Aufgabe 1 (12 Punkte)

Erklären Sie folgende Begriffe anhand eines von Ihnen gewählten Programmausschnittes und erläutern Sie den Programmcode in einem Kommentar.

a) abstrakte Methodendeklaration

b) Methodenaufruf

c) überladene Methode

d) Vererbung

e) Konstante

f) Kapselung

Aufgabe 2 (15 Punkte)

Für ein Theater soll ein Verzeichnis aller Veranstaltungen eines Jahres angelegt werden. Dazu sollen die Veranstaltungen in einer eigenen Klasse `Veranstaltungen` modelliert werden. Veranstaltungen können sein: `Konzert` oder `Schauspiel`, die jeweils in Unterklassen von `Veranstaltung` modelliert werden.

Alle Veranstaltungen beginnen um 20 Uhr und es findet höchstens eine Veranstaltung pro Tag statt. Das Theater hat immer freie Platzwahl. Bei Konzerten haben 200 Zuhörer Platz, sonst 250 Personen. Legen Sie die entsprechende Klasse `Veranstaltung` mit den notwendigen Attributen an.

Ergänzen Sie die Klasse `Veranstaltung`, um die Attribute `titel`, `datum` und `anzNochVerfügbarePlaetze`. Ein Zähler `anzVeranstaltungen` soll über die Anzahl der bisher instanziierten Veranstaltungen Buch führen.

Jede Klasse soll über einen Konstruktor verfügen, der alle Attribute der Klasse setzt.

Die polymorphe Methode `ticketVerkaufen` hat einen einzigen Parameter, der die Anzahl der angefragten Tickets angibt. Der Rückgabewert der Methode soll anzeigen, ob noch genügend Plätze frei waren oder nicht. Falls genügend Plätze frei waren, werden diese verkauft und entsprechend verbucht.

Geben Sie die Java-Implementierungen für **alle** oben genannten Klassen an. Wählen Sie geeignete Datentypen für die Attribute und Parameter und kapseln Sie die Attribute so gut wie möglich.

Die getter/setter-Methoden können Sie als gegeben voraussetzen.

Beachten Sie bei Ihrer Lösung die Prinzipien der OOP !

Aufgabe 3 (10 Punkte)

Ein Krankenhaus verwaltet seine Patienten mit einem Verwaltungsprogramm.

Jede/r Patient/In hat einen Name, Vorname, Geschlecht und ein Geburtsdatum. Gesetzlich Versicherte sind immer Mitglied einer Krankenkasse. Selbstzahler, auch bekannt als Privat-Patienten, können eine Krankenversicherung besitzen.

Die Klassen `Selbstzahler` und `GesetzlichVersicherter` sollen jeweils von der Klasse `Patient` abgeleitet sein.

Alle Leistungen des Krankenhauses werden über die Gebührenordnung abgerechnet, wobei gesetzlich Versicherte den einfachen und Selbstzahler der dreieinhalbfache Satz in Rechnung gestellt wird. Zu jedem Patienten wird ein Saldo geführt, der die Summe aus berechneten Leistungen und Zahlungen der Krankenkasse oder -versicherung darstellt. Dazu verfügt die Klasse `Patient` über die abstrakten Methoden `rechneLeistungAb` und `erstatteLeistung`.

Die Klassen `Selbstzahler` und `GesetzlichVersicherter` sollen jeweils von der Klasse `Patient` abgeleitet sein.

Hinweis: `Getter`- und `setter`-Methoden müssen Sie nicht angeben und können diese als vorhanden voraussetzen.

Geben Sie das UML-Klassendiagramm für **alle** oben genannten Klassen an. Wählen Sie geeignete Datentypen für die Attribute und Parameter und kapseln Sie die Attribute so gut wie möglich.

Alle oben genannten Angaben, die sich in einem UML-Klassendiagramm darstellen lassen, müssen auch in Ihrer Lösung vorkommen.

Aufgabe 4 (10 Punkte)

Gegeben sei folgendes UML-Klassen-Diagramm:

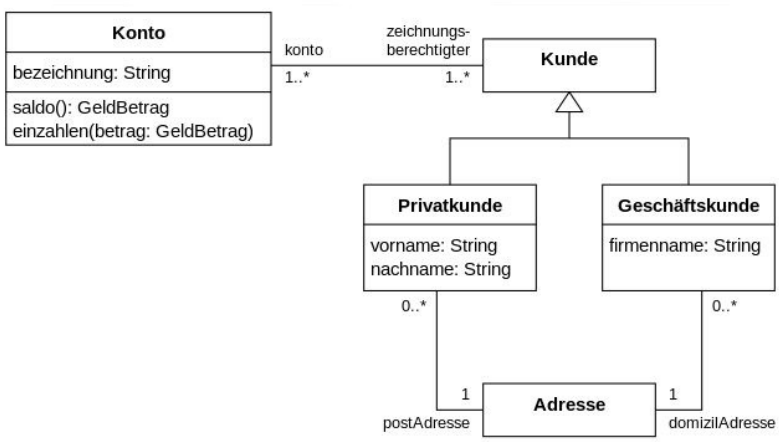

Abbildung 1:

Beachten Sie die Darstellung der Kardinalitäten und Assoziationen!

Hinweis: Konstruktoren, `getter`- und `setter`-Methoden müssen Sie nicht angeben und können diese als vorhanden voraussetzen.

a) Geben Sie die Klasse `Konto` als exakt entsprechende Java-Klasse an.

b) Geben Sie die Klasse `Privatkunde` als exakt entsprechende Java-Klasse an.

c) Geben Sie die Klasse `Adresse` als exakt entsprechende Java-Klasse an.

Aufgabe 5 (5 Punkte)

Betrachten Sie die folgenden Aussagen im Kontext von Klassen und Objekten.

Kreuzen Sie die Aussagen an, die *korrekt* sind:

☐ Konstruktoren besitzen oft einen Rückgabetyp.

☐ Konstruktoren besitzen oft eine Parameterliste.

☐ `this()` ist der Aufruf eines Konstruktors der eigenen Instanz.

☐ Variablen werden nur dann automatisch mit Default-Werten initialisiert, wenn es sich um lokale Variablen (z.B. in Methodenblöcken) handelt.

☐ Einer als `private` spezifizierten Klassen-Variablen kann man nur in Instanzen dieser Klasse einen Wert zuweisen.

Aufgabe 6 (8 Punkte)

Gegeben sei folgende Klasse `Test`.

```
1  public class Test {
2          private static int anzahl;
3          private static Auto a1, a2;
4          public static double quotient;
5          public static void main(String[] args) {
6                  quotient = 7.5;
7                  a2 = new Auto();
8                  a1 = a2;
9                  a2 = null;
10                 anzahl++;
11                 System.out.println("Anzahl: "+anzahl);
12         }
13 }
```

Es existiere außerdem eine Klasse `Auto` mit Standardkonstruktor. Es wird die `main`-Methode der Klasse `Test` aufgerufen.

Kreuzen Sie die Aussagen an, die *nicht korrekt* sind:

☐ Die Klasse `Test` besitzt mindestens einen Konstruktor mit leerer Parameterliste.

☐ Die Klasse `Auto` besitzt genau einen Konstruktor mit leerer Parameterliste.

☐ Die Variable `anzahl` ist eine Variablen eines primitiven Datentyps.

☐ Dadurch, dass in Zeile 9 `a2 = null;` gesetzt wird, haben beide Variablen `a1` und `a2` nun ein und denselben Wert.

☐ Dadurch, dass in Zeile 8 `a1 = a2;` gesetzt wird, kann das in Zeile 7 erzeugte Auto-Objekt bis zum Ende der Methode mindestens über eine Variable referenziert werden.

☐ Die Definition der Variablen `anzahl` kann nicht vor ihrer Deklaration erfolgen.

☐ Die Deklaration der Variablen `anzahl` kann nicht vor ihrer Definition erfolgen.

☐ Jede Instanz der Klasse `Test` hat einen individuellen Wert des Attributes `quotient`.

Zusatzblatt