

Dieses Übungsblatt dient der Vorbereitung auf die Klausur. Die Lösungen der Aufgaben sollen nicht abgegeben werden. Die Aufgaben werden in den Übungen der Woche vom 23.4.–27.4.2018 besprochen.

Übungsblatt 1

Aufgabe 1

Klassendiagramm

Modellieren Sie die folgende Situation als Klassendiagramm:

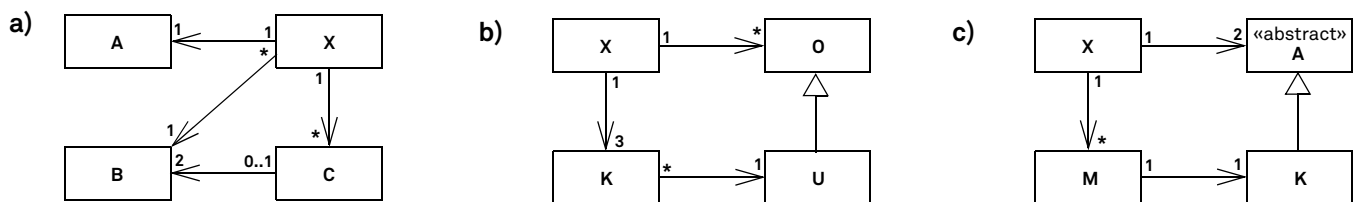
Studierende nehmen an Lehrveranstaltungen teil. Für jede Teilnahme an einer Lehrveranstaltung wird ein Prüfungsdatensatz angelegt. Für jede Lehrveranstaltung können bis zu drei Prüfungen notwendig sein. Jede Prüfung führt zu einer Note und hat zwei Prüfer und ein Prüfungsdatum. Jede Lehrveranstaltung ist einem Semester zugordnet und setzt eine Modulbeschreibung um, die aus einer Modulnummer und einem Text besteht.

Die Modellierung soll es ermöglichen, festzustellen, welcher Studierende in welchem Semester zu welchen Lehrveranstaltungen Prüfungen bei welchem Prüfer abgelegt hat. Ebenso soll es möglich sein, für jede Lehrveranstaltung festzustellen, welche Studierende welche Ergebnisse erzielt haben.

Aufgabe 2

Objektdiagramm

Gegeben sind die folgenden drei Klassendiagramme. Geben Sie für jedes Klassendiagramm ein Objektdiagramm an, das genau fünf Objekte enthält, von denen mindestens eines zur Klasse X gehört.



Aufgabe 3

Sequenzdiagramm

Gehen Sie davon aus, dass die in den folgenden Methoden benutzten Klassen und Methoden ihren Definitionen entsprechend verwendet und aufgerufen werden. Erstellen Sie jeweils ein **Sequenzdiagramm**, das den Ablauf eines Aufrufs der in der Teilaufgabe genannten Methode zeigt. Alle Methodenaufrufe innerhalb der Sequenzdiagramme sollen durch Pfeile visualisiert werden.

a) Visualisieren Sie einen Aufruf der Methode `remove`.

```
public Item remove(List l, int x, DataPool p) {
    Item it = null;
    if (x < l.size()) {
        it = p.detach(l.atPosition(x));
    }
    return it;
}
```

b) Visualisieren Sie einen Aufruf der Methode `average`.

```
public double average(Iterator<Data> it) {
    double result = 0.0;
    double count = 0.0;
    while (it.hasNext()) {
        result += it.next().getValue();
        count++;
    }
    return result / count;
}
```