

# Informatik I: Einführung in die Programmierung

Prof. Dr. Peter Thiemann  
Tim Schulte, Christoph-Simon Senjak  
Wintersemester 2018/2019

Universität Freiburg  
Institut für Informatik

## Übungsblatt 5

**Abgabe: Dienstag, 20.11.2018, 20:00 Uhr**

*Hinweis: Falsch benannte Dateien führen, seit dem zweiten Übungsblatt, zu einer Bewertung mit Null Punkten. Python Programme die nicht ausführbar sind werden ab dem nächsten Übungsblatt (Blatt 6) ebenfalls mit Null Punkten bewertet. Ab diesem Blatt gibt es zudem Punktabzug für PEP8-Verstöße, siehe hierzu den Python Style Guide<sup>1</sup>.*

**Aufgabe 5.1** (Fibonacci-Folge; Datei: `fibonacci.py`; Punkte: 4)

Schreiben Sie eine Funktion `fib(n: int) -> int`, welche die  $n$ -te Fibonaccizahl berechnet und zurückgibt. Die Fibonacci-Zahlen werden normalerweise wie folgt definiert:

$$\text{fib}(n) = \begin{cases} 0, & \text{falls } n = 0; \\ 1, & \text{falls } n = 1; \\ \text{fib}(n-1) + \text{fib}(n-2), & \text{sonst.} \end{cases}$$

Ähnlich wie bei der Primzahlberechnung (`primes.py`) soll sukzessive eine Liste der ersten  $n$  Fibonaccizahlen aufgebaut werden. Beginnen Sie hierzu mit der Liste `[0, 1]`, welche die nullte und erste Fibonaccizahl enthält. Berechnen sie anschließend, aus den beiden vorhandenen Werten, die dritte Fibonaccizahl. Dann die Vierte, die Fünfte, und so weiter, bis zur  $n$ -ten Fibonaccizahl. Diese soll schließlich zurückgegeben werden. Verwenden Sie in Ihrer Lösung keine `while`-Schleifen.

**Aufgabe 5.2** (Palindromische Zahlen; Datei: `palindromic.py`; Punkte: 3+3)

Eine palindromische Zahl liest sich in beide Richtungen gleich. Das größte Palindrom aus dem Produkt zweier zweistelliger Zahlen ist  $9009 = 91 \cdot 99$ . Finden Sie das größte Palindrom, das aus dem Produkt von zwei dreistelligen Zahlen besteht. Verwenden Sie in Ihrer Lösung keine `while`-Schleifen. Gehen Sie wie folgt vor:

- (a) Schreiben Sie eine Funktion `palindromic(n: int) -> bool`, welche für eine ganze Zahl  $n$  zurückgibt, ob diese palindromisch ist. Zum Beispiel:

```
>>> palindromic(9009)
True
>>> palindromic(101)
True
>>> palindromic(35)
False
```

Hinweis: Eine Sequenz `xs` kann mittels *Slicing* invertiert werden: `xs[-1::-1]`.

- (b) Schreiben Sie eine Funktion `max_palindrome() -> int`, welche die größte palindromische Zahl, die aus dem Produkt zweier dreistelliger Zahlen besteht, zurückgibt.

---

<sup>1</sup>[http://gki.informatik.uni-freiburg.de/teaching/info1\\_guide/styleguide.html#python-styleguide](http://gki.informatik.uni-freiburg.de/teaching/info1_guide/styleguide.html#python-styleguide)

**Aufgabe 5.3** (Größter gemeinsamer Teiler; Dateien: `gcd.py`, `while.txt`, Punkte: 4+4)

In dieser Aufgabe soll ein Programm zur Berechnung des größten gemeinsamen Teilers zweier oder mehrerer Zahlen implementiert werden. Verwenden Sie für die Lösung der folgenden Teilaufgaben `while`-Schleifen und begründen Sie jeweils, analog zur Vorlesung, warum diese abbrechen. Notieren Sie die Begründungen in der Datei `while.txt`.

- (a) Implementieren Sie die Funktion `gcd(m: int, n: int) -> int`, die den größten gemeinsamen Teiler zweier positiver, natürlicher Zahlen `m` und `n` berechnet. Verwenden Sie dazu die iterative Version des euklidischen Algorithmus ([https://de.wikipedia.org/wiki/Euklidischer\\_Algorithmus#Iterative\\_Variante](https://de.wikipedia.org/wiki/Euklidischer_Algorithmus#Iterative_Variante)). Es dürfen keine zusätzlichen Python-Module importiert werden.
- (b) Implementieren Sie die Funktion `cli_gcd()`, die beliebig viele positive, natürliche Zahlen mittels `input` von der Standardeingabe einliest und den größten gemeinsamen Teiler dieser Zahlen mittels `print` auf der Standardausgabe ausgibt. Ihr Programm soll so lange neue Zahleneingaben akzeptieren, bis eine 0 eingegeben wurde, die dann für die Berechnung des größten gemeinsamen Teilers nicht mehr berücksichtigt wird. Zum Beispiel:

```
>>> cli_gcd()
Calculate GCD from the input natural numbers.
Input nat. number (or 0 to quit): 294
Input nat. number (or 0 to quit): 84
Input nat. number (or 0 to quit): 126
Input nat. number (or 0 to quit): 0
GCD: 42
>>>
```

*Hinweis:* Sie dürfen bei der Bearbeitung dieser Aufgabe davon ausgehen, dass die Nutzereingaben sich stets mittels der `int`-Funktion in einen `int`-Wert konvertieren lassen.

**Aufgabe 5.4** (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet05` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, Bezug zur Vorlesung, Interessantes, benötigter Zeitaufwand, etc.).