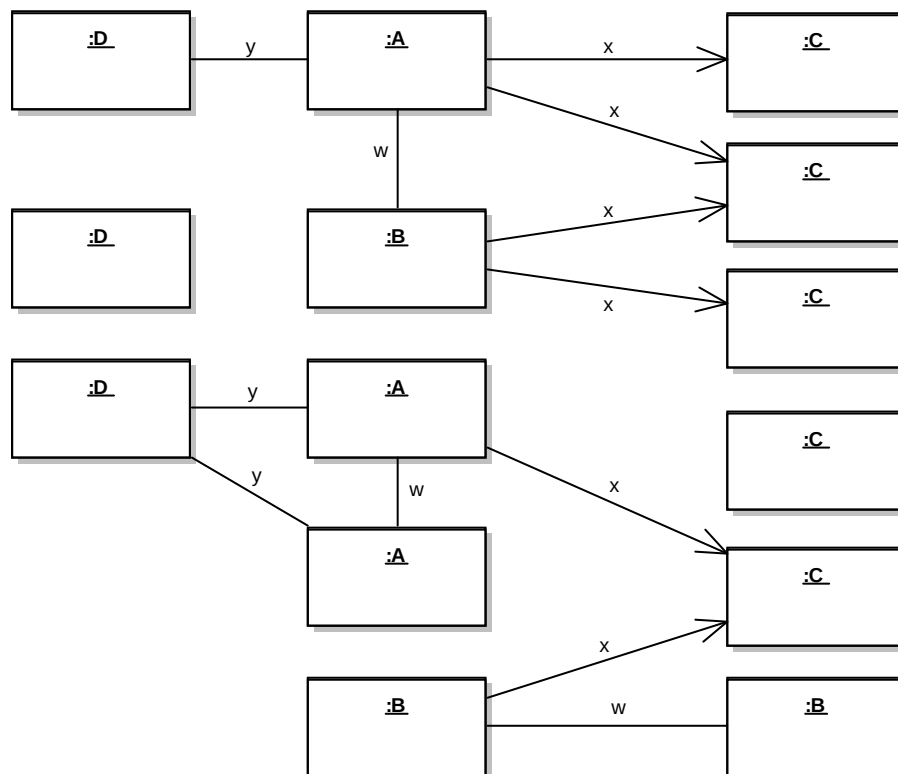


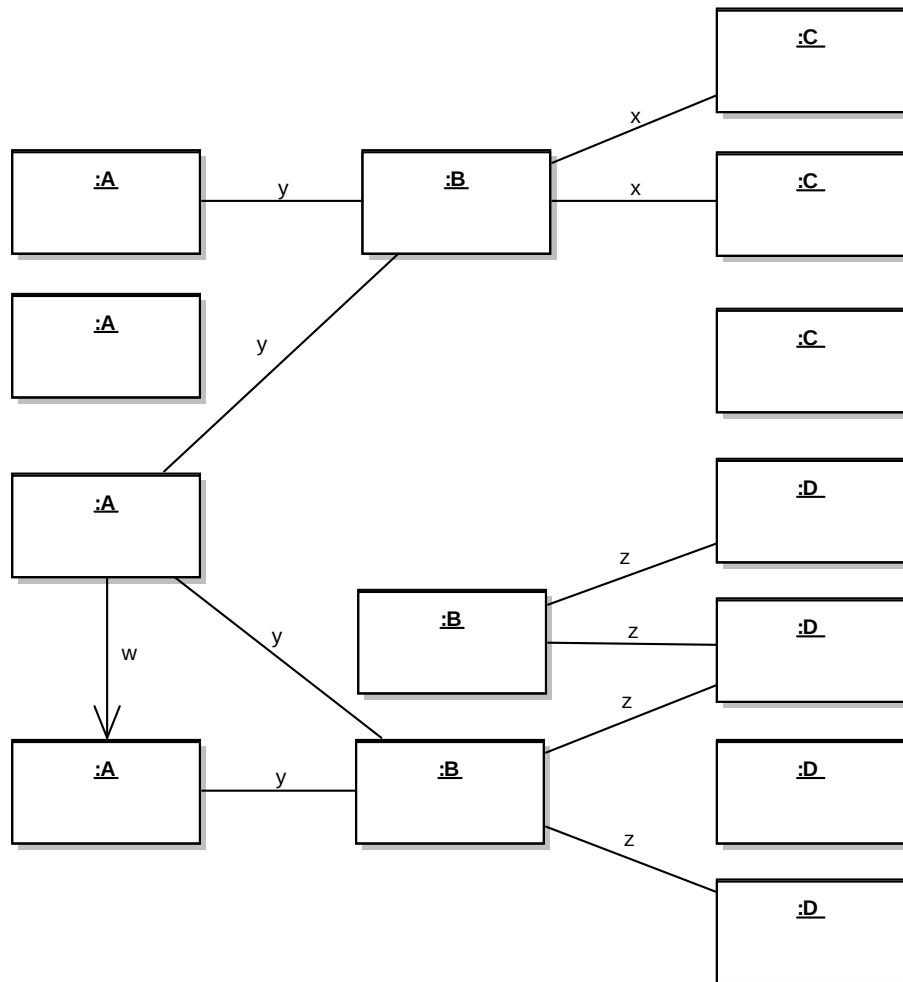
Aufgabe 1: Objektdiagramm

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 minimal und die maximal mögliche Anzahl an Beziehungen mit Objekten einer anderen Klasse darstellen. Der Name jeder Beziehung ist im Klassendiagramm eindeutig (es sollen also keine Beziehungen denselben Namen haben). Weiters sollen mögliche Generalisierungen bzw. XOR-Beziehungen erkannt werden.

- Objektdiagramm 1:



Objektdiagramm 2:



Aufgabe 2: Vergleich von Klassendiagrammausschnitten

Erklären Sie den Unterschied zwischen folgenden Klassendiagrammausschnitten:

a)



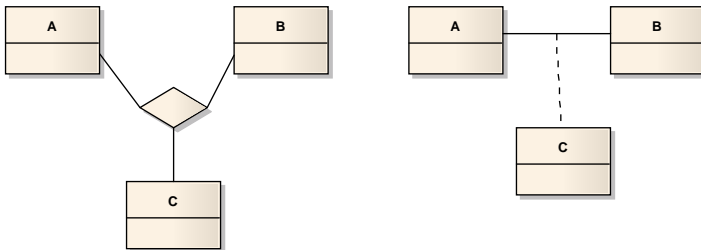
b)



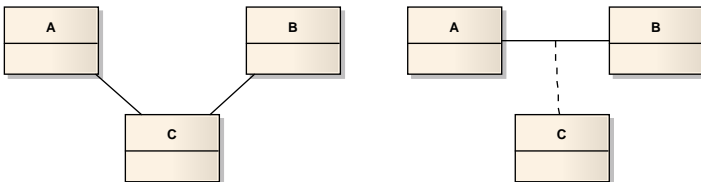
c)



d)



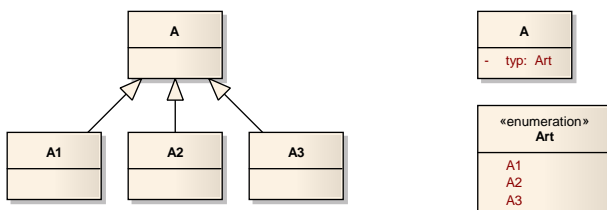
e)



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 Person {
    public String name;
}

class Mieter extends Person {
    private Date gebDat;
    private Mietobjekt mo;
    public Mietobjekt [] interessierteObjekte;
}

class Vermieter extends Person {
    private String wohnsitz;
    private Mietobjekt [] objekte;
}

class Mitarbeiter {
    public String name;
    public String telNr;
    private Anstellung anst;

    public Anstellung getAnst() {
        return anst;
    }
    public void setAnst(Anstellung anst) {
        this.anst = anst;
    }
}

abstract class Mietobjekt {
    public String adresse;
    private Vermieter besitzer;
    private Mieter [] mieter;
}

class Wohnung extends Mietobjekt {
    public int stockwerk;
}

class Geschäftsraum extends Mietobjekt {
    public int fläche;
}

class Hausverwaltung {
    public String adresse;
    public Mitarbeiter leiter;
    public Hashtable verwObjekte;
        // Key: Mietobjekt
        // Value: verantwortlicher (Typ: Mitarbeiter)
}

enum Anstellung { vollzeit , teilzeit }
```

Aufgabe 4: Konzert

Modellieren Sie folgenden Sachverhalt mittels UML Klassendiagramm gemäß nachstehenden Informationen. Verwenden Sie bei Bedarf abstrakte Klassen, Enumerationen und Aggregationen. Datentypen müssen Sie nicht darstellen, sofern diese nicht explizit im Text spezifiziert wurden.

Von jedem Konzert werden das Datum sowie der Veranstaltungsort gespeichert. Weiters wird vermerkt, um welche Musikgenres bei dem Konzert vertreten sind. Dabei gibt es Klassik, Rock, Pop und Heavy Metal. Auf einem Konzert können mehrere Bands spielen, die wiederum auf mehreren Konzerten spielen können. Dabei wird die Höhe der Gage, die eine Band für einen Auftritt erhält, gespeichert. Für eine Band speichern wir deren Namen sowie deren Musikgenre. Von Personen speichern wir deren Namen. Es gibt zwei Arten von Personen, Musiker und Manager. Von Musikern vermerken wir zusätzlich deren Künstlernamen und von Managern deren Telefonnummer. Ein Musiker kann in mehreren Bands spielen welche wiederum aus mehreren Musikern bestehen kann. Für die Zugehörigkeit eines Musikers zu einer Band speichern wir das von ihm gespielte Instrument. Eine Band wird von genau einem Manager betreut und ein Manager kann mehrere Bands betreuen. Für ein Musikstück speichern wir dessen Titel und Länge in Sekunden. Auf einem Konzert kann ein Musikstück von höchstens einer Band gespielt werden. Jedoch kann eine Band auf einem Konzert mehrere Musikstücke spielen. Weiters kann ein Musikstück von einer Band auf mehreren Konzerten gespielt werden.

Aufgabe 5: Regierung I

Bilden Sie die folgenden Sachverhalte mit einem Klassendiagramm ab:

Von Politikern speichern wir Name und Geburtsdatum. Jeder Politiker ist Mitglied von genau einer Partei. Eine Partei kann wiederum mehrere Politiker als Mitglieder haben. Von Parteien speichern wir deren Bezeichnung sowie deren Kürzel. Jede Partei hat einen Obmann der Politiker ist. Von einem Nationalrat speichern wir den Zeitpunkt dessen Bestellung. Ein Nationalrat hat genau 183 Abgeordnete die Politiker sind. Von jedem dieser Abgeordneten erfassen wir dessen Gehalt. Ein Politiker kann in mehreren Nationalräten bestellt sein. Weiters hat ein Nationalrat genau drei Politiker die Nationalratspräsidenten sind. Ein Politiker kann in höchstens zwei Nationalräten als Präsident fungieren. Für jeden dieser Präsidenten speichern wir ob der Präsident erster, zweiter oder dritter Nationalratspräsident ist.

Aufgabe 6: Regierung II

Erweitern Sie das Klassendiagramm aus Aufgabe 5 wie folgt:

Von einer Regierungsfunktion speichern wir deren Gehalt. Es gibt zwei verschiedene solche Funktionen, Minister und Bundeskanzler. Von der Funktion Minister erfassen wir zusätzlich das zugewiesene Ressort und für die Funktion Bundeskanzler erfassen wir die Adresse der Amtswohnung. Staatssekretäre sind formell nicht eine Regierungsfunktion, jedoch wollen wir auch diese Positionen als solche modellieren. Wir speichern von dieser Funktion zusätzlich eine Beschreibung der Aufgaben. Ein Minister und der Bundeskanzler haben höchstens einen Staatssekretär zugeordnet. Ferner ist ein Staatssekretär entweder einem Minister oder dem Bundeskanzler zugeordnet. Von einer Regierung speichern wir den Zeitpunkt der Bestellung sowie den Typ der Regierung. Hierbei gibt es Minderheitsregierungen, Alleinregierungen und Koalitionsregierungen. Ein Politiker kann eine Regierungsfunktion in mehreren Regierungen übernehmen. Weiters kann in einer Regierung eine Regierungsfunktion von bis zu zwei Politikern übernommen werden. Umgekehrt kann ein Politiker in einer Regierung mehrere Funktionen übernehmen.