



Übungen zur Computerphysik II
Wintersemester 2019/20

Übungsblatt 1

Ausgabe 28.10., Übungen 28.10.+4.11.+11.11., Abgabe bis 14.11.

Programmieren in Python, Kombinatorik

1. Aufgabe: "Hello, World!"

Wir wollen uns anhand kleiner Aufgaben mit der Programmierung in Python anfreunden. Eine gute Einleitung und Referenz findet sich unter

http://de.wikibooks.org/wiki/Python_unter_Linux.

- (a) Schreibe ein Programm, dass "Hello, World!" ausgibt.
- (b) Schreibe jeweils ein Programm, dass die Zahlen 1-42 ausgibt mit einer `for` und `while` Schleife.
- (c) Schreibe ein Programm, dass eine Tabelle zur Umrechnung von Celsius in Fahrenheit und Kelvin ausgibt für 0 bis 100 Grad Celsius.
- (d) Schreibe eine Funktion, die überprüft, ob eine Zahl eine Primzahl ist.
- (e) Was ist die größte Integer Zahl, die Python verarbeiten kann? Was macht Python mit größeren Zahlen?
- (f) Schreibe jeweils eine Funktion, die eine komplexe Zahl von der kartesischen in die Polarform bzw. von der Polarform in die kartesische Form umwandelt. Teste sie an einfachen Beispielen.
- (g) Schreibe eine Funktion, die den Betrag eines Vektors berechnet und diesen zusammen mit dem normierten Vektor zurück gibt. (Achtung: Nullvektor)
- (h) Schreibe eine Funktion, die das Kreuzprodukt zweier Vektoren berechnet.
- (i) Schreibe eine Funktion, die die Determinante einer 2x2 Matrix berechnet. Versuche eine Funktion zu schreiben, die die Determinante einer NxN-Matrix berechnet (z.B. rekursiv mit dem Laplaceschen Entwicklungssatz). Teste aus, wie groß die Matrix sein kann, um die Determinante noch in vernünftiger Zeit berechnen zu können.
- (j) (*) Kaffeeautomat: Schreibe ein Programm, das einen Kaffeeautomat simuliert. Er soll das Angebot anzeigen, Auswahl abfragen (mit/ohne Milch, mit/ohne Zucker,...), Geldbetrag einsammeln und Auswahl ausgeben ("Hier ihr Erdbeer Frappuccino").

2. Aufgabe: Kombinatorik

- (a) Schreibe eine rekursive und eine iterative Funktion zur Berechnung der Fakultät einer Zahl $n \in \mathbb{N}$. Teste die Funktionen und die Python-Funktion `math.factorial()` für große n .
- (b) Schreibe ein Programm, dass das Pascalsche Dreieck berechnet und ausgibt. Achtung: Die Formatierung ist nicht so einfach.
- (c) Finde heraus, wie man mit den *itertools* die Variationen und Kombinationen (jeweils mit und ohne Wiederholung) anzeigen lassen kann. Teste an einfachen Beispielen.
- (d) Die Fibonacci-Folge ist gegeben durch

$$f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2} (n > 1).$$

- i. Schreibe ein Programm, dass die ersten Zahlen der Fibonacci-Folge ausgibt.
- ii. Überprüfe, ob gilt

$$\sum_{i=0}^n f_i^2 = f_n f_{n+1}.$$

- iii. Bestimme den Grenzwert $\lim_{n \rightarrow \infty} f_{n+1}/f_n$. Wie nennt man den Wert?
 - iv. Der gleiche Wert ergibt sich durch den Kettenbruch $1 + \frac{1}{1 + \frac{1}{1 + \dots}}$. Stimmt das?
 - v. (*) Warum ist eine (simple) rekursive Implementierung der Fibonacci-Folge ungünstig? Wie sieht eine gute rekursive Implementierung aus?
- (e) Wieviele Münzen braucht man maximal und durchschnittlich, um alle Beträge zwischen 1 Cent und 99 Cent mit den Münzen 1,2,5,10,20,50 Cent zusammen zu bekommen?
 - (f) Schreibe ein Programm, das die Buchstabenhäufigkeit in einem eingelesenen Text ausgibt. Stelle das Ergebnis mit Matplotlib graphisch dar.