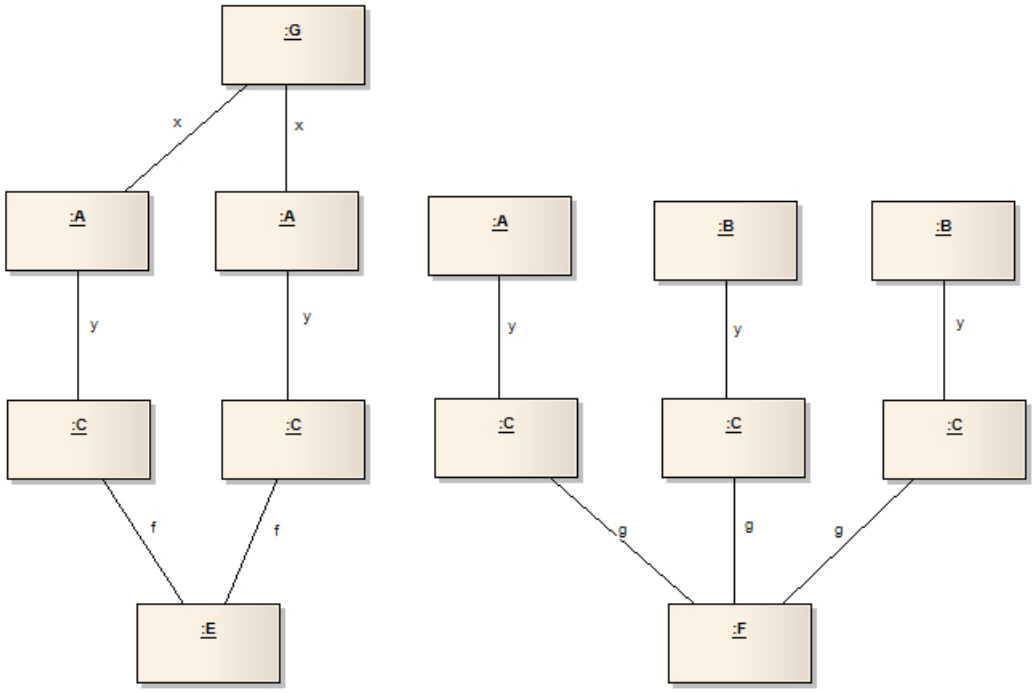
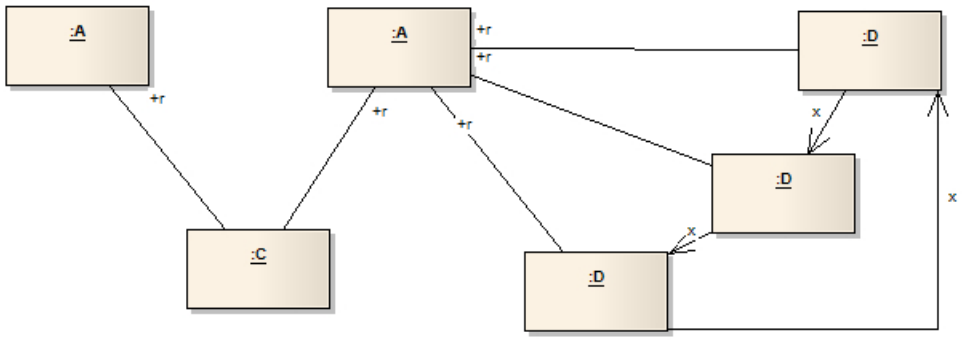
Aufgabe 1: Objektdiagramm

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:



Objektdiagramm 2:



Aufgabe 2: Vergleich von Klassendiagrammauschnitten

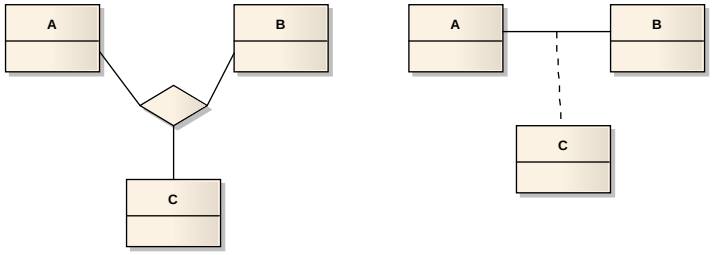
Erklären Sie den Unterschied zwischen folgenden Klassendiagrammausschnitten.

a)

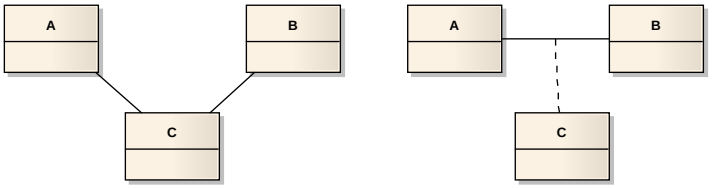
b)

c)

d)

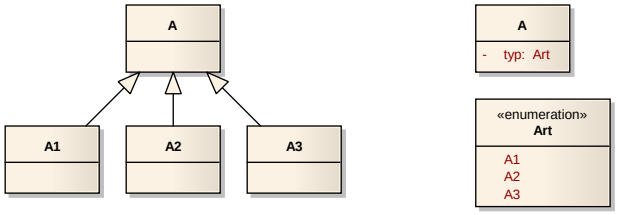


e)



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.

```
public abstract class Schiff {

    private String name;
    private int laenge;
    private int breite;
    private String[] bewaffnung;


    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getLaenge() {
        return laenge;
    }
    public void setLaenge(int laenge) {
        this.laenge = laenge;
    }
    public int getBreite() {
        return breite;
    }
    public void setBreite(int breite) {
        this.breite = breite;
    }
    public String[] getBewaffnung() {
        return bewaffnung;
    }
    public void setBewaffnung(String[] bewaffnung) {
        this.bewaffnung = bewaffnung;
    }

}

public class Segelschiff extends Schiff {

    private Flotte flotte;

    private int masten;
    private int segelflaeche;
    public Hashtable schlachtteilnahmen;


    public void setFlotte(Flotte f) {
        flotte = f;
    }


    public int getMasten() {
        return masten;
    }
```

```
    public void setMasten(int masten) {
        this.masten = masten;
    }
    public int getSegelflaeche() {
        return segelflaeche;
    }
    public void setSegelflaeche(int segelflaeche) {
        this.segelflaeche = segelflaeche;
    }

    public void addSchlacht(Schlacht s, String kapitaen) {
        schlachtteilnahmen.put(s,kapitaen);
    }
}

public class Flotte {
    private String seemacht;
    private Segelschiff flagschiff;
    public ArrayList schiffe;


    public String getSeemacht() {
        return seemacht;
    }
    public void setSeemacht(String seemacht) {
        this.seemacht = seemacht;
    }
    public void setFlagschiff(Segelschiff s) {
        flagschiff = s;
    }

    public void addSchiff(Segelschiff s) {
        schiffe.add(s);
    }
}

public class Schlacht {
    private Date datum;

    public Date getDatum() {
        return datum;
    }

    public void setDatum(Date datum) {
        this.datum = datum;
    }

}
```

Aufgabe 4: Modellierung einer Tankstelle

Eine Tankstellenkette möchte ihre Tankstellen in Zukunft elektronisch verwalten. Modellieren Sie folgenden Sachverhalt mittels UML Klassendiagramm gemäß folgenden Informationen:
Jede Tankstelle hat eine eindeutige ID und eine Adresse. Außerdem hat sie eine Menge von Zapfsäulen (maximal 12), die jeweils durch eine Nummer gekennzeichnet werden. Des Weiteren wird vermerkt, ob es sich um eine Selbstbedienungszapfsäule handelt oder nicht. Für jede Zapfsäule ist bekannt, welche Kraftstoffe bei ihr angeboten werden. Ein Kraftstoff wird eindeutig durch seinen Namen bestimmt, weiters wird die Oktanzahl vermerkt.
Jede Tankstelle archiviert sämtliche Tagespreise der Treibstoffe. Ein Tagespreis wird durch ein Datum, den Kraftstoff, für den er gilt, und durch die Tankstelle identifiziert. Weiters wird der Preis pro Liter vermerkt. Ein Kraftstoff-Kauf wird eindeutig identifiziert durch eine ID, und es wird vermerkt, an welcher Zapfsäule getankt wurde und welcher Kraftstoff zu welchem Tagespreis gekauft wurde. Außerdem wird noch gespeichert, wie viele Liter Kraftstoff abgegeben wurden.

Aufgabe 5: Modellierung eines Transportunternehmens I

Ein Transportunternehmer will die Abläufe in seinem Betrieb modernisieren und auf eine EDV-Lösung umstellen. Hierzu wurden in einer Besprechung mit dem Geschäftsführer der Aufbau und die Organisation des Unternehmens besprochen.
Folgendes Protokoll steht Ihnen zur Verfügung: "Ich beschäftige in meinem Betrieb derzeit rund 60 Fahrer und 6-7 andere Mitarbeiter (Planer und Verwaltungsmitarbeiter). Von diesen muss ich die Namen und die Sozialversicherungsnummer speichern können. Planer und Verwaltungsmitarbeiter sollen sich mit einem Benutzernamen und Passwort am System anmelden können. Von den Verwaltungsmitarbeitern ist ihr Tätigkeitsfeld bekannt. Dann muss ich auch meine Kunden abspeichern. Wichtig sind Name und die Rechnungsanschrift des Kunden.
Neben Kunden und Mitarbeitern will ich auch meine Fahrzeuge verwalten. Ich unterscheide zwischen verschiedenen Fahrzeugtypen wie etwa Kühlwägen, Kleintransporter, Ladebühnen oder Kastenwägen. Für die Fahrzeuge möchte ich das Kennzeichen und den Kilometerstand speichern. Es kann natürlich vorkommen, dass ein Fahrzeug repariert oder überprüft werden muss. Hierfür möchte ich die Werkstatt (Name und Adresse sind bekannt), den Mitarbeiter, der das Fahrzeug in die Werkstatt gebracht hat, sowie die Dauer und die Kosten der Reparatur speichern."
Modellieren Sie diesen Sachverhalt mittels UML Klassendiagramm.

Aufgabe 6: Modellierung eines Transportunternehmens II

Das Protokoll aus dem vorhergehenden Beispiel wird wie folgt ergänzt:
"Jedem Fahrer können Aufträge zugeteilt werden. Meist ruft dafür ein Kunde an und gibt uns bekannt, wann er ein Transportfahrzeug benötigt und wie groß das Paket ist. Aufgrund dessen weiß der Planer, welcher Fahrzeugtyp für den Auftrag benötigt wird. Ich möchte immer wissen, welcher Planer einen Auftrag einem Fahrzeug zugeordnet hat. Außerdem muss ein Kommentar mit zusätzlichen Informationen zum Auftrag gespeichert werden können und ich muss wissen, wann der Fahrer den Auftrag beginnt und beendet. Wichtig ist auch, dass ich immer weiß, welcher Fahrer mit welchem Fahrzeug welchen Auftrag erledigt hat.
Bei einem Auftrag ist immer ein Kunde Auftraggeber und es sind sowohl Abhol- als auch Lieferadresse bekannt. Der Auftrag kann direkt bei einem Verwaltungsmitarbeiter oder bei einer Partnerfirma, von der Name und Adresse bekannt sind, erfolgen."
Ergänzen Sie Ihr UML Klassendiagramm entsprechend.