

Aufgabe 1.1

Analyse: 1) genaue Bestimmung der Anforderungen
→ Pflichtenheft

2) Erstellung des Prototypen

3) OOA-Modell

Entwurf: 1) Grundsätzliche Struktur des Programms

2) welche Datenbank

3) welche Bibliotheken

Aufgabe 1.3

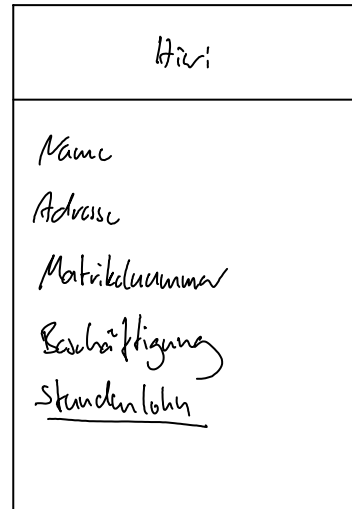
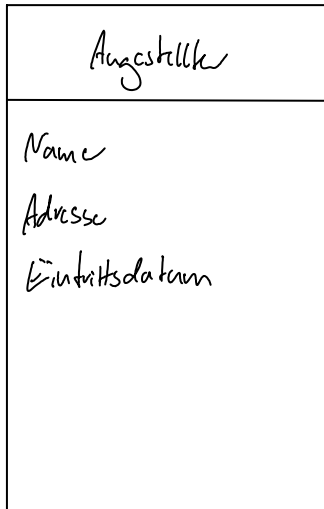
c) Anzuordnende Hauptwörter:

- Bibliothek
- Entleiher
- Woche
- Einheit
- Zeit
- System
- (• Mahnung)
- Anforderung
- Entleihername
- Benutzer / Anwender
- Thema
- Autor
- Suche

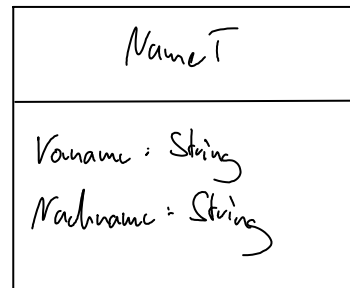
Liste der relevanten Klassen:

- Buch
- Zeitschrift
- Exemplar
- Mithglied
- Mitarbeiter

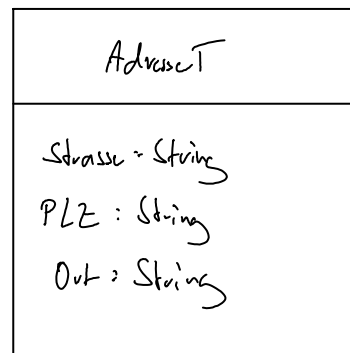
Aufgabe 1.4



Name: NameT



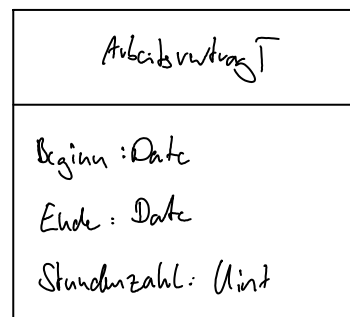
Adresse: AdresseT



Eintrittsdatum: Date

Matrikelnummer: String

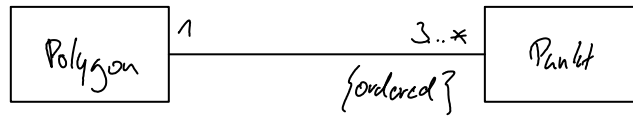
Beschäftigung: list of ArbeitsvertragT



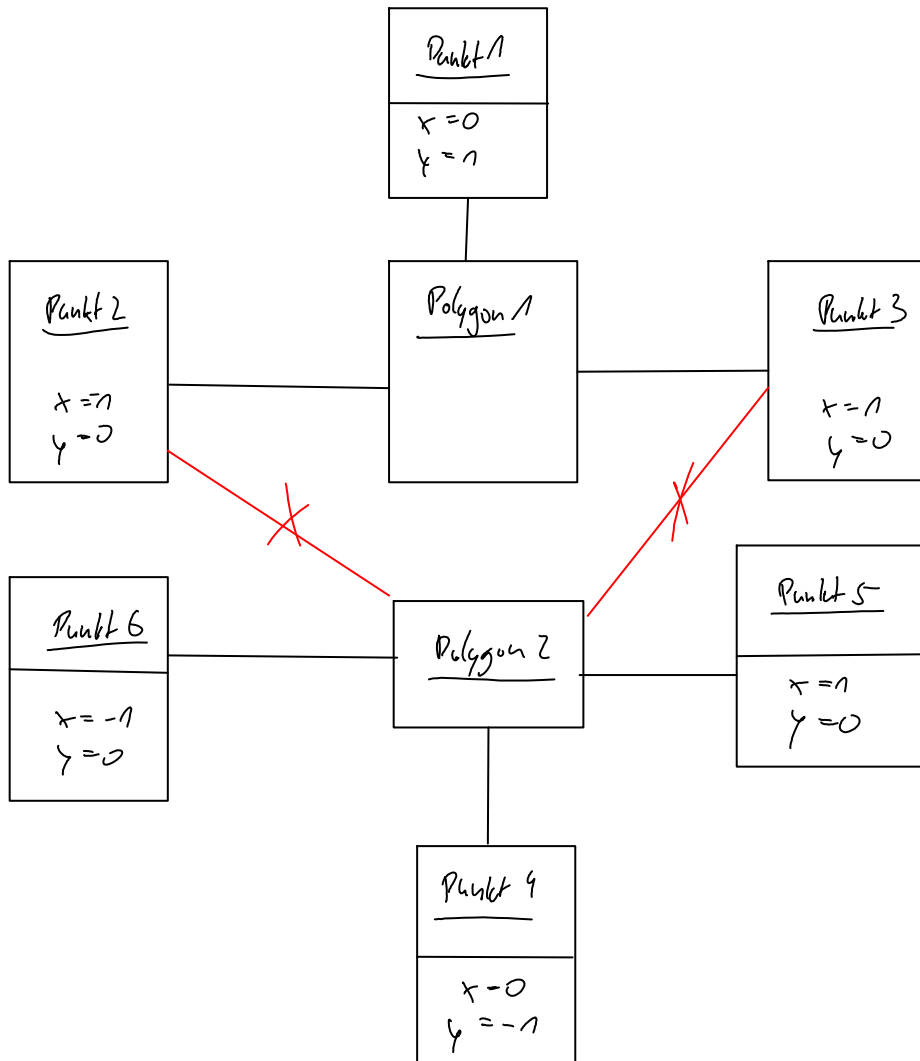
Stundenlohn: Fixed(2,2)

Aufgabe 2.1

a) Klassendiagramm

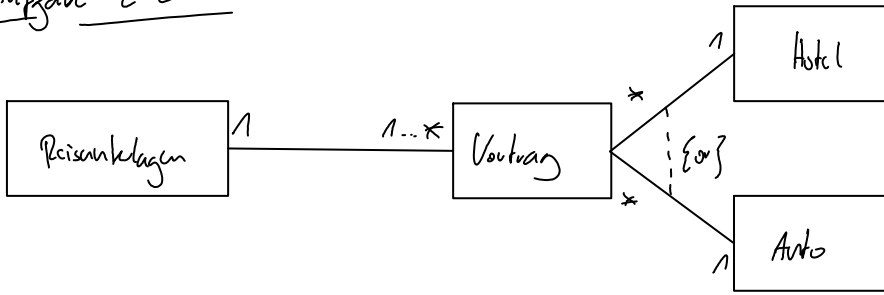


b) Objektdiagramm (1)



Aufgabe 2.2

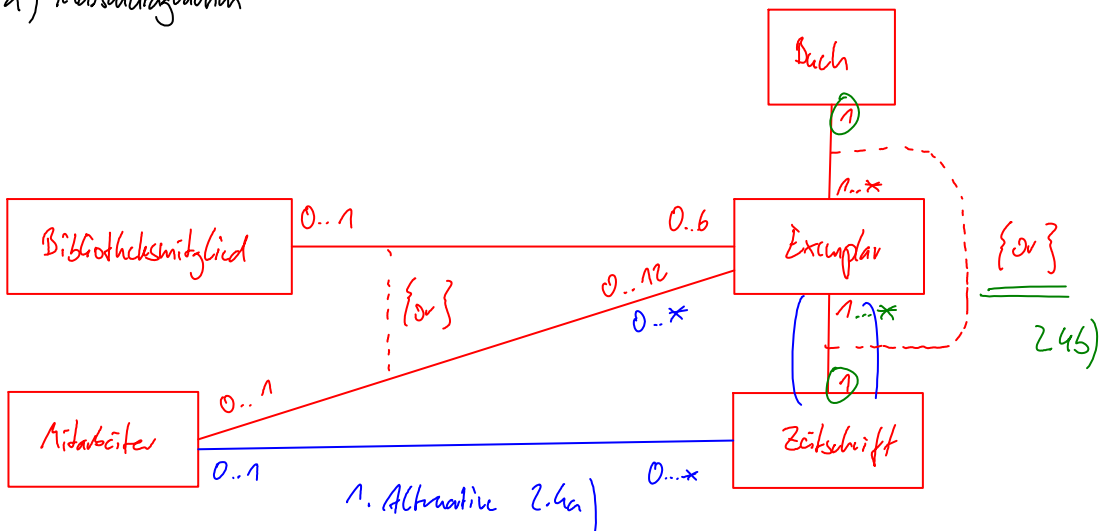
a)



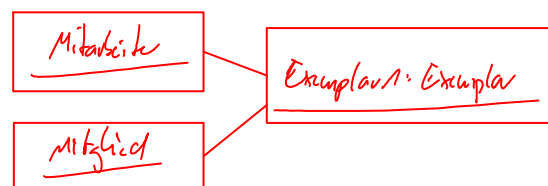
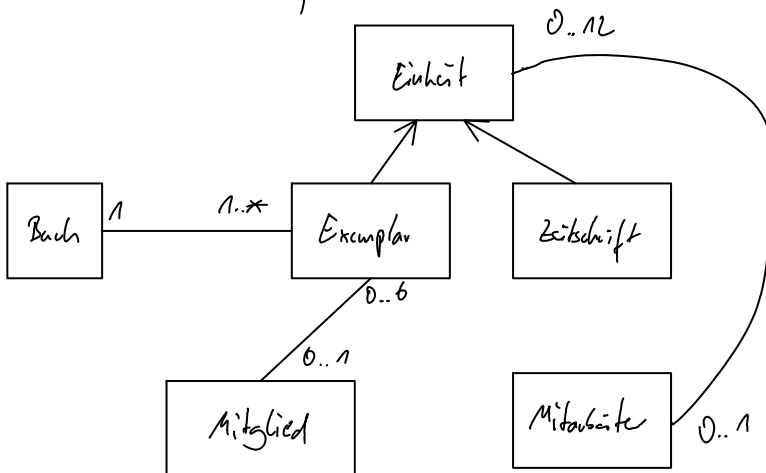
b) Nein, das ist nicht möglich. In einer oder-Assoziation dürfen in UML nicht zweimal die gleichen Rollen notiert werden.

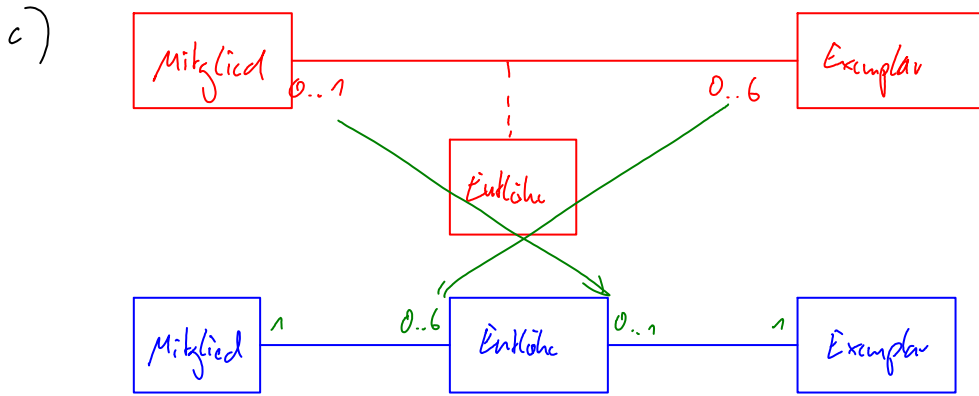
Aufgabe 2.4

a) Klassendiagramm



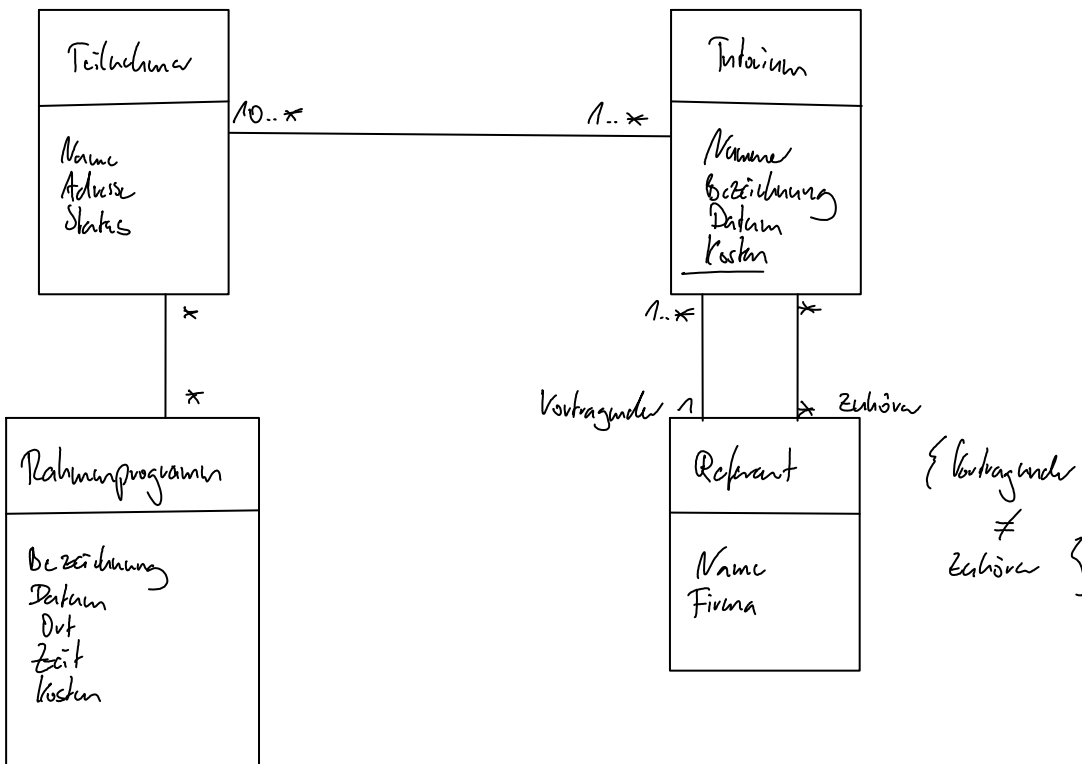
2. Alternative zu 2.4a)





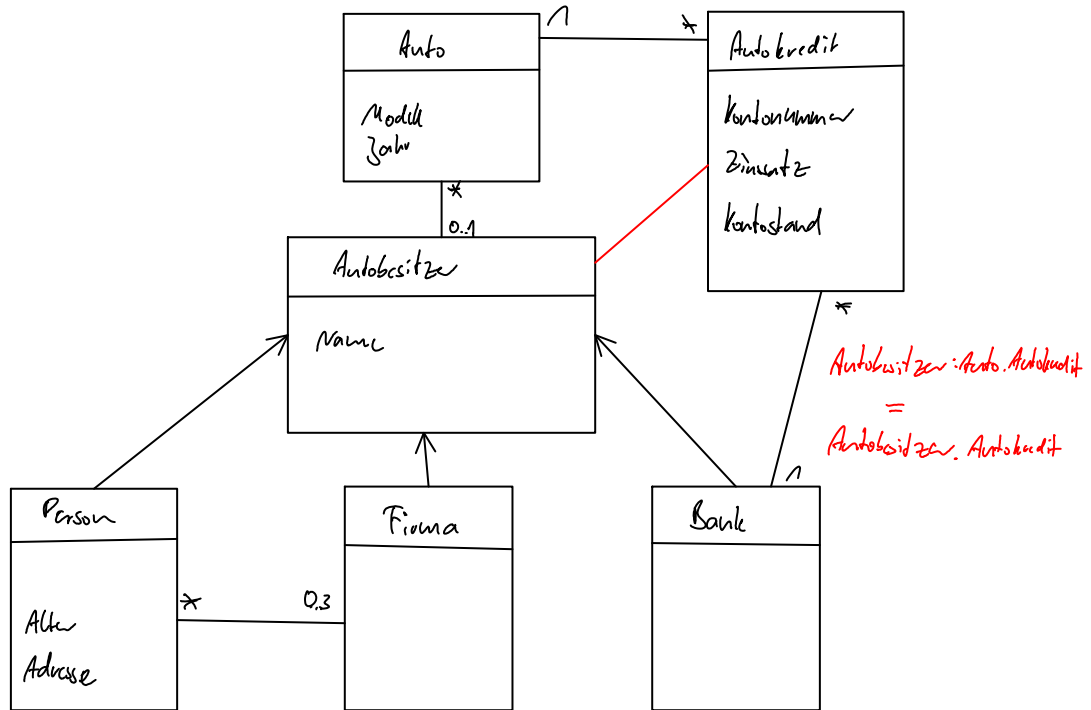
Aufgabe 2.5

a) Klassendiagramm

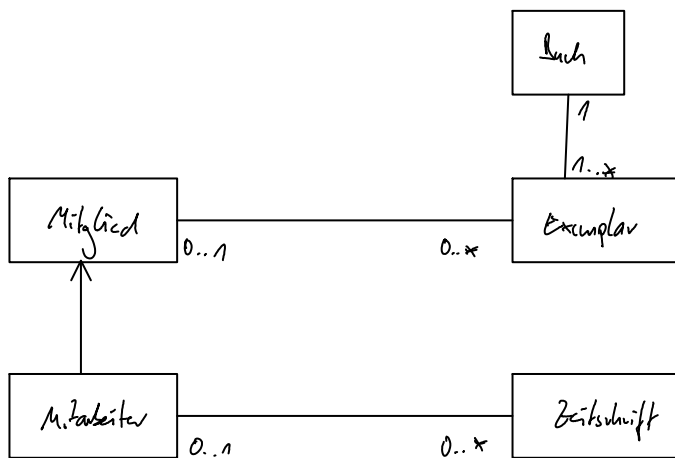


Aufgabe 3.1

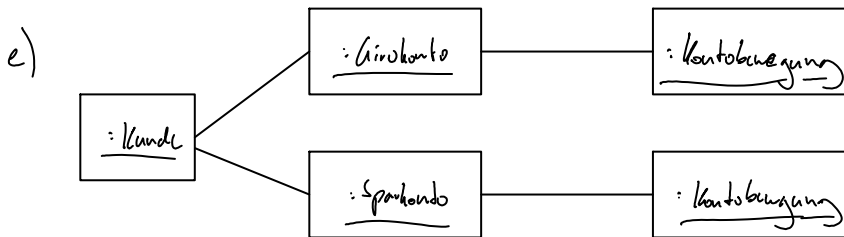
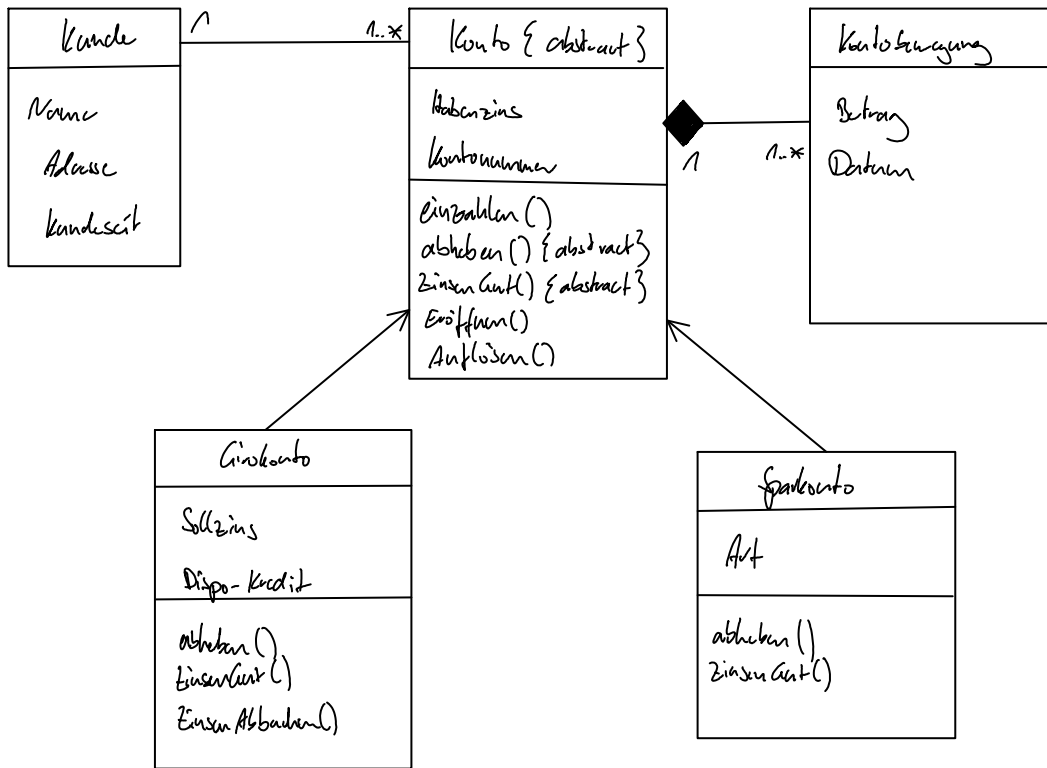
a) b)

Aufgabe 3.2

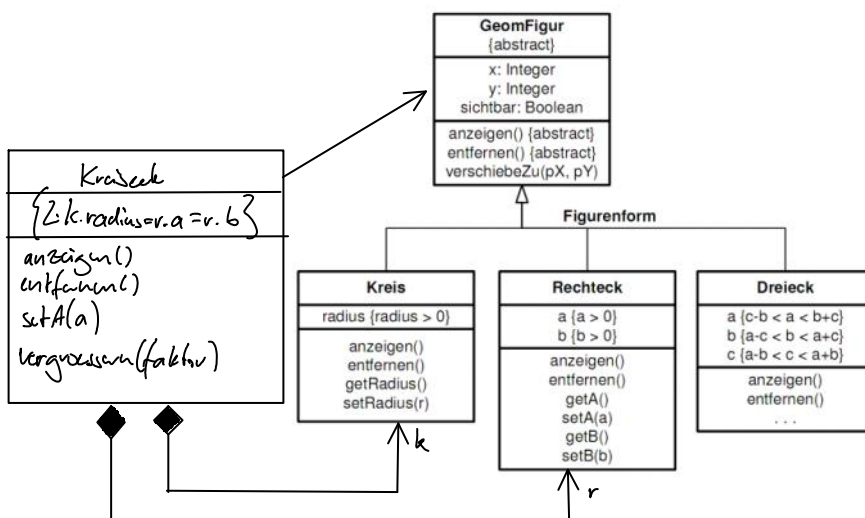
a)



Aufgabe 3.3



Aufgabe 3.4



c)

class Kreiseck extends GeomFigur

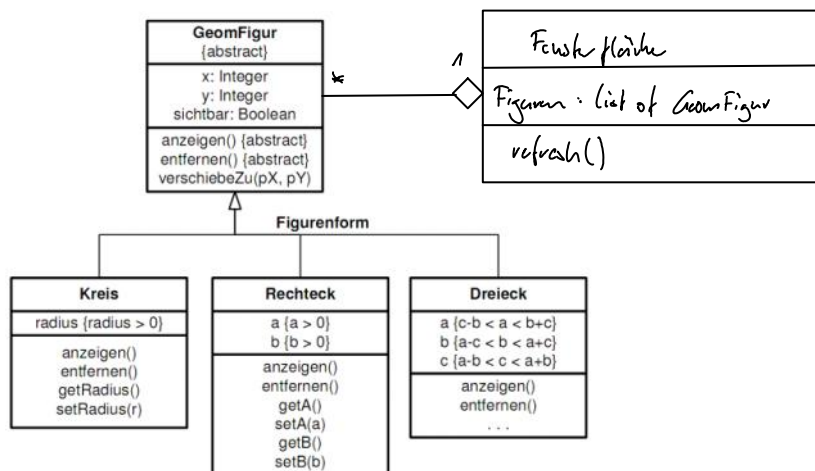
```

{
    Kreis k;
    Rechteck r;
    public void anzeigen()
    {
        k.anzeigen();
        r.anzeigen();
    }
    public void setA(int a)
    {
        k.setRadius(a/2);
        r.setA(a);
        r.setB(a);
    }
    public void vergroessern(float f)
    {
        setA(r.getA()*f);
    }
}

```

aufgaben identisch

d)



15.12.2008

Rechteck(x,y,a,b)

Kreis(x,y,radius)

public class Fensterflaeche

```

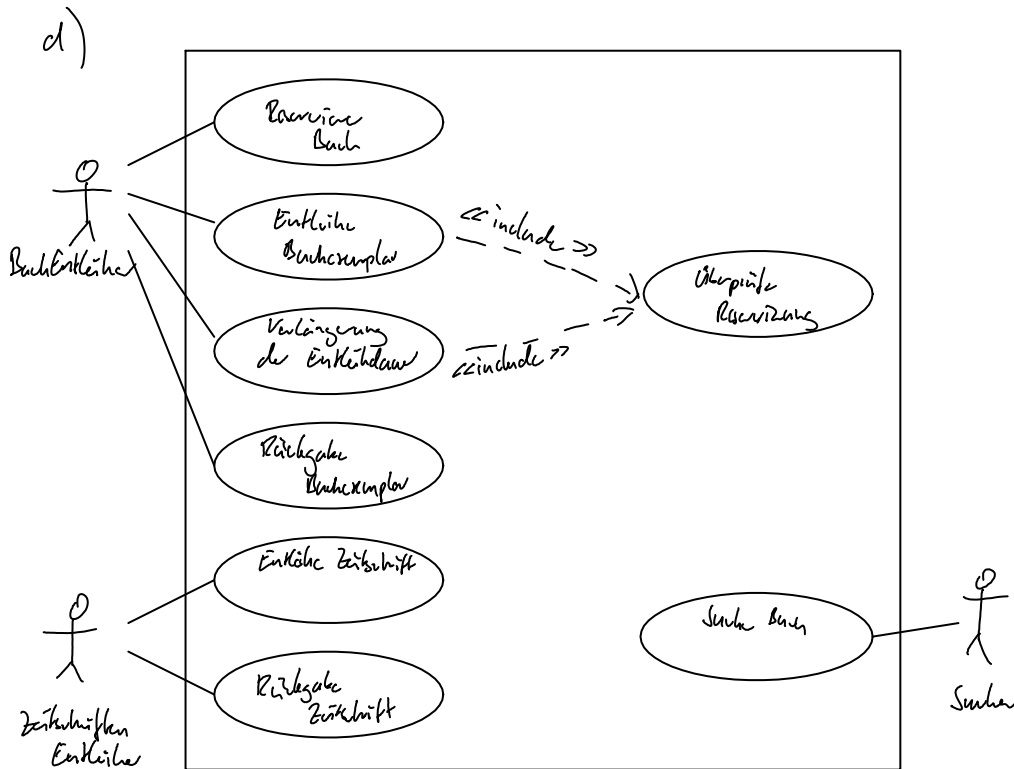
{
    GeomFigur[] Figuren;
    public void Fensterflaeche()
    {
        Figuren[0] = new Kreis(20,50,50);
        Figuren[1] = new Kreis(15,15,15);
        Figuren[2] = new Rechteck(20,20,66,66);
    }
    public void refresh()
    {
        for (int i=0; i<Figuren.length; i++)
            Figuren[i].anzeigen();
    }
}

```

Aufgabe 4.1

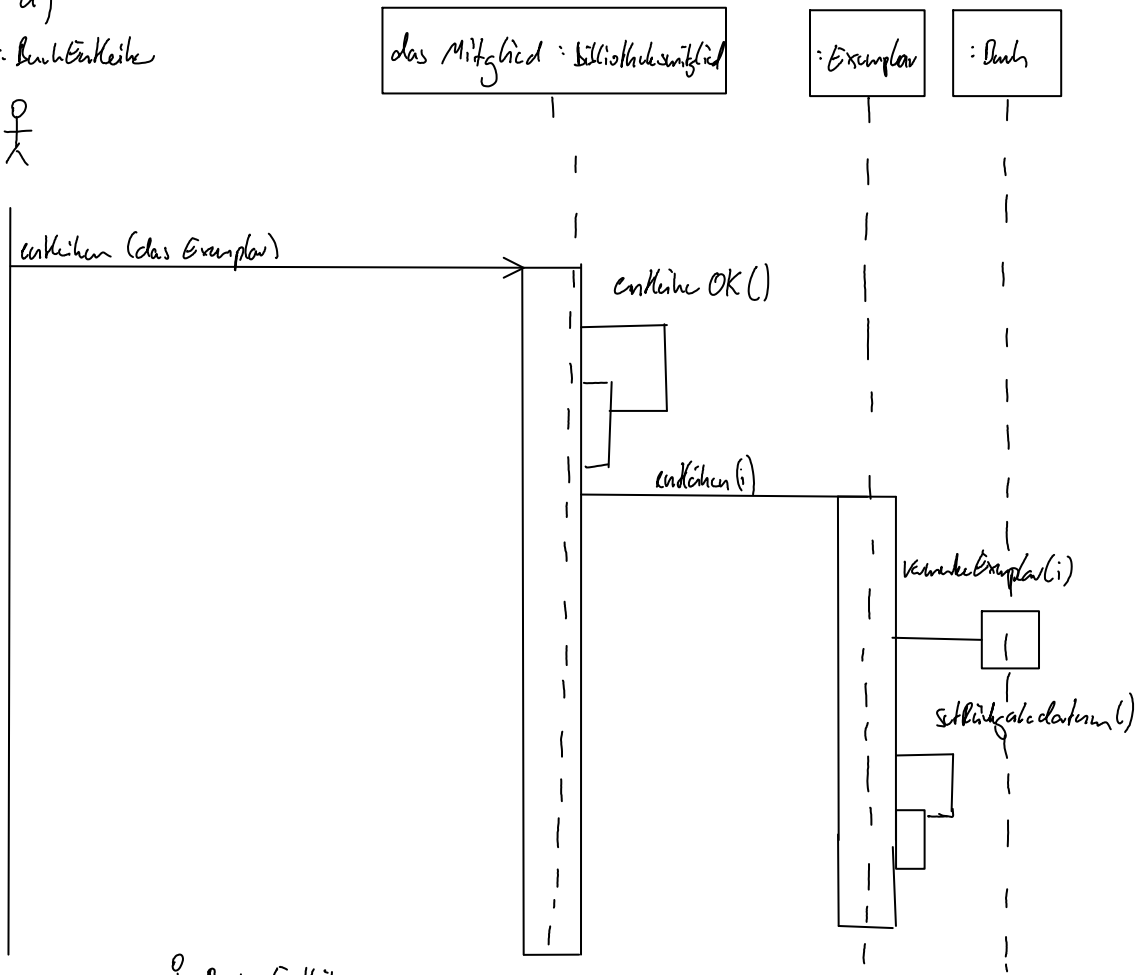
- a) Zu betrachtendes System: Bibliothek
 Externe Akteure: • BuchEntleiher
 • ZeitschriftenEntleiher
 • Sauber

- b) • Entleihe Buchexemplar
 • Rückgabe Buchexemplar
 • Reserviere Buch
 • Verlängerung Entleihdauer
 • Suche Buch
 • Entleihe Zeitschrift
 • Rückgabe Zeitschrift

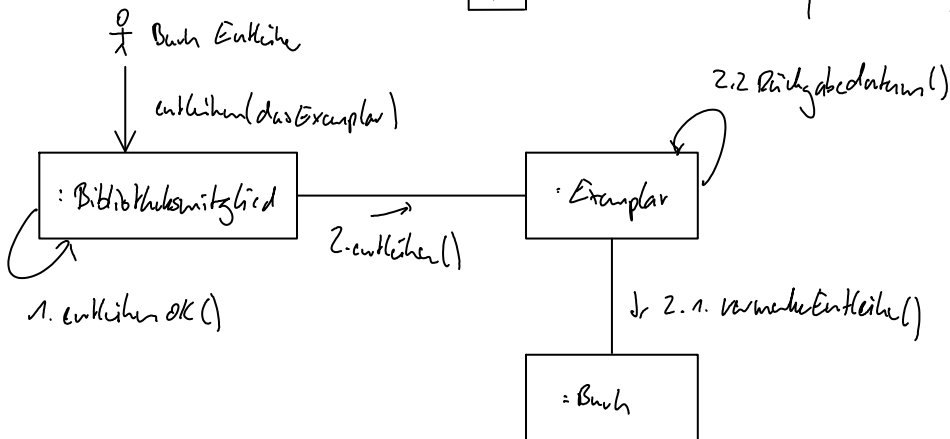


Aufgabe 4.2

a)
: BuchEntleiher

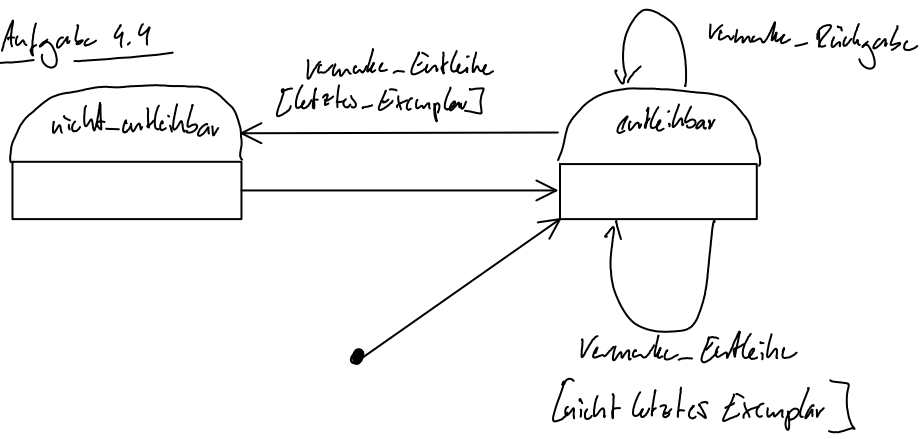


b)



12.01.2009

Aufgabe 4.4



Aufgabe 5.1

a) Zu betrachtendes System → Fahrkartenautomat

Externe Akteure:

- Fahrgast
- Service-Techniker

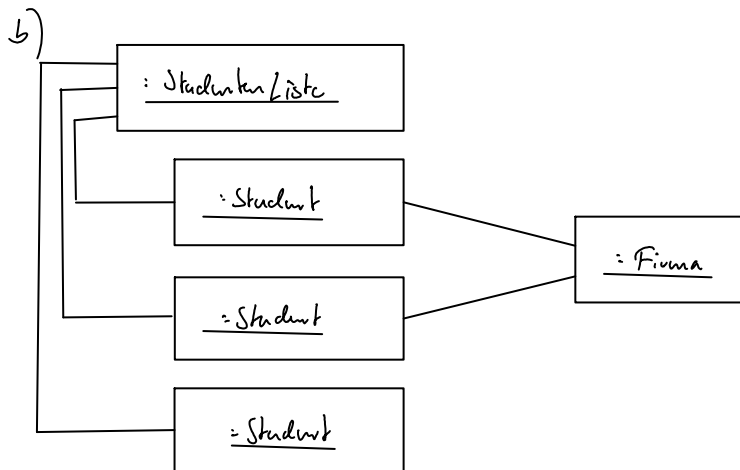
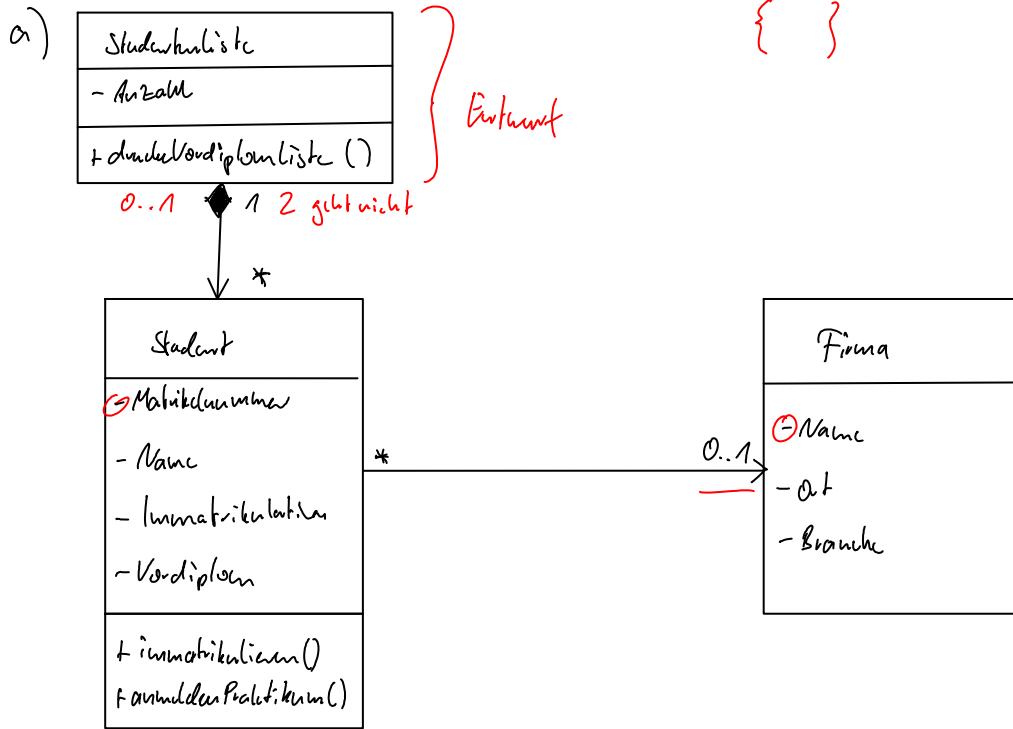
b) abg. Anwendungsfälle:

- Fahrkarte kaufen
- Fahrkartenautomat instandhalten

partielle Anwendungsfälle:

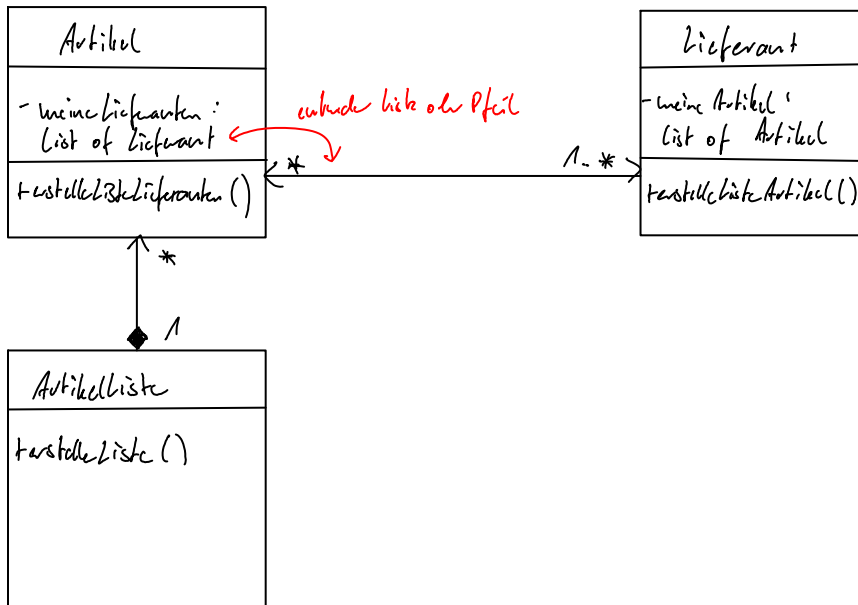
- Fahrziel und Fahrkartenart bestimmen
- Fahrpreis bezahlen

Aufgabe 6.1



Aufgabe 6.2

Ein Artikel wird von einem oder mehreren Lieferanten bezogen. Ein Lieferant liefert im Normalfall mehrere Artikel. Es soll eine Liste aller Artikel erstellt werden, die für jeden Artikel alle zugehörigen Lieferanten enthält. Außerdem soll für einen beliebigen Lieferanten eine Liste aller von ihm gelieferten Artikel erstellt werden.

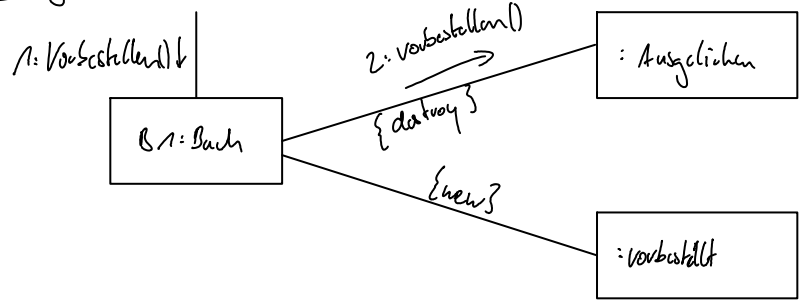


Aufgabe 6.3

```

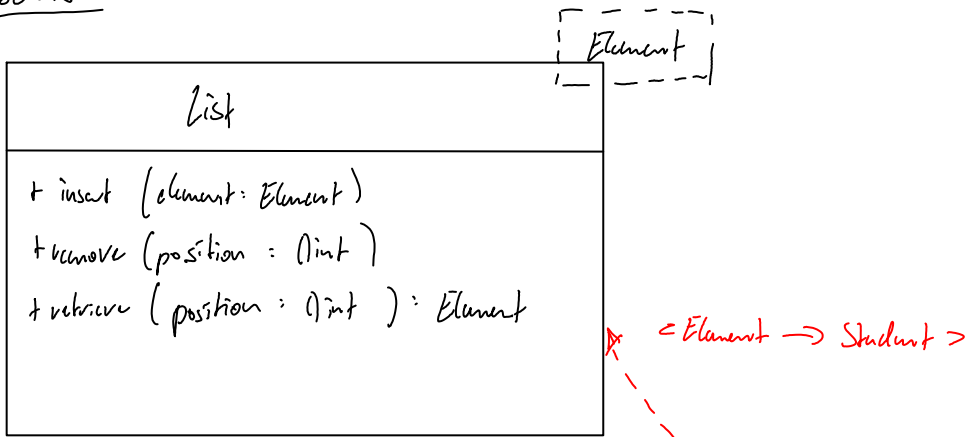
Literaturstelle neuLit;
neuLit = new Buch();
neuLit. erfassen(); // dynamische Bindung zu Buch-erfassen()
liste.einfügen(neuLit);
  
```

Aufgabe 6.4

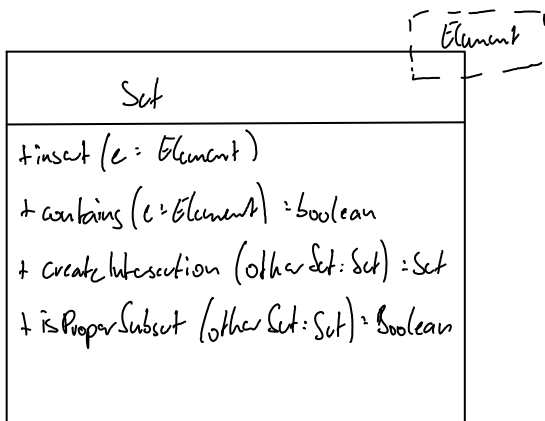


Aufgabe 6.5

a)



b)



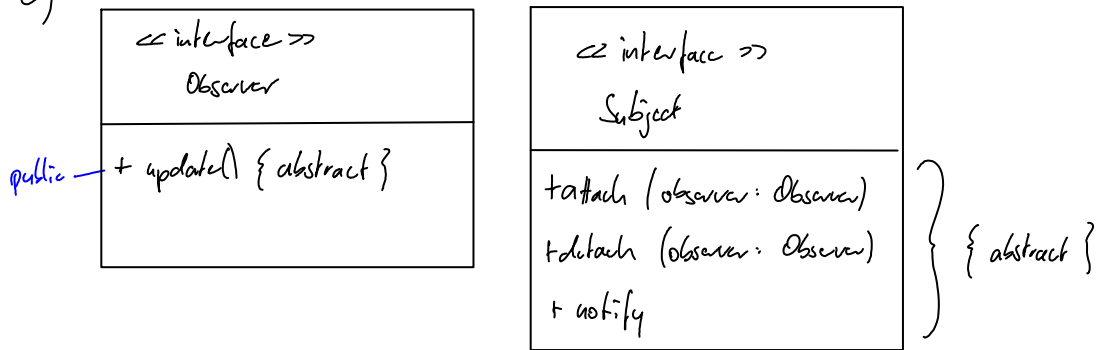
Aufgabe 7.1

a) Beobachter-Muster

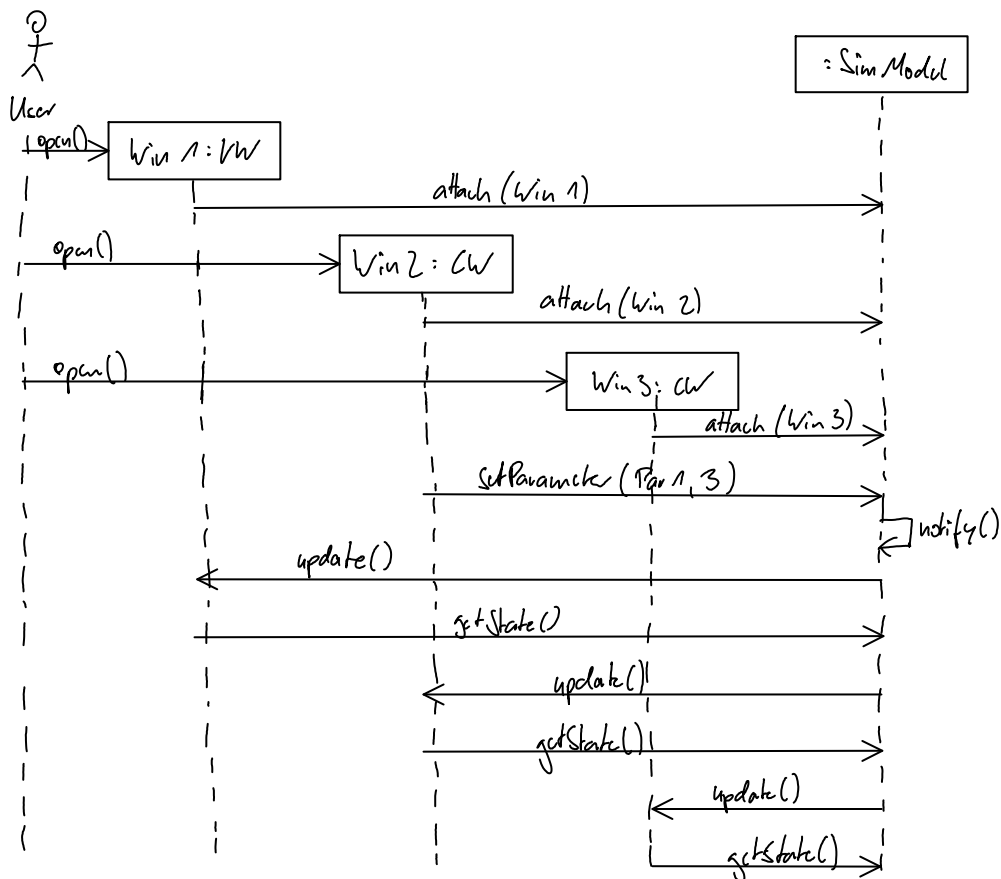
⇒ Bei Änderung eines Objekts (Objekt von Typ SimModel) alle davon abhängigen Objekte (Calibration Window, Visualization Window) benachrichtigt und aktualisiert werden.

b) siehe Aufgabenblatt

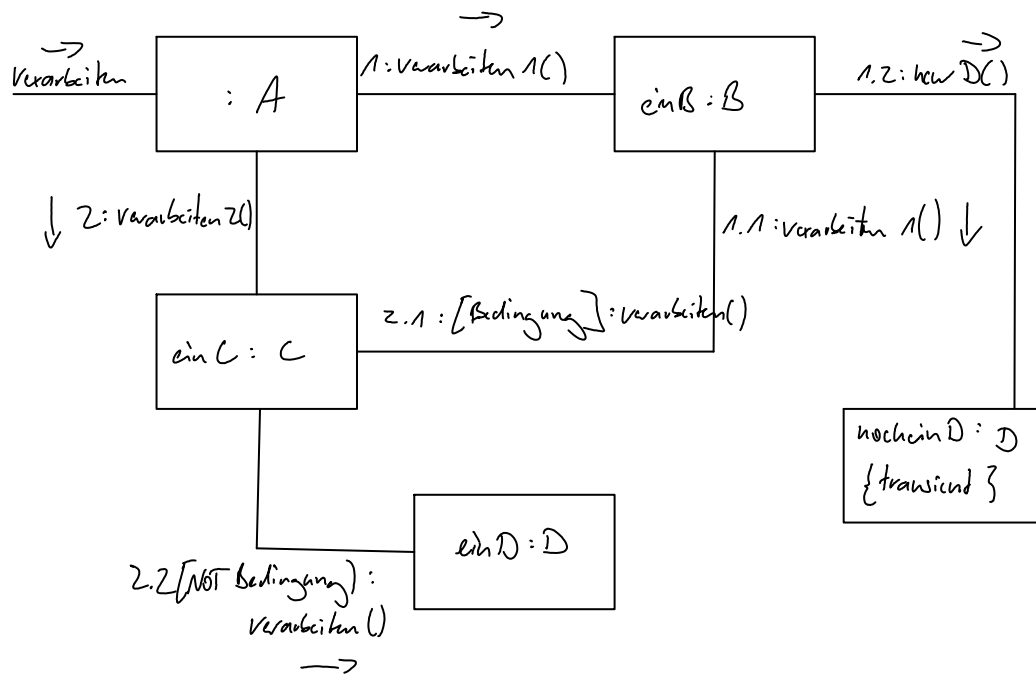
c)



d)

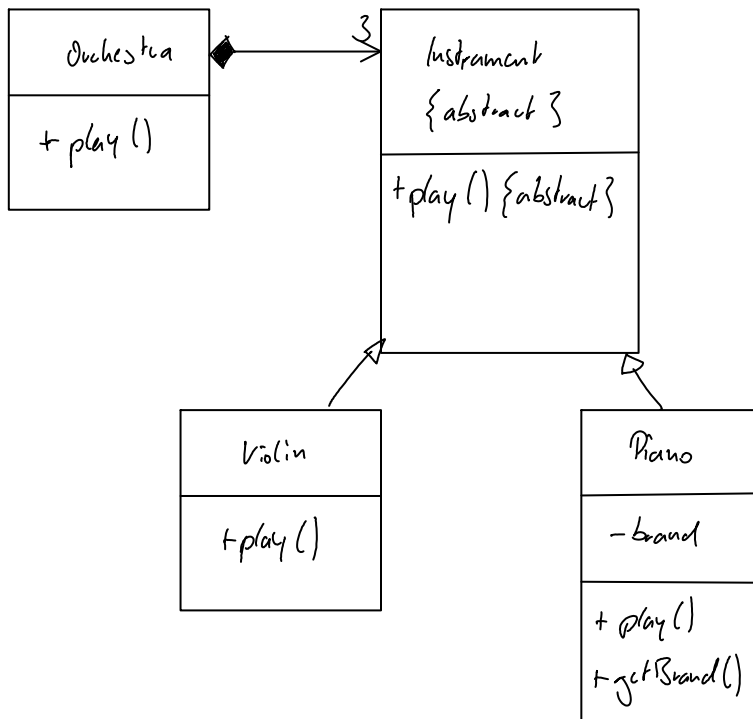


- Nachteil:
- e) • Funktionalität des Beobachter-Mustars in den Schnittstellen
Observer und Subject nur definiert, aber nicht implementiert wird.
- In einem Interface kann die Assoziation zwischen Observer und Subject nicht definiert werden, sie muss in den Unterklassen realisiert werden.

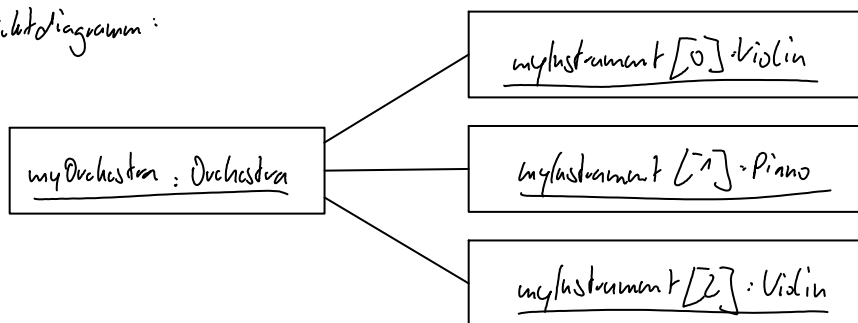
Aufgabe 8.2

Aufgabe 8.3

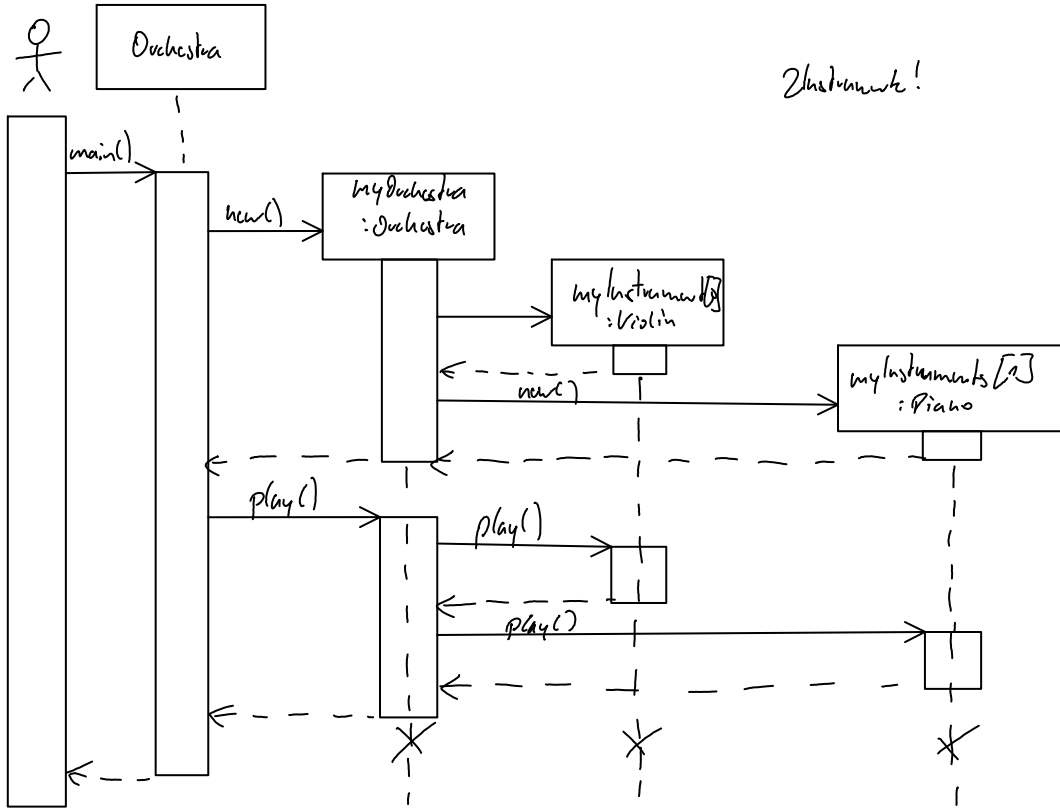
Klassendiagramm:



Objektendiagramm:



Sequenzdiagramm:



→ siehe Aufgabenblatt

Aufgabe 1.2

- a) Sitzplatz ist Klasse Eintrittskarte zuzuordnen (als Attribut SitzplatzNr.)
- b) Qualifizierte Assoziation mit qualifizierendem Attribut SitzplatzNr.

