

Zusammenfassung C-Einführung

1 Programmaufbau:

- Headerdateien
- Globale Konstanten
- Deklaration von Unterfunktionen
- main() – Hauptfunktion
- Definition der Unterfunktionen

```
1 // Programmbeginn -> Header
2 #include <stdio.h>
3
4 //globale Konstanten definieren
5 int iPi=3.141592653;
6 //Funktionen deklarieren
7 int fctAddieren(int, int);
8
9 //Hauptprogramm
10 int main()
11 {
12     //allgemeine Variablen initialisieren
13     int iErgebnis;
14     int iA, iB;
15     //Programmblöcke, Loops, Input,...
16     scanf("%d%d", &iA, &iB);
17     iErgebnis=fctAddieren(iA, iB);
18     fctAusgabe(iErgebnis);
19     //Programmende
20     return 0;
21 }
22 //Definition der Funktionen
23 int fctAddieren(int iZahl1, int iZahl2)
24 {
25     int iSumme;
26     iSumme=iZahl1*iZahl2;
27     return iSumme;
28 }
```

2 Header:

- Es gibt 2 Arten der Headerdateien
 - o in der C-Programmbibliothek bereits integriert, z.B. math.h, stdio.h, time.h,...
➔ Aufruf durch: #include <file>
 - o selbst verfasste Headerdateien, welche im selben Ordner wie das C-Programm liegen müssen
➔ Aufruf durch: #include "file"

```
1 // Headertitel
2 #ifndef MYHEADER_H
3 #define MYHEADER_H
4
5 //Funktionen deklarieren
6 int fctRechnen(int, int);
7
8 //Funktionen definieren
9 int fctRechnen(int iX, int iY)
10 {
11     //Code
12 }
13
14 //Beenden des Headers
15 #endif
```

3 Funktionen:

- beinhalten Kombinationen von Befehlen um bestimmte Aufgaben zu erledigen
- sie dienen zur Strukturierung des Codes, da der Code in kleine Teilaufgaben bearbeitet wird
- sie reduzieren den Code, da sie mehrfach aufgerufen werden können
- main() ist die Hauptfunktion, in der alle weiteren Funktionen aufgerufen werden
- in der C-Programmbibliothek sind bereits Standardfunktionen enthalten, z.B. int abs(int n)
- Deklaration von Funktionen: Rückgabetyt Name (Eingabetypen);
- Rückgabetypen hängen vom „return value“ ab
 - o Void ➔ die Funktion gibt nicht zurück
 - o Char ➔ die Funktion gibt ein Symbol zurück
 - o Int ➔ die Funktion gibt einen Wert aus
- Rückgabe von Parametern allg. durch „return“ möglich
- Inhalte sind lokal, globale Änderung durch Pointer möglich
 - o durch Pointer wird die Speicheradresse eines Parameters weitergegeben
- Definition einer Funktion: Rückgabetyt Name (Eingabetypen mit Parameternamen)

```
1 // Programmbeginn -> Header
2 #include <stdio.h>
3
4 //Funktionen deklarieren
5 int fctAddieren(int, int);
6 void fctAusgabe(int);
```

```
1 //Definition der Funktionen
2 int fctAddieren(int iZahl1, int iZahl2)
3 {
4     int iSumme;
5     iSumme=iZahl1*iZahl2;
6     return iSumme;
7 }
8 void fctAusgabe(int iAusgabe)
9 {
10     printf("%d", iAusgabe);
11     return;
12 }
```

4. Arrays

- definiert eine bestimmte Anzahl von Elementen desselben Datentyps
- ein- und mehrdimensional
- Deklaration:
 - o eindimensional: Datentyp Name [Größe]
 - o zweidimensional: Datentyp Name [Zeile] [Spalte]
- Zugriff durch Name [Position]
- Arrays können als Parameter für Funktionen eingesetzt werden → Pointer
- Rückgabewerte von Funktionen können indirekt Arrays sein → static
- Dynamische Arraygröße → Pointer

```
1  #include <stdio.h>
2
3  int main ()
4  {
5      int iArray[ 10 ];
6      int i,j;
7
8      //Werte in das Array schreiben
9      for ( i = 0; i < 10; i++ )
10     {
11         iArray[ i ] = i + 100; // Element i = i+100
12     }
13
14     //Ausgabe des Arrays
15     for (j = 0; j < 10; j++ )
16     {
17         printf("Element[%d] = %d\n", j, iArray[j] );
18     }
19
20     return 0;
21 }
```

Ausgabe:

```
1  Element[0] = 100
2  Element[1] = 101
3  Element[2] = 102
4  Element[3] = 103
5  Element[4] = 104
6  Element[5] = 105
7  Element[6] = 106
8  Element[7] = 107
9  Element[8] = 108
10 Element[9] = 109
```

Quellen:

http://wissrech.ins.uni-bonn.de/teaching/prama1_ws05/c-einfuehrung.pdf

http://www.tutorialspoint.com/cprogramming/c_functions.htm

<http://codepad.org>