

## 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 13

**Abgabe: Freitag, 06. Februar 2015, 18:00 Uhr**

**WICHTIGE HINWEISE:** Zur Bearbeitung der Übungsaufgaben legen Sie bitte ein neues Unterverzeichnis `sheet13` 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](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 13.1** (Skopus von Variablen; Punkte: 4+2+1; Dateien: `scope.txt`, `scope.py`)

Betrachten Sie den folgenden Quellcode:

```
foo, bar = "global", "global"

def decorator(f):
    global bar
    foo, bar = 2*('nonlocal',)
    def wrapper(*args, **kwargs):
        nonlocal foo
        foo += 'local'
        bar = 'local'
        print(10, foo, bar)
        return f(*args, **kwargs)
    print(12, foo, bar)
    return wrapper

@decorator
def add(x, y):
    foo = 'call'
    global foo
    return x+y

print(21, foo, bar)
print(22, add(foo, bar))
print(23, foo, bar)
print(24, add(foo, bar))
```

- (a) Spekulieren Sie, bevor Sie den Code ausführen, welche Ausgaben auf der Konsole erscheinen werden (Reihenfolge der Ausgaben und Werte der Variablen). Führen Sie dann den Code aus und geben Sie anschließend für jede der Ausgabe eine kurze Begründung an.
- (b) Was passiert, wenn Sie das `global`-Statement innerhalb der Funktion `add` auskommentieren? Was passiert, wenn Sie das `nonlocal`-Statement innerhalb der Funktion `wrapper` auskommentieren?
- (c) Wieso erhalten Sie eine `SyntaxWarning` und wie könnte man den Quellcode umschreiben, um diese Warnung nicht mehr zu erhalten? Nutzen Sie das WWW, um Hintergrundinformationen zu dieser Warnung zu erhalten.

**Aufgabe 13.2** (Comprehension; Punkte: 3+3; Datei: `comprehension.py`)

- (a) Berechnen Sie mit Hilfe von List-Comprehensions eine Liste mit sechzehn Fließkommazahlen. Diese vier mal vier Zahlen berechnen sich als Folge:

$$a_z := \left(\frac{1 + \sqrt{5}}{2}\right)^z \cdot a_0 \quad (1)$$

mit  $z, a_0 \in \{1, 2, 3, 4\}$  (Goldene Zahlenfolge). Dabei soll für jedes  $a_0$  über  $z$  iteriert werden, also alle vier Zahlen für  $a_0 = 1$  vor den vier Zahlen für  $a_0 = 2$  in der Rückgabeliste erscheinen, dann für  $a_0 = 3$  und schließlich für  $a_0 = 4$ .

- (b) Importieren Sie die Variable `names`, die in der bereitgestellten Datei `names_raw.py` definiert wird und einen unformatierten String mit Vor- und Nachnamen in eckigen Klammern enthält<sup>1</sup>. Nutzen Sie List-Comprehensions (zusammen mit weiteren Ihnen aus der Vorlesung bekannten Funktionen), um eine Liste mit 2-Tupeln zu erzeugen. Jedes dieser 2-Tupel soll aus dem Vor- und dem Nachnamen einer Person bestehen. Ihr Code sollte exakt eine Zeile lang sein (und keine Semikolons enthalten).

**Aufgabe 13.3** (Generatoren, Bellsches Dreieck; 3+2 Punkte; Dateien: `bell.py`, `bell.txt`)

Ergründen Sie auf folgender Wikipedia-Seite die Definition des “Bellschen Dreiecks”: [http://de.wikipedia.org/wiki/Bellsche\\_Zahl#Bellsches\\_Dreieck](http://de.wikipedia.org/wiki/Bellsche_Zahl#Bellsches_Dreieck)

- (a) Schreiben Sie eine Generatorfunktion `bell()`, mit deren Hilfe man über die (unendliche Anzahl von) Zeilen des “Bellschen Dreiecks” iterieren kann. Jede Zeile soll als Liste zurückgegeben werden beginnend mit `[1]`, `[1, 2]`, usw.
- (b) Schreiben Sie eine Funktion `iter_bell(n)`, die die Generatorfunktion aus Aufgabenteil a) benutzt, um die ersten  $n$  Bellschen Zahlen als Liste zurückzugeben.

**Aufgabe 13.4** (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet13` 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.).

<sup>1</sup>Vergleiche: [http://www.informatik.uni-freiburg.de/~ki/publications/by\\_author.html](http://www.informatik.uni-freiburg.de/~ki/publications/by_author.html)