

Praktikum Programmieren
Übungsblatt 1
Termin: 19. Oktober 2023

Einleitung

Um C++-Programme zu erstellen, zu übersetzen („kompilieren“), auszuführen und zu debuggen gibt es viele verschiedene Möglichkeiten. Grundsätzlich reicht es, einen Texteditor (z.B. „notepad“ unter Windows oder „vi“/„emacs“/„gedit“ unter Linux) zu verwenden und einen C++-Compiler (z.B. g++ oder clang) zu installieren.

Etwas einfacher ist aber insbesondere der Einstieg in die Programmierung mit einer Entwicklungsumgebung (einer sogenannten „Integrated Development Environment“ oder kurz „IDE“) an. Hier gibt es ein reichhaltiges Angebot an kommerziellen und nicht-kommerziellen Lösungen, wie z.B. „Visual Studio Code“, „CLion“, „Dev-C++“ oder „Code::Blocks“.

Für das Praktikum gibt es keine zwingende Vorgabe, welche Umgebung Sie nutzen. Recht verbreitet, leicht anwendbar und noch dazu für alle üblichen Plattformen verfügbar (Windows, Linux, Mac) ist Visual Studio Code. Wenn Sie diese Umgebung verwenden, können auch die Tutor:innen Sie am besten unterstützen.

Anleitungen zur Installation finden Sie unter den folgenden Adressen:

- Windows: <https://code.visualstudio.com/docs/cpp/config-mingw>
- MacOS: <https://code.visualstudio.com/docs/cpp/config-clang-mac>
- Linux: <https://code.visualstudio.com/docs/cpp/config-linux> (Debian/Ubuntu/Linux Mint)

Dieses erste Übungsblatt soll vor allem dazu dienen, dass Sie die Entwicklungsumgebung einmal einrichten und sich damit vertraut machen.

Übung 1 „Mein erstes C++-Programm“

In dieser Aufgabe sollen Sie ein erstes C++-Programm erstellen, übersetzen, eventuelle Fehler eliminieren und das Programm schließlich ausführen.

Geben Sie folgendes Programm ein:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char* argv[]) {
5     int Liste[20];
6     int Minimum;
7     int i;
8
9     // Füllen der Liste mit Zufallselementen
10    for (i = 0; i < 20; i++) {
11        Liste[i] = rand();
12    }
13 }
```

```
14 // Bislang kleinstes Element
15 Minimum = Liste[0];
16
17 // Kleinstes Element der ganzen Liste suchen
18 for (i = 1; i < 20; i++) {
19     if ( Liste[i] < Minimum ) {
20         Minimum = Liste[i];
21     }
22 }
23
24 // Ausgabe des Ergebnisses
25 printf("Das Minimum von ");
26 for (i = 0; i < 20; i++) {
27     printf("%i ", Liste[i]);
28 }
29 printf(" ist :\n%i\n", Minimum);
30 return 0;
31 }
```

Übersetzen Sie das Programm und führen Sie es aus.

Übung 2 „Testen von Programmänderungen“

Wenn Sie Aufgabe 1 erfolgreich ausgeführt haben, führen Sie verschiedene Modifikationen am Programm durch:

- Produzieren Sie Fehler im Programm und sehen Sie sich die vom Compiler erzeugten Fehlermeldungen an.
 - Ändern Sie Variablenamen in einer Deklaration, z.B. von `Minimum` nach `Min`.
 - Entfernen Sie ein Semikolon.
 - Ändern Sie ein Schlüsselwort (z.B. `four` statt `for`).
 - Entfernen Sie eine schließende geschweifte Klammer.
- Schreiben Sie das Programm so um, dass das größte statt dem kleinsten Element ausgegeben wird.
- Ändern Sie das Programm so, dass das kleinste Element aus einer Liste mit 40 Elementen ausgegeben wird.
- Verwenden Sie Umlaute in einem Kommentar oder einem Variablenamen.
- Was passiert, wenn Sie das 55. Element einer Liste mit 40 Elementen ausgeben?

Übung 3 „Debugging“

Starten Sie das Programm aus Aufgabe 1 mit dem Debugger und verfolgen Sie, wie sich die Werte von Variablen über den Ablauf des Programms hinweg verändern. (In Visual Studio Code müssen Sie dafür vor dem Start einen Breakpoint am Anfang der `main`-Funktion setzen, sonst läuft das Programm einfach durch.)