

Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel
Dr. Christian Becker-Asano, Dr. Stefan Wöflf
Wintersemester 2014/2015

Universität Freiburg
Institut für Informatik

Übungsblatt 11

Abgabe: Freitag, 23. Januar 2015, 18:00 Uhr

WICHTIGE HINWEISE: Zur Bearbeitung der Übungsaufgaben legen Sie bitte ein neues Unterverzeichnis `sheet11` im Wurzelverzeichnis Ihrer Arbeitskopie des SVN-Repositories an. Ihre Lösungen werden dann entsprechend dem ersten Übungsblatt 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/ws1415/info1/hinweise_uebungen.txt

Überprüfen Sie, dass Sie alle Lösungen ins Repository hochgeladen haben. Bewertet wird bei allen Aufgaben die letzte Version, die zur Deadline des Übungsblattes auf dem SVN-Server eingereicht ist.

Aufgabe 11.1 (`timeit`, Punkte: 3+3, Dateien: `test_sqrt.py`, `test_dict.py`)

Verwenden Sie das `timeit`-Modul, um die folgenden Fragestellungen zu beantworten. Verwenden Sie eines der beiden Templates, die Sie von der Webseite herunterladen, umbenennen und für die jeweilige Fragestellung adaptieren können. Jeder Test soll 3-mal wiederholt werden. Die Anzahl der Durchläufe (Loops) in jedem Testlauf legen Sie dabei so fest, dass die Gesamtlaufzeit pro Testlauf etwa zwischen 1 und 5 sec. liegt. Gestalten Sie die Ausgabe so, dass ersichtlich wird, für welchen Test die Laufzeiten gemessen wurden.

- (a) Sie wollen sehr häufig eine Funktion `spam_sqrt` verwenden, die die Funktion `sqrt` aus dem Modul `math` bis zu 10000 mal aufruft. Welche Implementierung ist besser: (1) eine, die das Modul `math` importiert und in der Definition von `spam_sqrt` die Wurzelfunktion über den Namen `math.sqrt` aufruft oder (2) eine, die aus dem Modul `math` die Funktion `sqrt` importiert und in der Definition von `spam_sqrt` die Wurzelfunktion über den Namen `sqrt` aufruft?
- (b) Gegeben sei ein Dictionary `dct` mit etwa 1000 Einträgen und eine Liste von *möglichen* Keys des Dictionary. Sie wollen für jeden Eintrag `k` der Liste, sofern er ein Schlüssel von `dct` ist, auf den Wert `dct[k]` zugreifen. Welche Implementierung ist besser: (1) eine, die über die Listenelemente iteriert und vor dem Zugriff auf `dct[k]` abfragt, ob `k` ein Key von `dct` ist, und nur dann auf `dct[k]` zugreift; oder (2) eine, die den Zugriff auf `dct[k]` mittels einer `try: ... except:-`Klausel „absichert“.

Testen Sie folgende Szenarien: etwa die Hälfte der Listeneinträge sind Keys des Dictionary, kein Listeneintrag ist ein Key, alle Einträge sind Keys.

Aufgabe 11.2 (`urllib` und `dict`; Punkte: 2+2; Datei: `gki_lectures.py`)

Auf der Webseite <http://www.informatik.uni-freiburg.de/~ki/teaching/index.html> sind alle Lehrveranstaltungen gelistet, die im laufenden Semester vom Lehrstuhl *Grundlagen der Künstlichen Intelligenz* angeboten werden.

- (a) Schreiben Sie eine Funktion `parse_teaching_website()`, die aus der o.a. Webseite ein `dictionary` generiert, welches als `keys` die jeweiligen Titel der dort eingetragenen Lehrveranstaltungen enthält. Als Werte eines jeden Eintrags im `dictionary` generiert Ihre Funktion Listen bestehend aus (1) dem Veranstaltungstyp (ein String), (2) einer Liste der Dozenten (jeweils ein String) und (3) der Webseite der Veranstaltung (ein String). Dieses `dictionary` wird am Ende mittels `return`-Anweisung an den Aufrufer der Funktion zurückgegeben.
- (b) Schreiben Sie nun eine Funktion `write_dict_to_file(dictionary)`, die den Inhalt der Rückgabe der Funktion `parse_teaching_website()` in eine neu anzulegende Datei `gki_lehre.txt` schreibt. Dabei beachten Sie bitte für die Formatierung des Textes folgendes Beispiel:

```
Kurs: Handlungsplanung
Typ: Lecture
Dozenten:
- Prof. Dr. Bernhard Nebel
- Dr. Robert Mattmüller
Webseite: http://gki.informatik.uni-freiburg.de/teaching/ws1415/aip/
```

Benutzen Sie in dieser Aufgabe das Modul `urllib` und geeignete, reguläre Ausdrücke.

Aufgabe 11.3 (`urllib` und `time`; Punkte 5+3; Datei: `lect_files.py`)

- (a) Schreiben Sie unter Verwendung des `urllib`-Pakets ein Python-Programm, das alle PDF- und Python-Dateien, die von der Webseite der Vorlesung

```
http://www.informatik.uni-freiburg.de/~ki/teaching/ws1415/info1/lecture.html
```

verlinkt werden, herunterlädt und in ein Unterverzeichnis `info1_lecture` abspeichert.

Das Ausführen des Programms soll nicht mit einem Fehler abbrechen, wenn auf eine der verlinkten Dateien nicht zugegriffen werden kann. Behandeln Sie also etwaige Fehler entsprechend.

Während der Ausführung des Programms soll auf der Konsole nur ausgegeben werden, welche Dateien heruntergeladen wurden oder ob ein Download nicht möglich war.

Alle Dateien sollen direkt im Verzeichnis `info1_lecture`, nicht in etwaigen Unterverzeichnissen hiervon abgespeichert sein.

Hinweis: Bitte laden Sie auf diesem Weg **nicht** die Vorlesungsaufzeichnungen herunter! Beachten Sie ferner: Der Inhalt der oben angegebenen URL ist nicht

statisch. Während der kommenden Woche werden weitere Dokumente verlinkt werden. Am kommenden Freitag werden wir ab 12:00 Uhr jedoch bis zur Abgabe des Übungsblattes keine Veränderung der Seite vornehmen.

- (b) Ergänzen Sie in einem zweiten Schritt Ihr Programm um ein boolesche Variable `verbose`. Wenn diese auf `True` gesetzt ist, soll nach dem Download jeder einzelnen Datei auf der Konsole ausgegeben werden, wie groß die heruntergeladene Datei ist und wie lange der Download gebraucht hat. Die Ausgabe sollte etwas wie folgt aussehen:

```
infoI11-handout.pdf    276KiB  276.0KiB/s  00:01
```

Am Ende des Downloads der Dokumente soll eine kurze Statistik ausgegeben werden, etwa so:

```
Downloaded 97 of 99 referenced documents  
(in total: 29998.6162 KiB in 19.76 sec)
```

Aufgabe 11.4 (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet11` 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.).