

## Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel  
Tim Schulte, Thorsten Engesser  
Wintersemester 2017/2018

Universität Freiburg  
Institut für Informatik

### Übungsblatt 14

**Abgabe: keine**

**WICHTIGER HINWEIS:** Bei diesem Übungsblatt handelt es sich um ein freiwilliges Übungsblatt, das weder korrigiert, noch in den Tutoraten besprochen wird. Die Bearbeitung des Blattes wird trotzdem empfohlen, da der Inhalt (Funktionale Programmierung, Iteratoren und Generatoren) klausurrelevant ist. Falls Sie Hilfe benötigen, können Sie diese im Betreuten Programmieren in der letzten Vorlesungswoche bekommen.

#### Aufgabe 14.1 (Dekoratoren)

Zum Testen von Programmen ist es häufig sinnvoll sicherzustellen, dass die an eine Funktion zur Laufzeit übergebenen Argumente vom richtigen Typ sind oder andere Bedingungen erfüllen. Eine Möglichkeit, dies zu realisieren, ist mittels Dekoratoren. Betrachten Sie dazu das folgende Beispiel und implementieren (und testen) Sie den zugehörigen Dekorator.

```
>>> @check_argument(lambda x: isinstance(x, int) and x >= 0)
... def factorial(n):
...     return 1 if n == 0 else n * factorial(n-1)
...
>>> factorial(2)
2
>>> factorial(2.0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 7, in helper
Exception: Failed argument check: factorial with arg 2.0
```

#### Aufgabe 14.2 (Spaltennamen)

- In Python unterscheidet man zwischen *Iteratoren*, *iterierbaren Objekten* und *Generatoren*. Erläutern Sie diese Begriffe jeweils kurz.
- Implementieren Sie einen Generator `colnames`, der sukzessive alle Strings der folgenden unendlichen Folge erzeugt, die z.B. zur Spaltenbezeichnung in Tabellenkalkulationsprogrammen Verwendung findet:

A, B, ..., Z, AA, AB, ..., AZ, BA, ..., ZZ, AAA, ...

Implementieren Sie zum Testen eine Funktion `test_colnames(k)`, die die ersten  $k$  Folgenglieder auf der Konsole ausgibt.

**Aufgabe 14.3** (Comprehension)

- (a) Ein pythagoreisches Tripel  $(x, y, z)$  besteht aus drei natürlichen Zahlen  $x$ ,  $y$  und  $z$ , so dass  $x^2 + y^2 = z^2$ . Berechnen Sie mit Hilfe von List-Comprehensions eine Liste mit allen pythagoreischen Tripel, so dass  $x \leq 100$ ,  $y \leq 100$  und  $z \leq 100$ .
- (b) Definieren Sie eine Funktion `powerset(s)`, die für eine beliebige Menge  $s$  die Potenzmenge berechnet und zurückgibt. Ihre Funktionsdefinition soll außer einer `return`-Anweisung keine weiteren Zeilen enthalten. Innerhalb des `return`-Anweisung dürfen allerdings Mengenoperationen sowie ein oder mehrere Set-Comprehensions benutzt werden.