

LU01.A13 - Dasselbe Problem in drei Paradigmen

Lösen Sie ein und dieselbe Aufgabe dreimal – prozedural, objektorientiert und funktional – und vergleichen Sie die drei Lösungen systematisch.

Erweiterte Aufgabe - Bearbeitung nach LU04

Diese Aufgabe gehört inhaltlich zu LU01 (Paradigmen), wird aber erst **nach LU04** bearbeitet. Erst dann stehen Ihnen mit `map`, `filter`, `reduce` und den Comprehensions die Bausteine zur Verfügung, die eine funktionale Lösung auch als solche erkennbar machen.

Sie zielt auf das Kompetenzniveau *erweitert* (AE1: *Ich kann aufzeigen, wie Probleme in den verschiedenen Konzepten (OO, prozedural und funktional) gelöst werden und diese miteinander vergleichen.*).

Die Aufgabenstellung

Für eine Klasse sollen aus einer Notenliste drei Kennzahlen ermittelt werden:

1. der Notendurchschnitt, auf zwei Stellen gerundet,
2. die Namen aller Lernenden mit einer Note unter 4.0,
3. die beste Note der Klasse.

Ausgangsdaten

```
noten = [  
    ('Alina', 5.5),  
    ('Ben', 3.5),  
    ('Chiara', 4.0),  
    ('Dario', 5.0),  
    ('Elif', 3.9),  
]
```

Erwarteter Output

```
Durchschnitt: 4.38  
Ungenuegend: ['Ben', 'Elif']  
Beste Note: 5.5
```

Anforderungen

Erstellen Sie **drei separate Python-Dateien** mit derselben Ausgabe:

1. Datei prozedural.py

- Kein Klassenkonstrukt, keine ausgelagerten Funktionen.
- Die Zwischenergebnisse werden in Variablen aufgebaut und in einer Schleife fortgeschrieben.
- Erlaubt (und erwünscht) sind Zähler, Akkumulatoren und `append`.

2. Datei objektorientiert.py

- Eine Klasse Klasse (oder `Notenliste`) hält die Noten als Attribut.
- Methoden: `hinzufuegen(name, note)`, `durchschnitt()`, `ungenuegende()`, `beste_note()`.
- Der Zustand lebt im Objekt und wird durch `hinzufuegen` verändert.

3. Datei funktional.py

- Nur Funktionen, die ihre Eingabe als Parameter erhalten und einen Wert zurückgeben.
- Keine globalen Variablen, kein `print` innerhalb der berechnenden Funktionen.
- Nichts wird verändert – wer eine Note ergänzt, erhält eine neue Liste zurück.
- **Keine for-Schleife.** Verwenden Sie die Bausteine aus LU04: `map`, `filter`, `reduce`, Lambda-Ausdrücke und Comprehensions.

Warum das Schleifenverbot in Teil 3?

Ohne diese Vorgabe unterscheidet sich die funktionale Fassung kaum von der prozeduralen – beide hätten eine Schleife, nur an einer anderen Stelle. Erst wenn die Iteration ganz im Sprachkonstrukt verschwindet, wird sichtbar, was der deklarative Stil eigentlich ändert.

`reduce` importieren Sie mit `from functools import reduce`.

Vergleichstabelle

Füllen Sie anschliessend die folgende Tabelle aus und begründen Sie jede Zelle mit einem Satz:

Kriterium	Prozedural	Objektorientiert	Funktional
Wo liegt der Zustand?			
Was passiert, wenn dieselbe Berechnung zweimal aufgerufen wird?			
Wie testet man die Berechnung einzeln?			

Kriterium	Prozedural	Objektorientiert	Funktional
Was muss man wissen, um eine einzelne Zeile zu verstehen?			
Was geschieht, wenn zwei Programmteile gleichzeitig darauf zugreifen?			
Anzahl Codezeilen			

Reflexionsfragen

1. In welcher der drei Fassungen lässt sich am einfachsten eine vierte Kennzahl ergänzen (z.B. der Median)? In welcher am schwierigsten?
2. Rufen Sie in der OO-Fassung `durchschnitt()` auf, fügen Sie dann eine Note hinzu und rufen Sie `durchschnitt()` erneut auf. Erklären Sie, warum derselbe Aufruf zwei verschiedene Ergebnisse liefert – und warum das in der funktionalen Fassung nicht passieren kann.
3. Die objektorientierte Programmierung wird oft der funktionalen gegenübergestellt. Zeigen Sie an Ihrem Beispiel, dass die OO-Fassung im Kern **imperativ** ist. Woran erkennt man das?
4. Was passiert in Ihrer funktionalen Fassung bei einer **leeren** Notenliste? Erklären Sie die Fehlermeldung und schlagen Sie eine Behandlung vor.
5. Nennen Sie für jedes der drei Paradigmen eine Situation aus Ihrem Lehrbetrieb oder aus einem früheren Modul, in der es die passende Wahl wäre.

Abgabe

Geben Sie die drei Python-Dateien, die ausgefüllte Vergleichstabelle und die beantworteten Reflexionsfragen in Moodle ab.

 © Kevin Maurizi

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m323/learningunits/lu01/aufgaben/paradigmenvergleich>

Last update: **2026/08/18 11:35**

