeitung	239
6.1 Streams und Files	240
6.1.1 Textstream und Binärstream	240
6.1.2 Datei öffnen und schließen	240
6.1.3 Struktur einer Textdatei anzeigen	242
6.2 Textdateien	243
6.2.1 Programm zur Ausgabe des eigenen Quelltextes	243
6.2.2 Text in eine Datei schreiben und wieder lesen	244
6.2.3 Gepufferte und ungepufferte Eingabe	245
6.3 Binärdateien	247
6.3.1 Strukturen als Datensätze	247
6.3.2 Direktzugriff auf bestimmte Datensätze	249
6.3.3 Hinzufügen von Datensätzen	251
6.4 Dateizugriff über den Handle	253
6.4.1 Handle als ungepuffertes Ein-/Ausgabesystem	253
6.4.2 Schreibender und lesender Zugriff	254
6.4.3 Datei kopieren	256
6.4.4 Anwendung mit struct und union	258
7 Lösungen zu allen Aufgaben	261
ASCII-Tabelle	290
Registerverzeichnis	291
Programmverzeichnis nach Abschnitten	293
Programmverzeichnis nach Alphabet	295
Sachwortverzeichnis	296