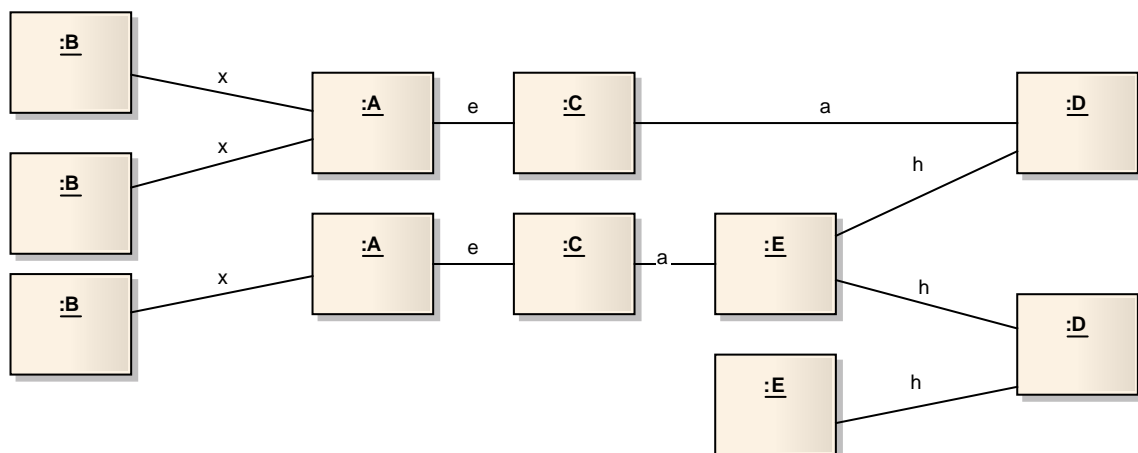
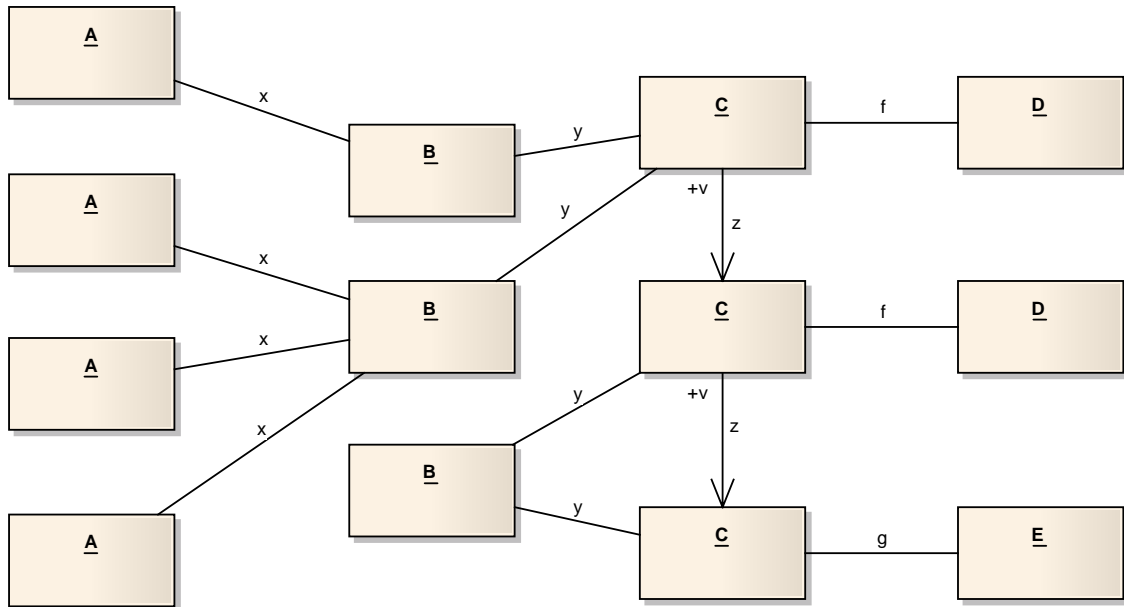


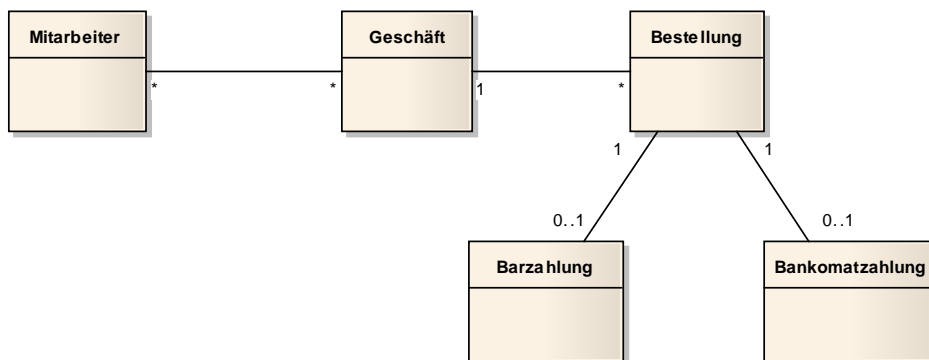
Aufgabe 1: Objektdiagramm

Entwerfen Sie zwei Klassendiagramme, zu dem diese Objektdiagramme konform ist. Beschriften Sie die Kardinalitäten der Beziehung mit 1, 0..1, * oder 1..*. Sie können davon ausgehen, dass dieses Objektdiagramm die höchst- und niedrigstzulässige Anzahl an Beziehungen mit Objekten einer anderen Klasse darstellt. Vergessen Sie nicht auf die Benennung der Assoziationen/Rollen. Achten Sie auf mögliche XOR-Einschränkungen, Generalisierungen und Navigationsrichtungen!



Aufgabe 2: Fehlerhafte Klassendiagramme

- (1) Gegeben ist das nachfolgende Klassendiagramm. Leider sind beim Kopieren des Diagramms einige Konstrukte verloren gegangen. Erweitern Sie das Klassendiagramm so, dass folgende Punkte korrekt abgebildet sind.
- a) Ein Mitarbeiter ist entweder Zweigstellenleiter oder Angestellter.
 - b) Eine Bestellung beinhaltet mehrere Massenartikel.
 - c) Ein Geschäft hat mehrere Zulieferer.
 - d) Eine Bestellung kann bar oder mit Bankomatkarte bezahlt werden.
 - e) Für jedes Beschäftigungsverhältnis gibt es einen eigenen Vertrag mit Beginndatum, Enddatum und Gehalt.

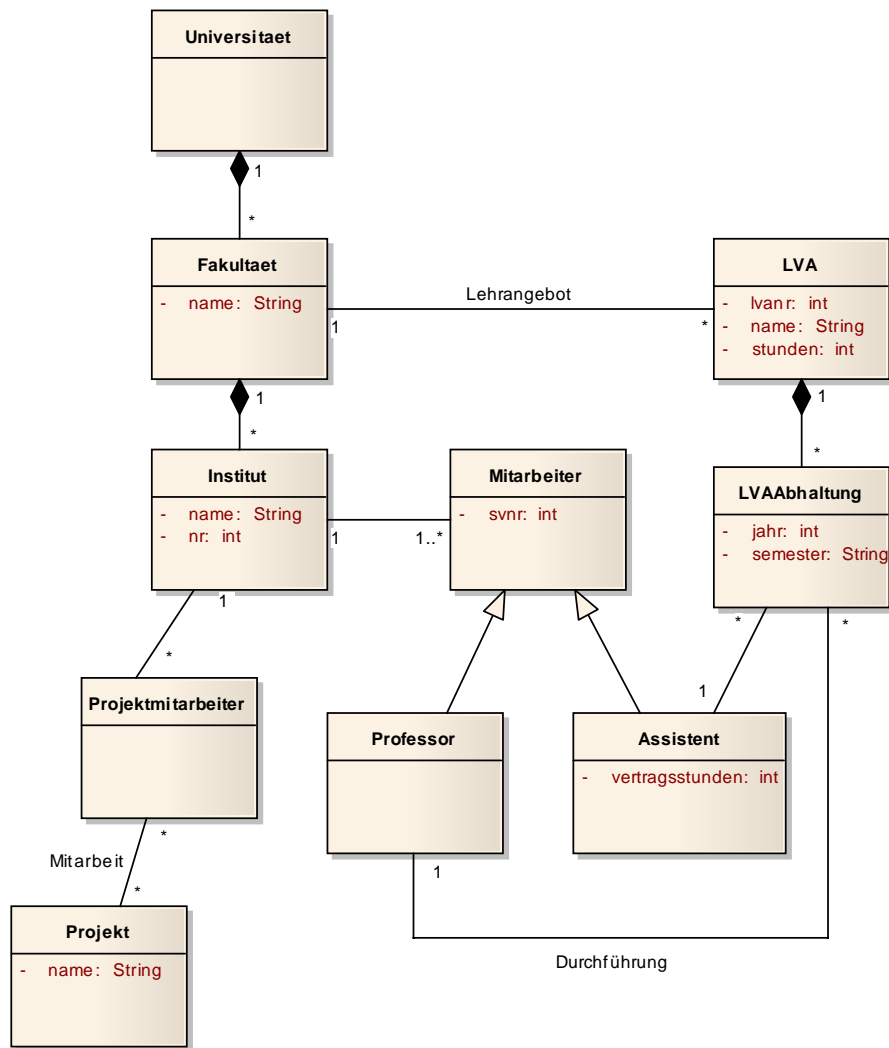


- (2) Gegeben ist folgendes UML Klassendiagramm, welches gewisse Aspekte der universitären Verwaltung abbildet. Bei der Modellierung sind leider einige Fehler passiert. Finden Sie 5 und korrigieren Sie diese im Diagramm.

Eine Universität besitzt eine oder mehrere Fakultäten, welche durch einen Namen eindeutig identifiziert werden. Fakultäten bestehen aus beliebig vielen Instituten, welche einen Namen und eine Institutsnummer haben. Mitarbeiter sind durch Name und SVNr ausgezeichnet und sind einem Institut zugeordnet.

Es wird zwischen Projektmitarbeitern, Professoren und Assistenten unterschieden. Assistenten haben ein Attribut Vertragsstunden. Projektmarbeiter arbeiten an beliebig vielen Projekten mit. Projekte können wiederum aus beliebig vielen Teilprojekten bestehen.

Das Lehrangebot einer Fakultät ergibt sich aus beliebig vielen Lehrveranstaltungen (LVAs). LVAs haben eine LVA-Nummer, einen Namen und eine Stundenanzahl. Lehrveranstaltungen werden entweder von einem Professor oder einem Assistenten abgehalten. Für eine Abhaltung werden Semester und Jahr gespeichert.



Aufgabe 3: Klassendiagramm: Hotelverwaltung

Modellieren Sie folgenden Sachverhalt mittels UML2-Klassendiagramm. Stellen Sie die beschriebenen (abstrakten) Klassen und ihre Attribute (inklusive deren Typen), sowie die Beziehungen zwischen den Klassen. Verwenden Sie Assoziationen, schwache und starke Aggregationen sowie Generalisierungsbeziehungen in Ihrem Diagramm dar.

Von einem Hotel sind Name, Adresse und Kategorie (*, **, ***, ****, *****) bekannt. Die Zimmer eines Hotels haben eine Zimmernummer und eine gewisse Bettenanzahl. Außerdem hat das Hotel Seminarräume und Ballsäle. Ein Raum kann ein Raucher- oder Nichtraucherzimmer sein. Von Mitarbeitern und Gästen sind Name und Adresse bekannt. Mitarbeiter arbeiten in einem oder in zwei Hotels. Es ist bekannt, wie viel sie jeweils verdienen. Gäste können Zimmer für einen gewissen Zeitraum (Start- und Enddatum) buchen. In einem Hotel können Veranstaltungen abgehalten werden, die sich aus Teilveranstaltungen zusammensetzen können. Diese finden entweder in Seminarräumen oder in Ballsälen statt. Die Teilnehmer an den Veranstaltungen werden genauso wie Gäste verwaltet.

Aufgabe 4: Reverse Engineering

Gegeben sei der unten angeführte Java Code. Führen Sie ein Reverse Engineering des Codes in ein UML-Klassendiagramm durch. Das heißt, Sie müssen ein UML-Klassendiagramm entwerfen, das semantisch dem Java Code entspricht. Konstruktoren brauchen Sie in Ihrem Diagramm nicht zu berücksichtigen. Modellieren Sie die Beziehungen zwischen den Klassen mittels Assoziationen.

```
import java.util.ArrayList; import java.util.Hashtable;
```

```
class Firma {  
    public String name;  
    public Mitarbeiter chef;  
    private ArrayList mitarbeiter;  
  
    public Firma (String name, Mitarbeiter chef) {  
        this.name = name;  
        this.chef = chef;  
        mitarbeiter = new ArrayList();  
    }  
}
```

```
class Projekt {  
    public ArrayList mitarbeiter;  
  
    public Projekt() {  
        mitarbeiter = new ArrayList();  
    }  
  
    void addMitarbeiter(Mitarbeiter m) {  
        mitarbeiter.add(m);  
    }  
}
```

```
abstract class Person {  
    private String vorname;  
    private String nachname;  
  
    public String getVorname() {  
        return vorname;  
    }  
  
    public void setVorname(String vorname) {  
        this.vorname = vorname;  
    }  
  
    public String getNachname() {  
        return nachname;  
    }  
  
    public void setNachname(String nachname) {  
        this.nachname = nachname;  
    }  
  
    public Person(String vorname, String nachname) {  
        super();  
        this.vorname = vorname;  
        this.nachname = nachname;  
    }  
}
```

```
class Mitarbeiter extends Person {  
    private Firma firma;
```

```

public Hashtable projekte;

public Mitarbeiter(String vorname, String nachname, Firma firma) {
    super(vorname, nachname);
    this.firma = firma;
    projekte = new Hashtable();
}

public void addProjekt(Projekt p, String infos) {
    projekte.put(p, infos);
}
}

```

Aufgabe 5: Klassendiagramm: Kino

Modellieren Sie folgenden Sachverhalt - der auch in der Übung aus Datenmodellierung in diesem Semester verwendet wird - mittels UML2-Klassendiagramm.

In einem Kino soll die Verwaltung in Zukunft automatisiert erfolgen. Ihre Aufgabe ist es nun dieses System entsprechend der folgenden Informationen zu modellieren.

Die Säle des Kinos sind durch einen Namen eindeutig identifiziert. Für jeden Saal ist die Größe der vorhandenen Leinwand bekannt sowie ob dieser Saal einen digitalen oder analogen Projektor besitzt. Optional kann zu einem Saal auch ein Plan als Bilddatei gespeichert werden. Jeder Saal hat 20 bis 200 Sitzplätze.

Von den Plätzen sind jeweils Reihe und Sitznummer bekannt, wobei die Kombination aus Reihe und Nummer einen Sitzplatz pro Saal eindeutig identifiziert (Ein bestimmter Sitzplatz des Kinos kann also beispielsweise durch Großer Kinosaal 1, Reihe 7, Platz 13 angegeben werden).

Die im Kino gespielten Filme haben eine eindeutige ID, Titel, Erscheinungsjahr, Spieldauer sowie Alter, ab dem sie freigegeben sind. Um vermerken zu können, wann welcher Film in welchem Saal gespielt wird, gibt es Filmslots. Jeder Filmslot ist abhängig von dem zugehörigen Saal. Zusätzlich müssen Datum und Uhrzeit zur Identifizierung bekannt sein. Für jeden Filmslot wird vermerkt, welcher Film gespielt wird. Ein Film wird zu einem Slot in maximal zwei Sälen gespielt.

Das Kino vermerkt alle registrierten Kunden. Für jeden Kunden gibt es eine eindeutige ID, weiters werden Vorname, Nachname, Adresse, Geburtsdatum, E-Mailadresse, das Datum der Registrierung und bisher gesammelte Treuepunkte gespeichert.

Die verkauften Tickets werden mit einer fortlaufenden Nummer gekennzeichnet. Weiters wird das Verkaufsdatum vermerkt. Jedes Ticket gehört zu einer Ticketgruppe, in welcher der Preis und die eindeutige Bezeichnung (z.B. Studentenkarte, Vollpreiskarte, ...) vermerkt werden. Für jedes Ticket ist natürlich bekannt, welcher Platz damit verkauft wird. Für welchen Film das Ticket gültig ist, wird durch eine Zuordnung zum entsprechenden Filmslot festgelegt. Weiters soll gespeichert werden, an welcher Kassa ein Ticket verkauft wurde. Für jede Kinokassa ist ihre eindeutige Schalternummer bekannt. Für Tickets, welche an registrierte Kunden verkauft werden, soll es möglich sein, zu speichern, welcher Kunde an welcher Kassa welches Ticket gekauft hat.

Aufgabe 6: Theoriefragen

Wiederholen Sie das Kapitel aus der Vorlesung, das sich mit Strukturmodellierung beschäftigt und beantworten Sie folgende Fragen.

- Welche Datentypen unterstützt UML? Wie können eigene Datentypen definiert werden? Was ist der Unterschied zwischen einer Klasse und einem Datentyp? Was sind Enumerationen und wofür werden sie eingesetzt?
- Was ist ein Metamodell und wofür wird ein Metamodell benötigt? Ist die UML erweiterbar? Wenn ja, welche Möglichkeiten gibt es?
- Was ist ein Paketdiagramm? Wozu dienen die Pakete und wofür werden sie eingesetzt?
- Wie können (abstrakte) Klassen, Generalisierungen, Assoziationen und starke Aggregationen in eine objekt-orientierte Programmiersprache wie Java übersetzt werden?