Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel — Universität Freiburg
Dr. Christian Becker-Asano, Dr. Stefan Wölfl — Institut für Informatik
Wintersemester 2013/2014

Übungsblatt 10

Abgabe: Freitag, 24. Januar 2014, 18:00 Uhr

Bei diesem Übungsblatt erwarten wir alle Lösungen zu den Aufgaben in der jeweils angegebenen Python-Datei im Unterverzeichnis `sheet10`. Beachten Sie dabei die bekannten Formatierungshinweise für Python-Dateien und vergessen Sie nicht, alle von Ihnen verwendeten Dateien ins SVN-Repository hochzuladen!

Aufgabe 10.1 (`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 10.2 (Tupel vs. Listen, Punkte: 4, Dateien: `test_list_tuple.py`, `ex10-2.txt`)

Tuples are faster than lists. Diese Feststellung findet sich in einigen Online-Dokumentationen zu Python, aber stimmt das? Um diese Frage zu analysieren, interessiert uns:

(a) Ist der Zugriff auf einen Listeneintrag `x = lst[k]` langsamer als der Zugriff auf einen Tupeleintrag `x = tpl[k]`?

(b) Ist die Erzeugung eines Listen-Slices `x = lst[k:n]` langsamer als die Erzeugung eines Tupel-Slices `x = tpl[k:n]`?

(c) Ist die Erzeugung einer Liste durch `lst = [x1, x2, ..., xN]` mittels Variablen `x1, x2, ..., xN` (den Variablen müssen natürlich vorher Werte zugewiesen sein) langsamer als die Erzeugung eines entsprechenden Tupel `tpl = (x1, x2, ..., xN)`?

Diskutieren Sie obige Fragestellung anhand eigener Testergebnisse. Verwenden Sie dazu das `timeit`-Modul und benutzen Sie eines der beiden bereitgestellten Templates, das entsprechend adaptiert werden muss. Gestalten Sie die Ausgabe so, dass ersichtlich wird, für welche der drei Fragen (a)–(c) und welchen Datentyp Ihre Zeitmessungen gelten.

Aufgabe 10.3 (`urllib` und `time`; Punkte 5+3; Datei: `gki_lectures.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/ws1314/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 abgspeichert sein.

Hinweis: *Bitte laden Sie auf diesem Weg *nicht* die Vorlesungsaufzeichnungen herunter! Beachten Sie ferner: Der Inhalt der oben angegeben URL ist nicht statisch. Während der kommenden Woche werden*

weitere Dokumente verlinkt werden. Am kommendem 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 boolsche 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 ausssehen:

```
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 10.4 (Erfahrungen; 2 Punkte)

Legen Sie im Unterverzeichnis `sheet10` eine Textdatei `erfahrung.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, benötigter Zeitaufwand nach Teilaufgabe, Bezug zur Vorlesung, Interessantes, etc.).