

# Beziehungen zwischen Objekten

M. Jakob

Gymnasium Pegnitz

2. Februar 2019

## Inhaltsverzeichnis

Aggregation und Referenzen

    Begriffsbildung

    Umsetzung in Java

Kommunikation zwischen Objekten — Sequenzdiagramme

Vererbung

# Gliederung

## Aggregation und Referenzen

Begriffsbildung

Umsetzung in Java

Kommunikation zwischen Objekten — Sequenzdiagramme

Vererbung

3/23

## In diesem Abschnitt

### Aggregation und Referenzen

Begriffsbildung

Umsetzung in Java

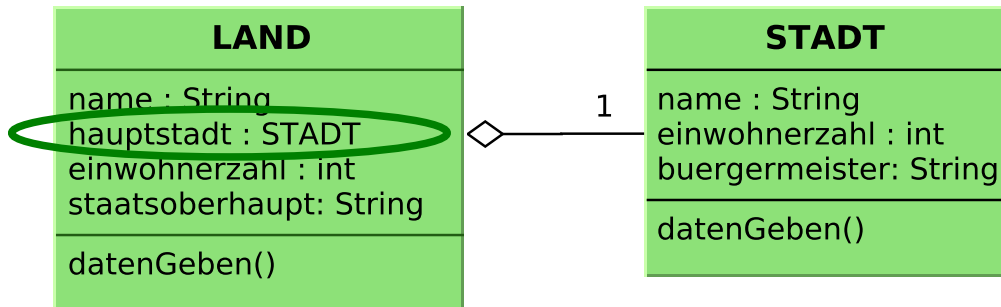
Kommunikation zwischen Objekten — Sequenzdiagramme

Vererbung

## Klassenbeziehungen

Zwischen verschiedenen Klassen können Beziehungen bestehen.

Beispiel:

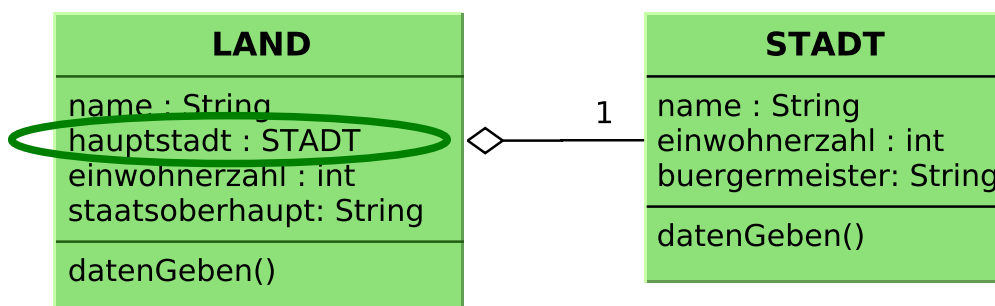


Die Klasse STADT tritt als Hauptstadt in der Klasse LAND auf. Die öffentlichen Methoden der Klasse STADT sind dann angebotene Dienste, die die Klasse LAND nutzen kann.

5/23

## Aggregation

Als Attribute sind in Java auch ganze Klassen erlaubt. Z.B. enthält die Klasse LAND das Attribut hauptstadt vom Datentyp STADT). Im Klassendiagramm wird das durch einen Verbindungslinie mit einer Raute symbolisiert und als **Aggregation** bezeichnet. An die Seite der **genutzten Klasse** wird die Anzahl der enthaltenen Objekte angegeben.



6/23

## Referenzattribute

Dabei werden nicht die Daten des Attributes hauptstadt in dem Speicherbereich des Landes abgelegt. Es wird nur eine Referenzadresse angegeben, unter der die Daten zu finden sind. Wenn die Stadtdatei aktualisiert wurden, sind damit die Daten der Hauptstadt ebenfalls aktualisiert.

Zelle	Inhalt
...	
10000	name
10001	„Deutschland“
10002	hauptstadt
10003	<b>10100</b>
10004	einwohnerzahl
10005	80000000
10006	staatsoberhaupt
10007	„Merkel“
...	
<b>10100</b>	name
10101	„Berlin“
10102	einwohnerzahl
10103	4000000
10104	buergermeister
10105	„Wowerit“

7/23

## In diesem Abschnitt

### Aggregation und Referenzen

#### Begriffsbildung

#### Umsetzung in Java

### Kommunikation zwischen Objekten — Sequenzdiagramme

### Vererbung

## Deklaration Attribute

Das Attribut `hauptstadt` wird als Referenzattribut der Klasse `Stadt` deklariert.

```
1 public class Land
2 {
3     private String name;
4     private Stadt hauptstadt;
5     private int einwohnerzahl;
6     private Person staatsoberhaupt;
```

9/23

## Anpassung der Getter- und Settermethoden

Die Datentypen der Übergabeparameter und Rückgabewerte müssen angepasst werden.

```
1     public Stadt hauptstadtGeben() {
2         return hauptstadt;
3     }
4     public void hauptstadtSetzen(Stadt
5         hauptstadt) {
6         this.hauptstadt = hauptstadt;
7     }
```

10/23

## Anpassung der Methode `datenAnzeigen()`

Auf das Referenzattribut `Hauptstadt` kann nicht mehr mit `System.out.println` ausgegeben werden. Vielmehr muss man die Daten anzeigen lassen, indem man der Klasse `STADT` die **Botschaft** übermittelt, dass die Daten des Objektes `hauptstadt` ausgegeben werden sollen. Das erfolgt mit der Anweisung `hauptstadt.datenAnzeigen()`.

```
1      public void datenAnzeigen() {
2          System.out.println("Landesdatendaten
3              :\n*****");
4          System.out.println("Staatsoberhaupt "
5              );
6          staatsoberhaupt.datenGeben();
          hauptstadt.datenAnzeigen();
          System.out.println("Ende
              Landesdatendaten:\n
                  *****");
7      }
```

11/23

## Übungen

- Ü 1.1: Die Klassen `Land` und `Stadt`
  - (a) Passe die Klassen `Land` und `Stadt` nach obigen Vorgaben an und teste die Klassen ausführlich.
  - (b) Das Klassendiagramm soll um die Klasse `PERSON` erweitert werden.
    - ▶ Ergänze das Klassendiagramm mit yEd. Welche Attribute aus `Land` bzw. `Stadt` werden Referenzattribute? ((Vorlage `KDStadtLand.graphml`))
    - ▶ Implementiere die Klasse `PERSON` vollständig ((Attribute, Getter- Setter-Methoden, Methode `datenAnzeigen()`))
    - ▶ Passe die Klassen `Land` und `Stadt` an die neue Situation an.

# Übungen

- Ü 1.2: BuchAutorVerlag

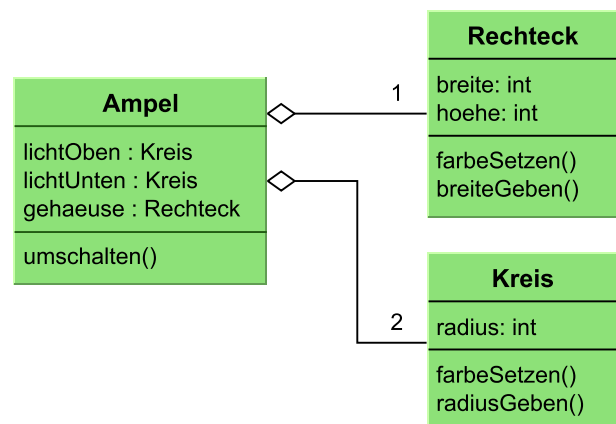
Implementiere das Java-Projekt nach folgender Vorlage:



# Beispiel — Ampel

## Aggregation

Objekte (hier Ampel) können als Attribute andere Objekte (hier Rechteck und Kreis) enthalten. Im Klassendiagramm wird das durch einen Verbindungslinie mit einer Raute bei der Aggregation symbolisiert. An der Seite der Referenzattribute wird die Anzahl der enthaltenene Objekte angegeben.



# Gliederung

Aggregation und Referenzen

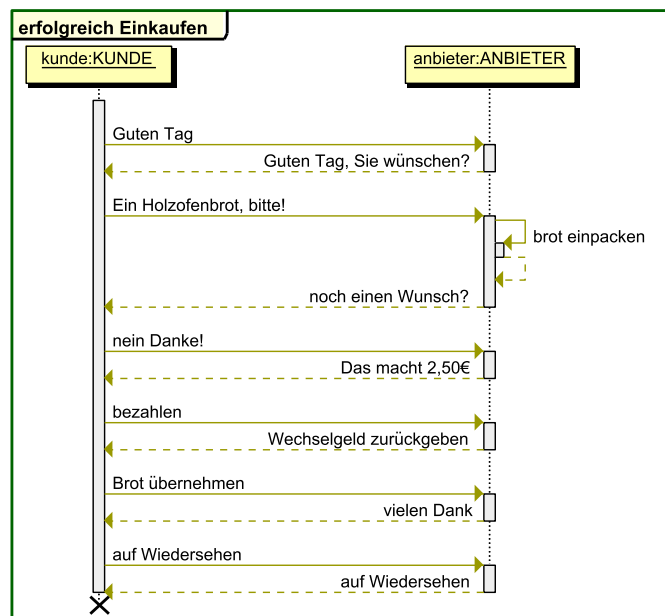
Begriffsbildung

Umsetzung in Java

## Kommunikation zwischen Objekten — Sequenzdiagramme

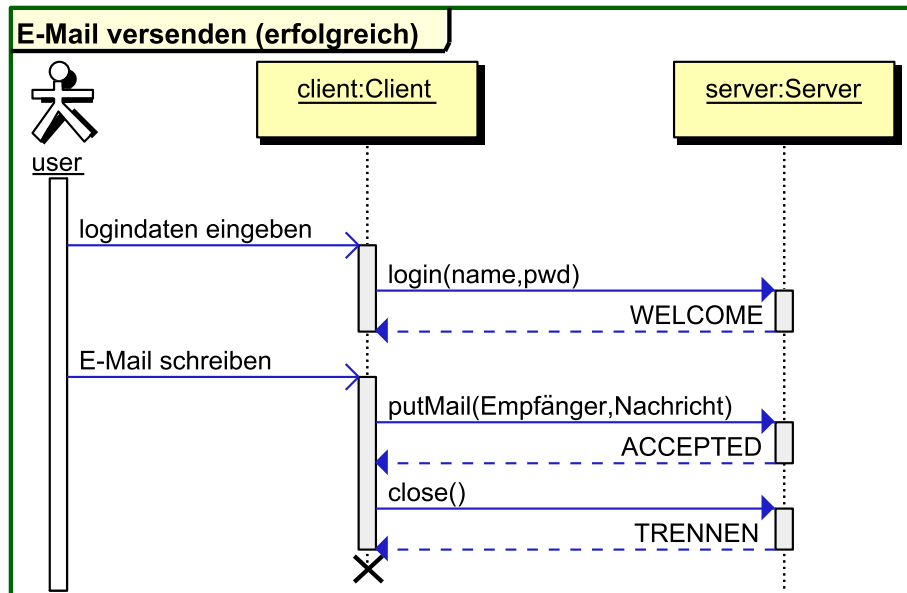
Vererbung

# Anwendungsbeispiel: Sequenzdiagramm beim Einkaufen





## Anwendungsbeispiel: Sequenzdiagramm zum E-Mail-Versand



17/23

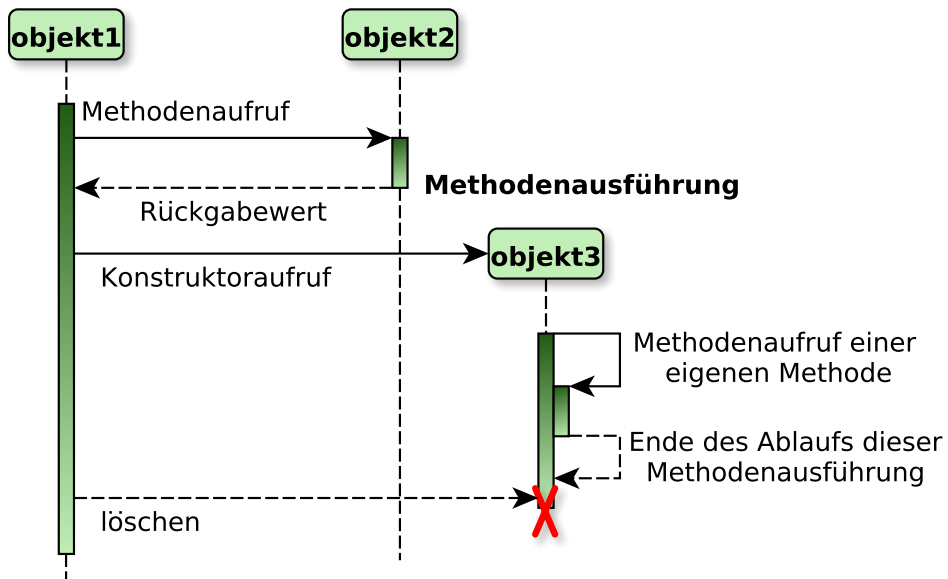
## Sequenzdiagramme ...

beschreiben die Kommunikation zwischen konkreten Objekten in konkreten Situationen.

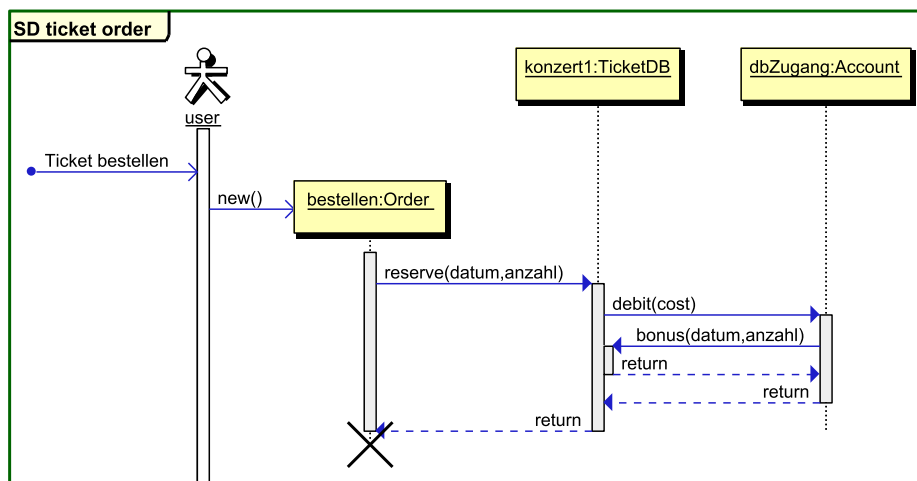
- ▶ Die Zeitachse schreitet im Diagramm von oben nach unten fort.
- ▶ Gestrichelte senkrechte Linien stellen die **Lebenszeit**, die Balken die **Aktivitätszeit** der Objekte dar.
- ▶ Durch waagrechte Pfeile werden die ausgetauschten **Botschaften** dargestellt. „Initialbotschaften“ (d.h. Methodenaufrufe) stehen auf durchgehenden, Antworten (Rückgabewerte) auf gestrichelte Linien.

18/23

## Hauptelemente eines Sequenzdiagramms



## Anwendungsbeispiel: Sequenzdiagramm Kartenbestellung



# Übungen

- Ü 2.1: Sequenzdiagramme zeichnen

Lehrer J teilt Lisa mit, dass sie bei Frau Schmitt die Entschuldigung von Sarah holen soll. Zeichne ein Sequenzdiagramm mit den Objekten `lehrerJ`, `lisa` und `frSchmitt`.

- Ü 2.2: Roland und Julia

`Arbeitsmaterial/Uebungen/Sequenzdiagramme/RolandUndJulia.pdf`

Erläutere den durch das Sequenzdiagramm von Roland und Julia dargestellten Ablauf.

- Ü 2.3: Stadt Land

`Arbeitsmaterial/Uebungen/Sequenzdiagramme/StadtLand/Sequenzdiagramme/Sequenzdiagramme.pdf`

21/23

---

# Gliederung

Aggregation und Referenzen

Begriffsbildung

Umsetzung in Java

Kommunikation zwischen Objekten — Sequenzdiagramme

Vererbung

22/23