

Programmierung in C

Grundlagen

Vorstellungsrunde

- Name
- Hobby/Beruf
- Schon mal was programmiert?
- Erwartungen an den Kurs
- Lieblingstier

C-Buch online:

http://openbook.rheinwerk-verlag.de/c_von_a_bis_z/

Entwicklungsumgebung:

Visual Studio Express 2015 für Windows Desktop

Microsoft Konto

Kontakt:

stefan@kallerhoff.de

mmmhhh...

Wenn auf diese Frage die Antwort rein zufällig ausgewählt würde, wie hoch wäre dann die Wahrscheinlichkeit das diese Frage richtig beantwortet würde?

a) 25%

b) 60%

c) 50%

d) 25%

Das erste Programm

```
#include "stdafx.h"

/* ich bin ein Kommentar und werde beim Übersetzen in Maschinsprache nicht beachtet */

// wenn ich nur über eine Zeile gehe, kann ich auch so aussehen

/* Hauptfunktion -> Einsprungpunkt für C-Programme */
int main()
{
    /* Ausgabe des Textes in Anführungsstrichen */
    printf("Hello World");

    /* Ein Zeichen wird eingelesen (nur damit die Anzeige nicht sofort wieder geschlossen wird) */
    getchar();

    /* Es wird aus der Hauptfunktion main() zurückgesprungen (Programm beendet) mit der Exit-Code 0 */
    return 0;
}
```

Das erste Programm

Die wichtigsten Escape-Sequenzen:

`\n` : Zeilenvorschub (new line)

`\r` : Wagenrücklauf (carriage return)

`\t` : Tabulator

`\a` : akustisches Signal (alert)

Das erste Programm

Aufgabe:

```
*****  
*           Hallo Welt           *  
*  Ich bin ein C Programm      *  
*****
```

Was ist Speicher?

Adresse	Inhalt
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

Die wichtigsten Datentypen:

int : Integer (ganze Zahlen)

float : Fließkomma-Zahlen

char : Schriftzeichen

void : Kein Typ (nix)

Wir rechnen

```
#include "stdafx.h"

/* hier werden die Variablen im Speicher angelegt */
int zahl1;
int zahl2;
int ergebnis;

int main()
{
    /* hier werden Variablen Werte zugewiesen */
    zahl1 = 6;
    zahl2 = 3;

    /* wir geben einen Text aus*/
    printf("Wir rechnen: \n");

    /* die Rechnung mit Zuweisung in die Ergebnis-Variable */
    ergebnis = zahl1 + zahl2;

    /* Ausgabe des Ergebnises */
    printf("Ergebnis der Rechnung: %d", ergebnis);

    getchar();

    return 0;
}
```

Die wichtigsten Rechenoperatoren:

+ : Addition (zahl1 + zahl2)

- : Subtraktion (zahl1 – zahl2)

* : Multiplikation (zahl1 * zahl2)

/ : Division (zahl1 / zahl2)

Aufgabe:

$$12 / 5 = ???$$

$$5 - 7 = ???$$

Was ist ein Zeiger?

Adresse	Inhalt
0	1
1	12
2	123
3	234
4	
5	
6	
7	1
8	
9	
10	
11	
12	
13	
14	
15	



Was ist ein Zeiger?

```
#include "stdafx.h"

/* hier werden die Variablen im Speicher angelegt */
int zahl1;
int zahl2;
int ergebnis;

int main()
{
    printf("Zahl 1 eingeben: ");
    /* zahl1 im Format %i = int wird eingelesen */
    scanf_s("%d", &zahl1);

    printf("\nZahl 2 eingeben: ");
    /* zahl2 im Format %i = int wird eingelesen */
    scanf_s("%d", &zahl2);

    printf("\nWir rechnen: ");

    /* die Rechnung mit Zuweisung in die Ergebnis-Variable */
    ergebnis = zahl1 + zahl2;

    /* Ausgabe des Ergebnises */
    printf("\nErgebnis der Rechnung: %d", ergebnis);

    /* zwei mal, da das letzte Enter-Tasten drücken aus 2 Zeichen besteht*/
    getchar();
    getchar();

    return 0;
}
```

Die wichtigsten Formatierungszeichen:

`%d` : Integer (Ganzzahl)

`%f` : Fließkommazahl (Float)

`%c` : Schriftzeichen (Character)

`%s` : String (Zeichenkette)

`%p` : void (nix)

Aufgabe:

Zahl 1: 4

Zahl 2: 3

$$4 + 3 = 7$$

$$4 - 3 = 1$$

$$4 * 3 = 12$$

$$4 / 3 = 1$$

Wenn ... dann

```
#include "stdafx.h"

/* hier werden die Variablen im Speicher angelegt */
int zahl1;
int zahl2;

int main()
{
    printf("Zahl 1 eingeben: "); /* zahl1 im Format %i = int wird eingelesen */
    scanf_s("%d", &zahl1);

    printf("Zahl 2 eingeben: "); /* zahl2 im Format %i = int wird eingelesen */
    scanf_s("%d", &zahl2);

    if ( zahl1 > zahl2 ) /* wenn zahl1 größer als zahl2 dann... */
    {
        printf("Zahl1 ist groesser als Zahl2");
    }
    else /* sonst... */
    {
        printf("Zahl1 ist nicht groesser als Zahl2");
    }

    /* zwei mal, da das letzte Enter-Tasten drücken aus 2 Zeichen besteht*/
    getchar();
    getchar();

    return 0;
}
```

Wenn ... dann

```
#include "stdafx.h"

int zahl1;
int zahl2;

int main()
{
    printf("Zahl 1 eingeben: ");
    scanf_s("%d", &zahl1);

    printf("Zahl 2 eingeben: ");
    scanf_s("%d", &zahl2);

    if (zahl1 > zahl2) /* wenn zahl1 größer als zahl2 dann... */
    {
        printf("Zahl1 ist groesser als Zahl2");
    }
    else /* sonst... */
    {
        if (zahl1 < zahl2) /* wenn zahl1 kleiner als zahl2 dann... */
        {
            printf("Zahl1 ist nicht groesser als Zahl2");
        }
        else /* sonst... */
        {
            printf("Zahl1 ist genau so gross wie Zahl2");
        }
    }

    getchar();
    getchar();

    return 0;
}
```

Die wichtigsten Vergleichsoperatoren:

== : gleich (**ACHTUNG** nur = ist falsch)

!= : nicht gleich

> : größer (>= größer gleich)

< : kleiner (<= kleiner gleich)

So lange ... bis

```
#include "stdafx.h"

int anzahl;

int main()
{
    printf("Wie oft: ");
    scanf_s("%d", &anzahl);

    while (anzahl > 0) /* so lange wie das stimmt */
    {
        anzahl = anzahl - 1;
        printf("\nNoch %d mal", anzahl);
    }

    printf("\nFertig");

    getchar();
    getchar();

    return 0;
}
```

Felder

```
int feld[10];
```

Adresse	Inhalt
0	
1	
2	
3	feld[0]
4	feld[1]
5	feld[2]
6	feld[3]
7	feld[4]
8	feld[5]
9	feld[6]
10	feld[7]
11	feld[8]
12	feld[9]
13	
14	
15	

Inkrement <-> *Dekrement*:

- `i++` : `i` wird, nachdem die Zeile ausgeführt wurde, um 1 erhöht ($i = i + 1$)
- `++i` : `i` wird, bevor die Zeile ausgeführt wird, um 1 erhöht ($i = i + 1$)
- `i--` : `i` wird, nachdem die Zeile ausgeführt wurde, um 1 verringert ($i = i - 1$)
- `--i` : `i` wird, bevor die Zeile ausgeführt wird, um 1 verringert ($i = i - 1$)

Genau so lange

```
#include "stdafx.h"

/* hier wird ein Feld mit der Länge 10 im Speicher angelegt */
int feld[10];

int main()
{
    int i;

    /* die Zahlen 0 bis 9 werden in feld abgelegt */
    for (i = 0; i < 10; i++)
    {
        feld[i] = i;
    }

    /* die Inhalte von feld werden ausgegeben */
    for (i = 0; i < 10; i++)
    {
        printf("feld[%d] = %d \n", i, feld[i]);
    }

    getchar();

    return 0;
}
```

Genau so lange

Aufgabe:

Die Zahlen von 1 bis 500 addieren!

Gauss hat das so gemacht (mit $n = 500$):

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

Stimmt das?

Viel Auswahl

```
int main()
{
    float zahl1, zahl2;
    float ergebnis = 0;
    char rechenZeichen;

    printf("Taschenrechner\n===== \n\n");

    printf("Bitte Zahl 1 eingeben: ");
    scanf_s("%f", &zahl1);
    getchar();

    printf("Bitte Zahl 2 eingeben: ");
    scanf_s("%f", &zahl2);
    getchar();

    printf("Berechnung wählen [+,-,*,/] ");
    rechenZeichen = getchar();

    switch (rechenZeichen)
    {
    case '+':
        ergebnis = zahl1 + zahl2;
        break;
    case '-':
        ergebnis = zahl1 - zahl2;
        break;
    case '*':
        ergebnis = zahl1 * zahl2;
        break;
    case '/':
        ergebnis = zahl1 / zahl2;
        break;
    default:
        printf(" Keine gültige Auswahl/n");
        break;
    }

    printf("\n %f %c %f = %f\n", zahl1, rechenZeichen, zahl2, ergebnis);

    getchar();
    getchar();

    return 0;
}
```

Funktionen, die Erste

```
#include "stdafx.h"

/* hier werden die Variablen im Speicher angelegt */
int zahl1;
int zahl2;

void addiere(void)
{
    /* zahl1 wird mit zahl2 addiert und das Ergebnis in zahl1 abgelegt */
    zahl1 = zahl1 + zahl2;
}

int main()
{
    printf("Zahl 1 eingeben: ");
    /* zahl1 im Format %i = int wird eingelesen */
    scanf_s("%d", &zahl1);

    printf("Zahl 2 eingeben: ");
    /* zahl2 im Format %i = int wird eingelesen */
    scanf_s("%d", &zahl2);

    printf("\nWir rechnen: ");

    addiere(); /* Funktion wird hier aufgerufen */

    printf("\nErgebnis der Rechnung: %d", zahl1);

    /* zwei mal, da das letzte Enter-Tasten drücken aus 2 Zeichen besteht*/
    getchar();
    getchar();

    return 0;
}
```

Speicherorganisation

Adresse	Inhalt
0	zahl2
1	zahl1
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

Die *globalen* Variablen werden auf einer Seite des Speichers angelegt. Eine geordnete Ablage ist nicht nötig (chaotisches Lager).

Dieser Teil des Speichers wird *Heap* (Haufen) genannt.

Funktionen, die Zweite

```
int zahl1, zahl2;

void rechne(void)
{
    /* lokale Variablen angelegt */
    int zwischenErgebnis1, zwischenErgebnis2;

    /* Berechnungen werden in zwischenErgebnis1 und zwischenErgebnis2 abgelegt */
    zwischenErgebnis1 = zahl1 + zahl2;
    zwischenErgebnis2 = zahl1 - zahl2;

    zahl1 = zahl1 * zwischenErgebnis1;
    zahl2 = zahl2 * zwischenErgebnis2;
}

int main()
{
    printf("Zahl 1 eingeben: ");
    scanf_s("%d", &zahl1);

    printf("Zahl 2 eingeben: ");
    scanf_s("%d", &zahl2);

    printf("\nWir rechnen: ");
    rechne(); /* Funktion wird hier aufgerufen */

    printf("\nErgebnis 1: %d", zahl1);
    printf("\nErgebnis 2: %d", zahl2);

    getchar();
    getchar();

    return 0;
}
```

Speicherorganisation

Adresse	Inhalt
0	zahl2
1	zahl1
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	zwischenErgebnis2
20	zwischenErgebnis1

Die *lokale* Variablen werden auf der anderen Seite des Speichers angelegt. Diese werden geordnet nach dem Prinzip LIFO (Last In – First out → Wäschestapel im Kleiderschrank) abgelegt.

Dieser Teil des Speichers wird *Stack* (Stapel) genannt.

push → etwas auf den Stack ablegen (schieben)

pop → etwas oben vom Stack nehmen (hervorholen)

Parameterübergabe

```
/* hier werden die Variablen im Speicher angelegt */
int ergebnis;

/* Funktion mit den Parametern ersteZahl und zweiteZahl*/
void subtrahiere(int ersteZahl, int zweiteZahl)
{
    /* Berechnung */
    ergebnis = ersteZahl - zweiteZahl;
}

int main()
{
    /* lokale Variablen */
    int zahl1;
    int zahl2;

    printf("Zahl 1 eingeben: ");
    scanf_s("%d", &zahl1);

    printf("Zahl 2 eingeben: ");
    scanf_s("%d", &zahl2);

    subtrahiere(zahl1,zahl2); /* Funktion wird mit Parameterübergabe aufgerufen */
    printf("\nErgebnis Zahl1 - Zahl2: %d", ergebnis);

    subtrahiere(zahl2, zahl1); /* Funktion wird mit gedrehten Parametern aufgerufen */
    printf("\nErgebnis Zahl2 - Zahl1: %d", ergebnis);

    /* zwei mal, da das letzte Enter-Tasten drücken aus 2 Zeichen besteht*/
    getchar();
    getchar();

    return 0;
}
```

Rückgabewert

```
/* Funktion mit den Parametern ersteZahl und zweiteZahl
   und einem Integer als Rückgabewert */
int subtrahiere(int ersteZahl, int zweiteZahl)
{
    int ergebnis; /* lokale variabel */

    ergebnis = ersteZahl - zweiteZahl;

    return ergebnis; /* Rückgabe von ergebnis*/
}

int main()
{
    /* lokale Variablen */
    int zahl1;
    int zahl2;
    int dasErgebnis;

    printf("Zahl 1 eingeben: ");
    scanf_s("%d", &zahl1);

    printf("Zahl 2 eingeben: ");
    scanf_s("%d", &zahl2);

    dasErgebnis = subtrahiere(zahl1, zahl2); /* Rückgabewert der Funktion in dasErgebnis */
    printf("\nErgebnis Zahl1 - Zahl2: %d", dasErgebnis);

    dasErgebnis = subtrahiere(zahl2, zahl1); /* Rückgabewert der Funktion in dasErgebnis */
    printf("\nErgebnis Zahl2 - Zahl1: %d", dasErgebnis);

    getchar();
    getchar();

    return 0;
}
```