BAUINFORMATIK

SS 2013 Vorlesung IV

Johannes Lange

Allgemeines

- NEU Allgemeine Punkte?
- Fragen?
- Test-Qicky
  - Vorbereitung ok?
  - Zu schwer/ zu leicht?
  - Zeit / Inhalt?
  - Ideen zur Verbesserung…

VBA (Visual Basic for Applications)

- Was machen wir?
  - Wiederholung
    - Benutzerdefinierte Dateitypen
    - Imperative und Objektorientierte Programmierung
    - Klassendiagramm Auto
  - Und dann:
    - Anwendung Objektorientierte Programmierung
    - Public, Private
    - .Set .With
    - Getter Setter

Benutzerdefinierte Datentypen

- Benutzerdefinierte Datentypen
  - Gruppierung von mehreren Variablen

```
Public Type PersAngaben
    LfdNr As Integer
    Vorname As String * 5    'Grössenvorgabe 5 Zeichen
    Name As String
End Type

Sub Test_Dateityp()
    Dim person As PersAngaben

    person.Vorname = "Peter Otto"
    person.Name = "Schubert"

    MsgBox (person.Vorname & " " & person.Name)
End Sub
```

Imperative Programmierung

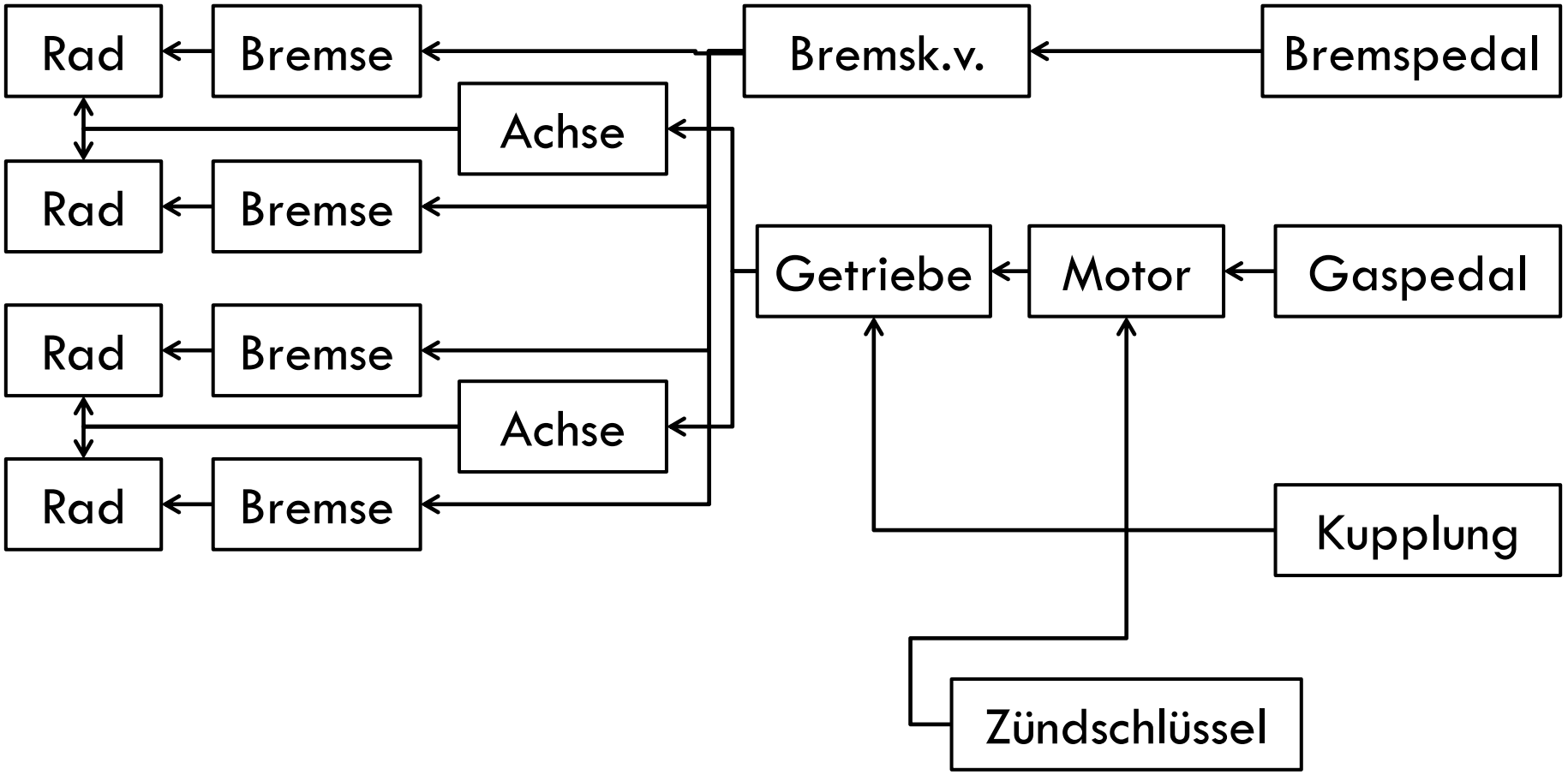
- Strukturierte Programmierung
  - Blöcke hintereinander
  - Schleife, Bedingung, etc.
- Prozedurale Programmierung
  - Funktion (Prozeduren)

```
Dim Wert As Integer

Wert = 0
For i=1 To 10
  Wert = Wert +i
Next
If Wert= 55 Then
   Wert=100
End If
…
```

Objektorientierte Programmierung

- „Welt“ durch Objekte modelliert
- Objekte
  - Sind gekapselt
  - Klar definierte Schnittstellen nach außen
  - Enthalten Methoden (Funktionen) und Attribute (Variablen, andere Objekte)
- Klassen sind die Baupläne der Objekte

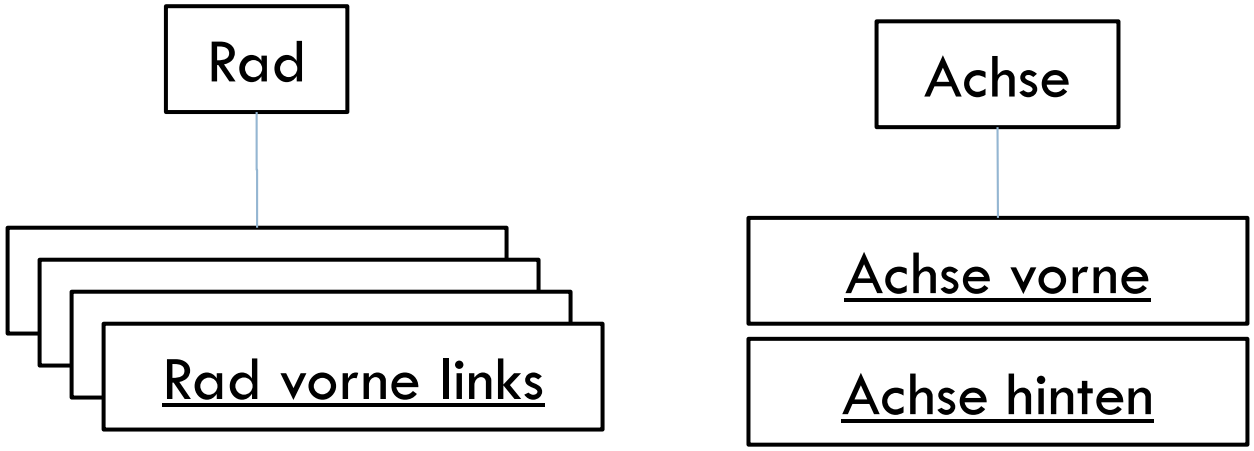
Was sind Objekte? Bsp.: Auto



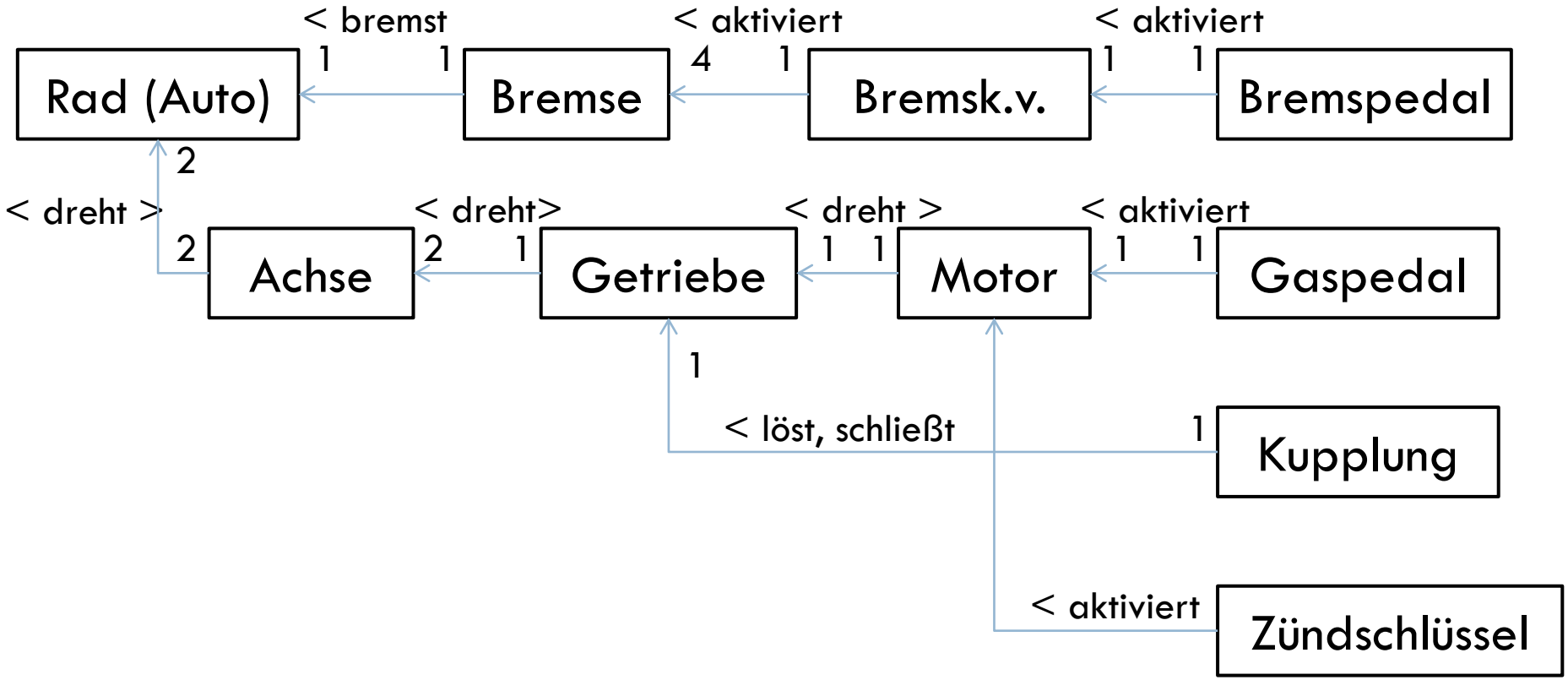
Klasse ←→Objekt

- Klasse ist der Bauplan eines Objekts
- Klasse kann beliebig viele Objekte bauen (Fabrik)

  = Klasse erstellt eine Instanz
- Das Objekt führt Funktion aus – nicht die Klasse

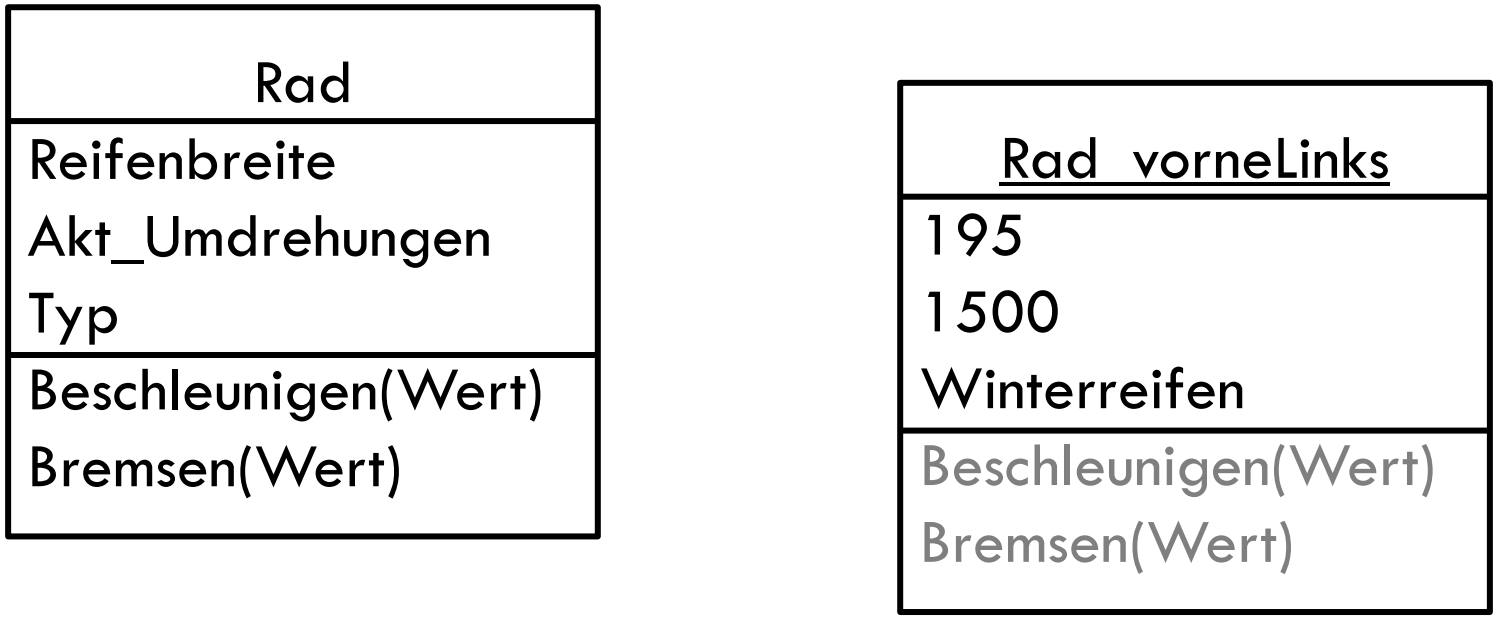


Analyse Klassendiagramm



Objekt / (Klasse als Bauplan)

- Klasse und Objekt haben jeweils einen Namen
- Attributen (Variablen, Member…)
- Methoden (Funktionen/Prozeduren)



1. Klasse erstellen

- Einfügen / Klassenmodul
- Öffentliche Variable erzeugen

```
Public OpenValue As String
```

- Öffentliche Funktion erzeugen

```
Public Function OpenFunction()
    MsgBox ("Public Funcion")
    OpenFunction = "ok"
End Function
```

2. Objekt der Klasse erzeugen

Excel

- Aufruf aus einem anderen Modul
- Objekt-Variable wird deklariert

```
Dim Obj_Klasse1 As Klasse1          'Nur Deklaration
```

- Objekt der Klasse *New* erzeugen
- Objekt der Variablen mit *Set* zuweisen

```
Set Obj_Klasse1 = New Klasse1
```

3. Mit Objekt arbeiten

- Mit dem „Punkt“ können jetzt Funktionen und Variablen aufgerufen werden

```
Obj_Klasse1.OpenValue = "Hallo Welt"
Testfkt = Obj_Klasse1.OpenFunction()
```

Initialize und Terminate

- Initialize wird beim Erzeugen eines Objekts aufgerufen (Konstruktor)

  → Voreinstellungen
- Terminate wird beim Zerstören eines Objekts aufgerufen (Destruktor)

  → Speichern, Zerstören…

Public / Private

- Variablen und Funktionen sind aufrufbar (gültig):
  - Public: in allen Modulen
  - Private: nur im aktuellen Modul

- Wozu?
  - Kapselung →Zugriff nur auf offizielle Schnittstellen
  - Gerade in der Objektorientierung wichtig

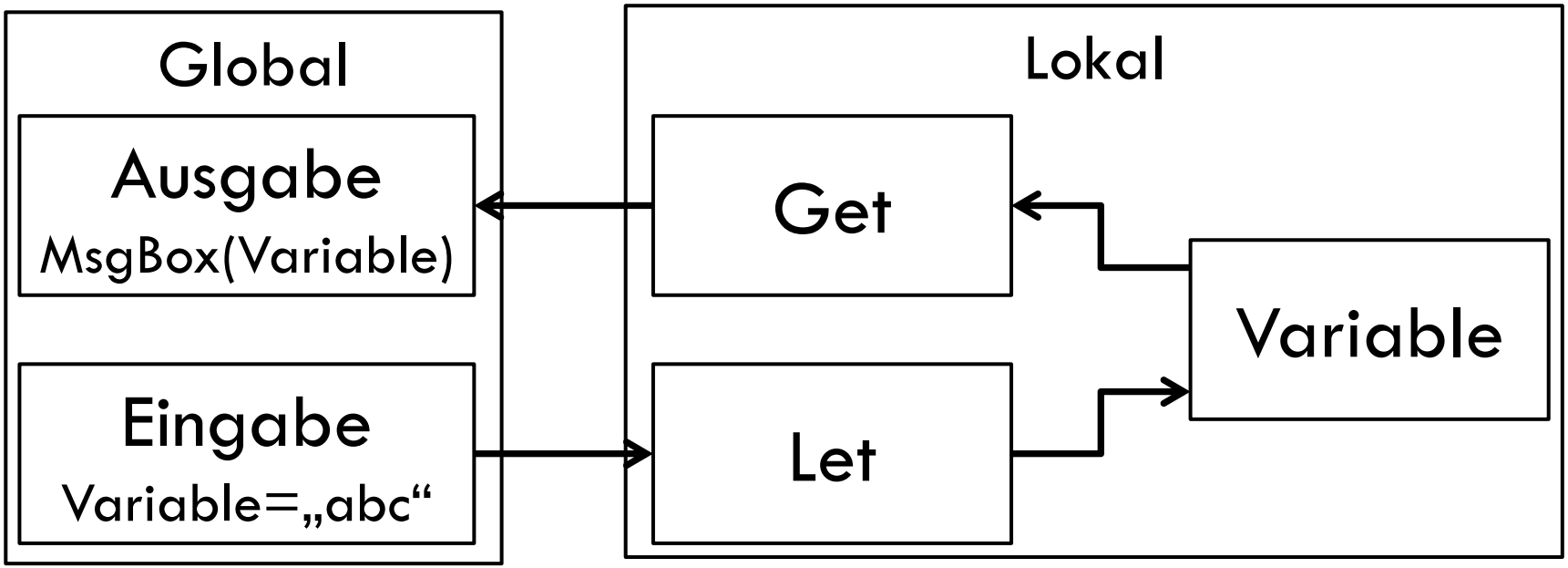
Property

Excel

- Variablen über get, let indirekt ansteuern

→ Kapslung der Variablen

→ Prüfung/Veränderung der Manipulation



Hilfsmittel „With“

Excel

- With (*klasse*)

→ Attribute und Methoden von Klasse beginnen mit „.“

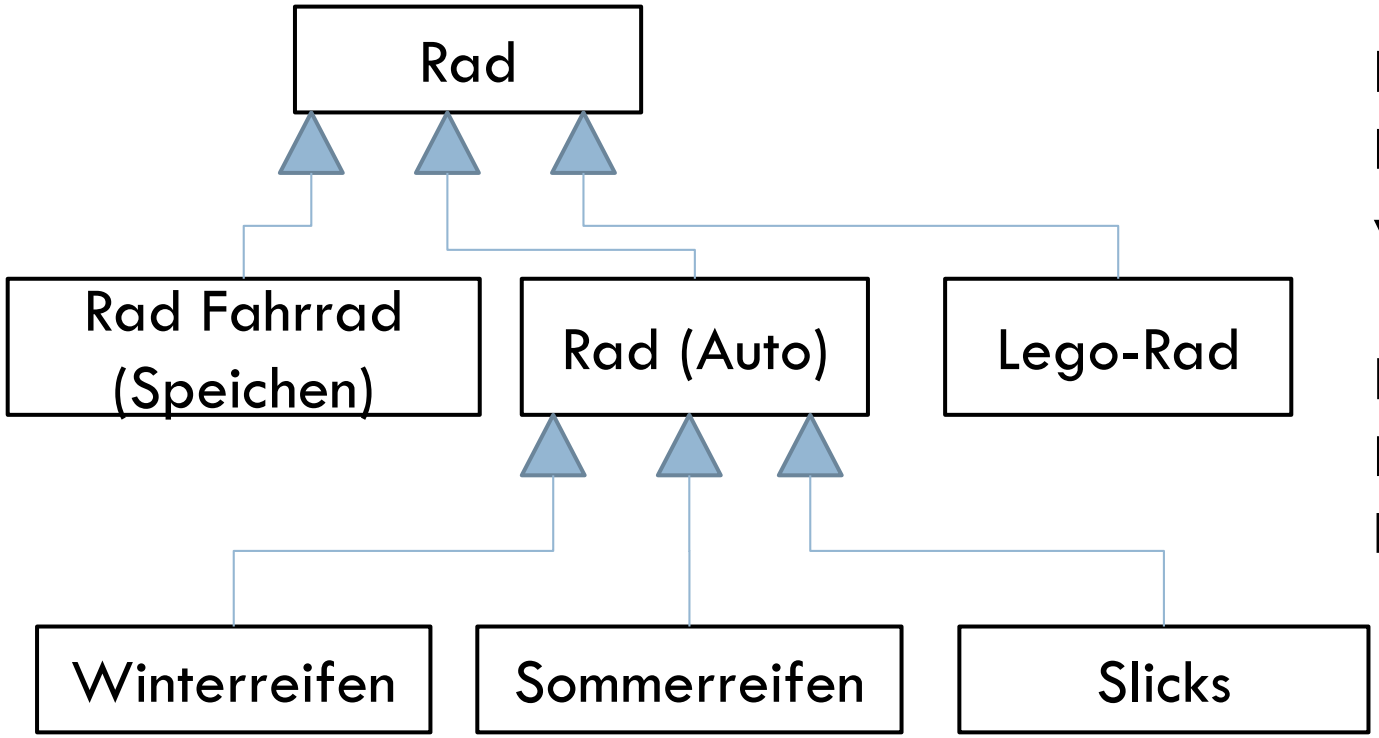
```
'Immer vollständig:
Rad_HintenLinks.Beschleunigen (10)
Rad_HintenLinks.Bremsen (5)
Rad_HintenLinks.Bremsen (3)

'Oder schöner:
With Rad_HintenLinks
     .Beschleunigen (10)
     .Bremsen (5)
     .Bremsen (3)
End With
```

Allgemein OO

- Grundstrukturen Objektorientierung
  - Kapselung
  - Vererbung ← in VBA nicht möglich
  - Polymorphie ← in VBA nicht möglich

→ Vererbung und Polymorphie sind in Visual Basic (VB) möglich!

Vererbung / Polymorphie



Erben:
Kinder erben von Eltern

Polymorphie:
Kindern können Eltern vertreten

→ Unterschied zwischen VBA und VB