

# **PROPÄDEUTIKUM INFORMATIK**

## **SOSE 2018**

Martin Mehlhose

# WIDERHOLUNG

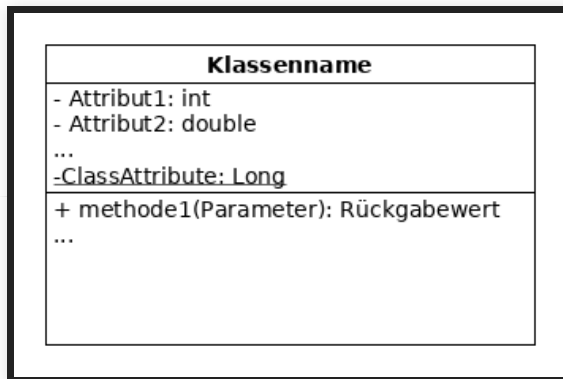
- Umrechnung Kommazahlen
- Sortieren von Daten
- Einstieg UML und Java Objekte

# UML

- UML-Diagramm repräsentiert eine Menge von Objekt-Klassen und deren Beziehungen untereinander
- Eine Objektklasse modelliert ein konkretes Objekt aus der realen oder imaginären Welt.
- Ein Objekt lässt sich beschreiben über seine Eigenschaften (Attribute) und sein Verhalten (Methoden)

# UML

- UML-Klasse teilt sich in 3 Bereiche
- 1. Klassenname
- 2. Klassen- und Objektvariablen
- 3. Klassen- und Objektmethoden



# UML

- Attribute und Methoden gelten entweder für konkrete Objekte (Instanzen) oder für die ganze Klasse
- Klassenobjekte und -methoden werden im UML unterstrichen dargestellt
- Ein + vor einem Attribut oder Methode sagt das dies public ist, ein - entsprechen private
- public bedeutet, dass Attribut/Methode kann von anderen Klassen angesprochen werden
- private Attribute können nur in der eigenen Klasse verwendet werden.

# UML

- Attribute und Methoden gelten entweder für konkrete Objekte (Instanzen) oder für die ganze Klasse
- Klassenobjekte und -methoden werden im UML unterstrichen dargestellt
- Ein + vor einem Attribut oder Methode sagt das dies public ist, ein - entsprechen private
- public bedeutet, dass Attribut/Methode kann von anderen Klassen angesprochen werden
- private Attribute können nur in der eigenen Klasse verwendet werden.

# UMSETZUNG IN JAVA

- Jede Klasse des UML bekommt eine eigene Klasse in Java
- Alle Attribute werden entsprechend als public oder private implementiert
- Attribute und Methoden die Alle Instanzen einer Klasse betreffen werden mit dem Schlüsselwort **static** eingeleitet
- Das aktuelle Objekt kann mit **this** angesprochen werden

# UMSETZUNG IN JAVA

- Java stellt den Standardkonstruktor **Klassenname()** bereit
- Um bei der Erstellung eines Objekts Integritätsbedingungen zu gewährleisten kann dieser überschrieben werden und die Parameterliste ergänzt werden
- Jedes private Attribut benötigt eine GET-Methode wenn es aus anderen Klassen gelesen werden soll, und eine SET-Methode, wenn es geändert werden soll



# UMSETZUNG IN JAVA

- Anlegen eines neuen neuen Objekts in der MAIN-Methode mit **Klassenname bezeichner = new Klassenname(Parameterliste)**
- Unsere Objektklasse fungiert nun also wie ein Datentyp
- Zugriff auf die public Attribute und Methoden mit Punktnotation: **bezeichner.methode()** bzw. bei static methoden **Klassenname.methode()**

# UMSETZUNG IN JAVA

- Besondere Objekt-Methode **toString()** wandelt ein Objekt in einen String um
- Die Standardimplementation reicht aber meist nicht aus und muss von uns ergänzt werden
- Dazu muss vor die Methode **@Override** geschrieben werden

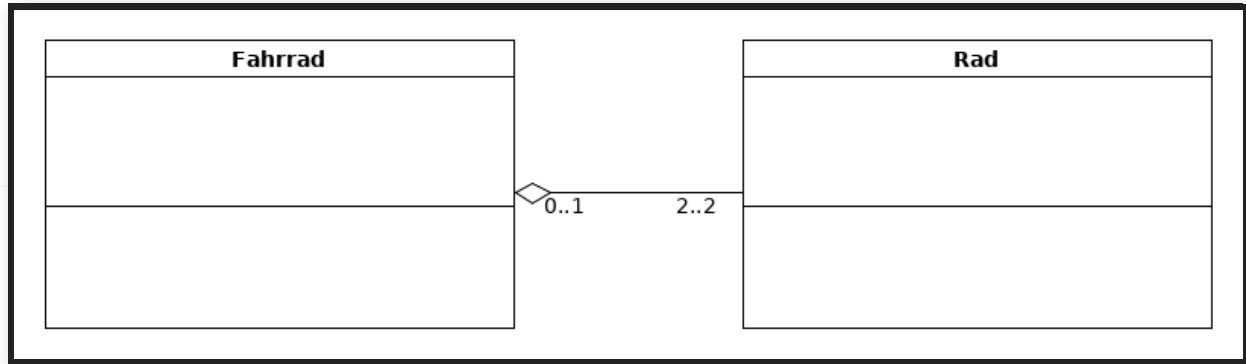
# UML - AGGREGATION

- Aggregation ist eine Möglichkeit komplexe Objekte zu bilden.
- Dabei ist ein Objekt Teil eines anderen Objekts.
- Zum Beispiel: Ein Rad ist Teil eines Fahrrads.
- Wird im UML durch eine Verbindungslinie dargestellt. An der Oberklasse in die das andere Objekt eingeht befindet sich ein Diamond Zeichen.

# UML - AGGREGATION

- Außerdem wird mit Multiplizitäten angegeben, wie viele Elemente an der Beziehung beteiligt sind.
- Multiplizitäten sind angegeben mit **min .. max** wobei  $\text{min} \geq 0$  und max eine beliebige Zahl oder \* für unendlich

# UML - AGGREGATION



# UML - VERERBUNG

- Eltern-Kind Beziehung zwischen zwei Klassen
- Kindklasse erbt alle Attribute und Methoden der Elternklasse
- Methoden die in der Kindklasse anders interpretiert werden sollen müssen mit @Override überschrieben werden

# UML - VERERBUNG

- Die Kindklasse wird in Java eingeleitet mit **public class Klassenname extends Elternklasse**
- Erster Aufruf im Konstruktor der Kindklasse muss der Konstruktor der Elternklasse mit **super(Parameterliste)** sein
- Danach kann auf die Methoden und Attribute zugegriffen werden, als ob es die eigenen wären.
- Vererbung spart Schreibarbeit und vermeidet ungewollte Copy-Paste-Fehler

# UML - VERERBUNG

