SAE	**OOP: Kapselung und spezielle Methoden**	Klasse:
L. Bung		Datum:

Aufgabe 1: Kapselung und Sichtbarkeit

Recherchieren Sie, was man in der objektorientierten Programmierung unter dem Begriff "Kapselung" versteht. Beantworten Sie dabei die folgenden Fragen:

a) Was bedeuten die Wörter "public" und "private"?

b) Wie werden Attribute und Methoden im UML-Klassendiagramm als public bzw. private gekennzeichnet?

c) Wie werden Attribute und Methoden in Python als public bzw. private gekennzeichnet?

d) Was sind "getter"- und "setter"-Methoden und wozu werden diese benötigt?

Aufgabe 2: Spezielle Methoden in Python

Recherchieren Sie, was die sogenannten "magic methods" in Python sind. Klären Sie dabei die Bedeutung folgender Methoden:

a) `__str__(self, other)`

b) `__add__(self, other)` bzw. `__sub__`, `__mul__` und `__div__`

c) `__eq__(self, other)` bzw. `__ne__`

d) `__lt__(self, other)`, `__gt__` sowie `__le__` und `__ge__`

Aufgabe 3: Geometrische Punkte in Python

Erstellen Sie eine Klasse `Punkt`, welche über eine x-Koordinate und eine y-Koordinate verfügt.

Nutzen Sie die in Aufgabe 2 gewonnenen Erkenntnisse, um folgenden Code funktionsfähig zu machen:

```
p1 = Punkt(2, 3)
p2 = Punkt(5, 5)
p3 = p1 + p2
print(p3)  #Erwartete Ausgabe: (7, 8)
p4 = p2 - p1
print(p4)  #Erwartete Ausgabe: (3, 2)
print(p3 == p4)  #Erwartete Ausgabe: false
```

Aufgabe 4: Bankkonto

a) Erstellen Sie eine Klasse `Bankkonto` mit den Attributen `kontonummer` und `kontostand` sowie den Methoden `einzahlen(betrag)`, `abheben(betrag)`.

b) Überlegen Sie sich, ob Attribute öffentlich sein sollten oder ob es sinnvoll wäre, eines auf privat zu setzen. Falls ja, erstellen Sie jeweils eine sinnvolle get- und set-Methode.

c) Setzen Sie den Code als UML-Klassendiagramm um.

Aufgabe 5: Online-Shop

Sie sollen eine Klassenstruktur für einen Online-Shop entwerfen, in dem Artikel verkauft und in einen Warenkorb gelegt werden können. Dabei sollen Sie selbst entscheiden, welche Attribute und Methoden öffentlich und welche privat sein sollten.

a) Erstellen Sie eine Klasse `Artikel` mit den Attributen `name` und `preis`. Überlegen Sie, ob die Attribute öffentlich oder privat sein sollen und erstellen Sie gegebenenfalls passende get- bzw. set-Methoden. Ermöglichen Sie zusätzlich, dass eine lesbare Darstellung des Artikels angezeigt werden kann.

b) Schreiben Sie eine Klasse `Warenkorb`, welcher mehrere `Artikel` enthalten soll. Überlegen Sie auch hier, wie die Sichtbarkeit der Artikel sein soll und modifizieren Sie Ihren Code gegebenenfalls entsprechend. Erstellen Sie zusätzlich die Methoden `artikel_hinzufügen(self, artikel)` und `artikel_entfernen(self, artikel)` sowie `gesamtpreis(self)`.

c) Implementieren Sie sinnvolle Versionen der Methoden `__str__(self)`, `__len__(self)` und `__add__(self, other)` für die Klasse `Warenkorb`.

d) Stellen Sie Ihren Code in Form eines UML-Klassendiagramms dar.