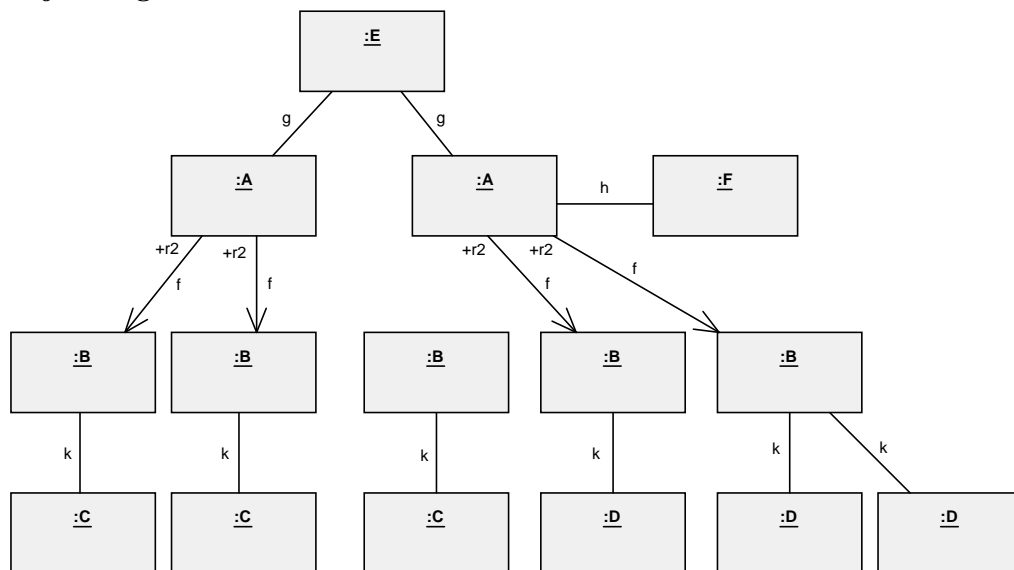
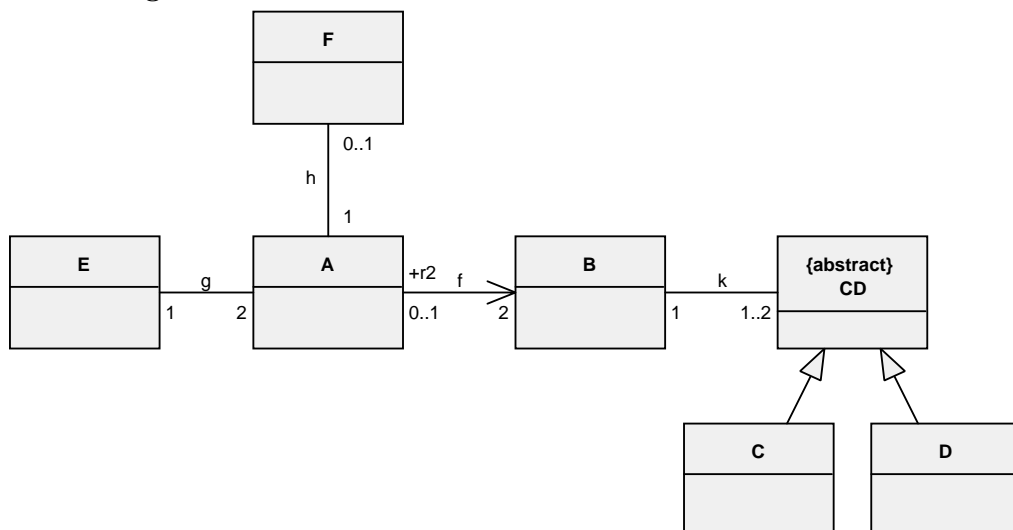
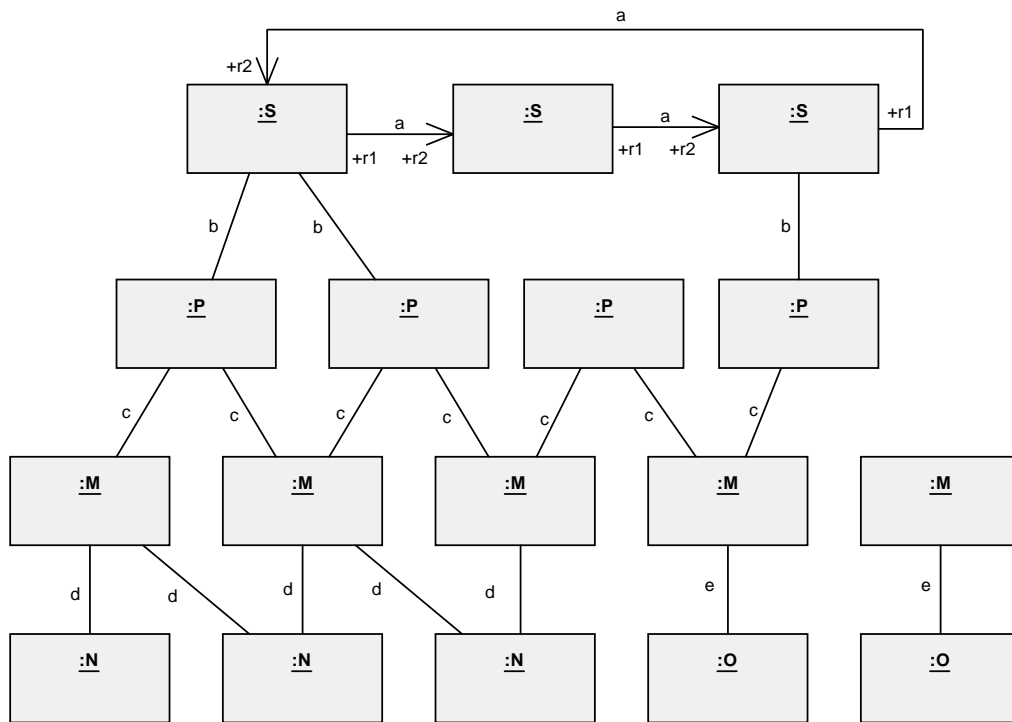


Aufgabe 1: Objektdiagramm

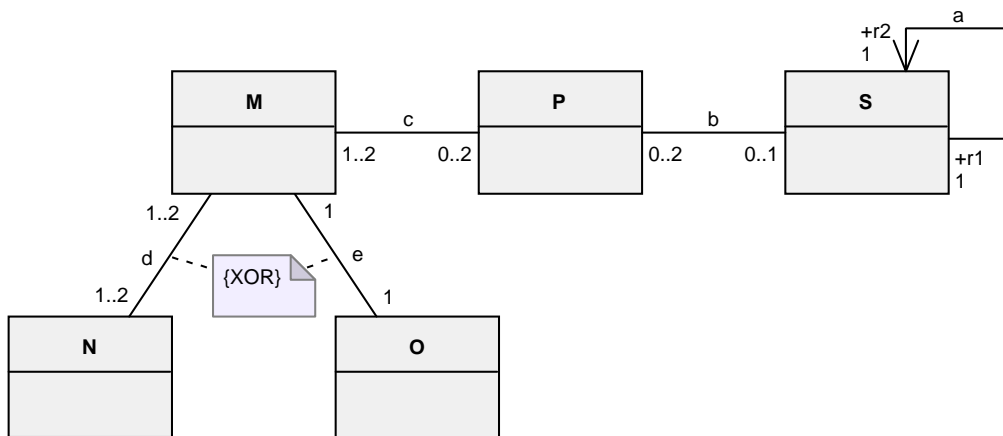
Entwerfen Sie zwei Klassendiagramme, zu denen nachfolgende Objektdiagramme konform sind. Wählen Sie die Kardinalitäten an den Assoziationsenden möglichst genau. Sie können davon ausgehen, dass diese Objektdiagramme die höchstzulässige Anzahl an Beziehungen mit Objekten einer anderen Klasse darstellen. Weiters sollen mögliche Generalisierungen bzw. XOR-Beziehungen erkannt werden.

• **Objektdiagramm 1:****Klassendiagramm 1:**

• Objektdiagramm 2:



Klassendiagramm 2:



Aufgabe 2: Vergleich von Klassendiagrammausschnitten

Erklären Sie den Unterschied zwischen folgenden Klassendiagrammausschnitten.

a)



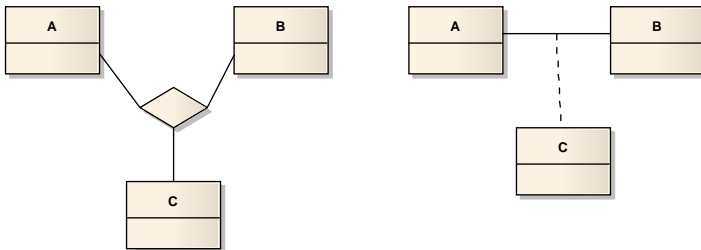
b)



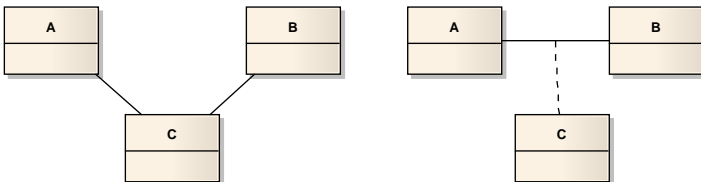
c)



d)



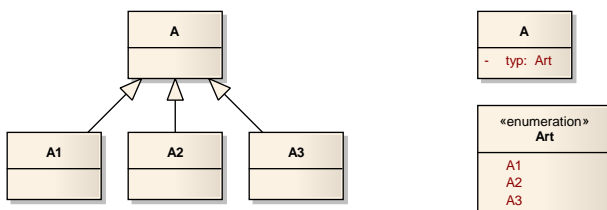
e)



f)



g)



Aufgabe 3: Reverse Engineering

Gegeben sei der unten angeführte Java ähnliche 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. Bilden Sie Referenzen möglichst durch Assoziationen ab.

```
abstract class Universitaetsangehoeriger {
    public String vorname;
    public String nachname;
    public int svnr;
}

class Student extends Universitaetsangehoeriger {
    public int matrnr;
    public LVA-Abhaltung [] abhaltungen;
}

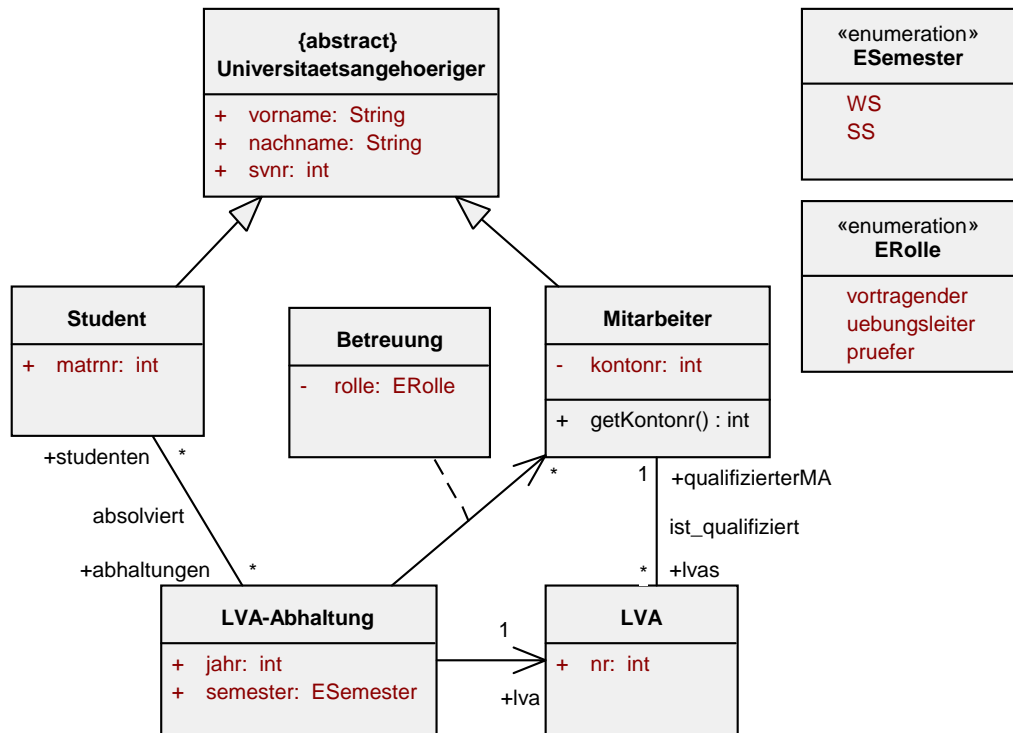
class Mitarbeiter extends Universitaetsangehoeriger {
    private int kontonr;
    public LVA [] lvas;
    public int getKontonr() {...}
}

class LVA-Abhaltung {
    public int jahr;
    public ESemester semester;
    public Student [] studenten;
    public LVA lva;
    public Hashtable betreuung;
        // Key: Mitarbeiter (Typ: Mitarbeiter)
        // Value: Rolle (Typ: ERolle)
}

class LVA {
    public int nr;
    Mitarbeiter qualifizierterMA;
}

Enumeration ESemester {
    WS;
    SS;
}

Enumeration ERolle {
    vortragender;
    uebungsleiter;
    pruefer;
}
```



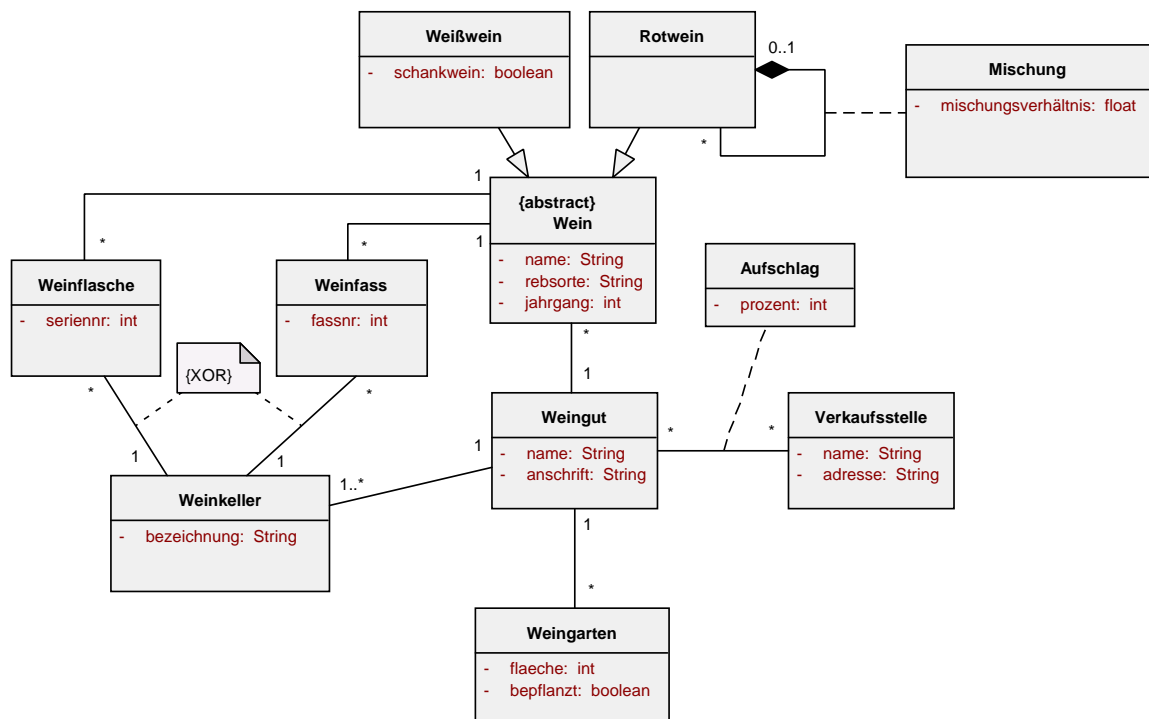
Aufgabe 4: Weingut

Modellieren Sie folgenden Sachverhalt mittels UML Klassendiagramm gemäß folgenden Informationen:

Jedes Weingut hat einen Namen und eine Anschrift. Ein Weingut ist im Besitz mehrerer Weingärten, von jedem Weingarten ist die Fläche bekannt und man weiß, ob ein Weingarten zur Zeit bepflanzt ist oder nicht. Neben den Weingärten verfügt ein Weingut auch über zumindest einen Weinkeller, ein Weinkeller gehört zu einem Weingut und hat eine Bezeichnung. In einem Weinkeller werden entweder mehrere Weinflaschen oder mehrere Weinfässer gelagert. Eine Weinflasche besitzt eine Seriennummer. In einer Weinflasche befindet sich ein bestimmter Wein, ein Wein ist Inhalt mehrerer Flaschen. Jedes Weinfass hat eine Fassnummer. In einem Weinfass wird ein bestimmter Wein aufbewahrt, jeder Wein ist in mehreren Fässern vorhanden.

Ein Weingut produziert mehrere Weine, ein spezieller Wein ist immer einem einzigen Weingut zuzuordnen. Von jedem Wein kennt man den Namen, die Rebsorte und den Jahrgang. Weiters gibt es genau zwei Arten von Weinen: Weißwein und Rotwein. Von einem Weißwein ist bekannt, ob dieser ein Schankwein ist oder nicht. Es gibt auch Rotweine, die aus mehreren, verschiedenen Rotweinen bestehen, Cuvee genannt. Das Mischungsverhältnis zwischen den verschiedenen Rotweinen muss bekannt sein.

Die Produkte eines Weingutes werden an mehreren Verkaufsstellen angeboten, eine Verkaufsstelle bietet natürlich die Produkte von mehreren Weinguten an. Eine Verkaufsstelle besitzt einen Namen sowie eine Adresse. Der Prozentsatz, den eine bestimmte Verkaufsstelle auf die Preise von einem bestimmten Weingut aufschlägt, muss gespeichert werden.



Aufgabe 5: Lego Sammlung I

Folgendes Beispiel ist Ihnen bereits in einer ähnlichen Form aus der LVA Datenmodellierung bekannt:

Ein passionierter Lego-Sammler und -Fan beauftragt Sie mit dem Entwurf eines Verwaltungssystems für Legosammlungen. Bilden Sie die folgenden Sachverhalte in einem Klassendiagramm ab:

Ein Sammler, von dem Name und Sozialversicherungsnummer (svnr) gespeichert werden, kann mehrere Bausteintypen besitzen. Jeder Bausteintyp besitzt eine eindeutige Nummer (nr) und eine Beschreibung (beschr). Bausteintypen gibt es in verschiedenen Farben (rot, blau, gruen).

Bausteintypen werden in Kisten mit eindeutigen Nummern (nr) und einer Beschriftung (beschr) aufbewahrt. Gehen Sie davon aus, dass jeder Bausteintyp genau einer Kiste zugeordnet ist, eine Kiste aber verschiedene Bausteintypen beinhalten kann.

Bausteintypen werden gewöhnlich in Form von Sets vertrieben, die alle eine eindeutige Produktnummer (pnr), eine Bezeichnung (bez) und ein Erscheinungsjahr (jahr) haben. Für jedes Set wird vermerkt, ob es momentan vergriffen ist oder nicht (vergriffen).

Es gibt genau zwei verschiedene Arten von Sets. Neben Sets, die im Handel einzeln erwerblich sind und von denen der Listenpreis (lpreis) gespeichert werden soll, gibt es auch Sets, die einzeln nicht gekauft werden können, sondern nur in Zusammenhang mit Produkten anderer Firmen erhältlich sind (z.B. als „gratis Beigaben“ zu Frühstücksflocken), so genannte Werbesets. Speichern Sie bei dieser Art von Werbesets den Namen der Firma (firma), die diese vertreibt.

Modellieren Sie diesen Sachverhalt mittels UML Klassendiagramm.

Aufgabe 6: Lego Sammlung II

Erweitern Sie das Klassendiagramm aus Aufgabe 5 wie folgt:

Jedes Set ist genau einem bestimmten Spielthema zugeordnet, wobei ein Spielthema natürlich mehrere Sets umfassen kann. Spielthemen werden durch ihre Bezeichnung (bez) eindeutig identifiziert und sind hierarchisch organisiert, d.h. zu jedem Spielthema kann es mehrere untergeordnete Spielthemen geben, wobei ein bestimmtes Spielthema aber höchstens ein direkt übergeordnetes Spielthema hat.

Jedes Set wird mit einer oder mehreren Bauanleitungen ausgeliefert. Bauanleitungen haben eine eindeutige Nummer (nr) und eine Seitenanzahl (seiten), und werden in speziellen Mappen aufgehoben. Mappen haben eine eindeutige Nummer (nr), eine Beschriftung (beschr) und können maximal zwanzig Bauanleitungen beinhalten. In einem Sammlerkatalog wird für jeden einzelnen Baustein von jedem einzelnen Sammler das Datum vermerkt, wann (datum) und zu welchem Preis der jeweilige Baustein erworben wurde.

Lösung für Beispiel 5 und 6:

