

Computerverleih

Die Konrad-Zuse-Schule Baku bietet ihren Schülerinnen und Schülern (im Folgenden Schüler genannt) die Möglichkeit, gebrauchte IT-Geräte auszuleihen. Für die Verwaltung aller nötigen Informationen rund um den Computerverleih soll eine Software erstellt werden.

Aufgaben

- 1 Im Material 1 ist das vorläufige UML-Klassendiagramm für den Softwareentwurf dargestellt.
- 1.1 Beschreiben Sie die Grundkonzepte der objektorientierten Softwareentwicklung. Es ist dabei darauf einzugehen, wie die Wirklichkeit in einer objektorientierten Software abgebildet wird (Stichworte: Klasse, Objekt, Attribut, Methode, Vererbung, Assoziationsarten, Navigierbarkeit). **(5 BE)**
- 1.2 Beschreiben Sie die im Klassendiagramm dargestellten Beziehungen jeweils an einem Beispiel. **(4 BE)**
- 1.3 Die Schülerin Kaja Njo leih am 28.09.2024 den PC mit Nr. 203 mit folgender Konfiguration aus: Dell M4500, 2,4 GHz, Leistungsindex 7. Der Computer wird voraussichtlich am 01.07.2025 zurückgebracht. Im aktuellen Schuljahr 2024/25 (01.08.2024 bis 31.07.2025) ist sie Schülerin der Klasse 11BGP3 der Schulform BG. Im vergangenen Schuljahr 2023/24 (01.08.2023 bis 31.07.2024) war sie Schülerin der Klasse 12SFE1 der Schulform SFE und hat den gleichen PC vom 01.02.2024 bis 30.06.2024 ausgeliehen und hat diesen zurückgebracht. Entwickeln und zeichnen Sie ein UML-Objektdiagramm unter Berücksichtigung des Klassendiagramms in Material 1.

Hinweis: Fehlende Attributwerte sind sinnvoll zu ergänzen.

(5 BE)

- 1.4 Implementieren Sie die Klassen `Person` und `Lehrkraft`.

Hinweise: Falls es sich bei der Person um einen Schüler handelt, werden in der Liste `meineKlassen` alle Klassen gespeichert, in denen er im Laufe seiner Zeit an der Schule war. Handelt es sich bei der Person um eine Lehrkraft, werden in der Liste alle Klassen gespeichert, in denen sie die Klassenleitung hatte. Jede neue Klasse wird immer an das Ende der Liste gesetzt. Die Methode `hinzufuegenAktuelleKlasse()` fügt nur dann die Klasse hinzu, wenn die übergebene Klassenbezeichnung existiert. Die `toString()`-Methode der Klasse `Lehrkraft` gibt die Werte für das aktuelle Schuljahr in folgender Form zurück:
`Bernd Beyer - Klassenlehrer/in der 11BGP1`

Eine Dokumentation der Klasse `String` und `List` ist in Material 2 zu finden.

(8 BE)

- 1.5 Bei der Berechnung der Gebühr einer Ausleihe gelten folgende Regeln: Es wird die tatsächliche Ausleihdauer in Tagen verwendet. Die Grundgebühr beträgt für 1 bis 14 Tage 1 €, danach 2 €. Die Gebühr für den Leistungsindex (Li) 1 und 2 beträgt 0 €, für 3 bis 6 beträgt sie 1 €, für 7 bis 10 beträgt sie 2 €. Die Gebühr für die Ausleihdauer beträgt 0,75 € pro angefangene 30 Tage. Die ersten 30 Tage sind kostenfrei. Schüler der Schulförm BBE zahlen nur die Grundgebühr. Lehrkräfte, die mehr als 20 Stunden unterrichten, bezahlen für jede Ausleihe 2€ mehr. Implementieren Sie die Methode berechneGebuehr() der Klasse Ausleihe.

Hinweis: Die Methode berechneGebuehr() wird bei der Rückgabe aufgerufen.

(6 BE)

- 1.6 Die Methode sucheFreieComputer() der Klasse Verwaltung gibt eine Liste aller ausleihbaren Computer zurück. Die Methode rueckgabeComputer() der gleichen Klasse aktualisiert die Ausleihe und gibt die entsprechende Leihgebühr zurück. Sollte kein entsprechender Computer ausgeliehen worden sein, wird -1 zurückgegeben. Das Datum der Ausleihe soll dabei aktualisiert werden. Implementieren Sie die Methoden sucheFreieComputer() und rueckgabeComputer() der Klasse Verwaltung.

(6 BE)

- 1.7 Von jedem Computer-Arbeitsplatz der Schule sollen Schüler sich Informationen in Bezug auf die Ausleihe anzeigen lassen können. Dazu wird eine Client Server Software entwickelt.

- 1.7.1 Um einen Überblick über gegebenenfalls verliehene Endgeräte auch von Zuhause zu ermöglichen, soll eine Client-Server-Lösung entwickelt werden. Eine typische Sitzung sieht wie folgt aus:

(> für Antworten vom Server, < für Kommando vom Client, <LF> Enter-Taste):
 >Auswahlmöglichkeiten: 1-freie Geräte, 2-freie Geräte sortiert nach Leistungsindex, 3#Gerät-Nr.-Informationen zum Gerät Nr. ausgeben, 4-Beenden<LF>
 <1<LF>
 >102: Laptop Dell M4500, 2x 2.8 Ghz, Li=7
 107: Laptop Acer 5630Z, 2.2 GHz, Li=2
 <3#1008<LF>
 >Gerät Nr. 1008 nicht vorhanden<LF>
 <3#108<LF>
 >108: Laptop Lenovo X1, 4x 1,9 GHz, Li=7<LF>
 <9<LF>
 >Falsche Eingabe<LF>
 <4<LF>
 >Server sagt tschüss<LF>

Entwickeln und implementieren Sie die Klasse AusleihInfoServer.

Hinweise: Erst wenn der aktuelle Clientdialog abgeschlossen ist, wird der Kontakt zu einem neuen Client ermöglicht. Der Text des Auswahlmenüs kann auf „Menü“ verkürzt werden. Die Port Nummer ist die 4712. Zur Umwandlung eines Strings in eine Integer-Zahl steht die Methode Integer.parseInt(s:String):int zur Verfügung.

(10 BE)

- 1.7.2 Die Methode `sucheAusComSorNacEnd()` der Klasse `Verwaltung` gibt eine Liste der momentan ausgeliehenen Computer zurück, sortiert nach Leihende. Die Liste ist aufsteigend sortiert, d.h. der Computer, dessen voraussichtliches Leihende zuerst kommt, steht am Anfang. Entwickeln und formulieren Sie den Algorithmus für diese Methode sprachlich.

(6 BE)

- 1.8 Computer, die eine bestimmte Anzahl an Tagen verliehen worden sind, haben eine höhere Chance auszufallen. Daher soll eine Methode implementiert werden, die für einen übergebenen Computer die Summe der insgesamt ausgeliehenen Tage ermittelt. Entwickeln und zeichnen Sie ein Sequenzdiagramm in Material 3 der Methode `berechneAusleihtage()` der Klasse `Verwaltung`.

(6 BE)

- 1.9 Diskutieren Sie die Vorteile der Verwendung von Hashfunktionen gegenüber CRCs für die Integrität von Daten. Beschreiben Sie weiterhin zwei zusätzliche kryptografische Eigenschaften von Hashfunktionen im Vergleich zu CRCs.

(4 BE)

- 2 Zur Verwaltung der Ausleihen wurde eine Datenbank entwickelt. Ein Entity-Relationship-Modell (ERM) finden Sie in Material 4.

- 2.1 Benennen Sie die Notationselemente des ERM jeweils an einem Beispiel und beschreiben Sie die vorhandenen Beziehungen mit ihren Kardinalitäten.

(4 BE)

- 2.2 Überführen Sie das ERM in das Relationale Modell. Beschreiben Sie jede notwendige Transformationsregel.

Hinweis: Alle Relationen sind in der Schreibweise `Relation (PK, Attribut, ..., FK#)` anzugeben.

(6 BE)

- 2.3 Einige Daten der Datenbank sollen ausgewertet werden.

- 2.3.1 Die Liste der Wartungen im letzten Jahr, bei denen in Beschreibung ‚HDD‘ oder ‚SSD‘ enthalten ist, sollen ausgegeben werden. Entwickeln Sie eine SQL-Anweisung.

(3 BE)

- 2.3.2 Entwickeln Sie die SQL-Anweisung, mit der die Tabelle `Wartung` erzeugt wird. Geben Sie auch die Primär- und Fremdschlüssel Definitionen an. Die Bezeichner der Schlüssel sollen sinnvoll festgelegt werden. Die Beschreibung soll aus maximal 200 Zeichen bestehen.

(3 BE)

- 2.3.3 Zur Aufteilung der Kosten der Wartungen auf die verschiedenen Schulformen soll die Anzahl der Ausleihen gruppiert nach Schulform, denen die ausleihenden Schüler zum Zeitpunkt der Ausleihe angehörten, ausgegeben werden. Die Anzahl soll für das Schuljahr 2023/24 ermittelt werden. Die Überschrift für die 1. Spalte soll „Anzahl Ausleihen“ lauten, danach soll die Schulform-ID und Schulform-Bezeichnung ausgegeben werden.
Entwickeln Sie die SQL-Anweisung.

Hinweise: Es ist davon auszugehen, dass ein Schüler in einem Schuljahr nur eine Schulform besucht. Das Schuljahr 2023/24 war vom 01.08.2023 bis 31.07.2024.

(4 BE)

- 2.3.4 Die drei am wenigsten ausgeliehenen Computer sollen aussortiert werden. Dazu soll eine Liste mit der ID, der Bezeichnung des Computers und der Summe der Tage, die der Computer insgesamt ausgeliehen worden ist, ausgegeben werden, aufsteigend sortiert nach der Summe der Ausleihstage.

Entwickeln Sie die SQL-Anweisung. Erörtern Sie ein mögliches Problem, das dabei auftreten kann.

Hinweise: Computer, die bisher nicht ausgeliehen wurden, müssen nicht beachtet werden.
Die Funktion `DATEDIFF('dd', start, ende)` gibt die Anzahl der Tage zurück, die zwischen `start` und `ende` liegen.

(4 BE)

- 2.3.5 Erläutern Sie die Funktionsweise von `LEFT JOIN`. Beschreiben Sie die Ausgabe, die folgende Abfrage erzeugt:

```
SELECT Computer.pcid, bezeichnung, von, bis
FROM Computer
LEFT JOIN Leiht ON Computer.pcid = Leiht.pcid
WHERE von IS NULL
```

(2 BE)

- 2.4 In der Tabelle in Material 5 wird von der Werkstatt das Wartungsmaterial verwaltet, das eingekauft und/oder eingebaut wurde.

- 2.4.1 Die Tabelle enthält Redundanzen und Inkonsistenzen. Zeigen Sie diese und beschreiben Sie mithilfe von Beispielen aus der Tabelle die verschiedenen Arten von Anomalien.

(4 BE)

- 2.4.2 Diskutieren Sie die Funktion des mehrfach auftretenden Werts 1070 bei der Einkaufs-Rechnungs-Nr..

(2 BE)

- 2.4.3 In der vorliegenden Tabelle sind `null`-Werte enthalten. Diskutieren Sie unter diesem Gesichtspunkt das Datenbankdesign. In der Diskussion ist auch auf alternative Tabellendesigns einzugehen, mit deren Vor- und Nachteilen, um `null`-Werte zu vermeiden.

(3 BE)

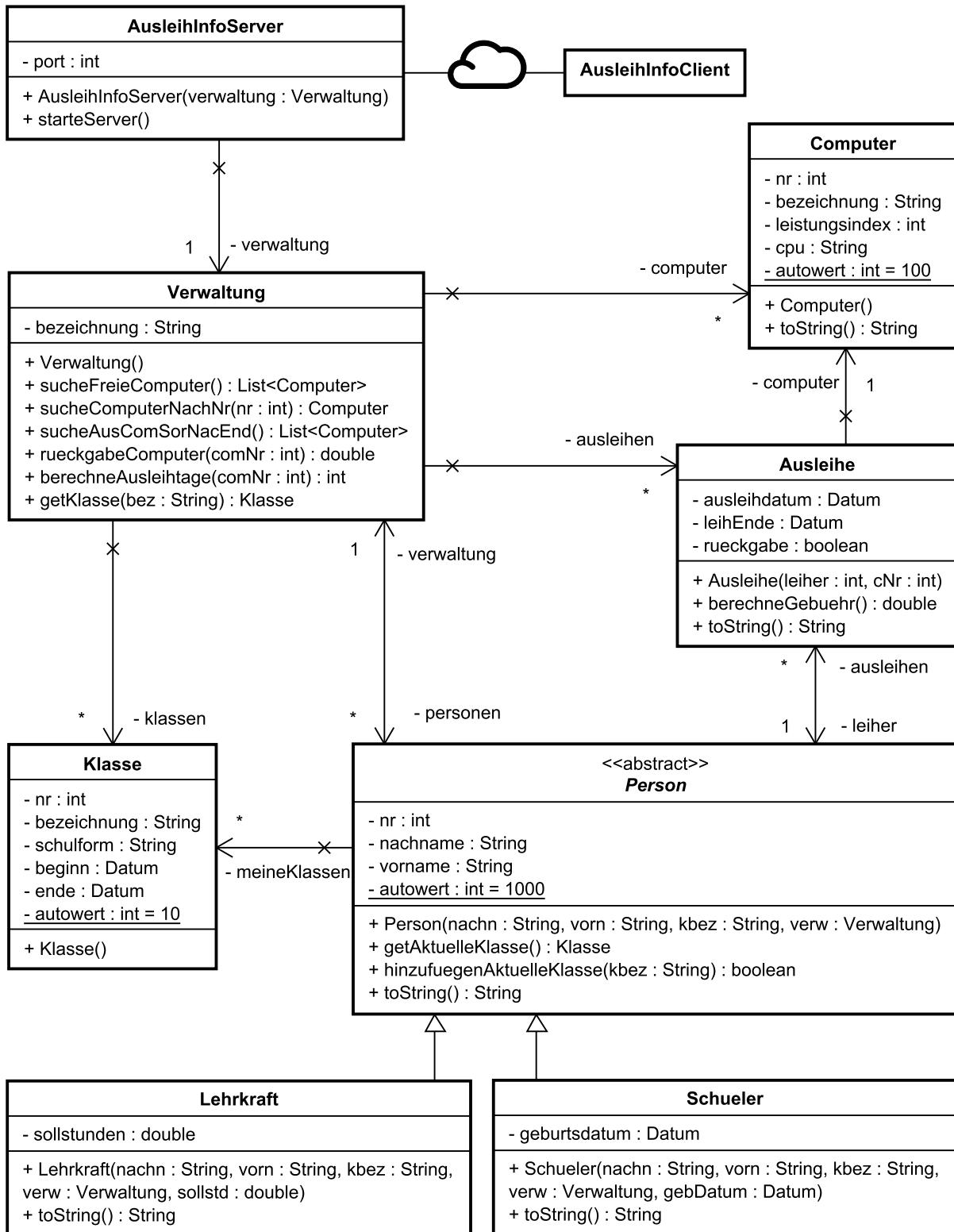
2.5 Die Datenbank soll um weitere Anforderungen erweitert werden:

- Ein Schüler kann eine Patin oder einen Paten (im Folgenden Pate genannt) für mehrere Schüler sein, um jenen bei Computerproblemen zu helfen. Ein Schüler darf maximal einen Paten haben.
- Zu jeder Wartung gehört genau ein Auftrag. Ein Auftrag wird mit einer eindeutigen ID, dem Erstelldatum, Fertigstelldatum und Beschreibung gespeichert. Jeder Auftrag wird von ein bis zwei Schülern aufgenommen.
- Jede Durchführung einer Arbeit gehört zu genau einer Wartung und hierzu werden eine eindeutige ID, die verwendete Zeit, das Datum der Ausführung, eine Bemerkung und der Typ der Tätigkeit gespeichert. Eine Durchführung wird von genau einem Schüler ausgeführt. Zu einer Wartung kann es mehrere Durchführungen geben.
- Der Typ der Tätigkeit wird mit einer eindeutigen ID, der Richtzeit und Bezeichnung gespeichert.

Entwickeln und zeichnen Sie die Erweiterung des ERM.

Hinweise: Das Diagramm ist mit notwendigen Entitätstypen und Beziehungen mit Kardinalitäten in der [min, max]-Notation darzustellen. Attribute müssen nur dargestellt werden, wenn sie neu hinzukommen.

(5 BE)

Material 1**UML-Klassendiagramm**

Material 1 (Fortsetzung)

Hinweise:

Klasse Verwaltung

- `sucheAusComSorNacEnd()` gibt eine Liste ausgeliehener Computer aufsteigend, sortiert nach voraussichtlichen Enddatum der Ausleihe, zurück.

Klasse Verwaltung

- `hinzufuegenAktuelleKlasse()` gibt `true` zurück, wenn die entsprechende Klasse gefunden und die Person der Klasse hinzugefügt wurde. Andernfalls `false`.

Klasse Computer

- `toString()` ergibt eine Zeichenkette der Form:
`bezeichnung, cpu, leistungsindex`

Auf alle Attribute kann mittels get-Methoden und auf die Attribute `voraussichtlicheLeihEnde` und `rueckgabe` der Klasse Ausleihe kann mittels set-Methoden zugegriffen werden.

Material 2

Klassendokumentationen

Klasse Datum

Datum()

erzeugt ein Datum-Objekt mit dem aktuellen Systemdatum

differenzInTagen(d1: Datum, d2: Datum): int

Gibt den Abstand in Tagen zwischen d1 und d2 als int-Wert zurück. Liegt d1 nach d2, ist der Wert positiv, liegt d1 vor d2 ist der Wert negativ. Bei Gleichheit ist der Wert 0.

Datum
+ Datum() + differenzInTagen(d1 : Datum, d2 : Datum) : int

Klasse String

equals(str: String): boolean

liefert true, wenn beide Strings gleich sind, andernfalls false.

split(str: String): String[]

Teilt einen String am Trennzeichen str. Die Teil-Strings werden in einem Feld zurückgeliefert.

charAt(pos: int): char

gibt den Character an der entsprechenden Position zurück.

String
+ equals(str : String) : boolean + split(str : String) : String[] + charAt(pos : int) : char

Klasse List

List<T>()

erzeugt eine generische Liste mit Elementen des Typs T.

add(obj: T)

hängt das Objekt obj vom Typ T am Ende der Liste an.

add(index: int, obj: T)

fügt das Objekt obj vom Typ T an der Position index in die Liste ein.

get(index: int): T

liefert das Listenelement an der Position index zurück bzw. null, falls index negativ oder größer gleich der Anzahl der momentan enthaltenen Elemente ist.

set(index: int, obj: T)

Ersetzt das Objekt an der Position index mit dem Objekt obj vom Typ t.

size(): int

liefert die Anzahl der Elemente in der Liste zurück.

List<T>
+ List<T>() + add(obj : T) + add(index : int, obj : T) + get(index : int) : T + set(index : int, obj : T) + size() : int

Material 2 (Fortsetzung)**Klasse ServerSocket**

`ServerSocket(localPort: int)`
 erzeugt einen Server-Socket und bindet ihn an Port localPort.
`accept(): Socket`
 wartet darauf, dass ein Client eine Verbindung aufbauen will. Wurde eine Verbindung aufgebaut, liefert die Methode das entsprechende Socket-Objekt.

ServerSocket
- localPort : int
+ ServerSocket(localPort : int) + accept() : Socket

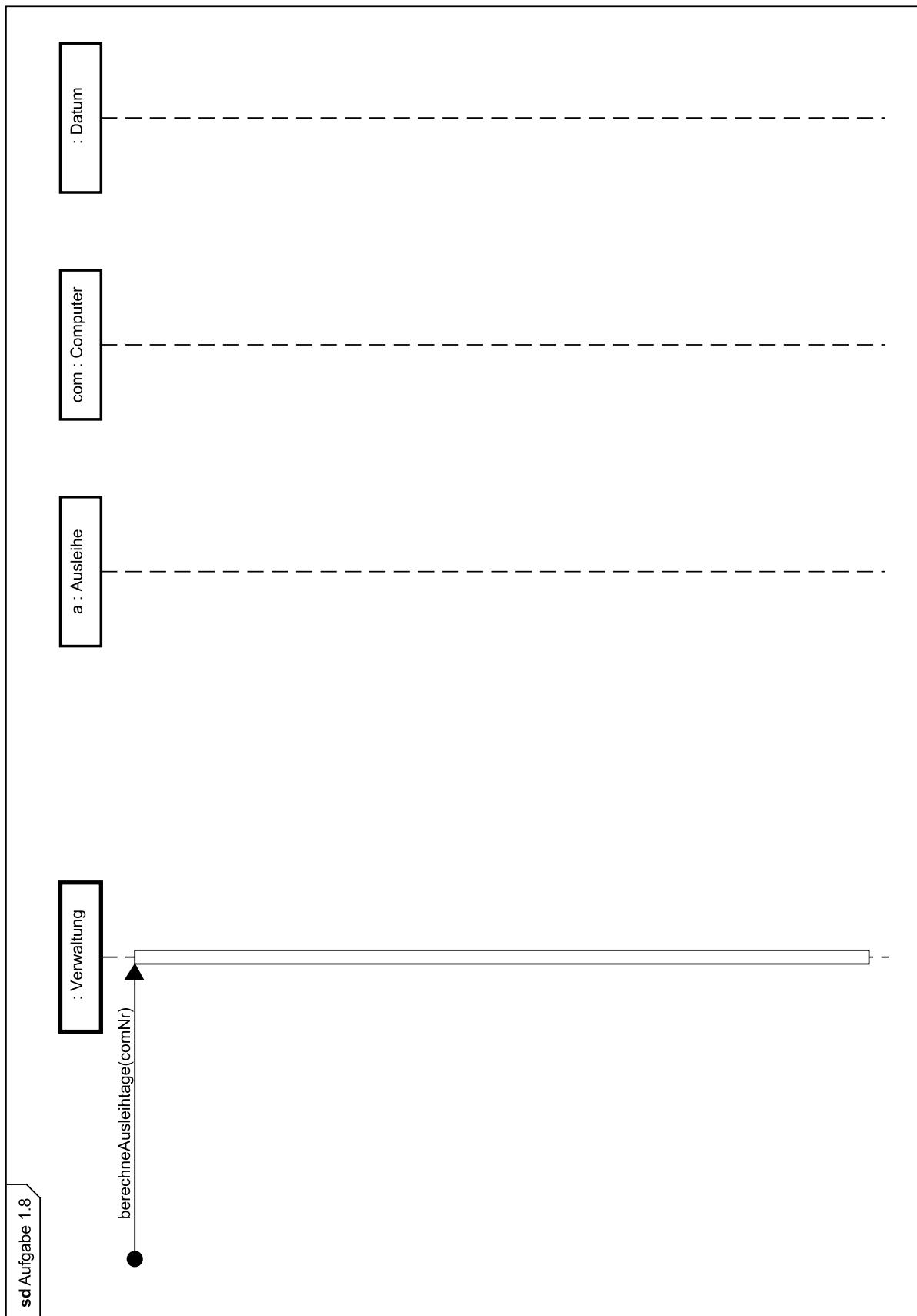
Klasse Socket

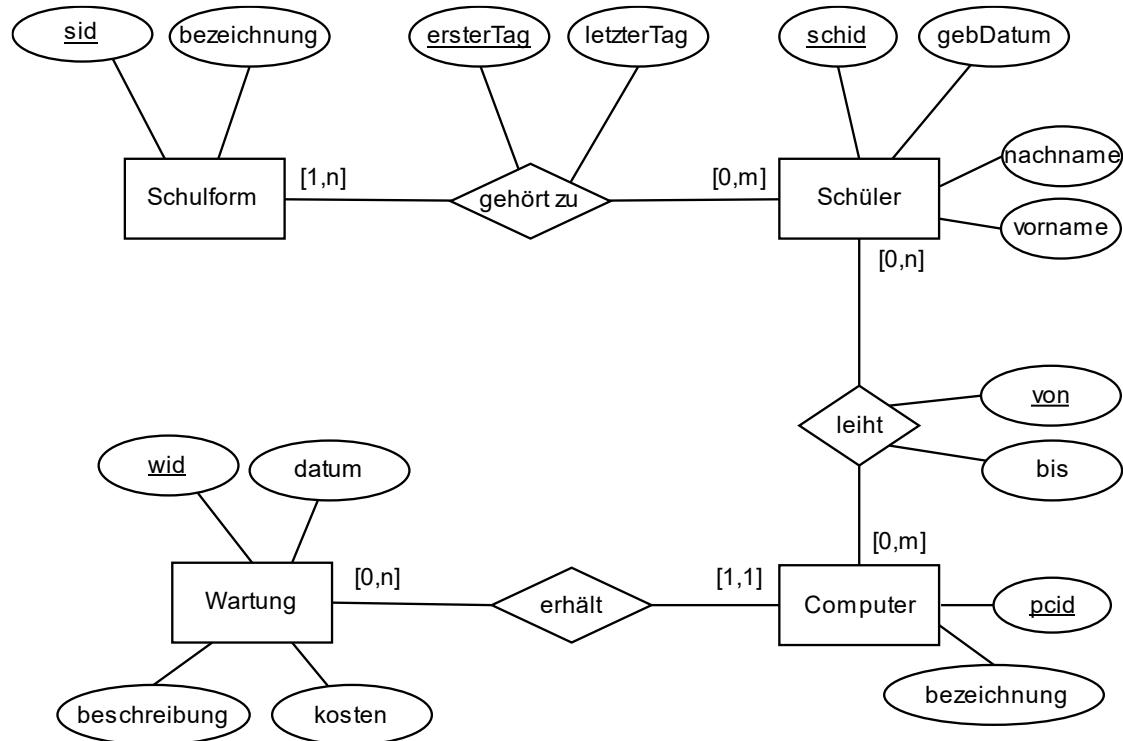
`Socket()`
 erzeugt einen Socket-Objekt.
`readLine(): String`
 liest eine Zeile vom Socket, bzw. liefert null, wenn der Socket nicht geöffnet ist. Eine Zeile wird durch ein Zeilenendezeichen abgeschlossen; das Zeilenendezeichen wird jedoch nicht in den zurückgegebenen String übernommen. Die Methode blockiert, bis eine komplette Zeile eingelesen ist.
`write(s: String)`
 schreibt einen String zum Socket; ist die Schnittstelle nicht geöffnet geschieht nichts.
`close()`
 löst die Verbindung auf.

Socket
+ Socket() + readLine() : String + write(s : String) + close()

Material 3

Vorlage für das Sequenzdiagramm der Methode berechneAusleihTage()



Material 4**ERM Computerverleih**

Hinweise: Die Wartung beinhaltet alle durchgeführten Arbeiten, wie Installation, Reparatur oder Aufrüstung eines Computers. In beschreibung werden alle dabei durchgeführten Tätigkeiten und verwendeten Materialien in Textform gespeichert.

Material 5**Ausschnitt der Tabelle für die Materialverwaltung der IT-Werkstatt**

Nr.	Bezeichnung	Preis	Eingang	Ausgang	Verwendet für Gerät Nr.	Verantwortlich für den Einkauf	ID	Einkaufs-Rechnungs-Nr.
...								
17	SSD Corsair 120GB	14,99	01.07.24	01.11.24	101	Karl Müller	3102	1070
18	SSD Corsair 240GB	24,99	01.07.24	null	null	Karl Mueller	3102	1070
19	Akku für DELL M4500	38,50	10.09.24	11.09.24	142	Peter Doban	3103	1071
20	SSD Corsair 120GB	14,99	10.09.24	null	null	Peter Doban	3103	1071
21	SSD Corsair 120GB	14,99	05.11.24	null	null	Karl Müller	3102	1075
22	Akku für Lenovo X1	52,82	12.12.24	null	null	Don John	4327	1098

Hinweise: Die Rechnungen des Einkaufs werden intern fortlaufend nummeriert. Der Einkauf wird bei verschiedenen Großunternehmen getätigkt.