

Praktikum Ingenieurinformatik

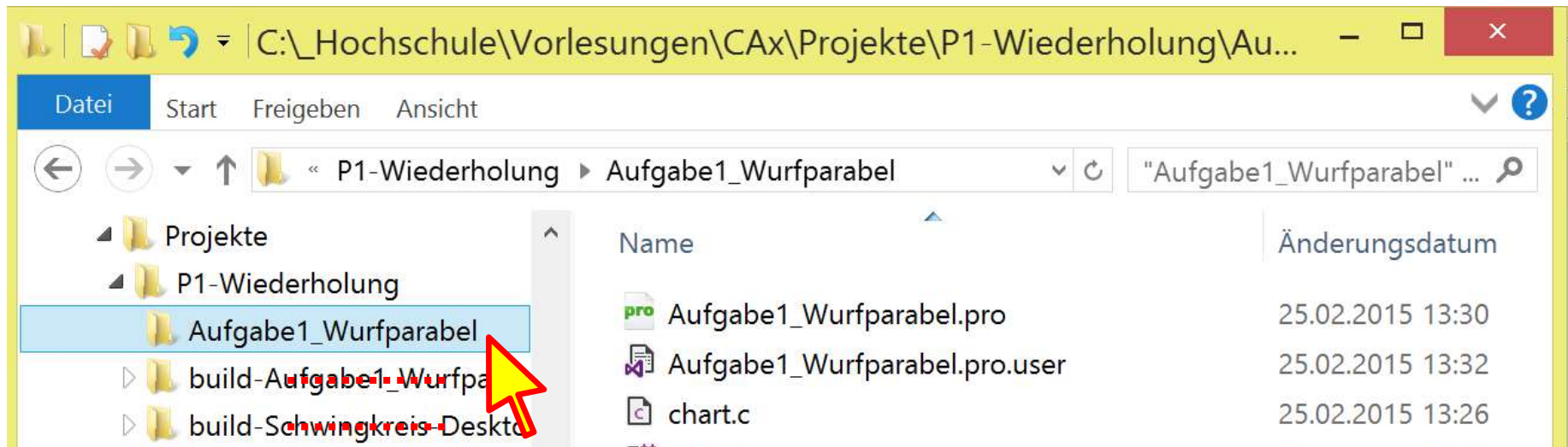
Termin 2

**Verzweigungen (if-else),
printf und scanf,
while-Schleife**

- 1. Import/Export von Projekten**
- 2. Verzweigungen (if-else-Anweisung)**
- 3. printf und scanf**
- 4. Übungsaufgaben (while-Schleife)**

1.1. Import/Export von Projekten

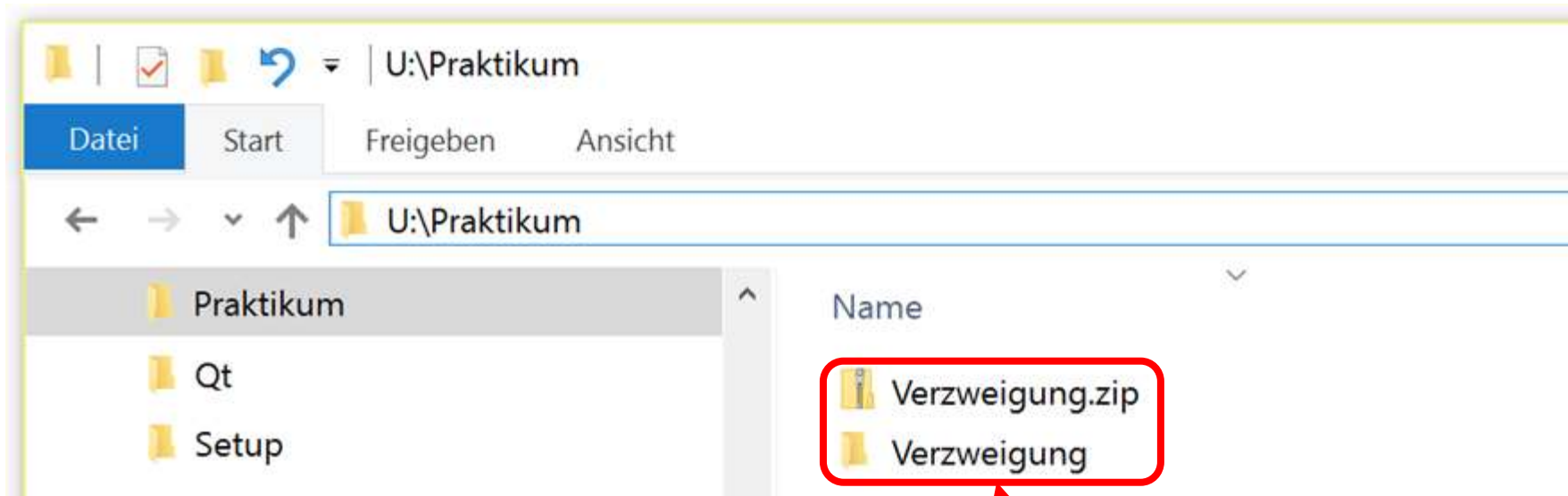
Frage: Wie kopiere ich ein C++-Projekt auf den USB-Stick, um es zur Weiterverarbeitung mit nach Hause zu nehmen?



- Qt Creator legt für jedes C-Projekt ein eigenes Unterverzeichnis an. Schließen Sie zuerst den Qt Creator.
- Nun öffnen Sie den Windows Explorer und kopieren das gewünschte Projektverzeichnis mit dem kompletten Inhalt auf Ihren USB-Stick (eventuell vorher ZIP-Archiv erstellen).
- Das „build-xxx-Verzeichnis“ muss nicht kopiert werden!

1.2. Import/Export von Projekten

Frage: Wie kopiere ich ein C++-Projekt vom USB-Stick auf meinen Rechner, um es dort mit Qt Creator weiter zu bearbeiten?



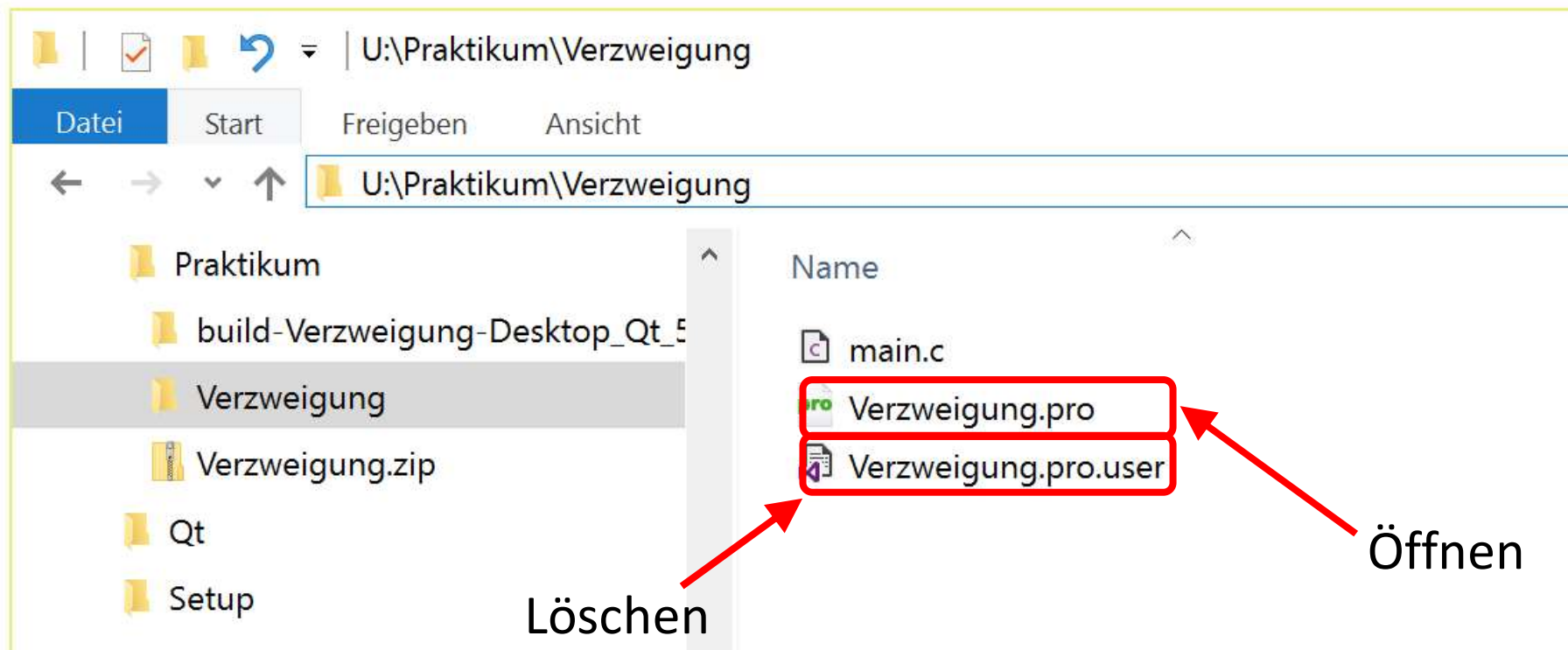
Konkretes Beispiel:

Kopieren Sie das Beispielpjekt „Verzweigung.zip“ vom Transferlaufwerk ins C++-Projektverzeichnis auf Ihrem Rechner, zum Beispiel ins Verzeichnis U:\Praktikum. Entpacken Sie dort die ZIP-Datei.

1.3. Import/Export von Projekten

Wechseln Sie nun mit dem Windows Explorer ins Projektverzeichnis (zum Beispiel U:\Praktikum\Verzweigung) und **löschen Sie dort – falls vorhanden – zunächst die Datei mit der Endung .pro.user!**

Öffnen Sie dann die Projektdatei mit der Endung .pro per Doppelclick.

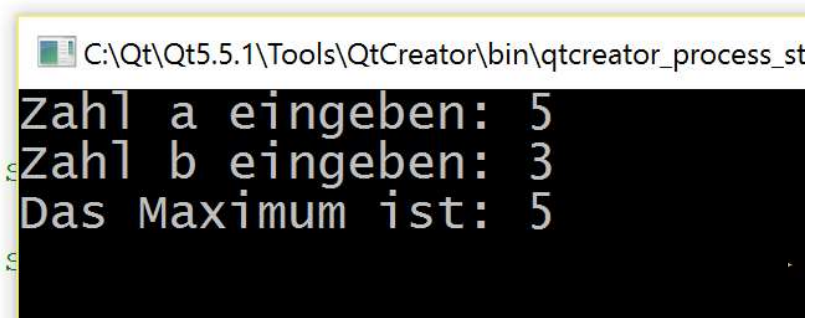


1.4. Import/Export von Projekten

Aufgabe:

Übersetzen Sie das C-Programm „Verzweigung“ und führen Sie es aus. Geben Sie unterschiedliche Werte für a und b ein. Zeigen Sie, dass alle drei Verzweigungs-Varianten ausgeführt werden können!

```
3  #include <stdio.h>
4
5  ▼ int main(void)
6  {
7      int a, b;
8      printf("Zahl a eingeben: "); scanf("%d", &a);
9      printf("Zahl b eingeben: "); scanf("%d", &b);
10 ▼  if(a == b)
11  {
12      printf("a und b sind gleich.\n");
13  }
14 ▼  else
15  {
16      if(a > b)
17          printf("Das Maximum ist %d", a);
18      else
19          printf("Das Maximum ist %d", b);
20  }
```



```
C:\Qt\Qt5.5.1\Tools\QtCreator\bin\qtcreator_process_st
Zahl a eingeben: 5
Zahl b eingeben: 3
Das Maximum ist: 5
```

1. Import/Export von Projekten
2. Verzweigungen (if-else-Anweisung)
3. printf und scanf
4. Übungsaufgaben (while-Schleife)

2.1. Verzweigungen (if-else-Anweisung)

Preisberechnung eines Mietwagens:

Der Benutzer soll dazu folgende Eingaben machen:

- Wagenkategorie, (1) klein oder (2) groß
- Mietzeit (Anzahl der Tage)
- Gefahrenen Kilometer

Ein kleiner Wagen hat 60 km frei und kostet 60 Euro pro Tag;
jeder zusätzlich gefahrene Kilometer kostet 0.40 Euro.

Große Wagen haben 90 km frei, kosten 90 Euro je Tag;
jeder zusätzlich gefahrene Kilometer 0.60 Euro.

Erstellen sie zunächst ein Struktogramm und anschließend ein C-Programm zur Berechnung des Mietpreises. Falscheingaben müssen nicht abgefangen werden.

1. Import/Export von Projekten
2. Verzweigungen (if-else-Anweisung)
3. **printf und scanf**
4. Übungsaufgaben (while-Schleife)

3.1. Formatierte Ausgabe mit printf

```
#include <stdio.h>
int main(void)
{
    int i = 10;
    printf("%d_____%d_____%d\n", i, i, i);
    i = i * 10;
    printf("%d_____%d_____%d\n", i, i, i);
    i = i * 10;
    printf("%d_____%d_____%d\n", i, i, i);
    return 0;
}
```

Ausgabe:

```
10____10____10
100____100____100
1000____1000____1000
```

3.2. Ausgabe von festen Texten

Im einfachsten Fall wird ein fester Text auf dem Bildschirm ausgegeben:

```
printf("Dies ist ein einfaches Beispiel");
```

Der Text kann auch Sonderzeichen (z. B. Zeilenumbrüche) enthalten:

```
printf("Ausgabe von \n zwei Zeilen!");
```

Die Zeichensequenz `\n` bewirkt einen Sprung an den Anfang der folgenden Bildschirmzeile. Weitere gebräuchliche Sonderzeichen sind:

<code>\n</code>	Sprung an den Anfang der folgenden Bildschirmzeile
<code>\b</code>	Gehe ein Zeichen zurück
<code>\a</code>	Akustisches Signal
<code>\r</code>	Sprung an den Anfang der aktuellen Bildschirmzeile
<code>\\</code>	Ausgabe des Gegenschrägstrichs "\" (Backslash)
<code>%%</code>	Ausgabe des Prozent-Zeichens "%"
<code>\"</code>	Ausgabe eines doppelten Anführungszeichens
<code>\t</code>	Sprung zur nächsten Tabulatorposition

3.3. Ausgabe aktueller Variablenwerte

Zur Ausgabe aktueller Variablenwerte werden in den Aufruf der Funktion `printf` entsprechende „Platzhalter“ eingefügt:

```
int x = 10;  
printf("Zahl %d wurde in x gespeichert.", x);
```

Auf dem Bildschirm erfolgt die Ausgabe „Zahl 10 wurde in x gespeichert“. Es wird also bei der Bildschirmausgabe der Platzhalter `%d` durch den aktuellen Wert von `x` ersetzt.

Es ist möglich, mehrere Variablen zugleich auszugeben:

```
int x = 123, y = 234;  
printf("x = %d und y = %d", x, y);
```

Die Ausgabe lautet nun „x = 123 und y = 234“. Für jede Variable ist ein eigener Platzhalter (hier: `%d`) notwendig. Die auszugebenden Variablen werden durch Kommas getrennt aufgelistet. Für jeden Platzhalter muss dabei eine Variable angegeben werden (hier: `x`, `y`).

3.4. Ausgabe unterschiedlicher Variablentypen

Es ist zu beachten, dass der Platzhalter zum Typ der auszugebenden Variablen passt (z. B. dient %d zur Ausgabe einer Variablen des Typs int, short oder long). Weitere Platzhalter sind:

%d, %i	int, short, long	Ganze Zahl
%x, %X	int, short, long	Ganze Zahl, Ausgabe als Hexadezimalzahl
%f	float, double	Fließkommazahl
%e, %E	float, double	Fließkommazahl, Ausgabe im Exponentialformat
%c	char	Einzelnes Zeichen (Buchstabe, Ziffer, ...)
%s	char*	Zeichenkette („String“)

```
int i = 10; double d = 22.22; char c = 'X';  
char* str = "abcdefg...";  
printf("Beispiel zu printf:\n");  
printf("%d, %f, %c\n", i, d, c);  
printf("%s", str);
```



```
C:\Windows\system32\cmd.  
Beispiel zu printf:  
10, 22.220000, X  
abcdefg...
```

3.5. Breite des Ausgabebereichs, Genauigkeit

Es kann die Breite des Ausgabebereichs angegeben werden. So gibt %10d eine ganze Zahl rechtsbündig in einem Bereich von 10 Zeichen Länge aus:

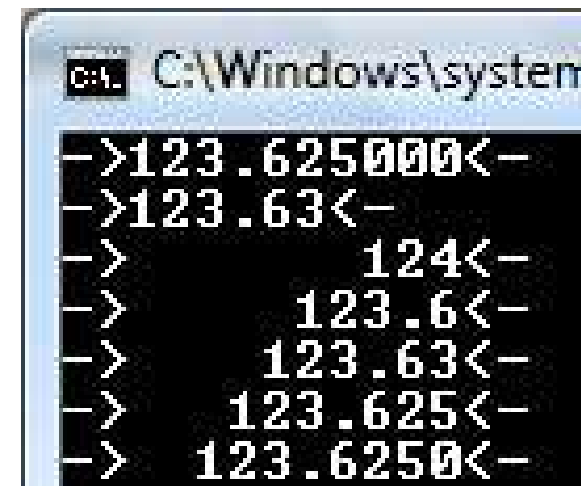
```
int i = 123;  
printf("->%d<-\\n", i);  
printf("->%4d<-\\n", i);  
printf("->%5d<-\\n", i);
```



```
C:\Windows\:  
->123<-  
-> 123<-  
->  123<-
```

Bei Fließkommazahlen kann zusätzlich die Genauigkeit eingestellt werden:

```
float f = 123.625;  
printf("->%f<-\\n", f);  
printf("->%.2f<-\\n", f);  
printf("->%10.0f<-\\n", f);  
printf("->%10.1f<-\\n", f);  
printf("->%10.2f<-\\n", f);  
printf("->%10.3f<-\\n", f);  
printf("->%10.4f<-\\n", f);
```



```
C:\Windows\system  
->123.625000<-  
->123.63<-  
->      124<-  
->      123.6<-  
->      123.63<-  
->      123.625<-  
->      123.6250<-
```

3.6. Übungsaufgabe

Tabellarische Ausgabe:

Ändern Sie das Programm vom Anfang dieses Kapitels so, dass die dargestellte Ausgabe auf dem Bildschirm erfolgt!

```
#include <stdio.h>
int main(void)
{
    int i = 10;
    printf(" --?-- \n", i, i, i);
    i = i * 10;
    printf(" --?-- \n", i, i, i);
    i = i * 10;
    printf(" --?-- \n", i, i, i);
    return 0;
}
```

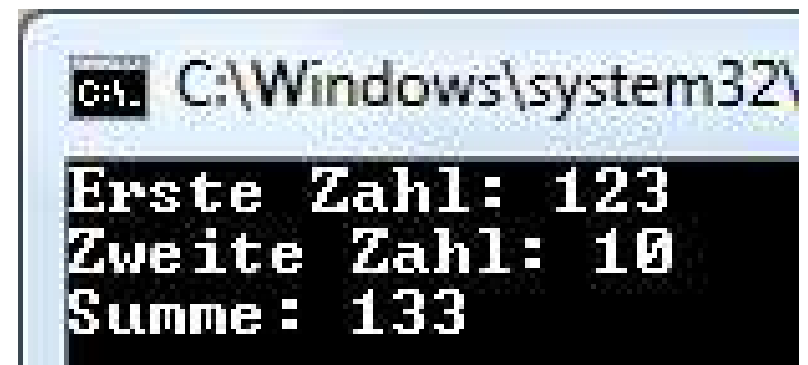
Ausgabe:

10	10	10
100	100	100
1000	1000	1000

3.7. Eingabe mittels scanf

Zur Eingabe von Variablen existiert die Funktion scanf. Wie bei printf wird mit einem „Platzhalter“ der Typ der einzugebenden Variablen angegeben. Die Variable selbst folgt als 2. Parameter (i. d. Regel mit vorangestellten &):

```
#include <stdio.h>
int main(void)
{
    int z1, z2, sum;
    printf("Erste Zahl: ");
    scanf("%d", &z1);
    printf("Zweite Zahl: ");
    scanf("%d", &z2);
    sum = z1 + z2;
    printf("Summe: %d", sum);
    return 0;
}
```



```
C:\Windows\system32\
Erste Zahl: 123
Zweite Zahl: 10
Summe: 133
```


3.8. Datentypen double und float

Bei der Ein- und Ausgabe von Fließkommazahlen (float und double) existiert ein wichtiger Unterschied zwischen printf und scanf:

- Bei der **Ausgabe** von **float-Variablen und double-Variablen** wird in beiden Fällen der Platzhalter **%f** verwendet.
- Bei der **Eingabe** von **float-Variablen** wird ebenfalls der Platzhalter **%f**, bei der Eingabe von **double-Variablen** aber **%lf** verwendet!

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    float f; double d;
```

```
    printf("Float-Zahl: ");
```

```
    scanf("%f", &f);
```

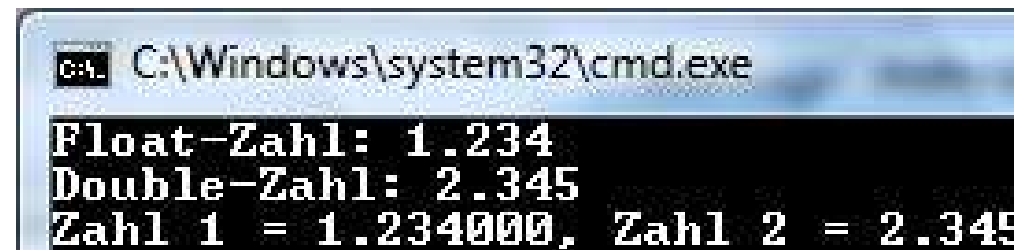
```
    printf("Double-Zahl: ");
```

```
    scanf("%lf", &d);
```

```
    printf("Zahl 1 = %f, Zahl 2 = %f", f, d);
```

```
    return 0;
```

```
}
```



```
C:\Windows\system32\cmd.exe
Float-Zahl: 1.234
Double-Zahl: 2.345
Zahl 1 = 1.234000, Zahl 2 = 2.345000
```

1. Import/Export von Projekten
2. Verzweigungen (if-else-Anweisung)
3. printf und scanf
4. Übungsaufgaben (while-Schleife)

4.1. Übungsaufgabe (while-Schleife)

Erweitern Sie das Programm aus Abschnitt 2 (Folie 8):

Das Programm soll den Anwender fragen, ob eine weitere Berechnung gewünscht wird (Eingabe von "1") oder nicht (Eingabe von "0").

Zusatzaufgabe: Verwenden Sie statt einer (abweisenden) while-Schleife eine (nichtabweisende) do-while-Schleife.

```
#include <stdio.h>
int main(void)
{
    int weiter = 1;
    while(weiter == 1)
    {
        /* ...hier findet die Berechnung statt... */
        printf("Nochmal (1) oder Ende (0)? ");
        scanf("%d", &weiter);
    }
    return 0;
}
```

*Hier den Berechnungsalgorithmus
aus Abschnitt 2 (Folie 8) einfügen!*



4.2. Zusatzübung (Stromrechnung)

Erstellung einer Stromrechnung:

Schreiben Sie ein C-Programm zur Erstellung einer einfachen Stromrechnung. Der Anwender wird aufgefordert, den Stromverbrauch in kWh einzugeben. Anschließend berechnet das Programm den Preis und gibt ihn zusammen mit dem Stromverbrauch am Bildschirm aus. Der Strompreis berechnet sich wie folgt:

- Für die ersten 250 kWh beträgt der Preis 0,15 Euro/kWh
- Für die nächsten 250 kWh beträgt der Preis jeweils 0,12 Euro/kWh
- Über 500 kWh beträgt der Preis jeweils 0,11 Euro/kWh

Beispiel:

Bei einem Verbrauch von 300 kWh muss man
 $250 \times 0,15 \text{ Euro} + 50 \times 0,12 \text{ Euro}$
bezahlen.

Erstellen Sie zuerst ein Struktogramm und danach das C-Programm!