LU07.A04 - Lange Funktion zerlegen

Zerlegen Sie die Funktion `bericht` in benannte Teilfunktionen. Die Ausgabe muss **zeichengenau** dieselbe bleiben.

Ausgangslage

```
BESTELLUNGEN = [
    {"kunde": "  Meier ", "artikel": "Maus",     "menge": 3, "preis": 25.0,
"status": "offen"},
    {"kunde": "keller",   "artikel": "Laptop",   "menge": 1, "preis":
1200.0, "status": "bezahlt"},
    {"kunde": "MEIER",    "artikel": "Tastatur", "menge": 2, "preis": 80.0,
"status": "bezahlt"},
    {"kunde": "Suter",    "artikel": "USB-Hub",  "menge": 5, "preis": 12.0,
"status": "storniert"},
    {"kunde": "keller ",  "artikel": "Monitor",  "menge": 1, "preis": 210.0,
"status": "offen"},
]


def bericht(bestellungen):
    zeilen = []
    total = 0.0
    anzahl = 0
    for b in bestellungen:
        if b["status"] != "storniert":
            wert = b["menge"] * b["preis"]
            if wert >= 100:
                wert = wert * 0.95
            total = total + wert
            anzahl = anzahl + 1
            name = b["kunde"].strip().lower().capitalize()
            zeilen.append(name + ": " + b["artikel"] + " = " +
str(round(wert, 2)) + " CHF")
    kopf = "Bestellungen (" + str(anzahl) + ")"
    strich = "-" * len(kopf)
    fuss = "Total: " + str(round(total, 2)) + " CHF"
    return "\n".join([kopf, strich] + zeilen + [strich, fuss])
```

Die Funktion liefert:

```
Bestellungen (4)
----------------
Meier: Maus = 75.0 CHF
Keller: Laptop = 1140.0 CHF
```

```
Meier: Tastatur = 152.0 CHF
Keller: Monitor = 199.5 CHF
----------------
Total: 1566.5 CHF
```

Detaillierte Aufgabenstellung

1. **Sicherheitsnetz zuerst:** Schreiben Sie einen Test, der die obige Ausgabe festhält. Er muss vor Ihrem ersten Umbau grün sein.
2. Wenden Sie mindestens vier Techniken aus LU07c an. Erwartet werden: Extract Function, Named Constant, Replace Loop with Pipeline, sprechende Namen.
3. Nach **jedem** einzelnen Umbau: Test laufen lassen und committen. Die Commit-Message nennt die Technik.
4. Am Schluss darf keine Funktion länger als etwa zehn Zeilen sein.

Abgabe

- die refactorierte Datei
- die Testdatei
- die Ausgabe von `git log –oneline` Ihrer Refactoring-Commits

Diese Aufgabe erzeugt genau das Material, das im Portfolio für **D1I** verlangt wird: Vorher/Nachher aus der Commit-Historie, gleiches Verhalten per Test belegt.

M323-LU07, M323-D1I

© Kevin Maurizi

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

Permanent link:
https://wiki.bzz.ch/modul/m323/learningunits/lu07/aufgaben/langefunktion?rev=1788945227

Last update: **2026/09/09 11:13**