

## Aufgabenblatt 4

### Musterlösung

## Vorlesung Verteilte Systeme

### Wintersemester 2025/26

#### Aufgabe 1: Simulator für Verteilte Systeme

```
import sys
import json

class Message:
    def __init__(self, src='', dst='', typ='', msg=None, **kwargs):
        if msg:
            self.__dict__ = json.loads(msg)
        else:
            self.src = src
            self.dest = dst
            self.body = { 'type': typ }
            for key, value in kwargs.items():
                self.body[key] = value
    def __getitem__(self, key):
        return self.body[key]
    def __str__(self):
        return json.dumps(self.__dict__)

def receive():
    msg = sys.stdin.readline()
    return Message(msg=msg) if msg else None

def send(msg):
    print(msg, flush=True)

me = receive()['name']
while msg := receive():
    message = msg['message']
    reply = 'World' if msg['message'] == 'Hello' else msg['message']
    send(Message(me, msg.src, 'echo_ok', in_reply_to=msg['id'], message=reply))
```

#### Aufgabe 3: Programmierung: Generische Proxy-Objekte

Die Lösung dieser Aufgabe finden Sie im Archiv [104Files.zip](#)<sup>1</sup> auf der Vorlesungswebseite.

#### Aufgabe 4: Parameterübergabe

Wird *call-by-reference* verwendet, wird `incr` ein Zeiger auf `i` übergeben. Damit wird `i` zweimal inkrementiert, das Endergebnis ist also in diesem Fall 2.

Bei *call-by-value* wird der Wert direkt übergeben und keine Referenz auf `i`. Daher bleibt `i` beim Aufrufer unverändert, also 0.

---

<sup>1</sup><http://www.bs.informatik.uni-siegen.de/web/wismueller/v1/vs/104Files.zip>

*Call-by-copy/result* bedeutet: Kopieren der Parameterwerte beim Aufruf der Prozedur, Zurückkopieren und Überschreiben der Aufrufparameter bei Beendigung der Prozedur. In diesem Fall wird  $i$  als zweimal als Wert übergeben (also jeweils 0) und jeweils inkrementiert, so dass beide Werte nun 1 sind. Wenn sie zurückkopiert werden, überschreibt die zweite Kopie die erste und der endgültige Wert ist nun ebenfalls 1.