

Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel
Tim Schulte, Thorsten Engesser
Wintersemester 2017/2018

Universität Freiburg
Institut für Informatik

Übungsblatt 6

Abgabe: Freitag, 1. Dezember 2017, 20:00 Uhr

WICHTIGE HINWEISE: Zur Bearbeitung der Übungsaufgaben legen Sie bitte ein neues Unterverzeichnis `sheet06` im Wurzelverzeichnis Ihrer Arbeitskopie des SVN-Repositories an. Ihre Lösungen werden dann in Dateien in diesem Unterverzeichnis erwartet. Beachten Sie bitte bei allen Aufgaben die *Hinweise zur Bearbeitung der Übungsaufgaben* unter der folgenden URL:

<http://gki.informatik.uni-freiburg.de/teaching/ws1718/info1/guide/hinweise.html>

Bewertet wird bei allen Aufgaben die letzte Version, die zur Deadline des Übungsblattes auf dem SVN-Server eingereicht ist.

Aufgabe 6.1 (Blindtext-Generator; Datei: `blindtext.py`; Punkte: 4+4+2)

Blindtexte wie „*Lorem ipsum dolor sit amet, consectetur adipiscing elit, ...*“ werden als Platzhalter im Layout verwendet, um einen Eindruck vom fertigen Dokument zu erhalten, oder zur Demonstration der Gestalt verschiedener Schrifttypen. In dieser Aufgabe sollen Sie einen *Blindtext-Generator* implementieren. Verwenden Sie hierzu das auf der Kurs-Website zur Verfügung gestellte Template. Die Idee ist es, aus einem Eingabestring zunächst eine Transitionstabelle zu erzeugen. Diese enthält für jeweils zwei Worte w_1 und w_2 die relative Häufigkeit, mit der w_2 auf w_1 folgt. Danach können Blindtexte, mit den entsprechenden Wahrscheinlichkeiten randomisiert, Wort für Wort aufgebaut werden. Ein *Wort* besteht hierbei entweder aus einem einzelnen Satzzeichen (, . ; ? !) oder einer endlichen Folge von Buchstaben des deutschen Alphabets.

- (a) Implementieren Sie eine Funktion `transitiontable(s)`, welche einen String `s` als Argument erhält und die *Transitionstabelle* für `s` zurückgibt. Wir repräsentieren diese als Dictionary, welches jedem Wort `w` aus `s` ein weiteres Dictionary zuordnet. Dieses enthält wiederum die relativen Worthäufigkeiten für alle Wörter, die in `s` auf `w` folgen (alle anderen Wörter haben die relative Häufigkeit 0 und müssen daher nicht explizit im Dictionary enthalten sein). Eine Ausnahme hierbei ist das letzte Wort, dessen Nachfolgewort wir als das erste Wort in `s` definieren.

Betrachten Sie hierzu das folgende Beispiel:

```
>>> s = """Wenn hinter Fliegen Fliegen fliegen,  
...     fliegen Fliegen Fliegen nach."""  
>>> tt = transitiontable(s)  
>>> from pprint import pprint  
>>> pprint(tt)  
{',': {'fliegen': 1.0},  
'.': {'Wenn': 1.0},  
'Fliegen': {'Fliegen': 0.5, 'fliegen': 0.25, 'nach': 0.25},  
'Wenn': {'hinter': 1.0},  
'fliegen': {'',': 0.5, 'Fliegen': 0.5},  
'hinter': {'Fliegen': 1.0},  
'nach': {'.': 1.0}}
```

Hinweis: Um Worte im Sinne der Spezifikation aus einem String zu extrahieren, können Sie die bereitgestellte Funktion `next_word` (eine leicht modifizierte Version der gleichnamigen Funktion aus Aufgabe 5.3) verwenden.

- (b) Implementieren Sie eine Funktion `blindtext(tt, n)`, welche für eine gegebene Transitionstabelle `tt` einen Blindtext generiert und diesen als String zurückgibt.

Als erstes Wort wird zunächst der Nachfolger eines der Worte „.“, „?“ oder „!“ bestimmt (wir gehen davon aus, dass mindestens eines dieser Worte in der Transitionstabelle enthalten sein muss). Danach wird sukzessive zu dem zuletzt ausgewählten Wort `w` eines seiner Nachfolgeworte aus `tt[w]` zufällig ausgewählt, bis ein Satz mit genau n Wörtern erzeugt wurde. Benutzen Sie hierzu die Funktion `random()` aus dem Modul `random` (zur Generierung einer Zufallszahl zwischen 0 und 1), um die Nachfolgewörter jeweils nach ihrer Wahrscheinlichkeitsverteilung auszuwählen. Siehe folgendes Beispiel:

```
>>> blindtext(tt, 10)
'Wenn hinter Fliegen fliegen, fliegen Fliegen Fliegen Fliegen nach'
```

In unserem Beispiel ist das erste Wort zwangsläufig „Wenn“ da es in der Transitionstabelle als einziges auf einen „.“ folgt. Anschließend folgt immer das Wort „hinter“ gefolgt von „Fliegen“. Auf „Fliegen“ folgt mit einer Wahrscheinlichkeit von 0.5 erneut „Fliegen“, mit 0.25 „fliegen“ und mit 0.25 „nach“, etc.

- (c) Ergänzen Sie das Template, sodass bei jedem Start des Programms ein Blindtext erzeugt und ausgegeben wird. Verwenden Sie hierzu die in (a) und (b) implementierten Funktionen. Ersetzen Sie zudem den Ausgangstext `INPUTTEXT` durch einen Text Ihrer Wahl¹, welcher mit einem terminierendes Satzzeichen (.?!) endet.

Aufgabe 6.2 (Mengen-Operationen; Punkte: 3+3+2; Datei: `sets.py`)

In dieser Aufgabe geht es um die Berechnung von Teilmengen und Potenzmengen. Die Teilaufgaben dürfen in beliebiger Reihenfolge bearbeitet werden, wobei die zuerst definierte Funktion in der Definition der anderen verwendet werden darf.

- (a) Definieren Sie eine Funktion `subsets(s, k)`, die angewendet auf eine Menge `s` und eine natürliche Zahl $k \geq 0$, die Menge gerade jener Teilmengen von `s` zurückgibt, die genau k Elemente enthalten.
- (b) Definieren Sie eine Funktion `powerset(s)`, die für eine beliebige Menge `s` die Potenzmenge berechnet und zurückgibt.
- (c) Schreiben Sie für die beiden Funktionen in (a) und (b) jeweils eine Testfunktionen (`test_subsets()` und `test_powerset()`) mit mindestens vier Testfällen.

Hinweis: In dieser Aufgabe sollen ausschließlich Mengen (keine Listen!) benutzt werden. Das Importieren von anderen Modulen ist bei der Bearbeitung beider Teilaufgaben nicht erlaubt. Achten Sie ferner darauf, dass die an diese Funktionen übergebenen Mengen nach Ausführung der Funktion noch dieselben Mengen sind! Denken Sie an die Dokumentation ihrer Funktionen mittels docstrings.

¹Eine Auswahl gemeinfreier Texte finden Sie z.B. auf <https://www.gutenberg.org/>

Aufgabe 6.3 (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet06` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, benötigter Zeitaufwand nach Teilaufgabe, Bezug zur Vorlesung, Interessantes, etc.).

Hinweis: Denken Sie daran in der kommenden Woche (vom 27.11. bis zum 1.12.) in Ihre Übungsgruppe zu kommen um ihren micro:bit abzuholen.