

# UML MODELLIERUNG

Dozent: Dr. Andreas Jäger

# UML

Unified Modeling Language

Dient zum erstellen von Software Entwürfen

→ Dient zur statischen Modellierung

→ Dient zur Verhaltensmodellierung

# Diagrammarten

## Statische Modellierung

- Klassendiagramme
- Paketdiagramme
- Kompositionstruktur
- ...

# Diagrammarten

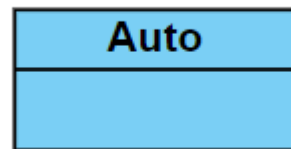
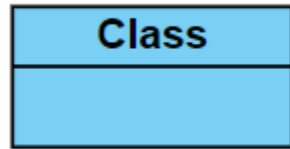
## Verhaltensmodellierung

- Use Case Diagramme
- Sequenzdiagramme
- Aktivitätsdiagramme
- Zustandsdiagramme (State)
- ...

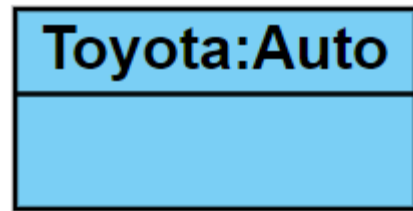
# ENTITÄTEN

# Klassendiagramme

Entität Klasse

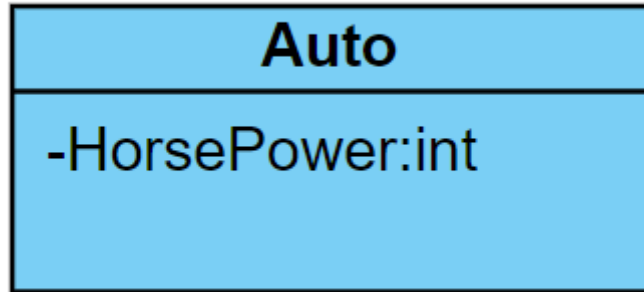


# Klassendiagramme



Instanz Toyota der Klasse Auto

# Klassendiagramme - Attribute

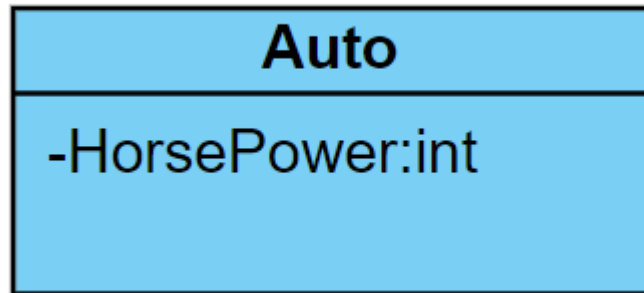


<sichtbarkeit> <name>:<typ>



# ATTRIBUTE

# Klassendiagramme - Attribute

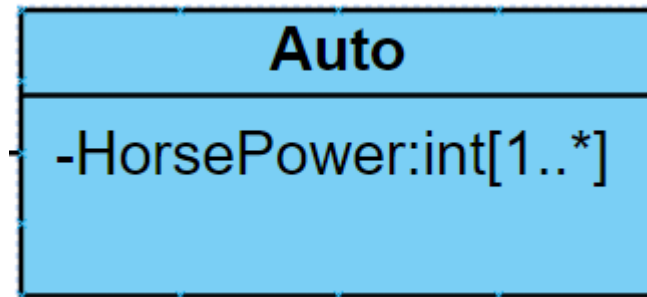
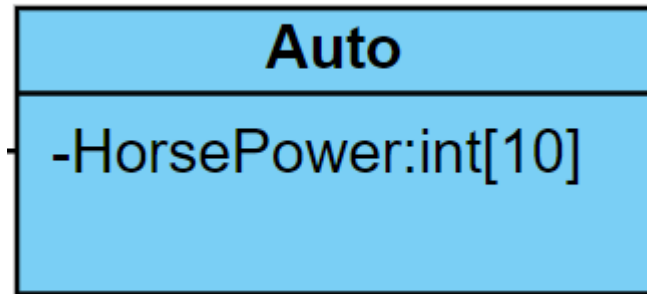


Sichtbarkeit ::= {+|-|#|~}

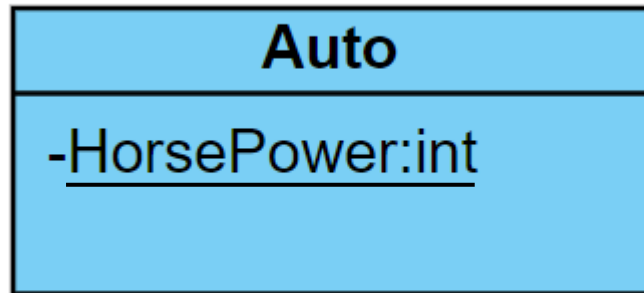
- Public
- Private
- Protected
- Package

# Klassendiagramme - Attribute

Multiplicity ist möglich

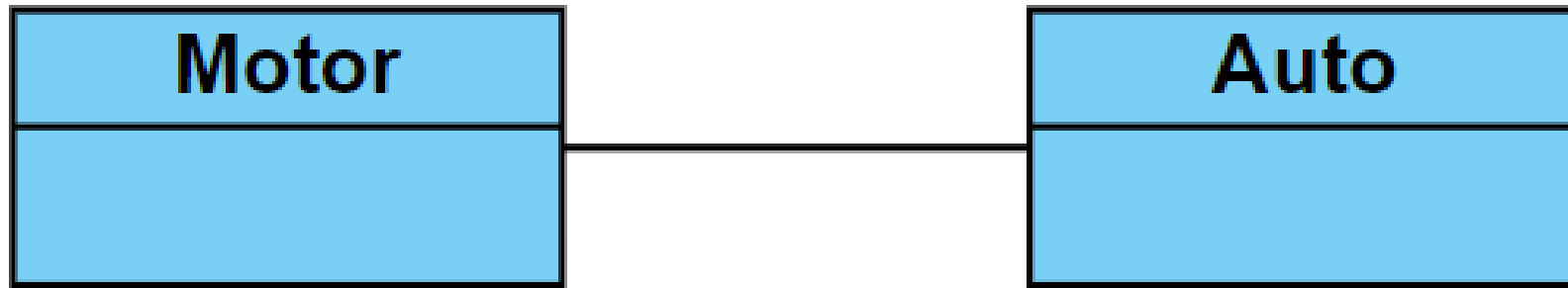


# Klassendiagramme - Attribute



Static attribute

# Attribute als Beziehung

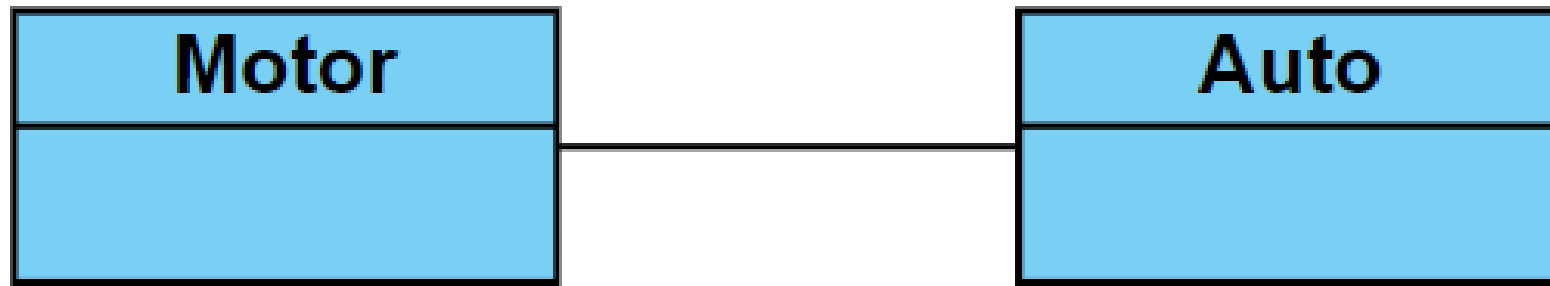


Ein Auto hat einen Motor.

Mehr Genauigkeit

Abhängigkeit wird visualisiert

# Attribute als Beziehung



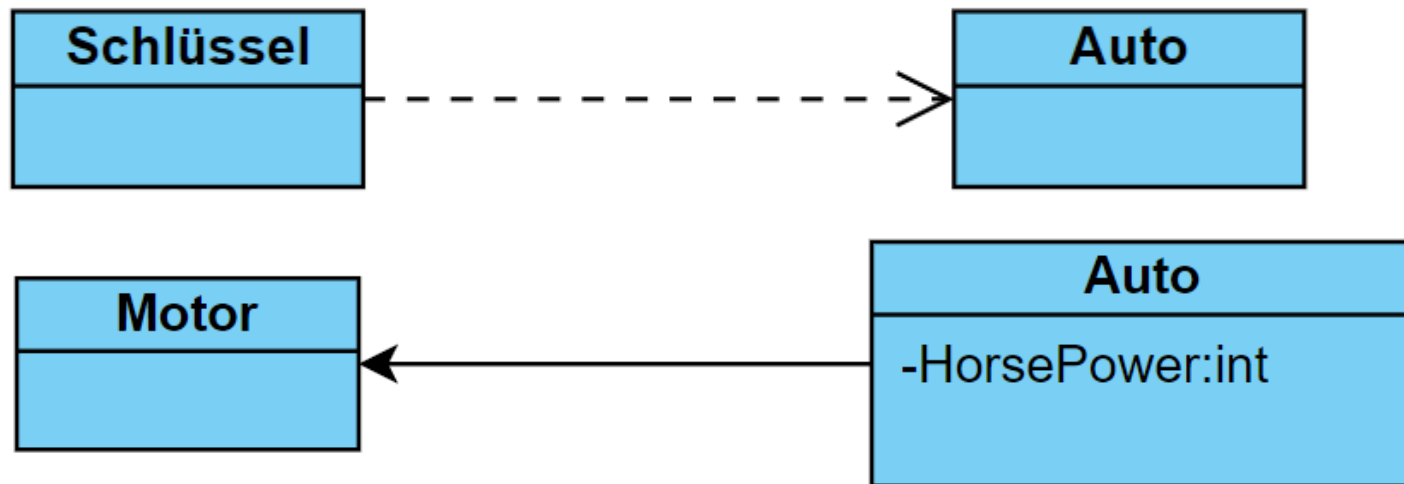
Ein Auto hat einen Motor.

Mehr Genauigkeit

Abhängigkeit wird visualisiert

# Dependency

Abbildung einer Beziehung zwischen 2 Klassen.  
Auto benutzt Motor



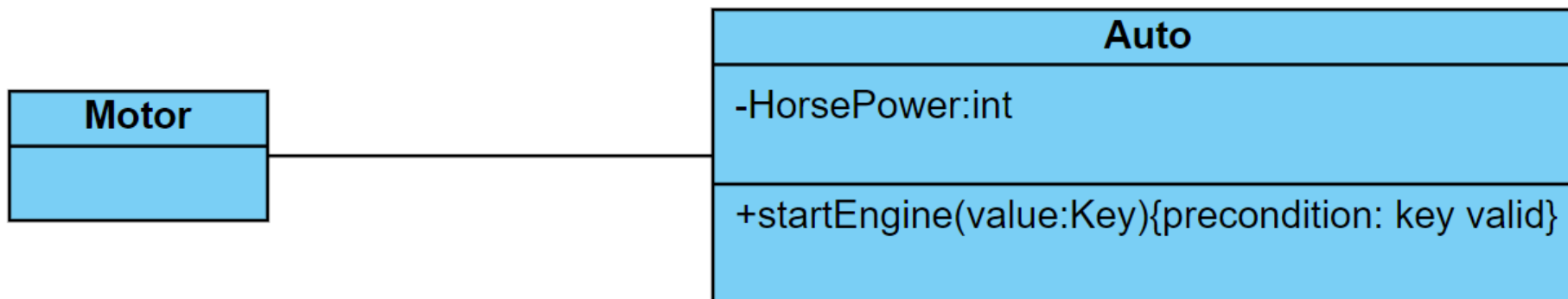
# Assoziation

Abbildung einer Beziehung zwischen 2 Klassen.

Auto besitzt Motor

Auto hat einen Motor

...



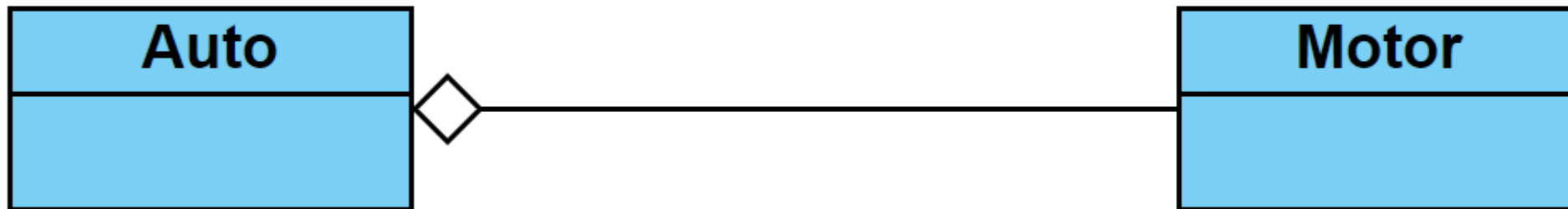


# Aggregation

Stärkere Bindung

... besitzt ein...

Auto besitzt einen Motor

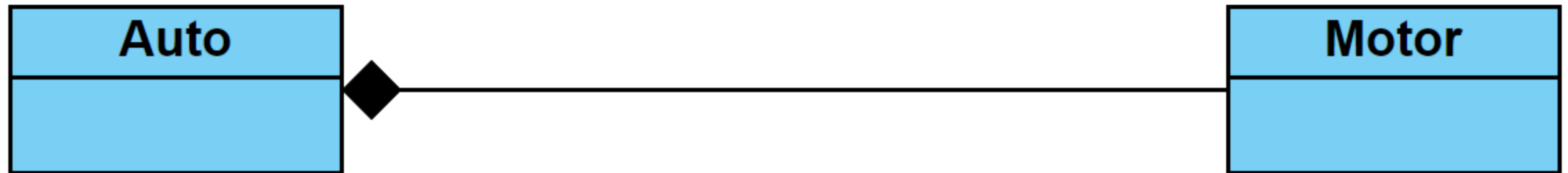


# Komposition

Sehr Starke Bindung

... Kann nicht ohne ... existieren

Das Auto kann nicht ohne Motor existieren

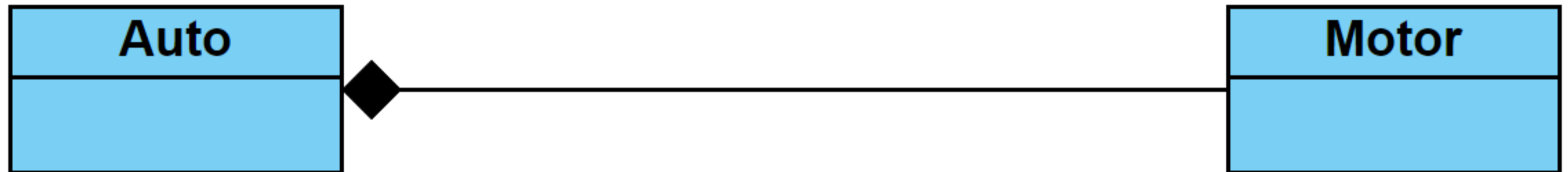


# Komposition

Sehr Starke Bindung

... ist Teil von...

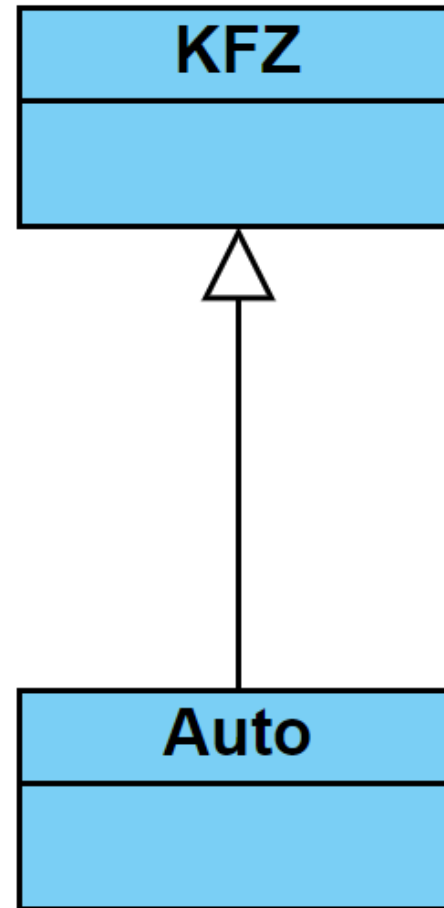
Der Motor ist Teil des Autos



# VERERBUNG

# Generalisierung

Vererbung Abbildung



# OPERATIONEN/METHODEN

# Operationen (Methoden)

<sichtbarkeit><name>(<parameter>):<rückgab  
etyp> {<eigenschaften>}

Auto
-HorsePower:int
+startEngine(value:Key){precondition: key valid}

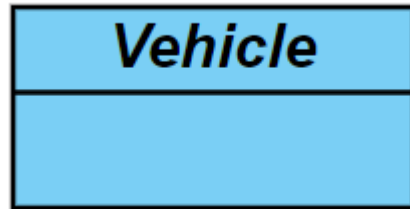
# Operationen (Methoden)

<sichtbarkeit><name>(<parameter>):<rückgab  
etyp> {<eigenschaften>}

Auto
-HorsePower:int
+startEngine(value:Key){precondition: key valid}



# Abstarkte Klassen



*kursiv*