Technische Hochschule Ingolstadt, WI
Prof. A. Schiendorfer, L. Lodes, M.Sc.

Software Development (Winter 24/25)

Übungsblatt 10

Bonuspunkte können bis 14.01.2025, 11:00 Uhr geltend gemacht werden.

Aufgabe 1 *(Testen und Debuggen, 2 Punkte)*

Beim sog. "Debuggen" geht es darum, die vorhandenen Fehler ("Bugs") in einem Programm zu finden und zu beheben. Der Begriff "Fehler" bezieht sich dabei nicht nur auf die falsche Verwendung der Programmiersprache und die daraus resultierenden Fehlermeldungen bei der Ausführung, sondern auch auf logische Fehler. Wenn ein logischer Fehler vorliegt, wird das Programm zwar ohne Fehlermeldung ausgeführt, jedoch meist mit falschen bzw. unerwarteten Ergebnissen. In dieser Aufgabe sollen sie das Debuggen von logischen Fehlern üben.

a) Überlegen Sie sich eine Strategie, mit der sie logisch fehlerhafte Teile von Code identifizieren können und beschreiben Sie diese kurz.

b) Betrachten Sie das folgende Codebeispiel. Beschreiben Sie kurz den/die logischen Fehler und erläutern Sie, wie Sie den/die Fehler gefunden haben. Zeigen Sie anschließend eine Lösung für den/die Fehler auf.

```python
def vector_addition(vectorA, vectorB):
    result = [None] * len(vectorA)

    for i in range(1, len(vectorA)):
        result[i] = vectorA[i] + vectorB[i]

    return result
```

c) Betrachten Sie das folgende Codebeispiel. Beschreiben Sie kurz den/die logischen Fehler und erläutern Sie, wie Sie den/die Fehler gefunden haben. Zeigen Sie anschließend eine Lösung für den/die Fehler auf.

```python
def calculate_overall_price(area, price_per_sqm=40, tax=0.15):
    net_price = area * price_per_sqm
    gross_price = net_price * tax
    return gross_price

area = 500
price_per_sqm = 85
tax = 0.10

overall_price = calculate_overall_price(area, price_per_sqm)
print(f"The overall price is {overall_price}")
```

Aufgabe 2 ***(Objektorientierte Programmierung, 3 Punkte)***

Betrachten Sie das folgende UML-Diagramm, welches die Klassen `Factory`, `Product` und zwei Subklassen von `Product` zeigt. Implementieren Sie die abgebildeten Klassen und Zusammenhänge in Python. Achten Sie insbesondere auf die eingezeichneten Assoziationen und Multipizitäten.