

Informatik I: Einführung in die Programmierung

Prof. Dr. Peter Thiemann
Hannes Saffrich, Michael Uhl
Wintersemester 2022

Universität Freiburg
Institut für Informatik

Übungsblatt 1

Abgabe: Montag, 24.10.2022, 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.

Webplattform

Für die Webplattform haben wir ein kurzes Einführungsvideo vorbereitet:

<https://youtu.be/n6fJp2u6Ws0>

Schauen Sie sich das Video an. *Wichtig:* Das Video ist von der Vorlesung des Vorjahres, d.h. die dort enthaltenen organisatorischen Informationen sind nicht mehr aktuell, aber die Funktionsweise der Webplattform und der Übungsabgabe ist immer noch dieselbe.

Präsenztutorate

Die Anmeldung zu den Präsenztutorate findet über den Chat¹ unserer Webplattform im Raum #2022WS-EiP-exercise-01 statt.

Software

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 (Ubuntu), MacOS und Windows zu Verfügung.

Sollten Sie bei der Installation Probleme haben, dann können Sie gerne im Chat oder in den Tutoraten fragen. Wir sind auch ganz lieb und beißen nicht :-)

Installation auf Ubuntu Linux 22.04

Öffnen Sie ein Terminal und geben Sie dort die folgenden Befehle ein:

```
sudo apt install python3.10 git
```

¹Auf <https://git.laurel.informatik.uni-freiburg.de> einloggen und oben im Menü auf “Chat” klicken.

Laden Sie sich Visual Studio Code von folgender Webseite herunter:

<https://code.visualstudio.com/sha/download?build=stable&os=linux-deb-x64>

Öffnen Sie ein Terminal in Ihrem Download-Verzeichnis und installieren Sie Visual Studio Code via

```
sudo apt install ./vscode.deb
```

Statt `vscode.deb` müssen Sie dabei den Dateinamen der heruntergeladenen Datei verwenden.

Installation auf Ubuntu Linux 21.04 oder 21.10

Öffnen Sie ein Terminal und geben Sie dort den folgenden Befehl ein:

```
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
```

Installation auf Windows

Python:

- Laden Sie sich Python 3.10 von folgender Webseite herunter:

<https://www.python.org/ftp/python/3.10.8/python-3.10.8-amd64.exe>

- Setzen Sie am Anfang der Installation einen Haken bei “Add python.exe to PATH”.
- Klicken Sie am Ende der Installation auf “Disable PATH length limit”.

Visual Studio Code:

- Laden Sie sich Visual Studio Code von folgender Webseite herunter:

<https://code.visualstudio.com/sha/download?build=stable&os=win32-x64-user>

Git (optional):

- Laden Sie sich das Git Versionskontrollsystem von folgender Webseite herunter:
<https://github.com/git-for-windows/git/releases/download/v2.38.0.windows.1/Git-2.38.0-64-bit.exe>
- Wählen Sie bei der Installation VSCode als den Standardeditor für Git aus.
- Wählen Sie bei der Installation “Use bundled OpenSSH” aus.
- Verwenden Sie bei allen anderen Optionen die Vorauswahl des Installers.

Konfiguration von Visual Studio Code

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;
- bei “Python > Linting: Pycodestyle Enabled” den Haken setzen;
- bei “Python > Linting > Pycodestyle Category Severity: E” den Wert “Warning” auswählen;
- bei “Python > Linting: Pycodestyle Args” auf “Add Item” klicken und dort
`--ignore=W292,E501,E704,W503,W504`
eingeben und auf “OK” klicken.
- bei “Security > Workspace > Trust: Enabled” den Haken *wegnehmen*.

Wenn Sie in Visual Studio Code zum ersten mal ein Python-Skript öffnen (eine Datei mit Endung `.py`), dann fragt Visual Studio Code, ob es zusätzliche Tools installieren soll. Stimmen Sie dem zu.

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; 7 Punkte; Dateien: `print1.py` und `print2.py`)

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

- (a) (3 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) (4 Punkte) Schreiben Sie ein Python-Skript `print2.py`, das folgende Ausgabe produziert:

```
HHH
HHH
HHH
```

Der String `'H'` darf dabei nur ein einziges Mal im `print`-Befehl vorkommen.

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

Schreiben Sie in Gruppen *von 1 bis 3 Leuten* 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. Gruppenaufgaben müssen von allen Gruppenmitgliedern abgegeben werden.

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 Zeitan-gabe 3.5 h steht dabei für 3 Stunden 30 Minuten.

Bei Gruppenaufgaben müssen alle Gruppenmitglieder die RZ-Accounts aller Grup-penmitglieder angeben, sodass wir Sie bei der Korrektur zuordnen können und unse-re Tutoren die Aufgabe nicht mehrfach korrigieren müssen. Fügen Sie hierzu in der NOTES.md-Datei unter der Zeile für den Zeitbedarf eine weitere Zeile hinzu die exakt das folgende Format hat:

Gruppe: xy123, xy234, xy345

Hierbei stehen xy123, xy234, und xy345 für die RZ-Accounts der Gruppenmitglieder. Der Buildserver überprüft ebenfalls, ob Sie das Format korrekt angegeben haben. Prüfen Sie, ob der Buildserver mit Ihrer Abgabe zufrieden ist, so wie es im Video zur Lehrplattform gezeigt wurde.