

Zusammenfassung des Handzettels für Programmieren in C

In der handschriftlichen Kopie werden mehr Abkürzungen verwendet. Alles **Grün** markierte dient zum lernen und wird nicht auf den Handzettel übertragen.

Version 1.1

Inhaltsverzeichnis

Operatoren (geordnet nach Priorität).....	3
Modulo.....	3
Typecast.....	3
Typen und Ihre Größen.....	4
Libraries.....	4
Kontrollstruktur Switch.....	4
Variablen.....	4
Deklaration.....	4
Definition.....	4
Initialisierung.....	4
IO.....	5
Ausgabe.....	5
Einlesen.....	5
Typdefinition der Platzhalter.....	5
Dateiverwaltung.....	5
ASCII.....	5
Arrays.....	5
Strings.....	5
Pointer.....	5
Beispiel.....	5
Funktionen.....	6
Call by Reference/Value.....	6
Beispielfunktionen.....	6
Aufrufe.....	6
Beispielprogramme.....	6

Vergleich zweier Arrays.....	6
rand.....	7
Vertauschung von Buchstaben (Buchstabensuppe).....	7

Operatoren (geordnet nach Priorität)

Operator(en) getrennt durch #	Kurzbeschreibung	Assoziativität	Beispiel
() [] -> .	Gruppierung Elementzugriff	→	(var) (var1 &&var2) obj.b
! ~ ++ -- (type) * & sizeof()	NOT Invertierung Pre/Post De/Inkrement Typecast Inhaltsoperator Adressoperator Größe in Byte	←	!var ~var ++var (int)var *var &var sizeof(array)
* / %	Mathematische Ausdrücke	→	2%2
+ -	Mathematische Ausdrücke	→	
<< >>	Shift	→	
< <= > >=	Shift gleich	→	
&	BITWEISE AND	→	
^	XOR	→	
	Bitweise OR	→	
&&	Logisches AND	→	
	Logisches OR	→	
?:	Ternärer Operator	←	Bedingung ? Ausdruck1 : Ausdruck2
= += -= /= %= *= != ^= &= <<= >>=	Kombinierte Zuweisung	←	
,		→	

Modulo

10%3=1

3%10=3

5%6=5

Typecast

int i=5;

int j=2;

float x = (float) i/(float)j;

Typen und Ihre Größen

Bezeichnung	Größe in Byte	Verwendung
true false		Zahl 0 Zahl ungleich 0
char	1	ASCII (A=65)
unsigned int	4	Vorzeichenlose Ganzzahlen
int	4	Ganzzahlen
float	4	Fließkommazahlen
double	8	Fließkommazahlen

Libraries

Library	Aufgaben	Beispielfunktionen
stdio.h	I/O	printf, scanf, fopen
stdlib.h	Standard-Funktionen	rand, atoi, malloc
time.h	Zeitfunktionen	Time
float.h	Minimum und Maximum für float	FLT_MAX_10_EXP +37

Quelle: http://www.acm.uiuc.edu/webmonkeys/book/c_guide/

Kontrollstruktur Switch

```
switch (char) {  
case 'a': /* do it */  
case 'b':  
/* do it */  
break;  
default: printf(„%s“, „error“); }
```

Variablen

Deklaration

```
int x;
```

Definition

```
x = 5;
```

Initialisierung

```
int x = 5;
```

IO

Ausgabe

Funktion: printf

Einlesen

Funktion: scanf

Zugriff auf Variable per Adresse (&)

Typdefinition der Platzhalter

%i int

%c char

%s string (Stoppt beim Element \0)

%d double und float?

\t%20d Sekunden

%lf double (long float)

Dateiverwaltung

fgets(bufferArray, 256, fp);

fputs(„string“, fp)

ASCII

A=65; a=97

Z=90; z=122

Arrays

Anfangsspeicheradresse eines Arrays speichert die Anfangsadresse des Felds, nicht veränderbar.

Adressbereich von `int feld[2]={0,1}` ist 2

Zugriff Element 0 ab erster Speicheradresse: `typ * Index (4byte*0)`

Zugriff Element 1 ab erster Speicheradresse: `typ * Index (4byte*1)`

Ein Array als Fkt.-Übergabeparameter übergibt nur die Anfangs-Speicheradresse.

Strings

`char zeichenkette[5] = {'H','a','l','l','o','\0'}; // \0: Zeichenkette zuende`

`char meinString[] = "Hallo";`

zeichenkette ist gleich meinString

Pointer

Pointer mit NULL initialisieren, da sonst Überschreibung einer Zufallsadresse

Beispiel

`int x=1, y=0;`

`int z[3]={0,1,2};`

```
int *ptr; //ptr is a pointer to int
ptr = &x; //ptr points to x
y = *ptr; //y is 1
*ptr = 0; // x is 0
ptr = &z; // ptr points to z[0]
*ptr=*(ptr+1); // ptr points to z[1]
```

Funktionen

Call by Reference/Value

Beispielfunktionen

```
void addOne(int zahl) { // Call by Value immer ohne Pointer
    zahl++;
}
void addTwo(int *zahl) { // Call by Reference ist immer mit Pointern
    *zahl=*zahl+2;
}
```

Aufrufe

```
addOne(zahl); //zahl bleibt gleich
addTwo(&zahl); //zahl wird geändert
```

Beispielprogramme

Vergleich zweier Arrays

```
#include <stdio.h>
#define MAX 3
int findeErstenUnterschied();
int main () {
    int meinArray[MAX]={0,1,2};
    int meinArray2[MAX]={0,1,6};
    findeErstenUnterschied(meinArray, meinArray2, sizeof(meinArray)/sizeof(int) );
}

/* Vergleich zweier Arrays
 * Im Namen von meinArray steckt die Anfangsadresse
 */
int findeErstenUnterschied(int meinArray[], int meinArray2[], int laenge) {
    int i=0;
    // for (; i<sizeof(meinArray); i++) { geht in einer Funktion nicht, Laenge wird nicht im Array uebergeben
    for (; i<laenge; i++) {
        if (meinArray[i] !=meinArray2[i]) return i;
    }
    return -1;
}
```

rand

```
#include <time.h>
srand(time(NULL));
randNum2 = rand()%10; // 0-9
randNum2 = rand()%10+1; // 1-10
```

Vertauschung von Buchstaben (Palindrom)

```
#include <stdio.h>

#define MAX_LEN 100

char myWord[MAX_LEN];
/**
 * change every char to equil char at the other side
 */
void charMix() {
    int i;
    int indexLen = countWordLen();
    for (i=0; i < indexLen/2; i++) {
        changeChar(i, indexLen-1-i);
    }
}

void changeChar(int p1, int p2) {
    printf("p1:%i, p2:%i\n", p1, p2);
    char tmp = myWord[p1];
    myWord[p1] = myWord[p2];
    myWord[p2] = tmp;
}

int countWordLen() {
    int i = 0;
    while(myWord[i] != '\0' && i < MAX_LEN) {
        i++;
    }
    return i;
}

void copyArrayToGlobal(char c[]) {
    int i;
    for (i=0; i<MAX_LEN; i++) {
        myWord[i] = c[i];
    }
}

int main () {
    char tmpWord[MAX_LEN] = "ABC.cba";
    copyArrayToGlobal(tmpWord);
    printf("Vor Mix: %s\n", myWord);
    charMix();
    printf("Hinter Mix: %s\n", myWord);
}
```