

LU07.L01 - Smell-Detektiv

Nr.	Smell	Problem	Technik
1	Kryptische Namen, Kommentar-Bedarf; zusätzlich <code>if ...: return True / return False</code>	<code>b[„a“]</code> und <code>b[„s“]</code> sagen nichts, die Bedingung muss entschlüsselt werden	Rename, Extract Variable; die Bedingung direkt zurückgeben
2	Magic Numbers	Woher kommen 2, 10, 7.0, 12.5, 24.9? Eine Preisänderung erfordert Suchen im Code	Replace Magic Number with Named Constant, besser noch eine Tariftabelle als Datenstruktur
3	Lange Parameterliste	Acht Parameter, beim Aufruf leicht zu vertauschen, Python meldet nichts	Introduce Parameter Object (frozen Dataclass)
4	Query und Modifier vermischt	Wer nur den Bestand lesen will, verändert dabei die Zugriffszähler; zwei Aufrufe haben verschiedene Wirkung	Separate Query from Modifier
5	Schleife mit Akkumulator	Beschreibt das <i>Wie</i> statt das <i>Was</i> , Zwischenvariable und <code>append</code> als Rauschen	Replace Loop with Pipeline
6	Globaler Zustand (unpure function)	Ergebnis hängt von der Programmgeschichte ab, nicht testbar ohne Zurücksetzen von <code>zaehler</code>	Zustand als Parameter und Rückgabewert, alternativ Closure

Hinweise

Ausschnitt 1 enthält zwei Smells. Beide Antworten sind richtig, wenn sie begründet sind:

```
def ist_kaufberechtigt(kunde):
    volljaehrig = kunde["alter"] >= 18
    aktiv = kunde["status"] == 1
    hat_guthaben = kunde["guthaben"] > 0
    return volljaehrig and aktiv and hat_guthaben
```

Ausschnitt 2 wird oft als «verschachtelte Bedingungen» genannt - das stimmt hier nicht, die Guard Clauses sind bereits sauber. Der Smell sind die Zahlen.

Ausschnitt 5 ist der Smell mit dem stärksten Modulbezug:

```
namen = [k["name"].upper() for k in kunden if k["aktiv"]]
```

M323-LU07, M323-D1B

© Kevin Maurizi

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

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu07/loesungen/smelldetektiv>

Last update: **2026/09/09 11:12**

