

Einführung in die Programmierung, WS 2015/2016 Blatt 4

Simon Hangl, Sebastian Stabinger, Benedikt Hupfaut, Johannes Kessler

2015-11-04

- Abgabe bis spätestens Dienstag 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 4: *1-mustermann-blatt4.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 sie. 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 Schleife zum Zählen verwendet wird.
- Wir schreiben eine Software zum Verwalten von Grundstücken und wollen eine Variable definieren welche ...
 - ... die Gesamtfläche des Grundstücks enthält.
 - ... die Anzahl der Grundstückseigentümer enthält.
 - ... den Anfangsbuchstaben des Nachnamens des Haupteigentümers des Grundstücks enthält.
 - ... angibt ob es sich um ein Baugrundstück handelt oder nicht.

Hinweis: Abgabe: 1-mustermann-a1.c

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

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 Stück C-Code, welches einige Variablen definiert, ist gegeben:

```
double a = 2.23e14;
float b = 0.034;
char c = 'C';
unsigned int d = 3147483647;
long e = 31474836472;
{ int sd = 234; }
int zero;
```

- 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
- Was fällt Ihnen auf, wenn sie **zero** ausgeben. Was läuft hier schief bzw. könnte schief laufen?

Hinweis: Sie müssen ihr Testprogramm möglicherweise ohne die Option `-Werror` compilieren.

Hinweis: Abgabe: 1-mustermann-a4.c