

Beziehungen zwischen Objekten

M. Jakob

Gymnasium Pegnitz

2. Februar 2019

1 Aggregation und Referenzen

1.1 Begriffsbildung

1.2 Umsetzung in Java

2 Kommunikation zwischen Objekten — Sequenzdiagramme

3 Vererbung

Gliederung

1 Aggregation und Referenzen

1.1 Begriffsbildung

1.2 Umsetzung in Java

2 Kommunikation zwischen Objekten — Sequenzdiagramme

3 Vererbung

In diesem Abschnitt

1 Aggregation und Referenzen

1.1 Begriffsbildung

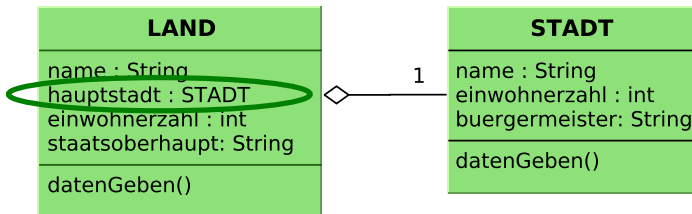
1.2 Umsetzung in Java

2 Kommunikation zwischen Objekten — Sequenzdiagramme

3 Vererbung

Klassenbeziehungen

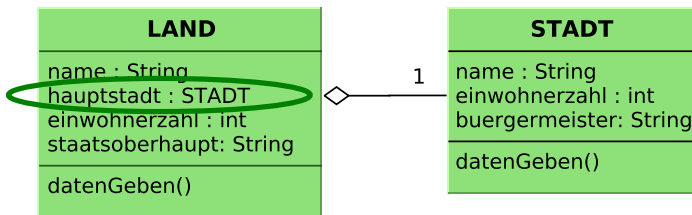
Zwischen verschiedenen Klassen können Beziehungen bestehen.
Beispiel:



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.

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 eine Verbindungslinie mit einer Raute symbolisiert und als **Aggregation** bezeichnet. An die Seite der **genutzten Klasse** wird die Anzahl der enthaltenen Objekte angegeben.



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 Stadt Daten aktualisiert wurden, sind damit die Daten der Hauptstadt ebenfalls aktualisiert.

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

In diesem Abschnitt

1 Aggregation und Referenzen

1.1 Begriffsbildung

1.2 Umsetzung in Java

2 Kommunikation zwischen Objekten — Sequenzdiagramme

3 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;
```

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 hauptstadt
5       ){
6       this.hauptstadt = hauptstadt;
7   }
```

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:\n
        *****");
3     System.out.println("Staatsoberhaupt");
4     staatsoberhaupt.datenGeben();
5     hauptstadt.datenAnzeigen();
6     System.out.println("Ende Landesdatendaten
        :\n*****");
```

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

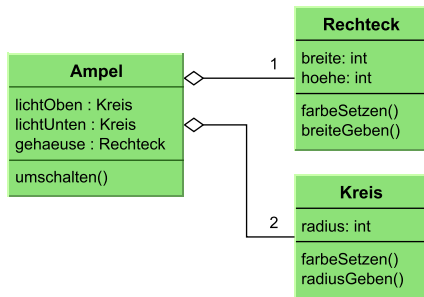
Implementiere das Java-Projekt nach folgender Vorlage:



Beispiel — Ampel

Aggregation

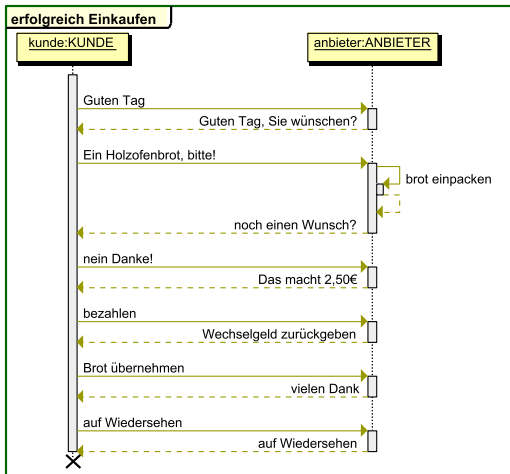
Objekte (hier `Ampel`) können als Attribute andere Objekte (hier `Rechteck` und `Kreis`) enthalten. Im Klassendiagramm wird das durch eine Verbindungslinie mit einer Raute bei der Aggregation symbolisiert. An der Seite der Referenzattribute wird die Anzahl der enthaltenen Objekte angegeben.



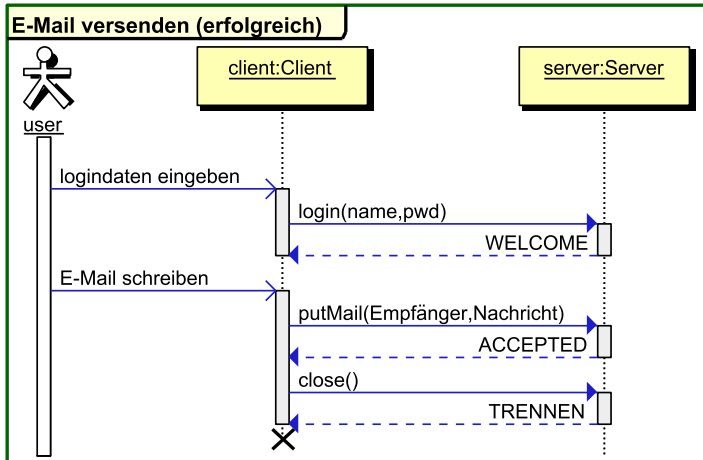
Gliederung

- 1 Aggregation und Referenzen
 - 1.1 Begriffsbildung
 - 1.2 Umsetzung in Java
- 2 Kommunikation zwischen Objekten — Sequenzdiagramme**
- 3 Vererbung

Anwendungsbeispiel: Sequenzdiagramm beim Einkaufen



Anwendungsbeispiel: Sequenzdiagramm zum E-Mail-Versand

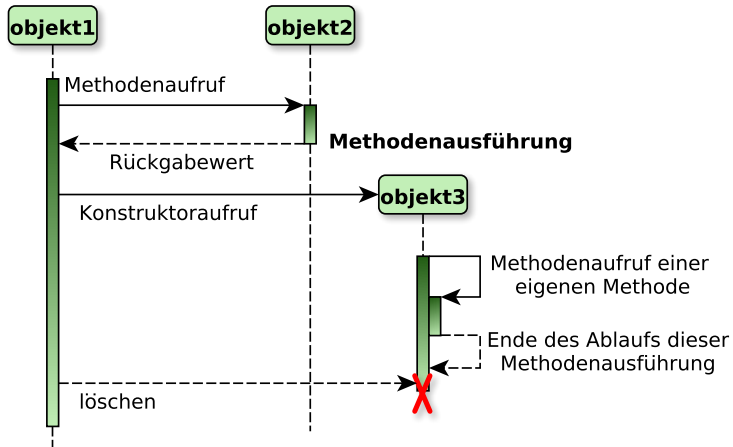


Sequenzdiagramme . . .

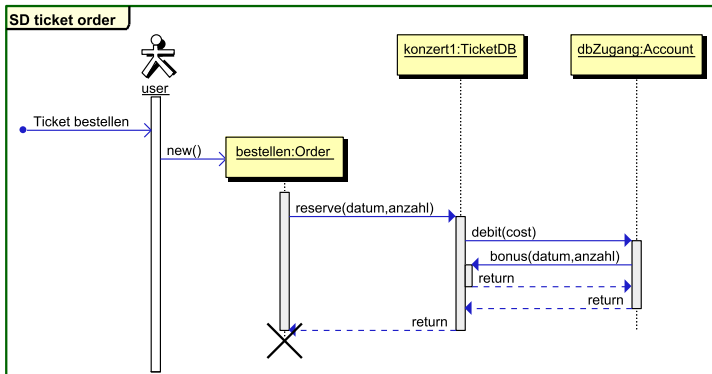
beschreiben die Kommunikation zwischen konkreten Objekten in konkreten Situationen.

- 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.

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

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

➔ Ü 2.3: Stadt Land

Gliederung

1 Aggregation und Referenzen

1.1 Begriffsbildung

1.2 Umsetzung in Java

2 Kommunikation zwischen Objekten — Sequenzdiagramme

3 Vererbung

