LU09.L02 - Refactoring

Musterlösung zu LU09.A02 - Refactoring. Es gibt mehrere gute Lösungen. Entscheidend ist, dass die Tests nach jedem Schritt grün bleiben.

Code Smells

Nr	Stelle	Code Smell	Prinzip
1	`n`, `m`, `d`, `e`, `s`, `g`, `a1`, `x`, `t`, `c`	nichtssagende Namen	Sprechende Namen
2	Zeilen 3–16	dreimal derselbe Ablauf für Summe, Schnitt und Runden	DRY
3	Zeilen 24–29	dreimal dieselbe Prüfung auf ungenügend	DRY
4	`4`, `1`	Genügend-Grenze und erlaubte Anzahl ungenügender Fächer als Magic Numbers	DRY
5	ganze Funktion	rechnet, entscheidet über die Promotion, formatiert und gibt aus	SRP
6	`# Mathe`, `# Deutsch`	Kommentare trennen Abschnitte, statt dass Funktionen sie benennen	SRP, sprechende Namen
7	`for g in m: s = s + g`	eigene Schleife statt `sum()`	KISS
8	`print(t)`	Nebenwirkung: Die Funktion lässt sich nicht ohne Bildschirmausgabe verwenden	SRP

Verbesserter Code

report.py

```
""" Erstellt das Zeugnis eines Lernenden """
PASS_MARK = 4.0
MAX_FAILED_SUBJECTS = 1


def round_to_half(value):
    """
    Rundet auf halbe Noten
    :param value: die Note
    :return: die gerundete Note
    """
    return round(value * 2) / 2


def average(grades):
    """
    Berechnet den Durchschnitt
    :param grades: Liste mit Noten
    :return: der Durchschnitt
```

```
    """
    return sum(grades) / len(grades)


def is_promoted(subject_grades):
    """
    Prüft, ob die Promotionsbedingungen erfüllt sind
    :param subject_grades: Dictionary mit Fach und Zeugnisnote
    :return: True, wenn der Schnitt genügt und höchstens ein Fach
ungenügend ist
    """
    failed = [grade for grade in subject_grades.values() if grade <
PASS_MARK]
    overall = round(average(list(subject_grades.values())), 1)
    return overall >= PASS_MARK and len(failed) <= MAX_FAILED_SUBJECTS


def report(name, math, german, english):
    """
    Erstellt das Zeugnis als Text
    :param name: Name des Lernenden
    :param math: Noten in Mathematik
    :param german: Noten in Deutsch
    :param english: Noten in Englisch
    :return: das Zeugnis
    """
    subject_grades = {
        'Mathe': round_to_half(average(math)),
        'Deutsch': round_to_half(average(german)),
        'Englisch': round_to_half(average(english)),
    }
    lines = [f'Zeugnis {name}']
    for subject, grade in subject_grades.items():
        lines.append(f'{subject}: {grade}')
    lines.append(f'Schnitt:
{round(average(list(subject_grades.values())), 1)}')
    lines.append('promoviert' if is_promoted(subject_grades) else
'nicht promoviert')
    return '\n'.join(lines)


if __name__ == '__main__':
    print(report('Mia', [4.5, 5.0], [5.5, 6.0], [4.5, 4.5]))
```

Die Ausgabe mit `print` steht jetzt im `__main__`-Block. Die Funktion `report` liefert nur noch den Text.

Zusätzliche Tests

Die Hilfsfunktionen lassen sich jetzt einzeln testen. Mit der alten Version war das nicht möglich, weil alles in einer Funktion steckte.

```
from report import report, round_to_half, is_promoted


def test_round_to_half():
    """ neue Hilfsfunktion: lässt sich jetzt einzeln testen """
    assert round_to_half(4.3) == 4.5
    assert round_to_half(4.2) == 4.0


def test_is_promoted_limit():
    """ Grenzfall: Schnitt genau 4.0 mit einer ungenügenden Note """
    assert is_promoted({'Mathe': 3.5, 'Deutsch': 4.0, 'Englisch': 4.5})
```

Ergebnis

```
$ pytest -q
......
6 passed

$ pylint --max-line-length=120 report.py test_report.py
Your code has been rated at 10.00/10
```

Beobachtung: pylint bewertete auch die alte Version mit 9.35/10. Die Note sagt also wenig über die Verständlichkeit aus. Die wichtigsten Verbesserungen (Namen, DRY, SRP) erkennt nur ein Mensch.

M450-LU09, M450-D1B, M450-D1A

Marcel Suter

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

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu09/loesungen/refactoring

Last update: **2026/11/23 09:19**