LU07.L04 - Lange Funktion zerlegen

Schritt 0: Sicherheitsnetz

```
# test_bericht.py
from bericht import BESTELLUNGEN, bericht

ERWARTET = (
    "Bestellungen (4)\n"
    "----------------\n"
    "Meier: Maus = 75.0 CHF\n"
    "Keller: Laptop = 1140.0 CHF\n"
    "Meier: Tastatur = 152.0 CHF\n"
    "Keller: Monitor = 199.5 CHF\n"
    "----------------\n"
    "Total: 1566.5 CHF"
)

def test_bericht_bleibt_gleich():
    assert bericht(BESTELLUNGEN) == ERWARTET

def test_stornierte_werden_ignoriert():
    nur_storno = [b for b in BESTELLUNGEN if b["status"] == "storniert"]
    assert "Bestellungen (0)" in bericht(nur_storno)

def test_rabattgrenze():
    knapp_darunter = [{"kunde": "x", "artikel": "a", "menge": 1, "preis":
99.0, "status": "offen"}]
    assert "= 99.0 CHF" in bericht(knapp_darunter)
```

Ergebnis nach dem Refactoring

```
RABATT_AB_BETRAG = 100
RABATT_PROZENT = 5


def ist_gueltig(bestellung):
    return bestellung["status"] != "storniert"


def zeilenwert(bestellung):
    wert = bestellung["menge"] * bestellung["preis"]
    if wert < RABATT_AB_BETRAG:
        return wert
    return wert * (1 - RABATT_PROZENT / 100)
```

```
def kundenname(bestellung):
    return bestellung["kunde"].strip().lower().capitalize()


def zeile(bestellung):
    return f"{kundenname(bestellung)}: {bestellung['artikel']} =
{round(zeilenwert(bestellung), 2)} CHF"


def bericht(bestellungen):
    gueltige = [b for b in bestellungen if ist_gueltig(b)]
    total = round(sum(zeilenwert(b) for b in gueltige), 2)
    kopf = f"Bestellungen ({len(gueltige)})"
    strich = "-" * len(kopf)
    return "\n".join([kopf, strich] + [zeile(b) for b in gueltige] +
[strich, f"Total: {total} CHF"])
```

Commit-Folge

```
refactor: Guard-Bedingung als ist_gueltig() extrahiert
refactor: Rabattgrenze und -satz als Konstanten benannt
refactor: Zeilenwert in zeilenwert() extrahiert
refactor: Namensaufbereitung in kundenname() extrahiert
refactor: Schleife durch Comprehension und sum() ersetzt
refactor: String-Verkettung durch f-Strings ersetzt
```

Worauf es ankommt

- `ist_gueltig`, `zeilenwert` und `kundenname` sind **pure** und einzeln testbar. Vorher war nichts davon prüfbar, ohne den ganzen Bericht zu erzeugen.
- Die Rabattregel steht an einer Stelle und heisst so, wie die Fachabteilung sie nennt.
- `bericht` liest sich jetzt wie eine Inhaltsangabe: filtern, summieren, formatieren.
- Der Test `test_rabattgrenze` deckt genau die Zeile ab, an der ein Vorzeichenfehler (> statt >=) unbemerkt bliebe.

Häufiger Fehler: `round()` zu früh anwenden. Wer in `zeilenwert` rundet, bekommt beim Total unter Umständen einen anderen Wert als die Vorlage. Runden Sie am Schluss - und prüfen Sie es mit dem Test statt mit dem Auge.

M323-LU07, M323-D1I

© Kevin Maurizi

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

Permanent link:
https://wiki.bzz.ch/modul/m323/learningunits/lu07/loesungen/langefunktion?rev=1788945248

Last update: **2026/09/09 11:14**