

Einführung in die Programmierung

Prof. Dr. Peter Thiemann
Marius Weidner
Simon Dorer, Timpe Horig

Universität Freiburg
Institut für Informatik
Wintersemester 2025

Übungsblatt 1

Abgabe: Montag, 20.10.2025, 09:00 Uhr

Wichtig: Bevor Sie mit den Aufgaben beginnen, melden Sie sich bitte jetzt für ein Tutorat auf [HisInOne](#) an.

Ankündigung: Am Mittwoch 15.10.2025 von 12:15 bis 13:00 Uhr findet in HS 00 026 μ -Saal G.-Köhler-Allee 101 eine Erklärung zum Abgabesystem statt.

Aufgabe 1.1 (Setup und Abgaberegeln)

In dieser Aufgabe sollen Sie die für die Vorlesung benötigte Software installieren und sich mit unserer Webplattform zur Abgabe der Übungen vertraut machen.

(a) **Setup**

Eine Anleitung zur Installation der benötigten Software finden Sie [hier](#).

(b) **Abgabesystem**

Eine Anleitung für unser Abgabesystem finden Sie [hier](#).

(c) **Abgaberegeln**

Die Abgaberegeln für die Übungen finden Sie [hier](#). Beachten Sie insbesondere, dass bei einem fehlerhaften Build die gesamte Abgabe mit 0 Punkten bewertet wird.

Sollten Sie bei Probleme oder Fragen haben, können Sie sich gerne im Chat¹ melden oder in den Tutorien fragen. Wir sind auch ganz lieb und beißen nicht :)

Aufgabe 1.2 (Wankelmütige Subtraktion; 4 Punkte; Datei `sub.txt` oder `sub.md`)

Gegeben sei die folgende umgangssprachliche, prozedurale Beschreibung:

Wankelmütige Subtraktion

- 1: Gegeben seien zwei Zahlen a und b aus der Menge der ganzen Zahlen \mathbb{Z}
 - 2: Falls $a > b$: Setze $a = a - b$
 - 3: Falls $a < b$: Setze $b = b - 1$
 - 4: Falls $a = b$: Gib a aus oder gib b aus und beende den Algorithmus
 - 5: Wiederhole den Vorgang ab Schritt 2.
-

¹Auf <https://chat.laurel.informatik.uni-freiburg.de> mit Ihrem RZ-Account (wie bei Ili-as) einloggen.

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

Entscheiden Sie außerdem, ob es sich bei der Beschreibung um einen Algorithmus handelt. Begründen Sie Ihre Antwort.

Aufgabe 1.3 (Ausdrücke; 3 Punkte; Datei: `expr.txt` oder `expr.md`)

In dieser Aufgabe geht es um Unterschiede zwischen dem *Skript-Modus* und dem *interaktiven Modus* von Python.

Im Skript-Modus können sie ein Python-Skript (Dateiendung `.py`) ausführen, indem sie es in Visual Studio Code öffnen und rechts oben auf das `▷`-Symbol klicken. Dabei öffnet sich ein Terminal, in dem das Skript ausgeführt wird und die Ausgabe angezeigt wird.

Im interaktiven Modus können Sie Python-Befehle direkt in der Konsole eingeben und erhalten sofort eine Antwort. Öffnen Sie hierzu ein Terminal und geben Sie den Befehl `python3.12` ein. Nun können Sie Python-Code hinter dem Prompt-Zeichen `>>>` eingeben. Beenden können Sie den Modus u.a. mit dem Befehl `exit()`.

- (a) (1.5 Punkte) Wenn Sie nacheinander die Ausdrücke `42` und `print(42)` in die interaktive Konsole eingeben, erhalten Sie die gleiche Ausgabe. Dennoch sind die beiden Ausdrücke nicht identisch. Versuchen Sie die Unterschiede herauszufinden, indem Sie die beiden Ausdrücke auch im Skript-Modus ausführen. Erklären Sie Ihre Erkenntnisse in eigenen Worten.
- (b) (1.5 Punkte) Wenn Sie `print(42 * ((1337 + 1) // 3) + (19 * 42))` in die interaktive Konsole eingeben, sehen Sie statt einer Zahl nur `...` als Ausgabe. Warum? Was passiert, wenn Sie den gleichen Ausdruck als Skript ausführen?

Sie müssen für diese Aufgabe kein Python-Skript abgeben, sondern lediglich die Datei `expr.txt` oder `expr.md` mit Ihren Erklärungen.

Aufgabe 1.4 (Print-Rätsel; 3 Punkte; Dateien: `print1.py`, `print2.py` und `print3.py`)

Ihr Python-Skript soll in den folgenden Teilaufgaben aus jeweils einer Zeile der Form `print(...)` bestehen, wobei Sie `...` durch einen Ausdruck ersetzen.

- (a) (1 Punkt) In der Vorlesung haben Sie *Hello-World-Programme* kennengelernt:

```
>>> print("Hello World!")  
Hello World!
```

Schreiben Sie ein Python-Skript `print1.py`, das die gleiche Ausgabe wie oben produziert, dabei jedoch *kein* Leerzeichen innerhalb eines Strings enthält. Außerhalb von Strings dürfen (und sollen) Sie Leerzeichen verwenden.

- (b) (1 Punkt) Betrachten Sie folgendes Python-Programm:

```
>>> print("XYXYXY\nXYXYXY\nXYXYXY\n")
XYXYXY
XYXYXY
XYXYXY
```

Schreiben Sie ein Python-Skript `print2.py`, das die gleiche Ausgabe wie das obige Programm erzeugt. Die Buchstaben `X` und `Y` sowie die Ziffern `0` bis `9` dürfen im gesamten Python-Skript höchstens ein Mal vorkommen.

- (c) (1 Punkt) Schreiben Sie ein Python-Skript `print3.py`, das die ersten 4 der letzten 4047 Ziffern der Zahl 4047^{4047} (im Dezimalsystem) berechnet und ausgibt. Angenommen, die letzten 4047 Ziffern der Zahl wären `27180470...`, dann soll die Ausgabe wie folgt aussehen:

```
2718
```

Aufgabe 1.5 (Erfahrungen; 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 `4.5 h` steht dabei für 4 Stunden 30 Minuten.

Der Build-Server überprüft ebenfalls, ob Sie das Format korrekt angegeben haben. Prüfen Sie, ob der Build-Server mit Ihrer Abgabe zufrieden ist, so wie es im Script zum Abgabesystem gezeigt wurde.