

# Informatik I: Einführung in die Programmierung

Prof. Dr. Peter Thiemann  
Tim Schulte, Christoph-Simon Senjak  
Wintersemester 2018/2019

Universität Freiburg  
Institut für Informatik

## Übungsblatt 4

**Abgabe: Dienstag, 13.11.2018, 20:00 Uhr**

### Aufgabe 4.1 (Konkatenation; Datei: `concatenated.py`; Punkte: 4)

Implementieren Sie eine Funktion `concatenated(xs: list) -> list`, welche eine Liste von Listen als Argument erhält, diese zu einer einzigen Liste zusammenfügt, und die resultierende Liste zurückgibt. Zum Beispiel:

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

### Aufgabe 4.2 (Filter; Datei: `overcooked.py`; Punkte: 6)

Gegeben sei eine Liste `recipes` mit Rezepten, wie folgt:

```
recipes = [("Sushi", ["Fisch", "Reis", "Nori"]),
           ("Sashimi", ["Fisch", "Reis"]),
           ("Pfannkuchen", ["Mehl", "Ei", "Milch"]),
           ("Burger", ["Brötchen", "Rind"]),
           ("Burger TS", ["Brötchen", "Rind", "Tomate", "Salat"]),
           ("Cheese Burger", ["Brötchen", "Rind", "Tomate", "Käse"]),
           ("Gemischter Salat", ["Salat", "Tomate", "Gurke"])]
```

Schreiben Sie eine Funktion `cookable(xs: list) -> list`, welche eine Liste `xs` an Zutaten (jede Zutat ist ein String) als Argument erhält und alle Rezepte aus `recipes` zurückgibt, welche mit den gegebenen Zutaten kochbar sind. Zum Beispiel:

```
>>> cookable(["Brötchen", "Tomate", "Gurke", "Salat", "Rind", "Brötchen"])
["Burger", "Burger TS", "Gemischter Salat"]
>>> cookable(["Fisch", "Reis", "Tomate"])
["Sashimi"]
```

### Aufgabe 4.3 (Primzahlen; Datei: `primes.py`; Punkte: 8)

Primzahlen sind natürliche Zahlen, die durch genau zwei Zahlen teilbar sind: durch 1 und durch sich selbst. Insbesondere ist 2 die kleinste Primzahl, eine größte Primzahl existiert nicht. Implementieren Sie eine Funktion `primes(n: int) -> list`, welche sukzessiv alle Primzahlen kleiner oder gleich  $n$  berechnet und diese in aufsteigender Reihenfolge als Liste zurückgibt. Implementieren Sie dazu die folgende Idee: Um zu überprüfen, ob eine Zahl  $n$  prim ist, reicht es, diese auf Teilbarkeit durch alle zuvor erzeugten Primzahlen  $\leq n$  zu überprüfen. Bereits erzeugte Primzahlen können (und sollten) in einer Liste zwischengespeichert werden. Sie können Ihre Funktion z.B. wie folgt testen:

```
>>> primes(1) == []  
True  
>>> primes(3) == [2, 3]  
True  
>>> primes(20) == [2, 3, 5, 7, 11, 13, 17, 19]  
True
```

**Aufgabe 4.4** (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet04` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, Bezug zur Vorlesung, Interessantes, benötigter Zeitaufwand, etc.).