Institut für Informatik
Dr. Tobias Neckel,
Alfredo Parra
Christoph Kowitz

Technische Universität München

Einführung in die wissenschaftliche Programmierung

Übungsblatt 7

1.) Telefonbuch (Hands-On)

Erweitern Sie das Telefonbuch von Blatt 4, sodass fehlerhafte Eingaben erkannt werden, d.h.,

i) ein Name darf nur Klein-, Großbuchstaben oder Leerzeichen beinhalten, und

ii) eine Telefonnummer darf nur Zahlen oder Leerzeichen beinhalten, darf aber zusätzlich mit „+“ anfangen.

Benutzen Sie dafür reguläre Ausdrücke, insbesondere die Funktion `re.match`.

2.) E-Mail Crawler (Hausaufgabe - Fortgeschritten)

In dieser Aufgabe soll ein Programm zum Auslesen von E-Mail Adressen auf Webseiten entwickelt werden. Das Programm liest eine Webseite ein und extrahiert sämtliche Links und E-Mail Adressen auf der Webseite. Die E-Mail Adressen werden gespeichert (und ausgegeben). Dann wird einem Link von der Webseite gefolgt um dort ebenfalls Links und E-Mail-Adressen zu extrahieren.

Mit dem Python Modul `urllib` können sehr einfach Webseiten als String ausgelesen werden. Dazu müssen Sie nur

```
site = urllib.urlopen('http://www.google.com').read()
```

aufrufen. Der Quelltext der Webseite steht dann in der Variable `site`.

E-Mail Crawler

Implementieren Sie nun einen E-Mail Crawler:

i) Schreiben Sie eine Funktion, welche aus einem String E-Mail Adressen extrahiert. Im (HTML-)Quelltext einer Webseite treten E-Mail Adressen entweder einfach als Wort (durch Leerzeichen begrenzt) auf, oder aber in der Form

```
<a href="mailto:example@test.com">Herr Example</a>
```

Benutzen Sie reguläre Ausdrücke und die Funktion `re.finditer`.

ii) Schreiben Sie eine Funktion, welche Webadressen (z.B. http://de.wikipedia.com, http://www.tum.de/) aus einem String extrahiert. Wieder treten Adressen in zwei Formen auf. Entweder einfach durch Leerzeichen begrenzt, oder in der Form:

```
<a href="http://www.facebook.com">Facebook</a>
```

iii) Starten Sie mit einer beliebigen Webseite und lesen Sie den Quelltext mit der `urllib` Bibliothek ein. Extrahieren Sie dann die E-Mail Adressen und speichern Sie diese in einer Menge (`set`). Extrahieren Sie die Links und speichern Sie diese in einer Liste. Entnehmen Sie dann ein Element der Liste mit Links und starten Sie den Vorgang erneut. Warum ist ein `set` der beste Datencontainer für unseren Zweck?

3.) Primzahlen (Hausaufgabe - Fortgeschritten)

Im folgenden Code werden unendlich viele Primzahlen ausgegeben:

```
import re

def isPrime(p):
    m = re.search(r'^1?$|^(11+?)\1+$', p)
    return not m

i = 2
counter = 0
while(counter < 100):
    if(isPrime('1' * i)):
        counter += 1
        print counter, i
    i += 1
```

Erklären Sie die Funktionsweise der Funktion `isPrime`.
Schreiben Sie das Programm so um, dass nur die ersten 100 Primzahlen ausgegeben werden.