

# Einführung in die Programmierung, WS 2016/2017 Blatt 3

Simon Hangl, Sebastian Stabinger, Alex Hirsch

2016–11–08

- Abgabe bis spätestens Montag 21:59:59 über OLAT (<https://lms.uibk.ac.at/olat/dmz/>).
- Bereiten Sie jede Aufgabe so vor, dass Sie Ihre Lösung im Proseminar präsentieren können!
- Benennen Sie Ihre Abgabe nach folgendem Schema:  
*Gruppennummer-Nachname-blattÜbungsblattnummer.tar.gz* Wenn Sie also Max Mustermann heißen und Gruppe 1 besuchen, heißt die Datei von Übung 3: *1-mustermann-blatt3.tar.gz*
- Compilieren Sie alle Programme mit den Optionen `-Wall -Werror -std=c99`
- **Vergessen Sie das Ankreuzen in OLAT nicht!**

## Feedback

Nutzen Sie die angebotenen Möglichkeiten, uns Feedback zu geben (eMail, Tutorium, Proseminar). Hier können Sie uns auf Probleme, notwendige Stoffwiederholungen, Unklarheiten, aber auch positive Dinge, die beibehalten werden sollten, hinweisen.

## Testen und Dokumentation

Stellen Sie sicher, dass alle Lösungen fehlerfrei kompilieren. Testen Sie Ihre Lösungen ausführlich (z.B. auf falsche Eingaben, falsche Berechnungen, Sonderfälle) und dokumentieren Sie diese. Dies hilft Ihnen bei der Präsentation und uns beim Nachvollziehen Ihrer Entscheidungen. Lesen Sie die Aufgaben *vollständig* durch.

## Aufgabe 1 (2 Punkte)

Es folgt eine Liste mit Beschreibungen der Funktion von Variablen. Geben Sie jeweils einen sinnvollen Namen und Typ für die Variable an und begründen Sie Ihre Entscheidung.

- Eine Variable welche innerhalb einer While-Schleife zum Zählen verwendet wird. `i`
- Wir schreiben eine Software zum Verwalten von Grundstücken und wollen eine Variable definieren welche ...

`area` – ... die Gesamtfläche des Grundstücks enthält.  
`owners` – ... die Anzahl der Grundstückseigentümer enthält.  
`lastNameInitial` – ... den Anfangsbuchstaben des Nachnamens des Haupteigentümers des Grundstücks enthält.  
`isBuildingSite` – ... angibt ob es sich um ein Baugrundstück handelt oder nicht.

Erklären Sie zudem was **Hungarian Notation** bedeutet.

**Hinweis:** Hungarian Notation kommt nicht in den Vorlesungsfolien vor. Sie sollen sich eigenständig darüber informieren.

**Hinweis:** Abgabe: 1-mustermann-a1.txt

## Aufgabe 2 (3 Punkte)

Schreiben Sie ein Programm in welchem die Variable `int n` mit einem Wert  $> 1$  initialisiert ist. Geben Sie alle Zahlen zwischen 1 und `n` aus wobei abwechselnd von Unten hinauf und von Oben herab gezählt werden soll.

z.B. Für `n=10` soll die Ausgabe folgendermaßen aussehen: 1 10 2 9 3 8 4 7 5 6

**Hinweis:** Es darf keine Zahl doppelt ausgegeben werden und es darf auch keine Zahl fehlen.

**Hinweis:** Abgabe: 1-mustermann-a2.c

```
while (n<=1){  
    print n  
    print  
    n- -  
}
```

### Aufgabe 3 (3 Punkte)

Schreiben Sie ein Programm welches die Fibonacci Zahlen kleiner  $n$  iterativ berechnet und ausgibt.

Die Fibonacci Zahlen sind folgendermaßen definiert: Die ersten beiden Zahlen sind 0 und 1. Die nächste Zahl ist jeweils die Summe der zwei vorangehenden Zahlen.

Also:  $f_i = f_{i-1} + f_{i-2}$  und  $f_0 = 0$  und  $f_1 = 1$ .

z.B. Für  $n = 10$  soll die Ausgabe folgendermaßen aussehen: 0 1 1 2 3 5 8

**Hinweis:** Vier Variablen (inklusive  $n$ ) und eine Schleife sind ausreichend für eine Implementierung. Alternativ ist auch eine Implementierung mit drei Variablen (inklusive  $n$ ), einer Schleife und Verzweigungen möglich.

**Hinweis:** Abgabe: 1-mustermann-a3.c

## Aufgabe 4 (2 Punkte)

Das folgende C-Program, welches einige Variablen definiert, ist gegeben.

```
#include <stdlib.h>

int main(void) {
    double a = 1.23e14;
    float b = 0.034;
    char c = 'X';
    unsigned int d = 3147483647;
    long e = 31474836472;
    { int sd = 232; }
    int zero;

    /* TODO print variables */

    return EXIT_SUCCESS;
}
```

- Geben Sie den Inhalt der Variablen **a**, **b**, **c**, **d** und **e** korrekt auf der Konsole aus
- Geben Sie **c** als Hexadezimalzahl aus
- Geben Sie **a** in Exponentialschreibweise aus
- Geben Sie **b** mit 10 Nachkommastellen aus. Fällt Ihnen hierbei etwas auf?
- Geben Sie **c** als Integer mit drei Stellen (evtl. führende Nullen) aus
- Erklären Sie die Lebensdauer der variable **sd**.
- Welchen Wert erwarten Sie für die Variable **zero**?

**Hinweis:** Kompilieren Sie Ihr Program mit den Optionen `-Wall -Werror -Wno-unused-variable`

**Hinweis:** Abgabe: `1-mustermann-a4.c`

## Aufgabe 5 (2 Punkte)

Definieren sie eine Aufzählungskonstante (enum) welche verwendet werden kann um zu speichern ob sich ein Student im **Bachelor**-, **Master**- oder **PhD**-Studium befindet.

Legen Sie eine Variable dieses Typs an und geben Sie je nach Inhalt dieser Variablen aus: “Der Student befindet sich im Bachelor Studium”, “Der Student befindet sich im Master Studium” oder “Der Student befindet sich im PhD Studium”.

***Hinweis:** Abgabe: 1-mustermann-a5.c*

## Aufgabe 6 (3 Punkte)

Lesen Sie vom Benutzer drei Fließkommazahlen mittels `scanf` ein und geben Sie die größte und kleinste dieser Zahlen auf dem Bildschirm aus.

***Hinweis:** Abgabe: 1-mustermann-a6.c*