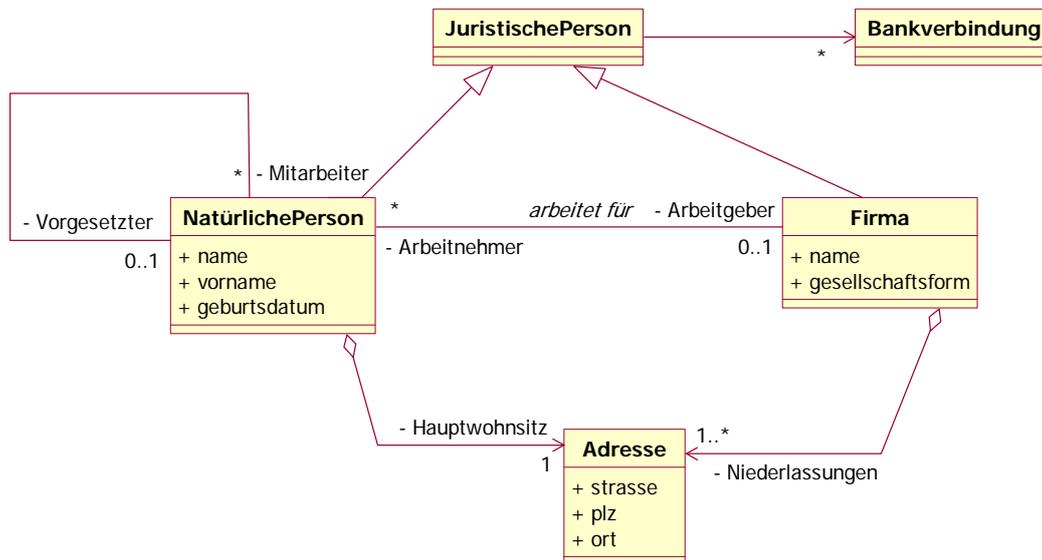


# Übungsblatt 2

## UML

### Aufgabe 1 Klassendiagramm

Gegeben ist das folgende Klassendiagramm:



- Sie sollen nun einem Kunden, der die UML-Darstellung nicht kennt, den Inhalt des Diagramms erklären. Beschreiben Sie hierzu (verbal) die Klassen mit Beziehungen, insbesondere auch die Bedeutung der Beziehungen. Verwenden Sie dazu nur Informationen aus dem Diagramm!
- Was können Sie zur Anzahl erfassbarer Adressen des Systems sagen?
- Was bedeutet das „Rauten“-Symbol in der Beziehung zwischen Firma und Adresse?
- Diskutieren Sie ein Verschieben der Klasse Adresse aus den gezeichneten Beziehungen in eine Beziehung zur Klasse JuristischePerson.

### Aufgabe 2 Fußgängerampel

Für eine Ampelanlage bestehend aus zwei Fahrzeugampeln, zwei Fußgängerampeln und Tastern (entsprechend dem Bild unten) soll die zu erstellende Software mit UML modelliert werden. Die folgenden Anforderungen sollen umgesetzt werden:

- Im Normalbetrieb sind die Ampeln A1 und A2 im Zustand Grün, die Ampeln A3 und A4 sind im Zustand Rot.
- Sobald an Taster T1 oder T2 gedrückt wird, wechseln nach der Zeit  $z_0$  die Ampeln A1 und A2 in Gelb und Rot über. Gelb dauert über die Zeit  $z_1$  an. Am gedrückten Taster zeigt eine Kontrolllampe unmittelbar an, dass der Tastendruck registriert wurde. Der Taster bleibt nun solange passiv.
- Sind die Ampeln A1 und A2 für die Zeit  $z_2$  im Zustand Rot, wechselt A3 und A4 auf den Zustand Grün. Die Kontrolllampe am Taster erlischt. A3 und A4 bleiben für die Zeit  $z_3$  im Zustand Grün. Nach Ablauf von  $z_3$  wechselt A3 und A4 wieder

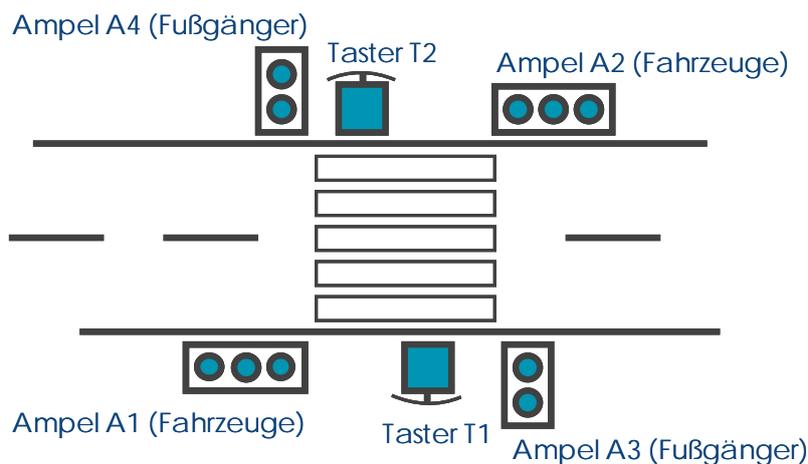
auf Rot und anschließend nach der Zeit  $z_4$  A1 und A2 über Gelb-Rot (Gelb-Rot dauert  $z_1$ ) auf Grün.

- Ein Timer liefert nach der Initialisierung auf eine Zeit  $z$  ein Signal (timeout) nach Ablauf dieser Zeit. Der Empfänger der timeout-Botschaft registriert sich hierzu beim Timer.

Die vorhandene Hardware besteht aus den Ampeln, den Tastern und einer Zentraleinheit, auf der sich der Rechner befindet. Als Software wurde vom Hersteller der Anlage folgende Klasse (einschl. der Implementierung) mitgeliefert:

```
class GeraeteControl {
    public void init() {} // Initialisierung
    public void lichtAn(lampenNr) {} // rot=1, gelb=2, grün=3
    // Kontrolllampe des Tasters = 1
    public void lichtAus(lampenNr) {} // wie lichtAn
}
class TasterControl extends GeraeteControl {
    abstract void taster();
}
```

Die Methode „taster“ wird mit Drücken des entsprechenden Tasters aufgerufen.



- Erstellen Sie das Klassendiagramm der Ampelanlage. Überlegen Sie, wo Generalisierung sinnvoll ist.
- Erstellen Sie das Sequenzdiagramm für das Szenario „Tastendruck“. Prüfen Sie anschließend das Klassendiagramm aus a). Beachten Sie, dass eintreffende Ereignisse in der jeweiligen Klasse als Methode umgesetzt werden.
- Erstellen Sie für die Fahrzeugampel und die Steuerung je ein State Chart.

Nur für Experten:

- Implementieren Sie die Ampelsteuerung. Implementieren Sie zunächst den Zustandsautomaten „konventionell“, d.h. mit einer enum-Variablen für den Zustand
- Implementieren Sie die Ampelsteuerung mit dem „State“-Entwurfsmuster.