
Übungsblatt 8 zur Vorlesung Grundlagen der Programmierung

Übung 1

Sortieren von Einträgen in Listen nach ihrer Länge

Erstellen Sie eine einfache Liste mit mindestens 10 Einträgen. Im Folgenden ein Beispiel:

```
text_list = ["Was", "sagt", "ein", "Informatiker", "wenn",  
            "er", "auf", "die", "Welt", "kommt", "?", "Hallo", "Welt", "!"]
```

Sortieren Sie die Wörter in der Liste nach ihrer Länge. Gleich lange Wörter sortieren Sie bitte nach dem Alphabet.

Hinweis: Es ist nicht gefordert ein besonders effizientes Sortier-Verfahren zu verwenden! Fokus der Aufgabe liegt auf den Operationen mit der Liste.

Übung 2

Einfache Matrix-Operationen

Sie können mit Listen recht einfach Matrizen und Vektoren in Python darstellen. Im Folgenden zwei Beispiele:

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
vektor = [4, 2, 7]
```

Realisieren Sie folgende Operationen als einfache Funktionen in Python für beliebige Matrizen und Vektoren:

- 1) Summe von zwei Matrizen
- 2) Multiplikation einer Matrix oder eines Vektors mit einer Zahl
- 3) Multiplikation einer Matrix mit einem Vektor (Achtung: die Anzahl Zeilen des Vektors müssen hierbei der Anzahl Spalten der Matrix entsprechen)
- 4) Bestimmung der Transponierten der Matrix bzw. des Vektors

Übung 3

Das Pascal'sche Dreieck und die Fibonacci-Folge – diesmal etwas effizienter implementiert

In der letzten Woche haben wir das Pascal'sche Dreieck sowie die Fibonacci-Folge recht ineffizient implementiert, da wir die Werte für jede Rekursion immer wieder vollständig berechnet haben.

Bitte überarbeiten Sie die beiden Lösungen der letzten Woche nochmals unter Verwendung der neu kennen gelernten Datenstrukturen. Nutzen Sie für die Implementierung des Pascal'schen Dreiecks zur Zwischenspeicherung der Ergebnisse eine **zweidimensionale Liste**.

Implementieren Sie die Bestimmung des n-ten Werts der Fibonacci-Folge bitte mit Hilfe eines **Dictionary**.

Übung 4

Schlechte Übersetzungsprogramme

Schreiben Sie ein einfaches Programm, welches Texte ganz simpel Wort für Wort aus dem Deutschen ins Englische oder Französische bzw. umgekehrt übersetzt.

Folgende Korrespondenzen sollen gelten:

Deutsch	Englisch	Französisch
Brot	Bread	pain
Wein	Wine	vin
Mit	With	avec
Ich	I	je
essen	eat	mange
trinken	drink	bois
freunde	friends	amis
Und	and	et
Von	of	du
Rot	red	rouge

Beispiel-Übersetzungen:

Deutsch: Ich trinke guten roten Wein

Englisch: I drink good red wine

Deutsch: Ich esse Brot

Französisch: Je mange pain

Englisch: I eat with friends

Französisch: Je mange avec amis

Übung 5

Chatten like Kant

Um unsere Freunde zu beeindrucken, möchten wir uns gerne beim Chatten so eloquent wie die großen Philosophen ausdrücken. Praktischerweise gibt es auf mobilen Geräten häufig einen Vorschlag-Mechanismus, der während der Texteingabe automatisch *wahrscheinliche* Kandidaten für die nächsten Worte auflistet (z.B. Microsoft SwiftKey oder Apple QuickType). Anstatt der üblichen Vorschläge würden wir hier aber nun gerne Wortvorschläge auf Basis des statistischen Auftretens von Wortfolgen aus philosophischen Werken erhalten. Ziel dieser Übung ist es daher ein kleines Programm zu schreiben, welches die Wortfolgen in einem philosophischen Werk analysiert und auf dieser Basis Wortkandidaten vorschlägt.

Zur Vereinfachung der Entwicklung habe ich Ihnen einen Programmrahmen sowie den Text des Werks „Kritik der reinen Vernunft“ von Immanuel Kant auf Stud.IP hinterlegt. Legen Sie bitte die Textdatei sowie das Python-Skript in denselben Ordner. Das Programm liest den Text ein und speichert ihn in einer Liste. Weiterhin habe ich eine Funktion ergänzt, mit der Sie einen Dictionary nach dem Wert (nicht nach seinem Schlüssel) sortieren lassen können.

Lösen Sie nun folgende Aufgaben:

- 1) Bestimmen Sie die 10 am häufigsten vorkommenden Worte mit mehr als zehn Zeichen im Werk „Kritik der reinen Vernunft“ und geben diese aus.

- 2) Bitten Sie den Nutzer ein Wort zu nennen und schlagen ihm die fünf Kandidaten des am wahrscheinlichsten darauf folgenden Worts vor. Die Kandidaten ermitteln Sie bitte durch Analyse des Texts.
- 3) Bitten Sie den Nutzer nun zwei Wörter zu nennen und schlagen ihm die fünf Kandidaten des am wahrscheinlichsten darauf folgenden Worts vor.

Im Folgenden sehen Sie eine Beispiel-Ausgabe des Programms:

```
*****
Chatten like Kant
*****

Die Hitliste der am häufigsten auftretenden Wörter mit mehr als 10 Zeichen im Werk:
erscheinungen
bedingungen
gegenstände
empirischen
möglichkeit
transzendentalen
erscheinung
vorstellung
transzendente

Geben Sie Ihr erstes Wort ein: transzendentalen
Vorschlag 1: transzendentalen ideen
Vorschlag 2: transzendentalen logik
Vorschlag 3: transzendentalen dialektik
Vorschlag 4: transzendentalen ästhetik
Vorschlag 5: transzendentalen analytik

Wählen Sie Vorschlag 1-5: 5

Vorschlag 1: transzendentalen analytik zweites
Vorschlag 2: transzendentalen analytik erstes
Vorschlag 3: transzendentalen analytik welche
Vorschlag 4: transzendentalen analytik fest
Vorschlag 5: transzendentalen analytik von
```

Abb. 1: Beispiel-Ausgabe des Vorschlagsprogramms

Lösungshinweis: am einfachsten ist es, wenn Sie sich in einer Struktur für jede mögliche Kombination aus zwei bzw. drei aufeinanderfolgenden Worten das Auftreten speichern. Nutzen Sie diese Struktur, um den wahrscheinlichsten Nachfolger für ein Wort bzw. Wortpaar zu bestimmen. Nutzen Sie hierfür Listen, Sets und/oder Dictionaries.