

## Informatik I: Einführung in die Programmierung

Prof. Dr. Peter Thiemann  
Hannes Saffrich, Simon Ging  
Wintersemester 2021

Universität Freiburg  
Institut für Informatik

### Übungsblatt 1

**Abgabe: Montag, 25.10.2021, 9:00 Uhr morgens**

Bevor es mit den eigentlichen Aufgaben losgeht, müssen Sie zunächst die für die Vorlesung benötigte Software installieren und sich mit unsere Webplattform zur Abgabe der Übungen vertraut machen.

Für die Webplattform haben wir ein kurzes Einführungsvideo vorbereitet:

<https://youtu.be/n6fJp2u6Ws0>

Schauen Sie sich das Video an; es enthält auch ein paar Details zu den Tutoraten, der Korrektur und der Übungsabgabe.

Für die Übungen in dieser Vorlesung benötigen Sie die folgende Software:

- Python 3.10 (die genaue Version ist wichtig!)
- Visual Studio Code
- git (optional)

Wir stellen hierbei Installationsanleitungen für Linux und MacOS zu Verfügung. Falls Sie Windows verwenden empfehlen wir entweder

- unser VirtualBox Image zu verwenden, welches ein Ubuntu 21.04 enthält auf dem bereits die benötigte Software installiert und konfiguriert ist; oder
- ein Ubuntu in Version 21.04 oder 21.10 als Zweitsystem neben ihrer Windows-plattform zu installieren.

Sie können das VirtualBox Image auch auf Linux oder MacOS verwenden, sollten Sie dort Schwierigkeiten bei der Installation haben.

VirtualBox ist eine sogenannte *Virtual Machine*, die es erlaubt ein Betriebssystem inklusive installierter Software auf einem anderen Betriebssystem auszuführen. Wenn Sie unser Image also mit VirtualBox ausführen, dann öffnet sich ein Fenster auf ihrem normalen Betriebssystem, in dem das fertig konfigurierte Ubuntu Linux läuft.

Es steht Ihnen natürlich frei die Software auf Windows oder einem anderen Linux selbst zu installieren und zu konfigurieren, aber dabei müssen Sie dann selbst herausfinden, wie das funktioniert.

### Installation über VirtualBox

Um unser vorkonfiguriertes VirtualBox Image zu verwenden, müssen Sie sich zu-

nächst die VirtualBox Software<sup>1</sup> installieren und unser Image<sup>2</sup> herunterladen. Anschließend müssen Sie die VirtualBox Software starten, dort das Image importieren und starten. Wir empfehlen das VirtualBox Image so zu konfigurieren, dass es mindestens 2 GB Arbeitsspeicher, 2 CPU Kerne und 128 MB Videospeicher zu Verfügung hat. Der Username und das Passwort auf dem Image ist jeweils **student**.

### Installation auf Ubuntu Linux 21.04 oder 21.10

Öffnen Sie ein Terminal, geben Sie dort den folgenden Befehl ein und drücken Sie dann **Enter**:

```
wget -q0- http://proglang.informatik.uni-freiburg.de/teaching/info1/2021/install-on-ubuntu.sh | bash
```

Hinweis: Das 0 in -q0- ist ein großes O und keine Null.

Wenn die Installation erfolgreich war, sehen sie am Ende folgende Ausgabe:

```
SUCCESS: All components were installed.
```

### Installation auf MacOS

Installieren Sie den *Homebrew* Package Manager, sofern Sie diesen noch nicht installiert haben. Öffnen Sie hierzu ein Terminal, geben Sie dort den folgenden Befehl ein und drücken Sie dann **Enter**:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Installieren Sie anschließend die benötigte Software, indem Sie die folgenden Befehle, wie zuvor, im Terminal ausführen:

```
brew install python@3.10
echo 'export PATH="/usr/local/opt/python@3.10/bin:$PATH"' >> ~/.zshrc
source ~/.zshrc
brew install --cask visual-studio-code
brew install git
```

### Konfiguration von Visual Studio Code

Wenn Sie die benötigte Software selbst installiert haben und *nicht* das VirtualBox Image verwenden, dann müssen Sie Visual Studio Code noch konfigurieren.

Starten Sie Visual Studio Code und installieren sie die **Python**-Erweiterung. Klicken Sie hierzu, in der Toolbar am linken Fensterrand, auf das *Extensions*-Icon, suchen Sie nach “Python” und klicken Sie dann auf den *install*-Button.

Als letztes müssen Sie noch einige Einstellungen setzen. Klicken Sie hierzu, wieder in der Toolbar am linken Fensterrand, auf das *Settings*-Icon und führen dort die folgenden Schritte aus:

- bei “Python > Analysis: Type Checking Mode” den Wert “basic” auswählen;
- bei “Python > Linting: Enabled” den Haken setzen;

---

<sup>1</sup><https://www.virtualbox.org/wiki/Downloads>

<sup>2</sup><https://archive.informatik.uni-freiburg.de/courses/proglang/2021-WS-Info1/EiP-2021WS-Ubuntu.ova>

- bei “Python > Linting: Pycodestyle Enabled” den Haken setzen. Anschließend öffnet sich ein Popup und bietet Ihnen an `pycodestyle` zu installieren. Stimmen Sie zu;
- bei “Python > Linting > Pycodestyle Category Severity: E” den Wert “Warning” auswählen;
- bei “Security > Workspace > Trust: Enabled” den Haken *wegnehmen*.

### Dateiformate für Ihre Abgabe

Die einzelnen Aufgaben auf den Übungsblättern sind stets mit vollständigen Dateinamen annotiert. Bitte halten Sie sich an die Namen und die dazugehörigen Dateiformate. Beispielsweise soll Aufgabe 1.1 als Datei `algorithms.txt` abgegeben werden.

Die Dateiformate sind immer in *plaintext* und haben die Endungen `.txt` und `.md` für Text und `.py` für Python-Code. Das heißt es sind insbesondere keine PDFs, keine Word-Dokumente und auch keine Bildschirmfotos erlaubt ;-)

Eine simple Möglichkeit diese Kriterien zu erfüllen ist es einfach alle Dateien mit Visual Studio Code zu erstellen bzw. zu bearbeiten. Wenn Sie zum Beispiel die Endung `.txt` verwenden, dann erkennt Visual Studio Code, dass es sich nicht um ein Python-Skript handelt und verhält sich wie ein normaler Texteditor.

### Aufgabe 1.1 (Algorithmen; 5 Punkte; Datei: `algorithms.txt`)

Handelt es sich bei der folgenden umgangssprachlichen, prozeduralen Beschreibung um einen Algorithmus?

Entscheiden Sie dazu, ob die Beschreibung die Eigenschaften Präzision, Effektivität, statische Finitheit, dynamische Finitheit und Terminierung (siehe Folien) erfüllt. Begründen Sie jeweils kurz Ihre Antwort.

*Gegeben seien zwei positive Zahlen  $a, b$ . Setze  $k = 0$ . Solange  $k$  kleiner ist als  $b$ , führe folgende Schritte durch: Addiere  $b$  zu  $a$  hinzu ( $b$  bleibt unverändert). Falls  $a$  nun gerade ist, erhöhe  $k$  um 1. Ist  $k$  größer oder gleich  $b$ , gib  $a$  aus.*

### Aufgabe 1.2 (Print-Rätsel; 12 Punkte; Dateien: `print1.py` und `print2.py`)

Ihr Python-Skript darf pro Teilaufgabe nur einen einzelnen `print`-Befehl beinhalten.

- (a) (6 Punkte) Schreiben Sie ein Python-Skript `print1.py`, das folgende Ausgabe produzieren:

```
Test
Test
Test
```

Der String "Test" darf dabei nur ein einziges Mal im `print`-Befehl vorkommen.

Folgende Python-Skripts sind also *keine* Lösung:

```
- print("Test\nTest\nTest")  
  
- print("Test")  
  print("Test")  
  print("Test")
```

- (b) (6 Punkte) Schreiben Sie ein Python-Skript `print2.py`, das folgende Ausgabe produziert:

```
000  
000  
000
```

Der String "0" darf dabei nur ein einziges Mal im `print`-Befehl vorkommen.

**Aufgabe 1.3** (Gruppenarbeit: Ada Lovelace; 6 Punkte; Datei: `lovelace.txt`)

Schreiben Sie (wenn möglich) zu dritt eine Kurzbiografie über Ada Lovelace. Heben Sie ihre Beiträge zur Informatik hervor.

Die Kurzbiographie soll in etwa 1500 Zeichen bzw. 200 Wörter lang sein. Ziel dieser Aufgabe ist in erster Linie mit ihrer Gruppe warm zu werden. Sobald wir etwas mehr Python gelernt haben, wird es dann auch interessantere Gruppenaufgaben geben.

Schreiben Sie in die erste Zeile Ihres Textes die RZ-Account Kürzel (xy123) Ihrer Gruppenmitglieder, sodass unsere Tutoren die Aufgabe nicht mehrfach korrigieren müssen. Jedes Gruppenmitglied soll die Gruppenarbeit hochladen.

**Aufgabe 1.4** (Erfahrungen; 2 Punkte; Datei: `NOTES.md`)

Notieren Sie Ihre Erfahrungen mit diesem Übungsblatt (benötigter Zeitaufwand, Probleme, Bezug zur Vorlesung, Interessantes, etc.).

Editieren Sie hierzu die Datei `NOTES.md` im Abgabeordner dieses Übungsblattes auf unserer Webplattform. Halten Sie sich an das dort vorgegebene Format, da wir den Zeitbedarf mit einem Python-Skript automatisch statistisch auswerten. Die Zeitanzeige 3.5 h steht dabei für 3 Stunden 30 Minuten.