LU07.A07 - Sicherheitsnetz zuerst

Schreiben Sie Characterization Tests für eine Funktion, deren Verhalten niemand mehr kennt - und refactoren Sie sie erst danach.

Repository: m323-lu07-a07-characterization

Ausgangslage

Diese Funktion läuft seit zwei Jahren im Notensystem. Dokumentation gibt es keine, Tests auch nicht.

```
def notenschnitt(punkte, maximum):
    if maximum == 0:
        return 1
    note = 1 + 5 * (sum(punkte) / len(punkte)) / maximum
    if note > 6:
        note = 6
    return round(note * 2) / 2
```

1. Verhalten erkunden

Rufen Sie die Funktion mit verschiedenen Eingaben auf und notieren Sie, was herauskommt. Denken Sie an die Ränder: leere Liste, `maximum = 0`, mehr Punkte als das Maximum, exakt die Höchstpunktzahl.

2. Verhalten festhalten

In `main_test.py` sind sechs Tests vorbereitet, die noch `0` als erwarteten Wert enthalten. Tragen Sie den **beobachteten** Wert ein - auch dann, wenn er Ihnen falsch vorkommt. Ein Characterization Test hält fest, was *ist*, nicht was sein sollte.

Raten Sie nicht. Ein geratener Wert fällt sofort durch, weil die Funktion etwas anderes liefert.

Auch der Absturz gehört dazu: `test_leere_punkteliste` erwartet eine Ausnahme. Tragen Sie ein, welche tatsächlich auftritt.

3. Refactoren

Erst jetzt: sprechende Namen, benannte Konstanten, die Rundungsregel als eigene Funktion. Nach

jedem Schritt `pytest`. Alle sieben Tests bleiben grün.

4. Bewerten

Notieren Sie in `BEFUND.md`, welche Ergebnisse Sie für fachlich falsch halten und was zu tun wäre. **Ändern Sie das Verhalten nicht** - das wäre ein Bugfix und gehört in einen eigenen Commit nach dem Refactoring.

Bewertung

Teil	Punkte
Tests (`main_test.py`)	14
pylint (`main.py`)	5

Die Punkte gibt es nur, wenn der eingetragene Wert dem tatsächlichen Verhalten entspricht **und** die Funktion nach dem Refactoring immer noch so reagiert.

Leitfragen

- Was passiert bei `notenschnitt([], 20)`? Ist das ein Verhalten, das man testen kann?
- Warum liefert `notenschnitt([10], 0)` den Wert `1` und nicht `1.0`? Spielt das eine Rolle?
- Was bedeutet `round(note * 2) / 2` fachlich?
- Der Test hält ein Verhalten fest, das vielleicht ein Bug ist. Warum ist das trotzdem der richtige erste Schritt?

M323-LU07, M323-D1I

© Kevin Maurizi

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

Permanent link:
https://wiki.bzz.ch/modul/m323/learningunits/lu07/aufgaben/characterization?rev=1788954014

Last update: **2026/09/09 13:40**