

The background of the slide is a complex network of thin grey lines connecting various sized circles. Some circles are solid blue, some are solid dark blue, and some are white with a dark blue center. The overall effect is a sense of interconnectedness and technology.

# ASSOZIATION

---

Vincent Uhlmann  
IT-Akademie Dr. Heuer GmbH

# ASSOZIATION

- Assoziationen beschreiben Beziehungen zwischen Klassen
- Klassen können miteinander interagieren



# UNIDIREKTIONALE ASSOZIATION

- Unidirektionale Assoziationen beschreiben Beziehungen zwischen Klassen, bei denen eine Klasse die andere kennt
- Dargestellt mit einer Linie und einem Pfeil am Ende der Linie
- Ausgehend von der Klasse, die die andere Klasse kennt
- Die Person kennt die Adresse



# UNIDIREKTIONALE ASSOZIATION

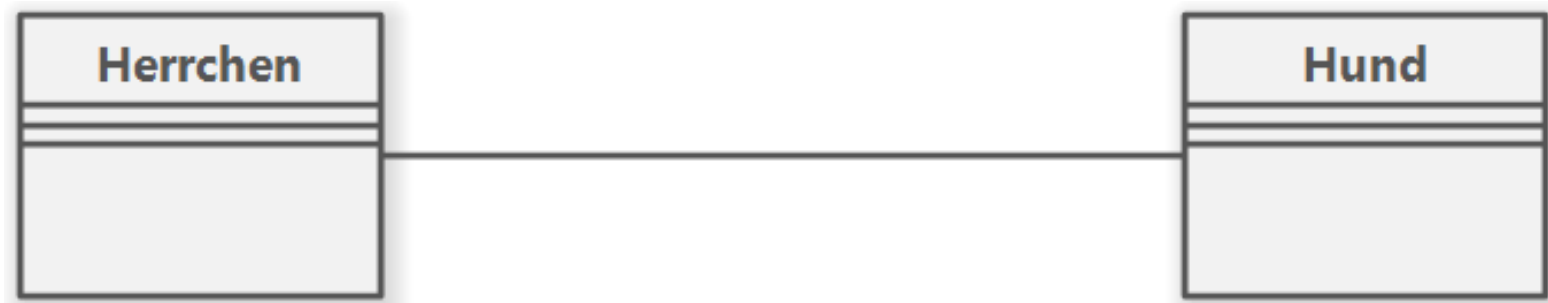
```
public class Person
{
    private Adresse _adresse;

    public Person(Adresse adresse)
    {
        _adresse = adresse;
    }
}
```

```
public class Adresse
{
}
```

# BI-DIREKTIONALE ASSOZIATION

- Bidirektionale Assoziationen beschreiben Beziehungen zwischen Klassen, bei denen sich die Klassen gegenseitig kennen
- Dargestellt mit einer Linie ohne Pfeile
- Das Herrchen kennt den Hund
- Der Hund kennt das Herrchen



# BI-DIREKTIONALE ASSOZIATION

```
public class Herrchen
{
    private Hund _hund;

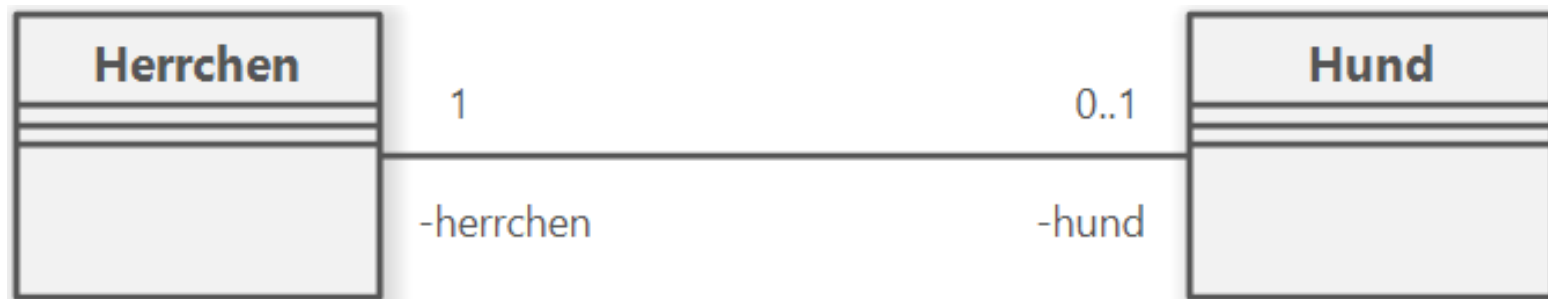
    public void SetHund(Hund hund)
    {
        _hund = hund;
    }
}
```

```
public class Hund
{
    private Herrchen _herrchen;

    public Hund(Herrchen herrchen)
    {
        _herrchen = herrchen;
    }
}
```

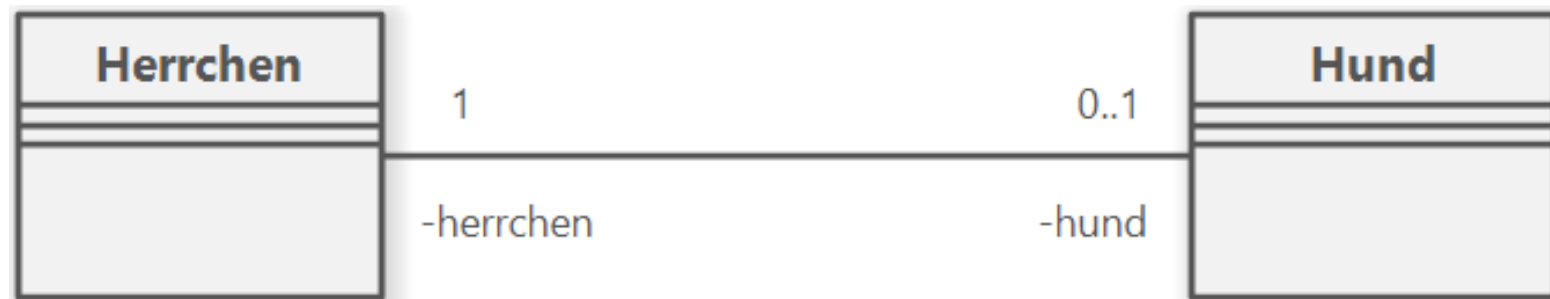
# ASSOZIATION

- Rollennamen und Sichtbarkeiten können verwendet werden, um die Assoziationen zu kennzeichnen
- Die Multiplizität gibt an, mit wie vielen Objekten der gegenüberliegenden Klasse ein Objekt in Beziehung stehen kann
- In den Klassen selbst werden keine Variablen/Attribute angelegt



# ASSOZIATION

- Ein Herrchen hat keinen oder einen Hund
- Ein Herrchen hat eine (1) private Variable vom Typ Hund
- Ein Hund hat ein Herrchen
- Ein Hund hat eine (1) private Variable vom Typ Herrchen

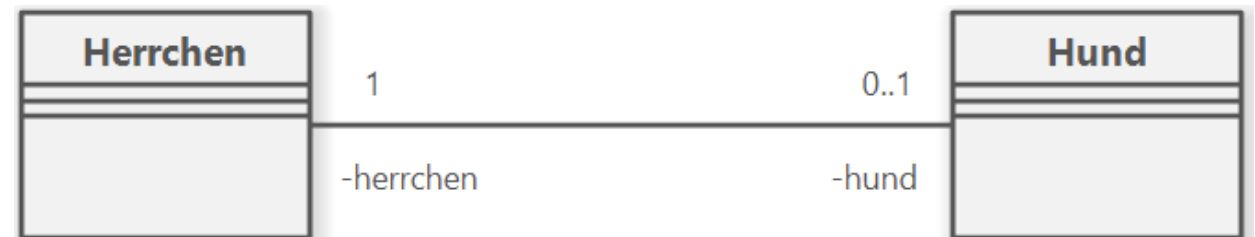




# MULTIPLIZITÄT

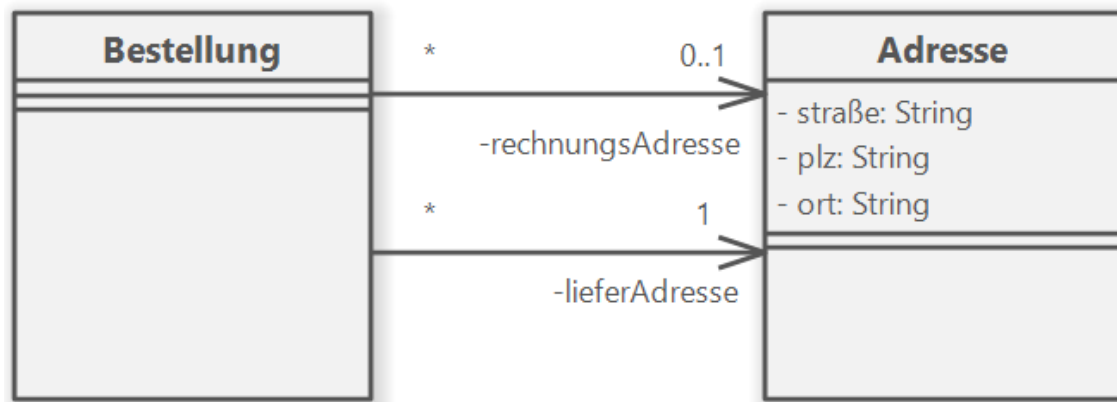
- Multiplizitäten definieren Anzahl der Instanzen, die miteinander in Beziehung stehen
- Werden entlang der Assoziationslinie dargestellt

- 1: Genau einer
- 0..1: Keiner oder einer
- 1..5: Einer bis fünf
- \*: Keiner, einer oder mehrere
- 0..\*: Keiner, einer oder mehrere
- 1..\*: Mindestens einer



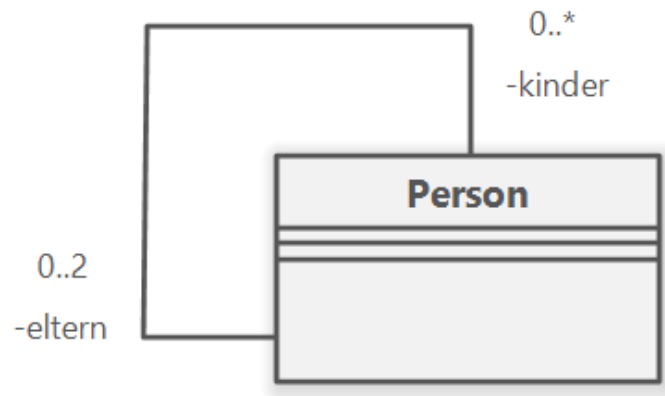
# ASSOZIATION

```
public class Bestellung
{
    private Adresse _lieferAdresse;
    private Adresse _rechnungsAdresse;
}
```



# REFLEXIVE ASSOZIATIONEN

- Reflexive Assoziationen treten auf, wenn ein Objekt eine Verbindung zu einem Objekt derselben Klasse hat
- Beispiele: Die Eltern einer Person sind Personen-Objekte, Kinder einer Person sind ebenfalls Personen-Objekte



```
public class Person
{
    private Person[] _eltern;
    private Person[] _kinder;
}
```

# STATISCHE ASSOZIATIONEN

- Definiert eine Verbindung zwischen Klassen unabhängig von Instanzen dieser Klassen
- Eine statische Assoziation zwischen Klassen wird mit dem Modifizierer `<<static>>` versehen





# AGGREGATION

---

Vincent Uhlmann  
IT-Akademie Dr. Heuer GmbH

# AGGREGATION

- Beschreibt eine Teile-Ganzes-Beziehung mit selbstständigem Teil
- Drückt eine stärkere Bindung zwischen Objekten aus als eine Assoziation
- Die Lebensdauer des Ganzen ist von den Teilen unabhängig
- Das Ganze bleibt bestehen, auch wenn Teile entfernt werden oder nicht mehr existieren

# AGGREGATION

- Aggregation wird durch eine Linie mit einer unausgefüllten Raute dargestellt
- Die Raute wird an der aggregierenden Klasse (dem Ganzen) platziert



# AGGREGATION

- Die Klasse ist nicht für die Erstellung der Schüler verantwortlich
- Eine Klasse kann keine oder viele Schüler haben
- Ein Schüler gehört zu einer oder keiner Klasse
- Die Klasse wäre auch ohne Schüler eine Klasse





# AGGREGATION

- Eine Vorlesung hat Studenten
  - Die Vorlesung aggregiert Studenten
  - Die Vorlesung ist nicht für die Erzeugung der Studenten verantwortlich
  - Ein Student kann an mehreren Vorlesungen teilnehmen
- Eine Abteilung hat Mitarbeiter
  - Die Abteilung aggregiert Mitarbeiter
  - Die Abteilung ist nicht für die Erzeugung der Mitarbeiter verantwortlich
  - Ein Mitarbeiter kann in verschiedenen Abteilungen arbeiten

The background of the slide is a complex network of thin grey lines connecting various sized circles. Some circles are solid blue, some are solid dark blue, and some are white with a dark blue outline. The overall aesthetic is modern and technological.

# KOMPOSITION

---

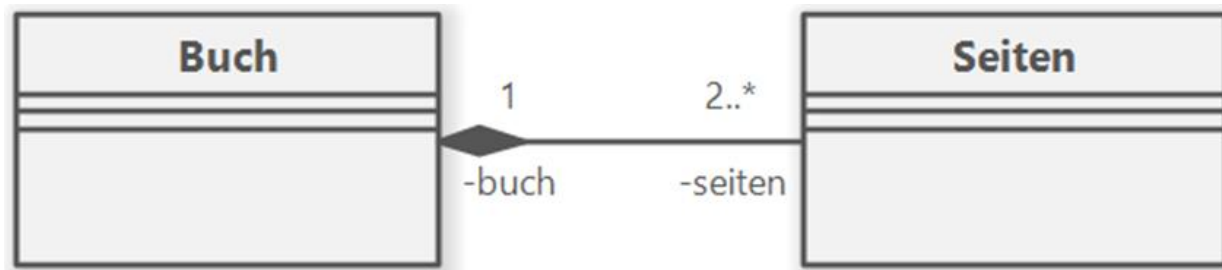
Vincent Uhlmann  
IT-Akademie Dr. Heuer GmbH

# KOMPOSITION

- Spezialfall der Aggregation
- Beschreibt eine Teile-Ganzes-Beziehung mit unselbstständigem Teil
- Teil ohne das Ganze nicht denkbar
- Untrennbare Verbindung zwischen Teil(en) und Ganzem
- Stärkste Form der Bindung

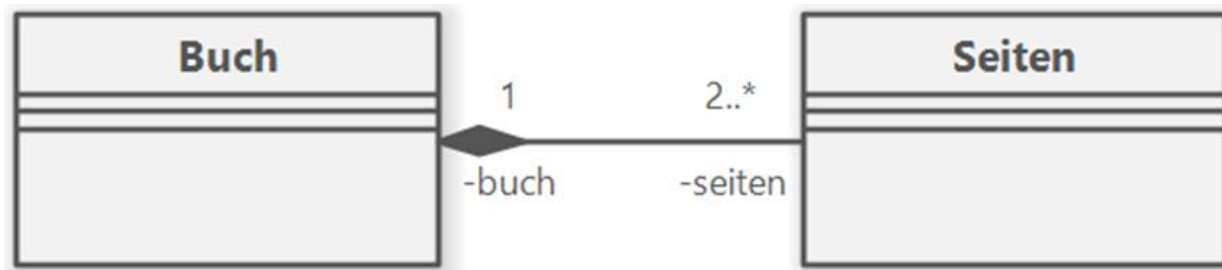
# KOMPOSITION

- Komposition wird durch eine Linie mit einer ausgefüllten Raute dargestellt
- Wird das Ganze zerstört, existieren die Teile nicht mehr
- Wird ein Teil entnommen, kann das Ganze zerstört werden
- Multiplizität ist auf der Seite des Ganzen immer 1
- Das Ganze ist für die Erzeugung der Teile verantwortlich



# KOMPOSITION

- Die Komposition ist eine lebenslange Assoziation
- Sie beschreibt zudem ein ausschließliches Enthaltensein
- Die Seite aus einem Buch kann keine Seite in einem anderen Buch werden



# KOMPOSITION

- Baum und Blatt
  - Der Baum erzeugt die Blätter
  - Wird ein Blatt entnommen, kann es an keinem anderen Baum angebracht werden
- Haus und Räume
  - Durch den Bau eines Hauses entstehen Räume
  - Würden die Räume entnommen werden, würde das Haus aufhören zu existieren