SAE	**OOP: Vermischte Übungen**	Klasse:
L. Bung		Datum:

Zur Wiederholung und Vertiefung der bisherigen Inhalte finden Sie auf diesem Arbeitsblatt vermischte Übungen. Sie können sich aussuchen, welche der folgenden Aufgaben Sie bearbeiten. Wählen Sie die Aufgaben, bei denen Sie bisher die größten Schwierigkeiten hatten oder gefehlt haben.

Sie können auch Aufgaben von den bisherigen Übungsblättern nacharbeiten.

Aufgabe 1: Klassen- und Objektdiagramme in UML

In einer kleinen Stadtbibliothek werden Bücher und deren Ausleihe durch Benutzer verwaltet. Ein Benutzer ist eine registrierte Person, die Bücher ausleihen kann. Zu jedem Benutzer werden Name, Ausweisnummer und E-Mail-Adresse gespeichert. Ein Buch besitzt einen Titel, einen Autor und eine ISBN. Wenn ein Benutzer ein Buch ausleiht, wird ein Ausleihvorgang erstellt. Dabei werden das Datum der Ausleihe und (falls das Buch schon zurückgegeben wurde) das Rückgabedatum gespeichert. Jeder Ausleihvorgang erhält außerdem eine eindeutige Ausleih-ID.

a) Erstellen Sie ein UML-Klassendiagramm[1] der Situation. Modellieren Sie dabei die Klassen `Benutzer`, `Buch`, `Ausleihvorgang` mit geeigneten Attributen und Methoden.

b) Erstellen Sie ein UML-Objektdiagramm für folgende konkrete Situation:

Am 8. Mai 2025 leiht sich die Benutzerin Anna Weber (Ausweisnummer: 10234, E-Mail: `anna.weber@mail.de`) das Buch "1984" von George Orwell (ISBN: 9780451524935) aus. Die Ausleihe erhält die ID A5678.

Aufgabe 2: Klassen in Python

a) Schreiben Sie eine Klasse `Student` und eine Klasse `Kurs`.

b) In der Klasse `Kurs` sollen eingeschriebene Studenten in einer Liste verwaltet werden.

c) Schreiben Sie Methoden, um einen Studenten in einen Kurs ein- und auszuschreiben.

d) Erstellen Sie mehrere Objekte und testen Sie ihre Klassen ausgiebig.

[1] (z.B. wieder mit `https://draw.io`)

Aufgabe 3: Kapselung in Python

a) Schreiben Sie eine Klasse `Benutzer` mit dem öffentlichen Attribut `nutzername` und dem privaten Attribut `passwort`.

b) Implementieren Sie die nötigen getter- und setter-Methoden.

c) Schreiben Sie eine Methode `istLangGenug(neuesPasswort: str)`, welche überprüft, ob das gegebene Passwort länger als 8 Zeichen ist.

d) Schreiben Sie eine Methode `hatGrossbuchstaben(neuesPasswort: str)` und eine Methode `hatKleinbuchstaben(neuesPasswort: str)`, welche überprüfen, ob das gegebene Passwort einen Großbuchstaben bzw. Kleinbuchstaben enthält.

> **Tipp**
>
> Die Python-Methoden `isupper()`[2] und `islower()`[3] könnten hilfreich sein.

e) Rufen Sie Ihre geschriebenen Methoden in der Setter-Methode auf und ändern Sie das Passwort nur, wenn es sicher genug ist.

Aufgabe 4: Vererbung in UML

Erstellen Sie ein UML-Klassendiagramm[4], welches die folgende Situation modelliert.

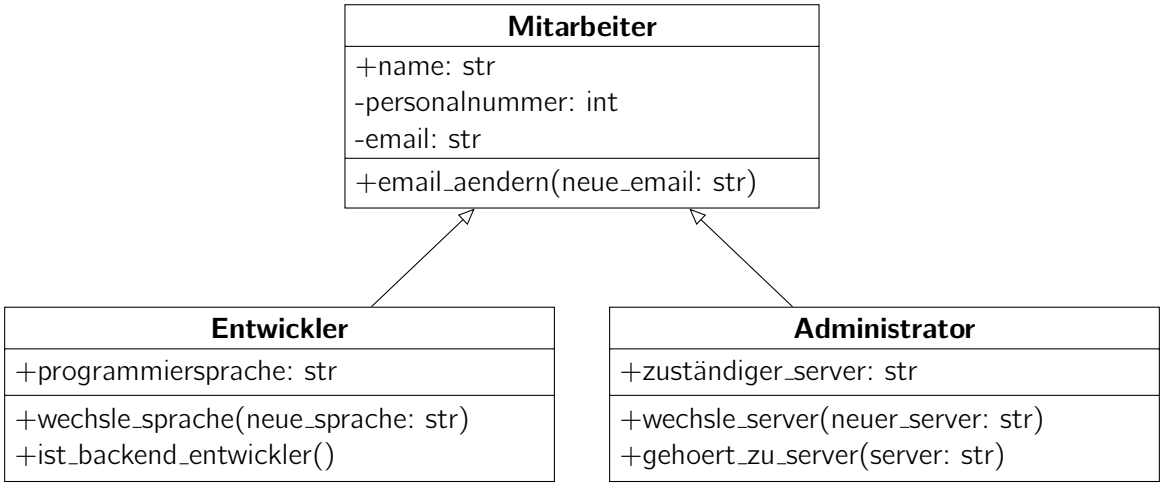
Ein IT-Dienstleister entwickelt ein Ticketsystem zur Bearbeitung interner Anfragen. Alle Anfragen werden als Objekte der Oberklasse `Ticket` verwaltet. Diese enthält allgemeine Informationen wie die Ticket-ID, eine Beschreibung und das Erstellungsdatum. Je nach Art der Anfrage gibt es spezialisierte Tickettypen: Ein `Fehlerticket` wird bei Störungen oder Fehlermeldungen verwendet. Es verfügt über das zusätzliche Attribut `prioritaet` (z.B. "hoch", "mittel", "niedrig"). Ein `ServiceTicket` stellt eine Serviceanfrage dar (z.B. Softwareinstallation oder Computerwartung). Es verfügt über das zusätzliche Attribut `angeforderte_leistung` (z.B. "VPN-Zugang einrichten").

Fügen Sie bei der Modellierung geeignete Methoden zu den Klassen hinzu. Wählen Sie außerdem den korrekten Datentyp für die Attribute und überlegen Sie, ob diese public oder private sein sollten.

[2] `https://www.w3schools.com/python/ref_string_isupper.asp`
[3] `https://www.w3schools.com/python/ref_string_islower.asp`
[4] (z.B. wieder mit `https://draw.io`)

Aufgabe 5: Vererbung in Python

Setzen Sie das folgende UML-Klassendiagramm in Python um.



Die Methode `ist_backend_entwickler()` soll `True` zurückgeben, wenn die Programmiersprache des Entwicklers Java oder Python ist (ansonsten `False`).

Die Methode `gehoert_zu_server(server)` soll überprüfen, ob der Administrator dem übergebenen Server zugewiesen ist.

Achten Sie bei der Implementierung auf die Sichtbarkeit und implementieren Sie gegebenenfalls getter- und setter-Methoden.