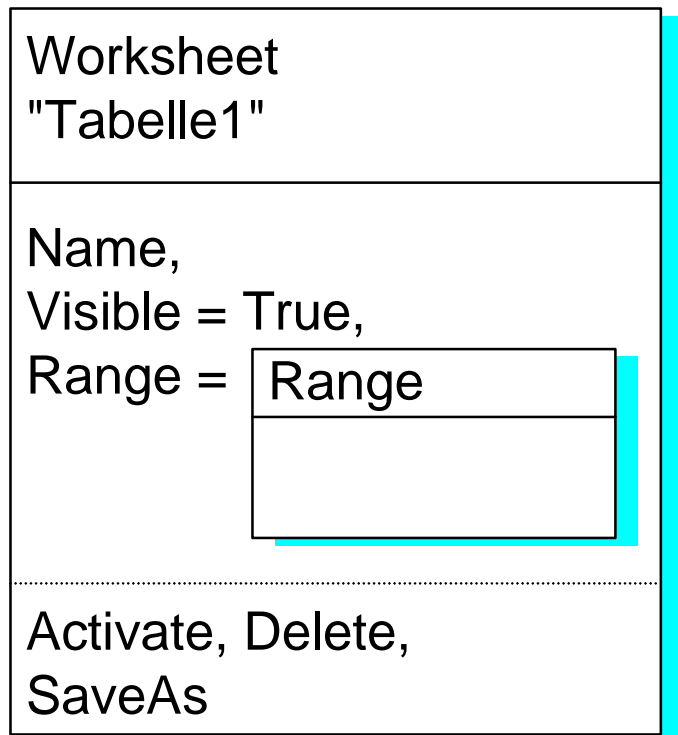


# Objekt und Auflistung

## Objekt

Beispiele:

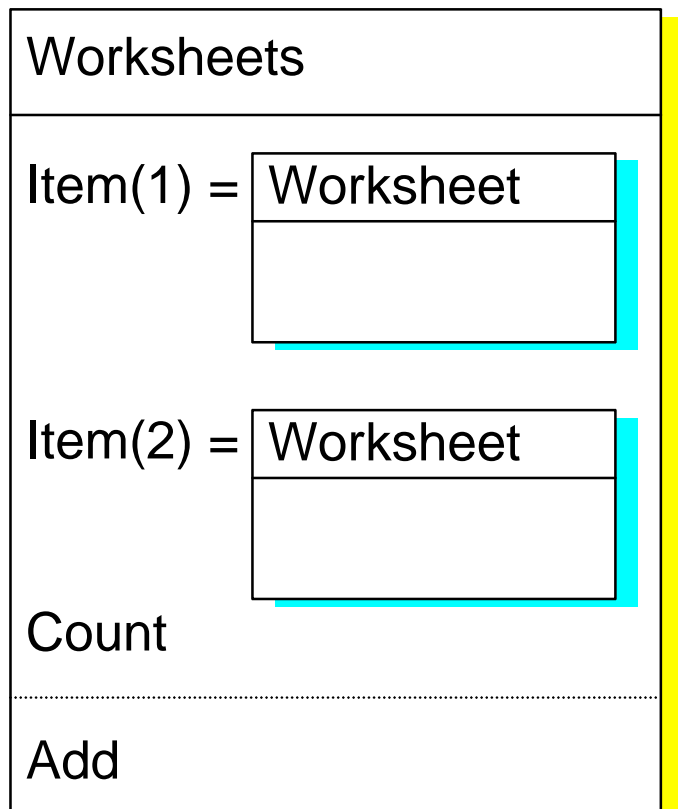
- Application
- Range



## Auflistung und Objekt

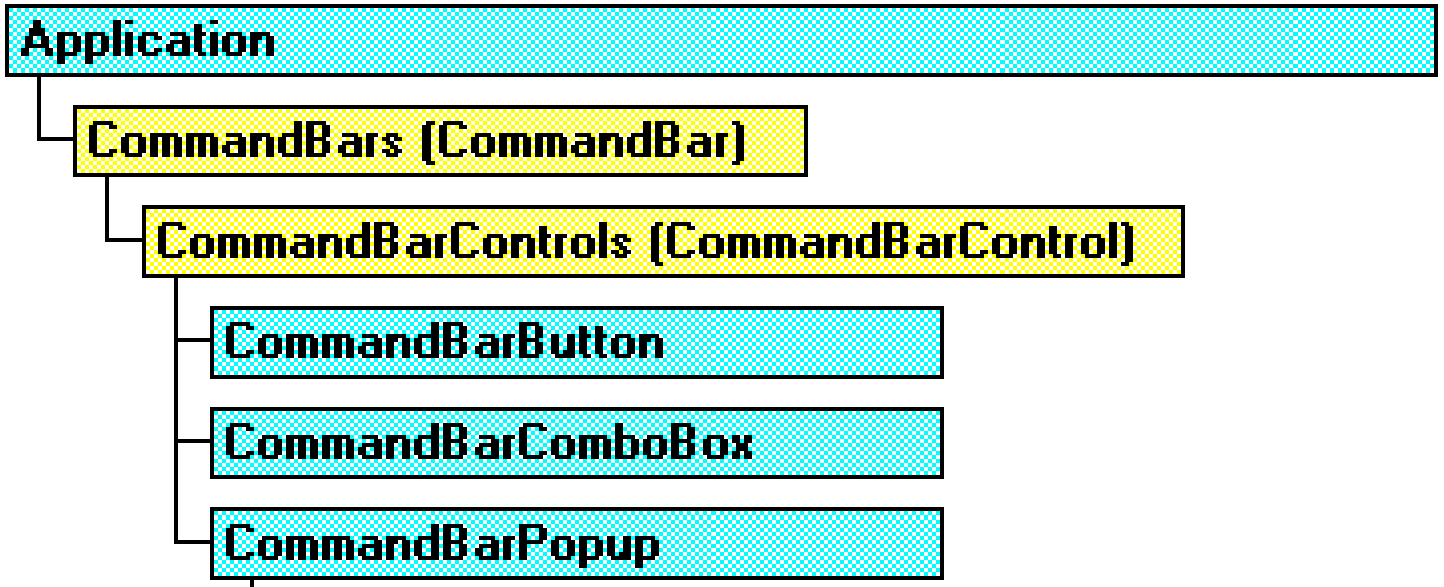
Beispiele:

- Worksheets
- Workbooks
- CommandBars
- CommandBarControls



# Objekt und Objektmodell

## Objektmodell



## Zugriff auf einzelnes Objekt

⇒ Punktoperator

## Zugriff auf alle Objekte

⇒ For Each ... Next-Schleife

```
Dim BefehlsLeiste As CommandBar
Dim BefehlsFläche As CommandBarControl
For Each BefehlsLeiste In CommandBars
    ...
    For Each BefehlsFläche In befehlsLeiste.Controls
        ...
    Next BefehlsFläche
    ...
Next BefehlsLeiste
```

# Objektvariable

\ **Tabelle1** / Tabelle2 / Zwischenresultate / Tabelle3 /

**Fangfrage:** Welche Variable enthält das Objekt  
"Zwischenresultate" am Ende des folgenden Programms?

```
Sub Test()  
    Dim Tab1 As Worksheet  
    Dim Tab2 As Worksheet  
  
    For Each Tab1 In Worksheets  
        If Tab1.Name = "Zwischenresultate" Then  
            Exit For  
        End If  
    Next Tab1  
  
    For Each Tab2 In Worksheets  
        If Tab2.Name = "Zwischenresultate" Then  
            Exit For  
        End If  
    Next Tab2  
  
End Sub
```

' (a)

' (b)

' (c)

# Objektvariablen und Verweise

Eine **Objektvariable** ist eine Variable, die einen Verweis (Zeiger) auf ein Objekt speichern kann.

Ein **Verweis** entspricht der Anfangsadresse eines Speicherbereiches, in dem sich ein Objekt befindet.

⇒ speichert *nicht* das Objekt selbst!

⇒ verweist nach der Vereinbarung auf **“Nothing”**

## Beispiel

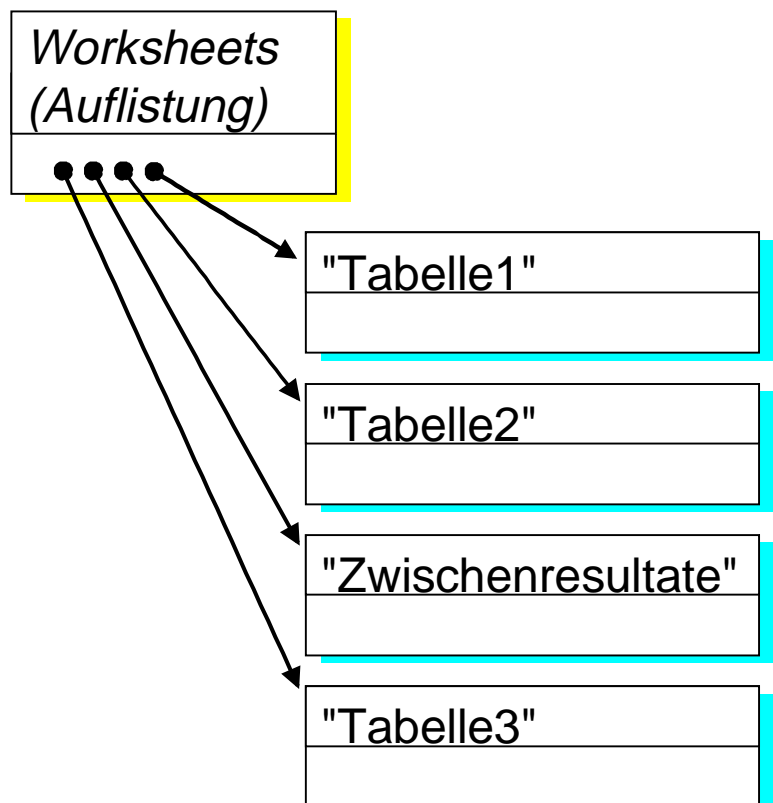
Objekte, Objektvariable und Verweise im vergangenen Beispiel

Objektvariablen:

Tab1  $\perp$

Tab2  $\perp$

Objekte:



# Variablenvereinbarung

Objektvariable werden wie 'normale' Variable mit den Schlüsselwörtern **Dim**, **Private** oder **Public** vereinbart.

Zulässige Datentypen für Objektvariablen sind *spezifische* Objekttypen oder der *allgemeine* Objekttyp **Object**.

## Beispiele

<b>Dim</b> Blatt <b>As</b> Worksheet	<i>'spezifisch; Prozedurebene</i>
<b>Dim</b> Blatt <b>As</b> Object	<i>'allgemein; Prozedurebene</i>
<b>Public</b> Blatt <b>As</b> Worksheet	<i>'spezifisch; Projektebene</i>
<b>Private</b> Blatt <b>As</b> Object	<i>'allgemein; Modulebene</i>

# Variablenzuweisung

Die Zuweisung an eine Objektvariable muss immer mit dem Schlüsselwort **Set** eingeleitet werden.

## Beispiel

	A	B	
1			
2			

```
Dim Zelle1 As Range  
Dim Zelle2 As Range
```

```
Set Zelle1 = Worksheets(1).Cells(1, 1)  
Set Zelle2 = Worksheets(1).Cells(1, 2)  
Zelle1.Value = "Text1"  
Zelle2.Value = "Text2"
```

```
Set Zelle2 = Zelle1  
Zelle1.Value = "Text3"  
Zelle2.Value = "Text4"
```

### Standardeigenschaften:

```
Zelle2 = Zelle1  
entspricht
```

```
Zelle2.Value = Zelle1.Value
```

# Argumente und Funktionswerte

## Funktionswerte (Zwischenresultate II.xls)

Wir lagern den Programmcode, der das Tabellenblatt "Zwischenresultate" sucht, in eine Prozedur aus.

```
Function sucheBlatt(TabName As String) As Worksheet
    Dim Blatt As Worksheet
    For Each Blatt In Worksheets
        If Blatt.Name = TabName Then
            Set sucheBlatt = Blatt
        End If
    Next Blatt
End Function
```

Prozeduraufruf

```
Dim ZRBlatt As Worksheet
Set ZRBlatt = sucheBlatt("Zwischenresultate")
```

## Argumente (vertausche.xls)

```
Sub vertauscheBlätter(Blatt1 As Worksheet, _
                    Blatt2 As Worksheet)
    ...
```

**End Sub**

Prozeduraufruf

```
Dim Tabelle1 As Worksheet
Dim Tabelle2 As Worksheet
Set Tabelle1 = Worksheets("Tab5")
Set Tabelle2 = Worksheets("Tab2")
vertauscheBlätter Tabelle1, Tabelle2
```

ohne Set

# Vergleichsoperator

Objektvariablen können mit dem Operator **Is** auf Gleichheit geprüft werden (*nicht* mit dem Operator =).

Jede Objektvariable kann mit **Nothing** verglichen werden.

## Beispiel (Zwischenresultate II.xls)

**Makro:** Erstelle eine Arbeitsblatt "Zwischenresultate", **falls** es noch nicht existiert. **Sonst** lösche es.

```
Sub Tabellenblatt_Klick()  
Dim ZRBlatt As Worksheet  
    Set ZRBlatt = sucheBlatt("Zwischenresultate")  
    If ZRBlatt Is Nothing Then  
        Worksheets.Add.Name = "Zwischenresultate"  
    Else  
        ZRBlatt.Delete  
    End If  
End Sub
```

Achtung bei Standardeigenschaften!

```
If Zelle1 = Zelle2 Then  
    entspricht nicht  
If Zelle1 Is Zelle2 Then
```



---

# Objekte erstellen

---

## vier Möglichkeiten, Objekte zu erstellen

- 1) Viele Objekte werden beim Start der Anwendung (z.B. Excel) erstellt
  - ✓ Application-Object
  - ✓ Workbooks-Auflistung (mit einem Workbook-Objekt)
  - ✓ CommandBars-Auflistung (mit CommandBar-Objekten)usw.
  
- 2) Die **Add**-Methode einer Auflistung erzeugt neue Objekte und fügt sie zur Auflistung hinzu.
  - ✓ Worksheets.**Add**.Name = "Zwischenresultate"usw.
  
- 3) Mit dem **New**-Schlüsselwort können benutzerdefinierte Objekte erstellt werden.  
⇒ **Set** VektorA = **New** clsVektor
  
- 4) Die **CreateObject**-Funktion erzeugt (ActiveX-) Objekte fremder Anwendungen (z.B. Access) und gibt Verweise darauf zurück.  
**Set** AccessApp = **CreateObject**("Access.Application")

# Telefonverzeichnis.xls

## Aufgabe

- Eingabe von Namen und Telefonnummern
- Verwalten der Einträge in einem Verzeichnis
- Erstellen einer alphabetischen Liste aller Einträge

## Benutzerschnittstelle

### Eingabebereiche

- Range("name")
- Range("telefon")

name	telefon
Heidi	43 43 03
Hinzufügen	Suchen
Liste	

### Verarbeitung

- hinzufügen\_Klick()
- suchen\_Klick()
- auflisten\_Klick()

anzahl	4
liste_name	liste_telefon
Gerald	74 12 91
Heidi	43 43 03
Peter	58 92 29
Sabine	78 98 12

### Ausgabebereiche

- Range("anzahl")
- Range("telefon")
- Range("liste\_name")
- Range("liste\_telefon")

# Abstraktion

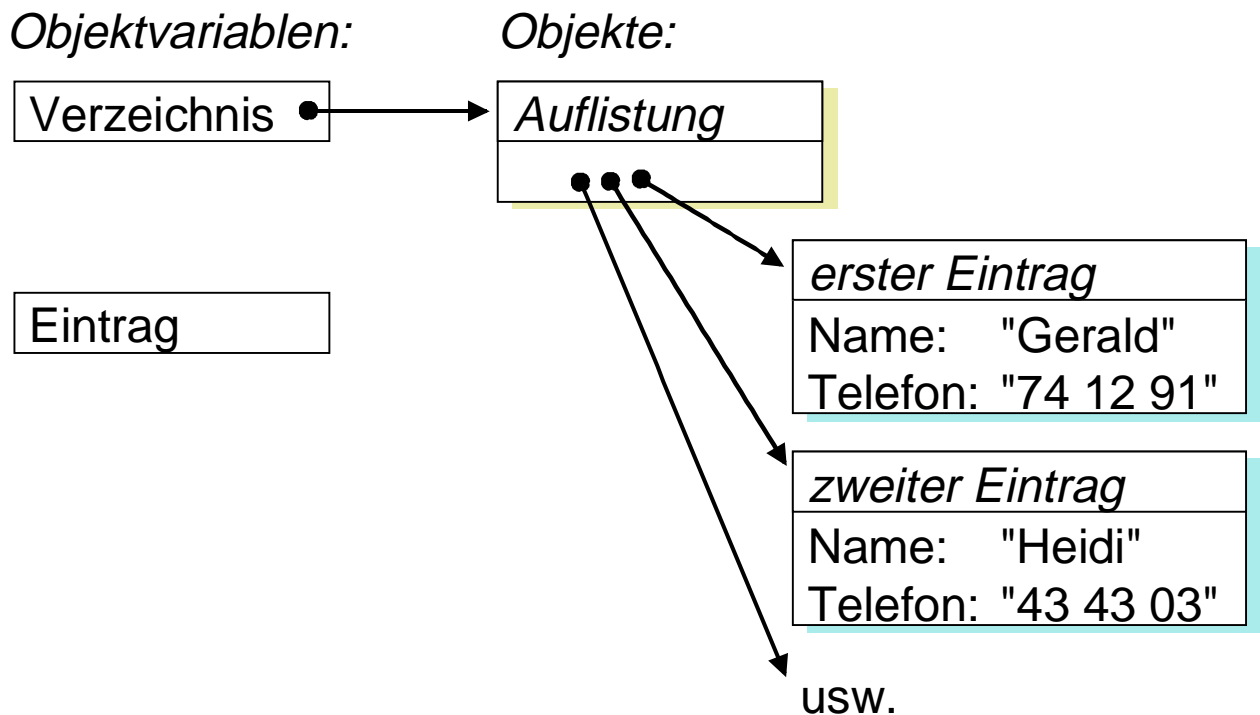
Ein **Objektmodell** beschreibt die Objekte einer Anwendung und ihre Beziehungen untereinander.

Ein **Objekt** ist eine Abstraktion eines Gegenstandes oder eines Konzeptes.

Als **Abstraktion** bezeichnet man die selektive Auswahl bestimmter Aspekte eines Gegenstandes oder eines Problems, mit dem Ziel, wichtige Aspekte zu isolieren und unwichtige zu ignorieren.

⇒ Karteikasten mit Karten

# Objektmodell



## Wir brauchen ...

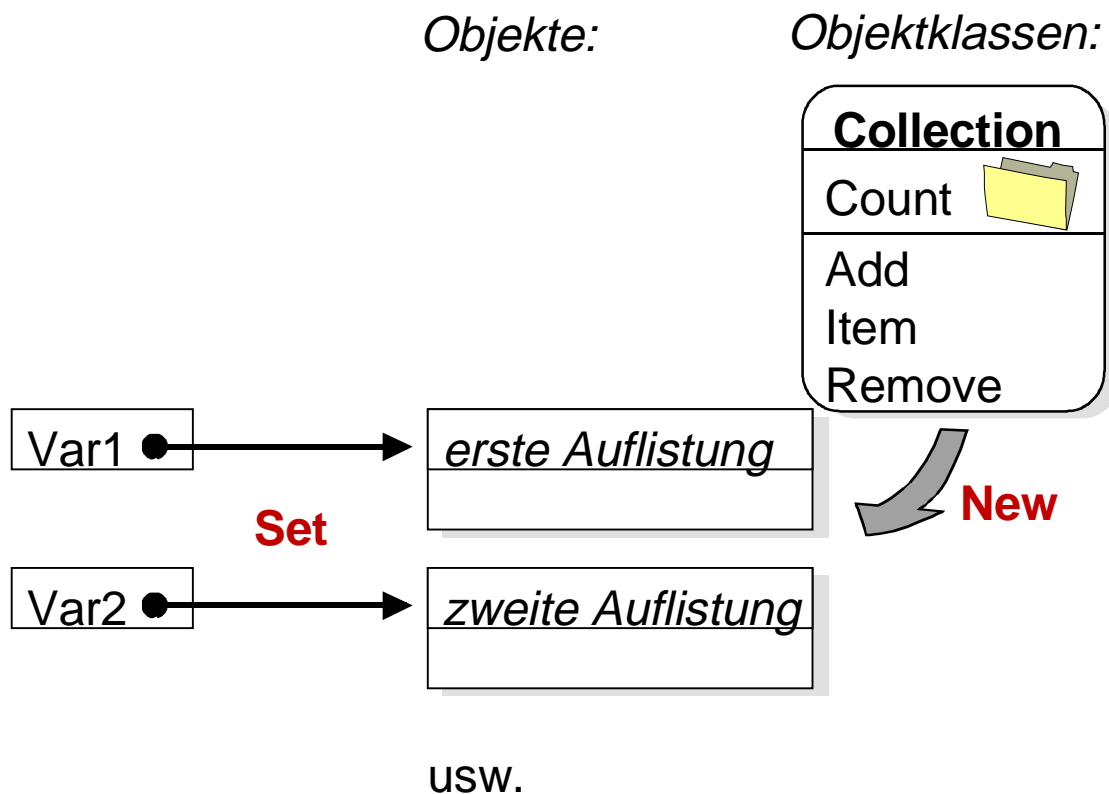
- ⇒ mehrere 'Eintrag'-**Objekte**, welche die Karten des Karteikartensystems repräsentieren
- ⇒ ein 'Auflistung'-**Objekt**, welches den Karteikasten repräsentiert
- ⇒ eine **Objektvariable** 'Eintrag', um auf die 'Eintrag'-**Objekte** zugreifen zu können.
- ⇒ eine **Objektvariable** 'Verzeichnis', um auf das 'Auflistung'-**Objekt** zugreifen zu können
- ⇒ einen **Algorithmus**, der die Einträge im Verzeichnis alphabetisch **sortiert**.

# Objektklasse und Objekt

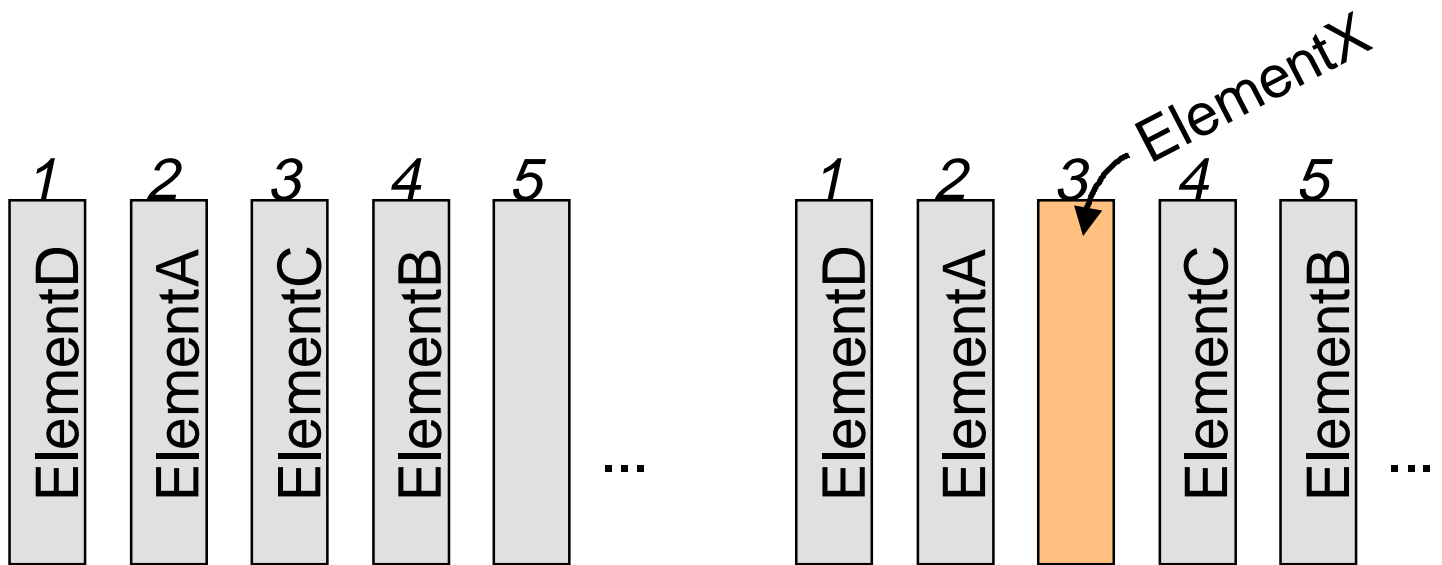
Eine **Objektklasse** ist eine Beschreibung einer Gruppe von Objekten mit gleichen Eigenschaften und gleichem Verhalten. Den *Zustand* beschreibt man durch eine Menge von Daten, das *Verhalten* durch eine Menge von Prozeduren.

## Beispiel

Visual Basic stellt die Objektklasse **Collection** für Auflistungsobjekte zur Verfügung



# Objektklasse “Collection”



Ein **Collection**-Objekt ist

- eine *geordnete* Folge  
(→Vorgänger und Nachfolger)
- von *Elementen*, auf die  
(→beliebige Datentypen, auch gemischt)
- als Einheit Bezug genommen werden kann.

## Eigenschaften

- **Objekt.Count** (schreibgeschützt)

## Methoden

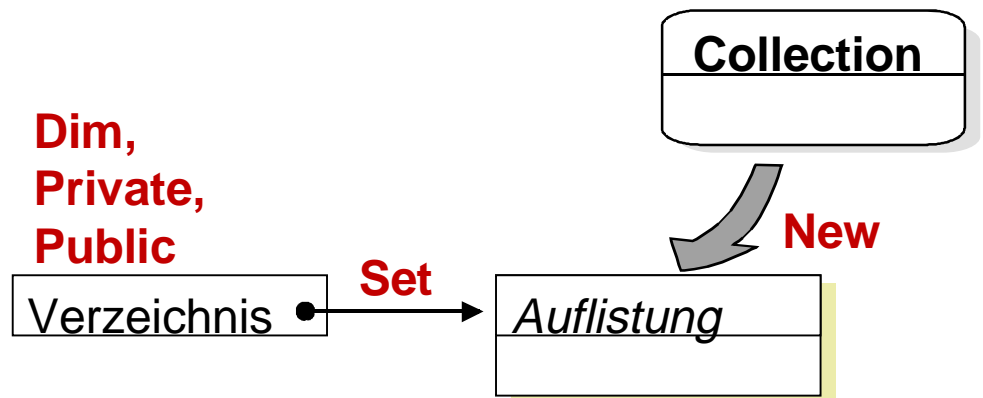
- **Objekt.Add** (item, key, before, after) (benannte Argumente)
- **Objekt.Item**(Index)
- **Objekt.Remove** Index

# Neue Objekte erstellen

Ein einzelnes Objekt, das man aufgrund einer Objektklasse erstellt, nennt man **Instanz** der Klasse.

Die **New**-Anweisung erzeugt eine Instanz einer Objektklasse

## drei Schritte



- 1) Objektvariable vereinbaren
- 2) Objekt erstellen und der Variablen zuweisen
- 3) über Variable auf Methoden und Eigenschaften zugreifen

## Beispiel Telefonverzeichnis.xls

- 1) Im Vereinbarungsabschnitt

```
Private Verzeichnis As Collection
```

- 2) Innerhalb einer Prozedur

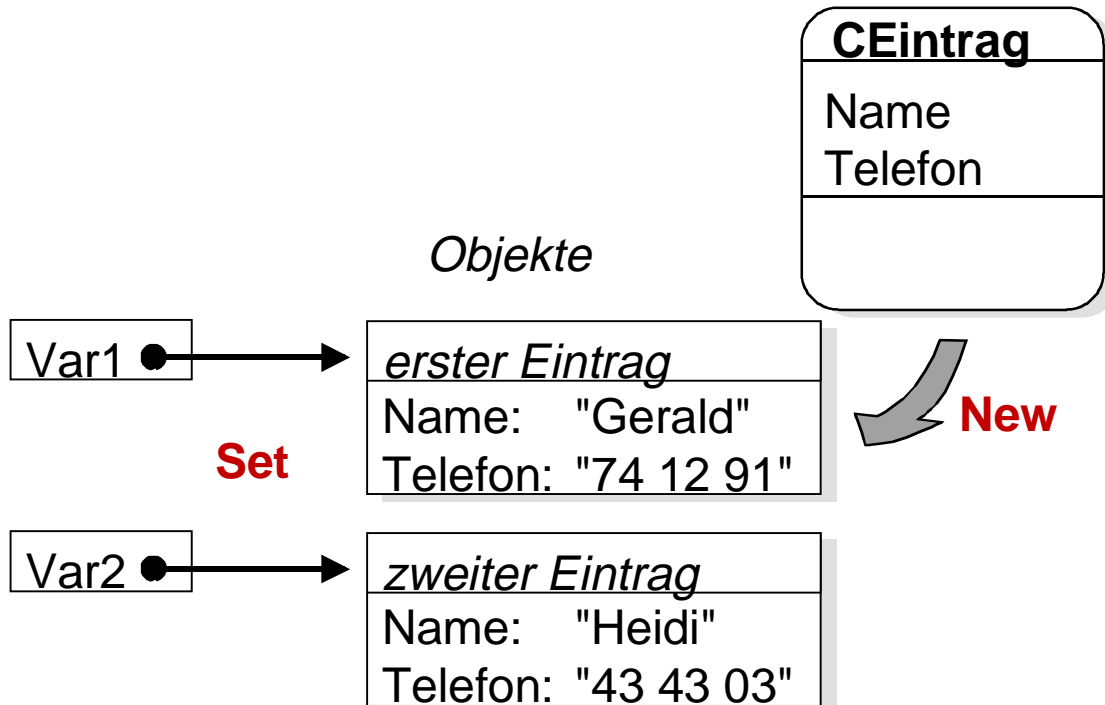
```
Set Verzeichnis = New Collection
```

- 3) Innerhalb einer Prozedur

```
z.B.: Range ("anzahl") = Verzeichnis.Count
```

# Benutzerdefinierte Objektklasse

## Beispiel: Objektklasse "CEintrag"



Ein **Klassenmodul** ist ein Modul, das die Definition einer Objektklasse enthält, einschliesslich der Definitionen seiner Eigenschaft und Methoden.



# Benutzerdefinierte Objektklasse “CEintrag”

- 1) neues Klassenmodul erstellen
- 2) Klassenmodul benennen



- 3) Eigenschaften und Methoden implementieren

**Eigenschaften** sind im wesentlichen öffentliche Variablen,  
**Methoden** sind öffentliche Prozeduren.

## Option Explicit

```
' --- EIGENSCHAFTEN
```

```
Public Name As String
```

```
Public Telefon As String
```

```
' --- METHODEN
```

---

# Neue Objekte erstellen

---

## wieder drei Schritte

1) Objektvariable deklarieren

```
Dim neuerEintrag As CEintrag
```

2) Objekt erstellen und der Variablen zuweisen

```
Set neuerEintrag = New CEintrag
```

3) über Variable auf (Methoden und) Eigenschaften zugreifen

```
neuerEintrag.Name = Range("name")
```

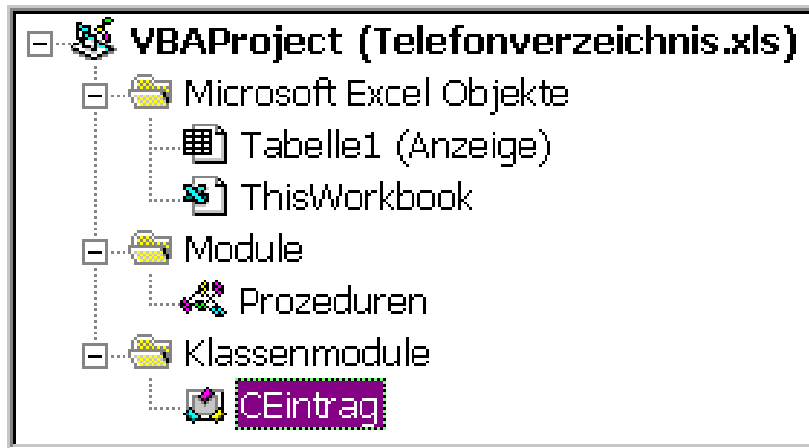
```
neuerEintrag.Telefon = Range("telefon")
```

neues Objekt in Auflistung verwalten

```
Verzeichnis.Add neuerEintrag
```

# Implementation des Telefonverzeichnisses

## Projekt

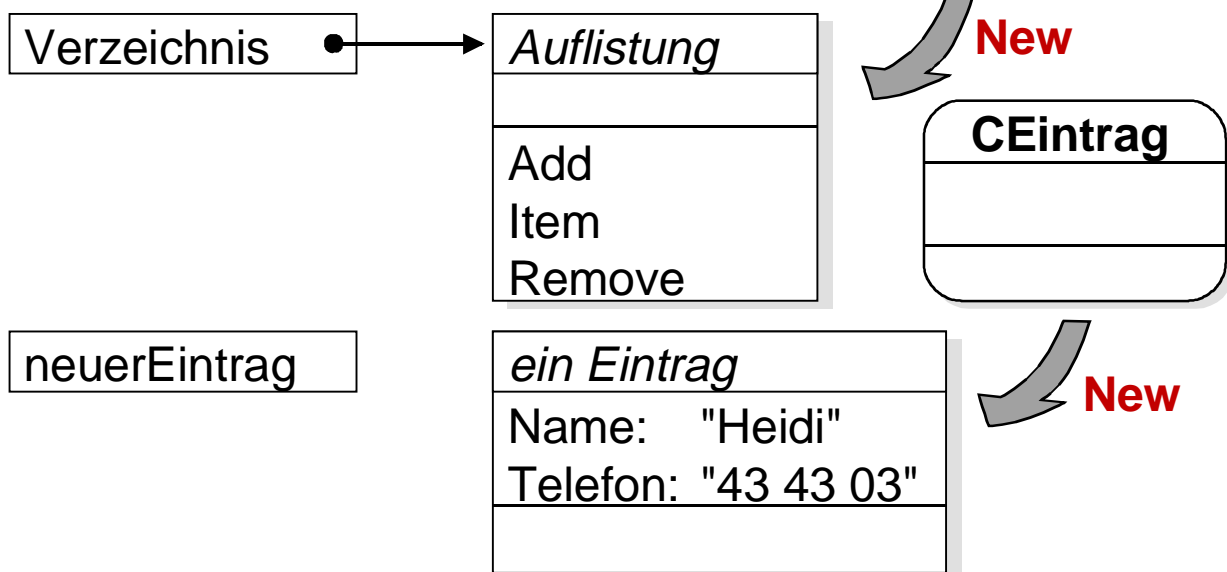


## Ablauf

*Objektvariablen:*

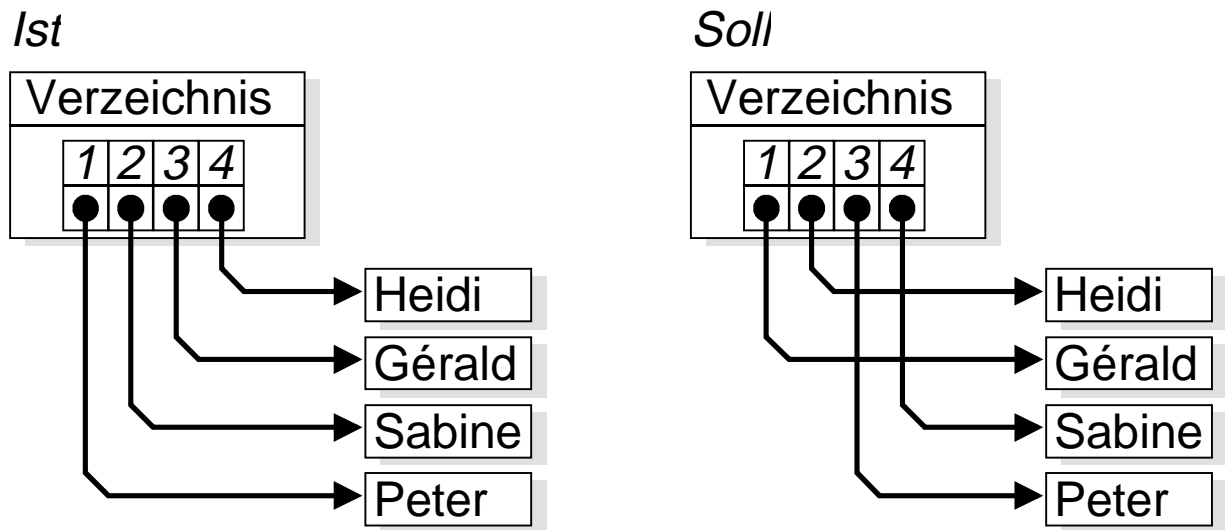
*Objekte:*

*Objektklassen:*



# Sortieralgorithmus

## Beispiel



## Sortieralgorithmus

### sortiere durch Auswählen

erstelle neues Verzeichnis (ohne Einträge)

SOLANGE noch Einträge im alten Verzeichnis

**suche kleinsten Eintrag** im alten Verzeichnis

füge Eintrag am Ende des neuen Verzeichnis ein

entferne Eintrag aus altem Verzeichnis

ersetze altes Verzeichnis durch neues Verzeichnis.

**suche kleinsten Eintrag** im Verzeichnis

setze kleinster Eintrag = erster Eintrag

FÜR ALLE Einträge im Verzeichnis

FALLS aktueller Eintrag < kleinster Eintrag

setze kleinster Eintrag = aktueller Eintrag.