

👤 Aufgabe 1: Wiederholung: Klassen- und Objektdiagramme

Ein Café möchte ein digitales Verwaltungssystem entwerfen. Erstellen Sie hierfür ein Klassendiagramm¹:

- Ein Produkt soll mit Namen, Preis und Beschreibung gespeichert werden. Die Methode `zeigeInfo()` soll die Produktinformationen als Text ausgeben.
- Eine Bestellung hat eine Bestellnummer, ein Datum und einen Gesamtpreis. Sie soll die Methoden `berechneGesamtpreis()` und `zeigeDetails()` haben.
- Ein Kunde hat einen Namen, eine Kundennummer und eine Email.

Entwerfen Sie zusätzlich ein passendes Objektdiagramm für die folgende Situation:

- Das Produkt "Cappuccino" kostet 3,50€ und hat die Beschreibung "Espresso mit Milchschaum".
- Die Bestellung mit der Nummer 20240312 wurde am 12.03.2024 aufgegeben und hat eine Gesamtsumme von 7,00€.
- Die Kundin "Lisa Meier" hat die Kundennummer 1023 und die E-Mail-Adresse "lisa.meier@email.de".

👤 Aufgabe 2: Vorstellungsrunde

- a) Schreiben Sie eine Klasse `Person`, welche die Attribute `name` und `alter` besitzt.
- b) Fügen Sie eine Methode `vorstellen()` hinzu, die eine Begrüßung mit den gespeicherten Attributen ausgibt.
- c) Erstellen Sie zwei konkrete Personen und lassen Sie sie vorstellen.
- d) Schreiben Sie ein Programm, das den Nutzer nach seinem Namen und Alter fragt. Lassen Sie das Programm daraus ein Objekt vom Typ `Person` erstellen und rufen sie die Methode `vorstellen()` auf.

¹Sie können für die Diagramme wieder draw.io nutzen.

Aufgabe 3: Patientenverwaltung: Implementierung

Nachdem Sie in der letzten Stunde das Patientenverwaltungssystem für das Krankenhaus geplant haben, soll es nun konkret umgesetzt werden.

Nutzen Sie Ihre erstellten Objekt- und Klassendiagramme und implementieren Sie diese in Python.

Aufgabe 4: Brüche in Python²

In Python gibt es keinen Datentyp für Brüche wie $\frac{13}{7}$. Implementieren Sie das nebenstehende Klassendiagramm in Python, so dass die folgenden Programmaufrufe möglich werden:

```
b = Bruch(4, 6)
b.erweitern(3)
b.kuerzen(2)
b.zaehler
b.nenner
```

Bruch
zaehler: int nenner: int
erweitern(k: int) kuerzen(k: int)

Erstellen Sie einige Objekte und Testen Sie das Erweitern und Kürzen der Brüche.

Bonus: Erstellen Sie zusätzlich eine Methode `vollstaendigKuerzen()` mit entsprechender Implementierung.

Aufgabe 5: Zähler²

Ein Zähler ist ein Gerät, mit dem man hochzählen kann, und den man gezielt wieder auf Null setzen kann. Man benutzt solche Geräte z.B. bei Verkehrszählungen.

- Modellieren Sie eine Klasse `Zaehler`, mit der man Objekte erzeugen kann, die sich wie Zähler in der Wirklichkeit verhalten. Berücksichtigen Sie vorerst noch nicht, dass es für die Zahl eine Obergrenze gibt. Stellen Sie die modellierte Klasse mit einem Klassendiagramm dar. Implementieren und testen Sie anschließend die entwickelte Klasse.
- Wenn man eine Uhr simulieren möchte, dann benötigt man einen Sekunden- bzw. Minutenzähler, der so zählt:
0, 1, 2, 3, ..., 58, 59, 0, 1, 2, 3, ..., 58, 59, 0, 1, 2, ...
Ein Stundenzähler zählt entsprechend:
0, 1, 2, 3, ..., 23, 0, 1, 2, 3, ..., 23, 0, 1, 2, ...
Modellieren und implementieren Sie auch für Zählsituationen mit einer Obergrenze eine geeignete Klasse.

²Klaus Becker & Heiko Jochum