

Übungen zu Graphikprogrammierung in C++

Was man wissen muss:

- *Java*
- *Unix (sunhalle) und der Umgang mit einem Editor*
- *theoretische Grundlagen im Umgang mit dem GNU GCC Compiler*

Die Lösungen dieses Blattes sollten gezippt an traub@in.tum.de geschickt werden. Das Entzippen muss automatisch in ein Unterverzeichnis mit eurer Kennung erfolgen.

Aufgabe 1 (H) Einfaches Programm mit dem GNU GCC

Als erste Aufgabe soll ein Programm geschrieben werden, das

```
> Hello World
```

auf die Kommandozeile ausgibt (siehe Merkblatt 1) und mit dem GNU GCC Kompiler übersetzt wird. Hierbei gilt es eine einfache main-Funktion zu schreiben. Jedes C++ Programm benötigt eine main-Funktion. Das minimalste C++ Programm ist `int main() {}`. Das Programm startet indem diese Funktion ausgeführt wird. Der Rückgabewert(`int`) gibt einen Wert an das System zurück. Ein von Null unterschiedlicher Wert signalisiert einen Fehler. In dem Aufruf

```
int main(int argc, char* argv[])
```

beschreibt `argc` die Anzahl der Argumente (inklusive dem Dateinamen des ausführbaren Programms) und `argv[x]` die Argumente selbst, wobei $x = 0$ der Dateiname ist.

Das Programm soll so erweitert werden, dass der Dateinamen und ein zusätzliches Argument mittels `cout << argv[0] << " " << argv[1] << endl;` ausgegeben werden.

Das implementierte Programm soll mit

```
g++ Hello.cpp -o Hello
```

kompiliert werden. Nach dem Kompilieren kann das Programm mit `./Hello` ausgeführt werden.

Aufgabe 2 (H) Verwenden der Standardausgabe

In C++ gibt es dieselben Typen von Variablen als in Java. Der aus Java bekannte Typ `int` ist in C++ als `int` definiert. Ein weitere Datentyp ist `char*` der als ein Array von Zeichen interpretiert werden kann. Im Grunde ist es ein Pointer auf eine Struktur vom Typ `Charakter`, aber Pointer werden erst später detailliert eingeführt.

In einem Programm sollen Sie Ihren Namen und Ihr Geburtsdatum dreimal hintereinander auf die Standardausgabe ausgeben. Hierbei soll speziell der Unterschied zu der Methode `printf` aus der C Programmiersprache verdeutlicht werden. Mit `cout` muss man nicht auf bestimmte Datentypen achten, sondern der Operator `<<` bewirkt, dass die Variable, die auf den Operator folgt direkt ausgegeben wird. Das System entscheidet, wie diese Variable interpretiert wird.

Aufgabe 3 (H) Mathematische Operationen

Ein weiterer Datentyp ist `float`. Dieser stellt 32 Bit Gleitkommazahlen dar. Des weiteren können 64 Bit Gleitkommazahlen durch `double` deklariert werden.

```
#include <header>
```

erlaubt es bestimmten Operationen, die in externen Dateien implementiert sind zu verwenden. Die Headerdatei beinhaltet nur das Interface. Die Implementierung ist in den dazugehörigen `.cpp` Dateien. Diese werden erst beim Binden dazugefügt. Der GNU GCC linkt standardmäßig gegen die `libm (-lm)`, die die Implementierung aller Standardklassen enthält. Ein Beispiel, das in allen vorangegangenen Programmen verwendet wurde ist `#include <iostream>`. Dies stellt Operationen wie z.B. die Ausgabe auf den Bildschirm zur Verfügung. Eine weitere Datei ist die

```
#include <cmath>
```

Diese stellt komplexere arithmetische Operationen zur Verfügung. Ein Beispiel ist die Berechnung des Cosinus und des Sinus.

In einem Programm sollen alle Werte des Cosinus und des Sinus von 0 bis 360 Grad im Intervall von 5 Grad ausgegeben werden.

Aufgabe 4 (H) Namespaces

In allen vorangegangenen Programmen wurde eine Zeile als selbstverständlich hingenommen.

```
using namespace std;
```

Was passiert, wenn man diese Zeile in dem HelloWorld Programm löscht?

Der namespace definiert einen Namensraum in dem bestimmte Funktionen- und Variablennamen gültig sind. Entweder muss dieser Namensraum global in der Datei gesetzt werden (`using namespace std;`) oder er muss lokal vor jedem Aufruf einer Funktion, die sich in diesem Namensraum befindet, spezifiziert werden (`std::cout`).

Das HelloWorld Programm soll wieder lauffähig gemacht werden, ohne die Zeile `using namespace std;` zu verwenden.

Was man gelernt haben sollte:

- *kompilieren mit g++*
- *int main() {..}*
- *include statements*
- *cout / << / endl*
- *using namespace std;*