Programmierkurs Python I – WS 09/10

Übung 8

---

1 Flache Listen (3 Punkte)

Schreibe eine Funktion, die für verschachtelte Listen eine entsprechende flache Liste berechnet. Verschachtelte Listen können (verschachtelte) Listen als Elemente haben. Die Elemente der flachen Liste sollen dieselben atomaren Elemente der verschachtelten Liste sein (in derselben Reihenfolge). Die Implementierung kann – muss aber nicht – Iteratoren bzw. Generatoren verwenden.

```
>>> flatten([1, [2, [3, 4], 5], [[6]]])
[1, 2, 3, 4, 5, 6]
```

Hinweis: Mit `isinstance(x, list)` kann man testen, ob `x` eine Liste ist.

2 Köpfe (2 Punkte)

Schreibe einen Iterator (oder Generator) `head`, der über die ersten n Elemente eines anderen Iterators iteriert. (Sollte der Iterator weniger als n Elemente haben, sollen einfach alle Elemente des Iterators aufgezählt werden.) Das Programm soll auch für „unendliche“ Iteratoren terminieren.

```
class Numbers:
    def __init__(self):
        self.n = -1
    def __iter__(self):
        return self
    def next(self)
        self.n += 1
        return self.n

>>> list(head(10, Numbers()))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

3 Cycle als Generator (1 + 1 + 1 Punkte)

Schreibe einen Generator `Cycle`, der die Elemente eines Sammelobjekts in unendlicher Folge aufzählt.

```
>>> list(head(10, Cycle([1,2,3])))
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1]
```

Bonus #1: Schreibe das Programm so, dass es auch mit Mengen funktioniert. Die Reihenfolge der Elemente ist beliebig.

Bonus #2: Schreibe das Programm so, dass es auch Iteratoren als Eingabe akzeptiert.

4 Binäre Bäume (Bonus, 3 Punkte)

Ändere die Beispiel-Implementierung von Binären Bäumen so, dass kein Generator sondern ein Iterator verwendet wird.

Abgabe bis Donnerstag, 21.01.2010, 14:00 Uhr