

BAUINFORMATIK

SS 2013 Vorlesung III

Johannes Lange

Allgemeines

2

- NEU Allgemeine Punkte?
- Fragen?

- Test-Qicky nächste Woche
 - ▣ Lernen mit Fragen auf den Übungszetteln
 - ▣ Beispielklausur
 - ▣ Dauer 15 Minuten, direkt am Anfang, hier
 - Ohne Hilfsmittel
 - Bitte pünktlich kommen!

VBA (Visual Basic for Applications)

3

- Was machen wir?
 - ▣ Wiederholung
 - Dialoge
 - Excelfunktionen
 - Benutzerdefinierte Dateitypen
 - ▣ Imperative und Objektorientierte Programmierung
 - Klassendiagramm Auto
 - Public private
 - .Set .With
 - Getter Setter

Programmelemente



4

Dokumentation

- Kommentare, Struktur des Codes

Datenspeicher

- Variablen, Konstanten
- Felder
- Benutzerdef. Datentypen
- Klassen (OO)

Programmstruktur und -ablauf

- Sub und Funktion
- Verzweigung
- Schleife

Office Anwendung

- Excel

Bearbeitung und Prüfung

- Operatoren
- Prüffunktionen (neu)
- Textfunktionen
- Verzeichnisse und Dateien
- Zeiten

Gui

- Steuerelemente & Dialoge

Dialoge



5

- Dialoge
 - Standard-Dialoge
 - MessageBox
 - InputBox
 - Load/Save-Dialoge
 - Eigene Dialoge mit Steuerelementen

MsgBox

InputBox

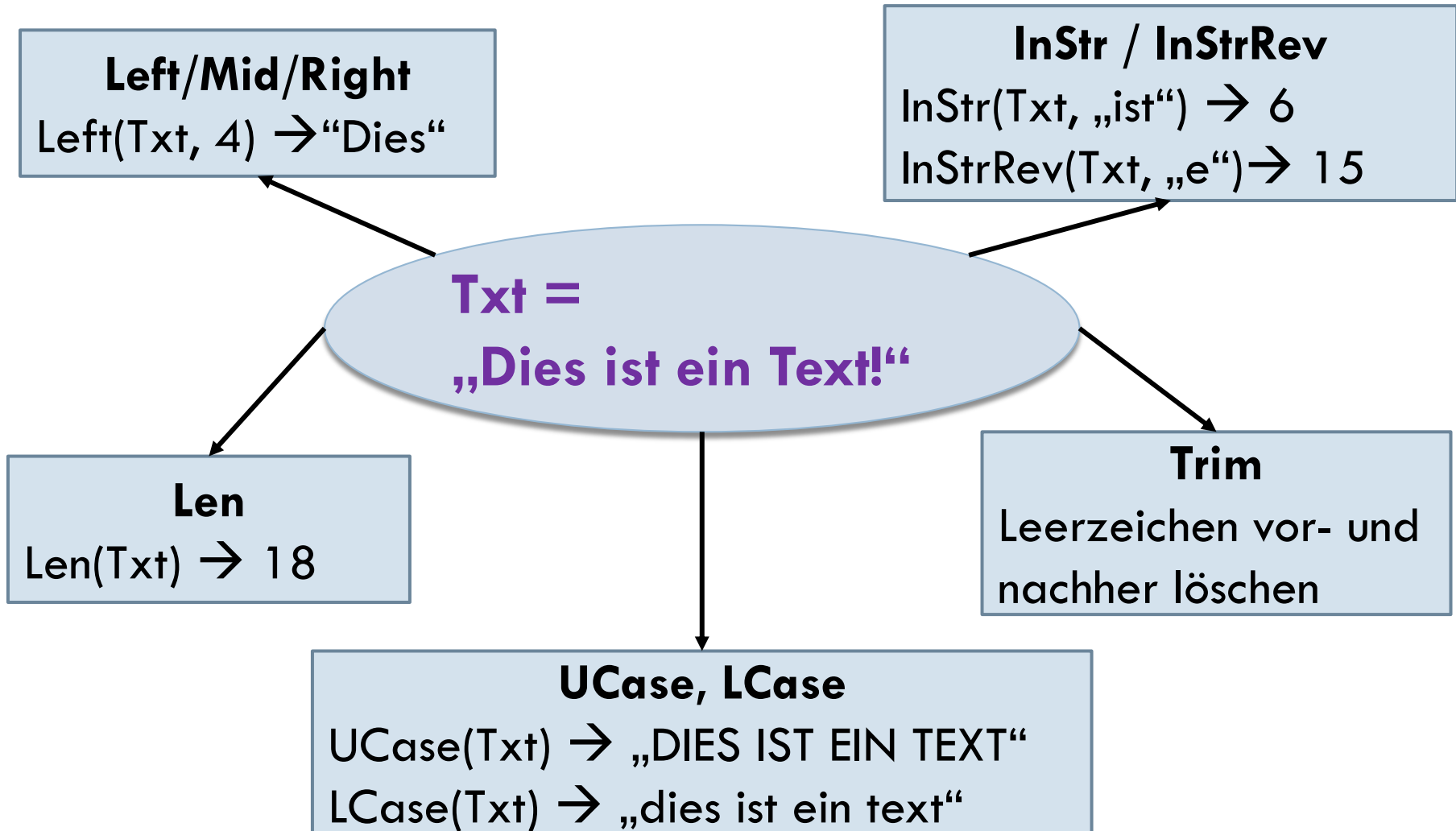
Load - Dialog

Freier Dialog

Textfunktionen



6



Application, Workbook, Sheet, Range

7

- Application Anwendung
- Workbook File
- Sheet Arbeitsblatt
- Range / Cells Felder
 - ▣ Name Name
 - ▣ Select Bereich markieren
 - ▣ Activate Ansicht
 - ▣ Value Wert abfragen
 - ▣ Cut Ausschneiden
 - ▣ Copy Kopieren

Position in Excel



8

□ ActiveCell

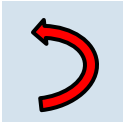
- Aktives Feld in Excel
- Details Abfragbar über `.Address` `.Row` `.Column`

```
Sub exceltest2 ()  
    Position = ActiveCell.Address  
    Position = ActiveCell.Row  
    Position = ActiveCell.Column  
    Position = ActiveCell.Parent.Name  
    Position = ActiveSheet.Parent.Name  
End Sub
```

□ ActiveSheet

- Aktives Tabellenblatt
- `ActiveCell.Parent` → `ActiveSheet`
- `ActiveSheet.Parent` → `File`

Offset (Excel)



9

□ Offset: Relative Zellenwahl

`ActiveCell.Offset(x,y).Select`

! Vorsicht:

- Negativer Offset möglich, aber
 - Prüfe dass Zellenposition existiert.
- Eingefügte Zeilen/ Reihen verschieben den Offset

Benutzerdefinierte Datentypen

10

Excel

- Benutzerdefinierte Datentypen
 - Gruppierung von mehreren Variablen

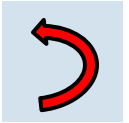
```
Public Type PersAngaben|
    LfdNr As Integer
    Vorname As String * 5    'Grössenvorgabe 5 Zeichen
    Name As String
End Type
```

```
Sub Test_Datentyp()
    Dim person As PersAngaben

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

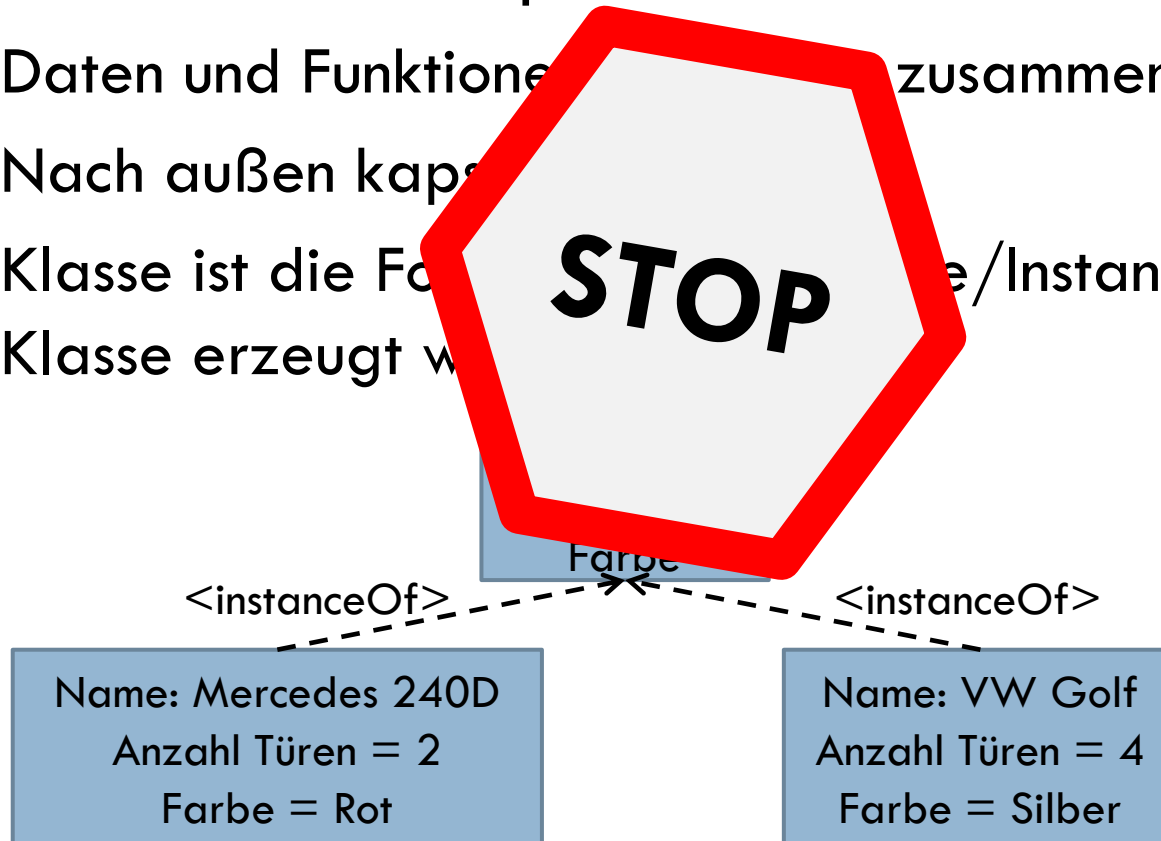
    MsgBox (person.Vorname & " " & person.Name)
End Sub
```

Klassen (OO)



11

- VBA ist Objektorientiert (Klassenstruktur)
- Was bedeutet Objektorientiert?
 - ▣ Daten und Funktionen zusammenschließen
 - ▣ Nach außen kapseln
 - ▣ Klasse ist die Fabrik/Instanzen der Klasse erzeugt werden



Imperative Programmierung

12

- 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
```

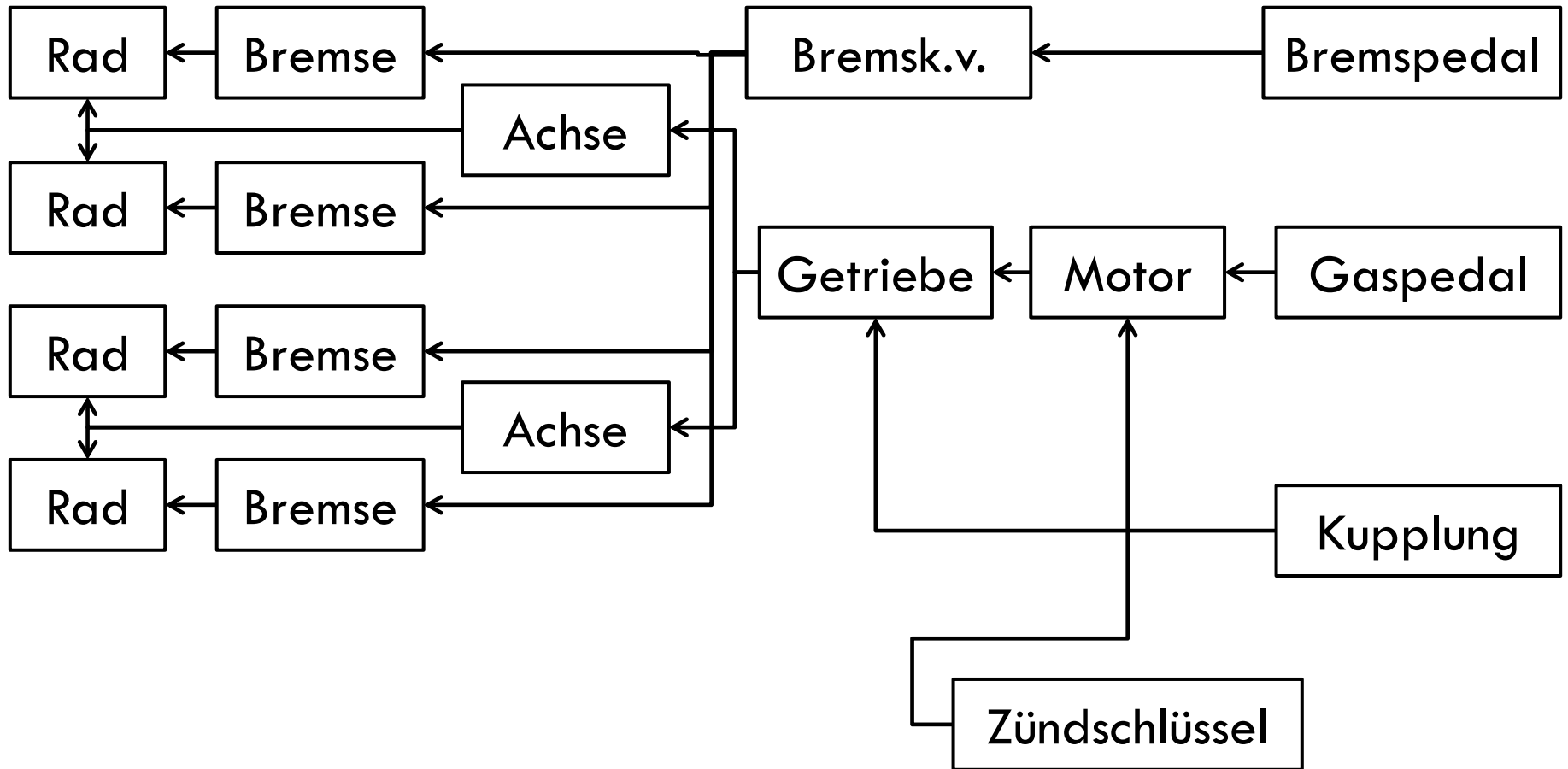
```
    Wert=100
```

```
End If
```

```
...
```

Was sind Objekte? Bsp.: Auto

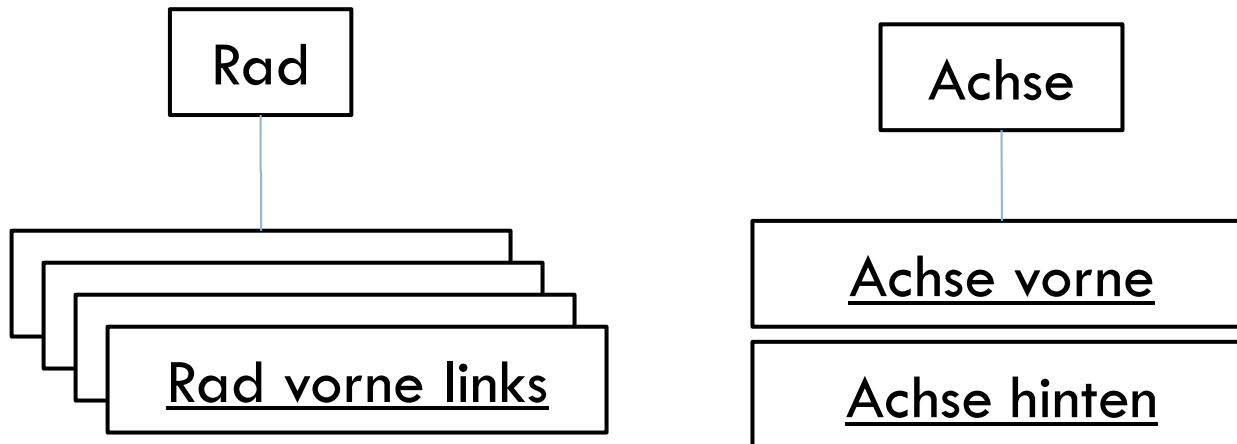
13



Klasse \leftrightarrow Objekt

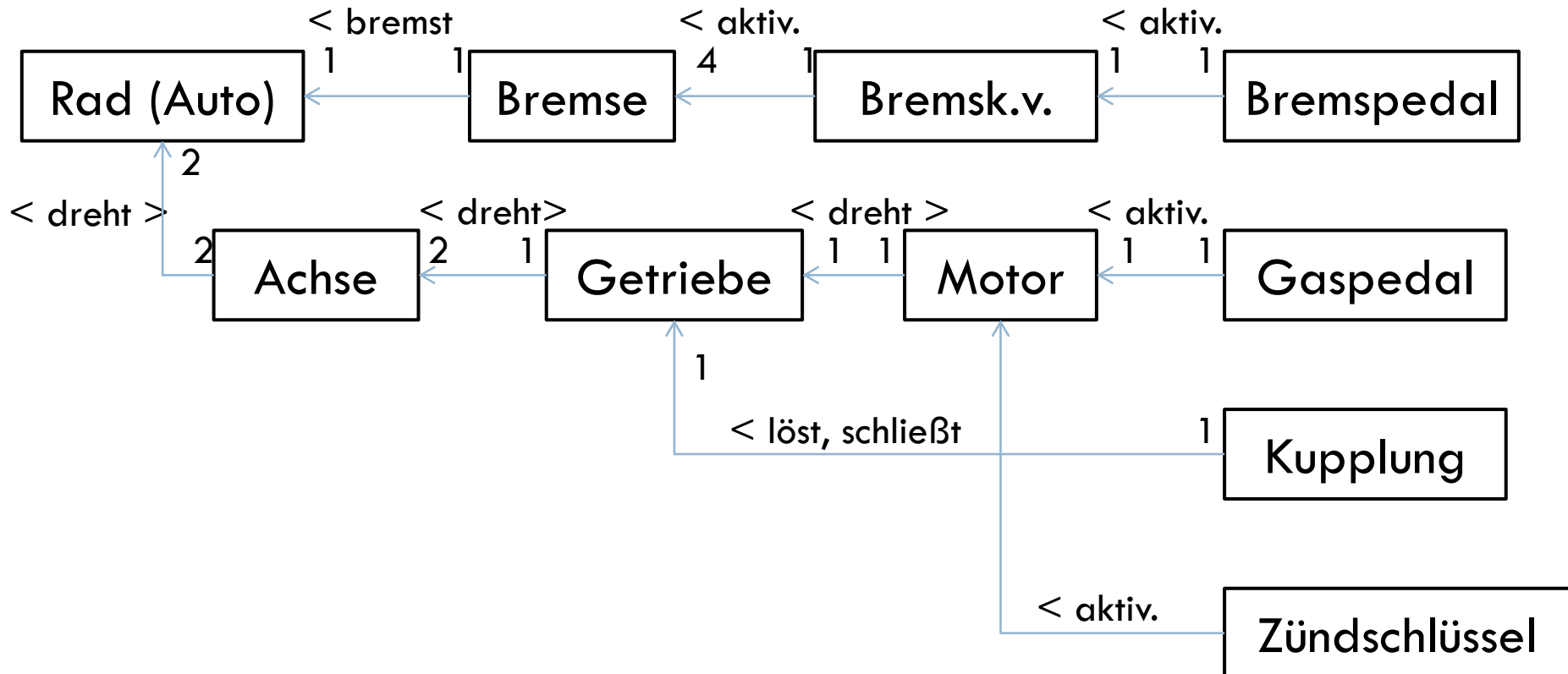
14

- 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

15



Objekt / (Klasse als Bauplan)

16

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

Rad
Reifenbreite
Akt_Umdrehungen
Typ
Beschleunigen(Wert)
Bremsen(Wert)

<u>Rad_vorneLinks</u>
195
1500
Winterreifen
Beschleunigen(Wert)
Bremsen(Wert)

Objekt erzeugen

17

- Objekt-Variable wird deklariert
- Objekt-Variable wird mit set und new erzeugt

```
'Rad-Objekt erzeugen  
Dim Rad_HintenLinks As Rad  
Set Rad_HintenLinks = New Rad
```

Mit Objekt arbeiten



18

- Klasse ist ein eigenes Modul
- Ein Objekt von dieser Klasse wird instanziiert

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

ObjektName.Funktionsaufruf()

Bsp.: Rad_LinksVorne.Beschleunigen(100)

Initialize und Terminate



Excel

19

- Initialize wird beim Erzeugen eines Objekts aufgerufen (Konstruktor)
- Terminate wird beim Zerstören eines Objekts aufgerufen (Destruktor)

Public / Private



20

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

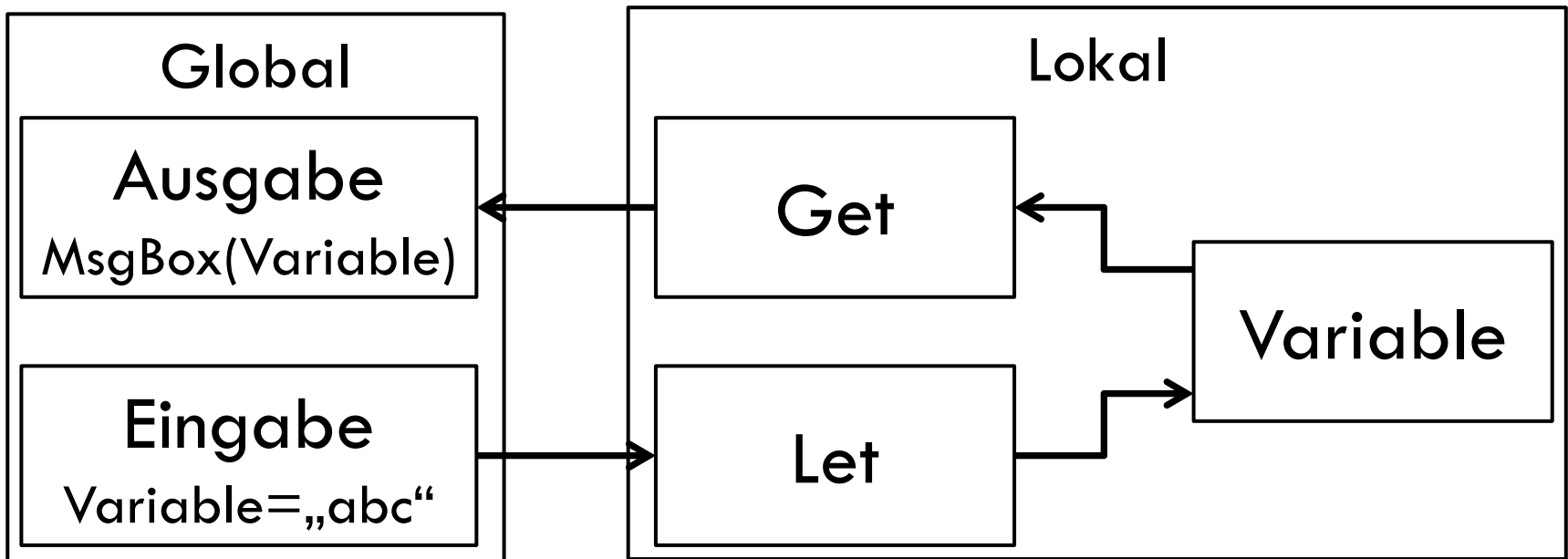
- Wozu?
 - Kapselung → Zugriff nur auf offizielle Schnittstellen
 - Gerade in der Objektorientierung wichtig

Property



21

- Variablen über get, let indirekt ansteuern
- Kapslung der Variablen
- Prüfung/Veränderung der Manipulation



Hilfsmittel „With“



22

□ 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 ○○

23

□ Grundstrukturen Objektorientierung

- Kapselung

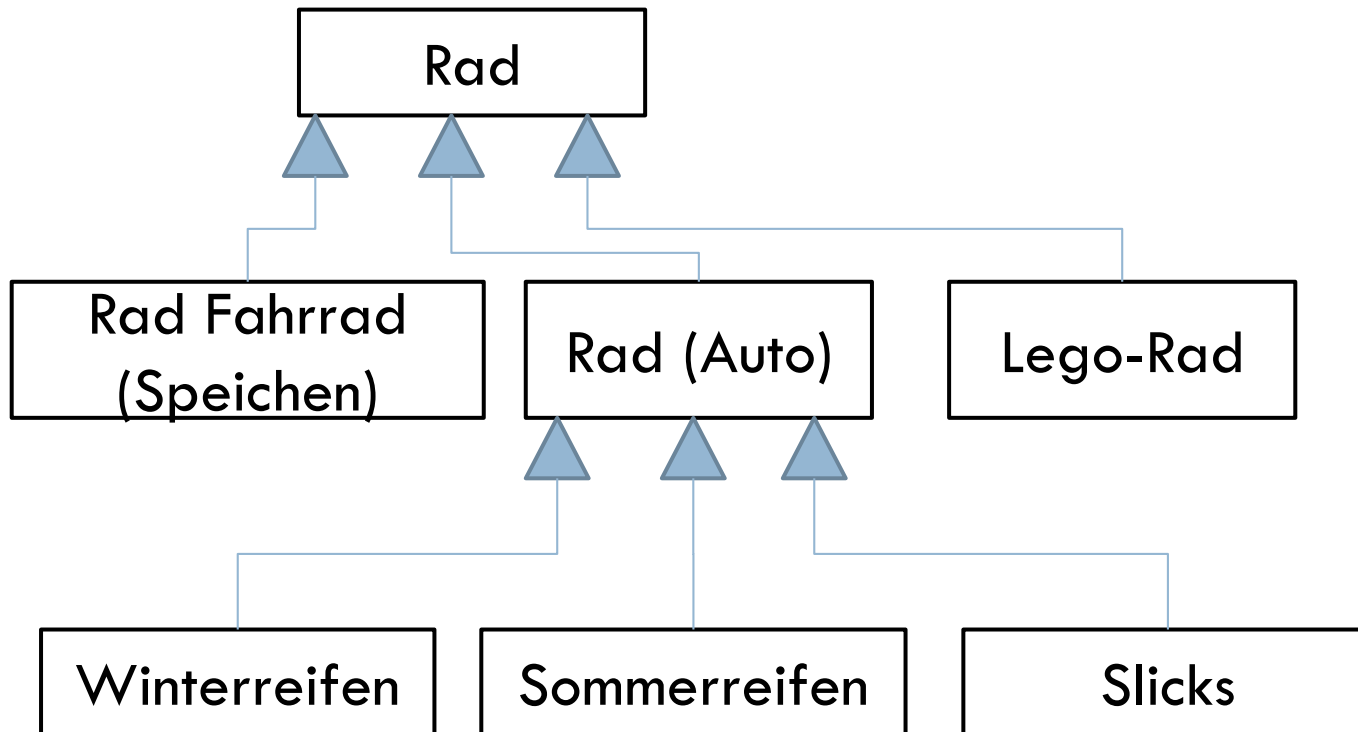
- Vererbung ← in VBA nicht möglich

- Polymorphie ← in VBA nicht möglich

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

Vererbung / Polymorphie

24



Erben:
Kinder erben
von Eltern

Polymorphie:
Kindern können
Eltern vertreten

→ Unterschied zwischen VBA und VB