BAUINFORMATIK

SS 2013 Vorlesung II

Johannes Lange

Allgemeines

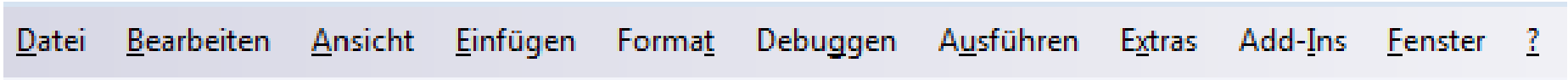
- Allgemeine Punkte
  - Anpassung der Termine im Skript
- Gibt es Fragen?
  - Heute ausführliche Wiederholung!

VBA (Visual Basic for Applications)

- Was machen wir?
  - VBA Entwicklungsumgebung (IDE)
  - Wiederholung Grundlagen
  - Grundlegende Sprachelemente von VBA
    - Dialoge und Dialogboxen
    - Textfunktionen
    - Zeiten, Dateien und Verzeichnisse
  - Spezielle Funktionen in Office (Excel)

VBA Editor

- Rundgang durch den VBA Editor



Programmelemente

Dokumentation

- Kommentare

Datenspeicher

- Variablen, Konstanten
- Felder

Programmstruktur und –ablauf

- Sub und Funktion
- Verzweigung
- Schleife

Bearbeitung und Prüfung

- Operatoren
- (Prüffunktionen)

Gui

- (Steuerelemente)

Weiter…

Kommentare

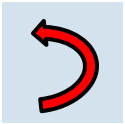
- Werden im Code nicht ausgeführt
- Verständnis
  - für andere (Teamwork)
  - für in 3 Monaten…
- Kommentare beginnen mit Hochkomma

```
' ****************************************
'  Berechnungsprogramm Einfeldträger
'  (C) Vorlesung Bauinformatik
' ****************************************
```

- Allgemeine Infos (über Makro, Funktion …)
- Spezielle Infos (Algorithmen, Änderungen, …)

→ **Kommentare 1/3 des Gesamt-Code**

Strukturierung des Codes



- Kommentare
- Einrücken
- Zerlegung in Funktionen und Makros
- Nomenklatur von Variablen und Funktionsnamen
- Gültigkeiten reduzieren
- Keine Sprünge „goto“

- Highlighting

Variablen & Konstanten

- Variablen speichern Werte und Texte

```
Dim i As Integer     ' Deklaration [1]
i = 1                ' Zuweisung   [2]
```

- Regeln
  - Erste Zeichen ein Buchstabe
  - Leerzeichen und Sonderzeichen nicht erlaubt
  - Reservierte Namen nicht erlaubt
- Variablentypen
  - VBA ist typfrei verwendbar
  - Wichtigsten Typen:

    byte, boolean, integer, long, double, string

Variablen & Konstanten

Excel (lokalglobal)

- Gültigkeit:
  - Global vor dem Sub (für alle Makros)
  - Lokal nach dem Sub (nur für dieses Makro)

- Konstanten
  - Feststehende Werte

```
'Relative Position: hier konstant festgelegt
Const relPos = 13
```

Operatoren

- Operatoren sind die Rechenbefehle des Programms
  - Rechenoperatoren

    + - * / ^
  - Vergleichsoperatoren

    = > < >= <=
  - Verbindungsoperatoren

    &
  - Logikoperatoren

    And, Or…

Verzweigung

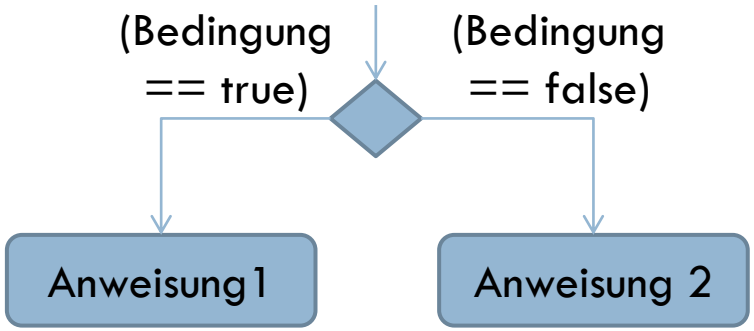
- Entscheidung zwischen zwei oder mehr Wegen

```
If (IsNumeric(loc_value)) Then
    checkInput = loc_value
Else
    MsgBox ("Überprüfen Sie bitte Ihre Eingaben.")
End If
```

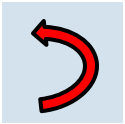
If (Bedingung) Then Anweisung1

[Else Anweisung2]

EndIf



Verzweigung



- Mehrfach-Verzweigung mit Select Case

```
diceNr = InputBox("Geben Sie eine Wüfelzahl ein:")

Select Case diceNr
Case 1
    MsgBox ("Sie haben eine Eins gewürfelt")
Case 2, 3
    MsgBox ("Sie haben eine Zwei oder Drei gewürfelt")
Case 4 To 6
    MsgBox ("Sie haben eine Vier bis Sechs gewürfelt")
Case Else
    MsgBox ("Das kenn ich nicht!")
End Select
```

Schleifen

- Schleifen wiederholen einen Programmabschnitt
- For Schleife
  - For:

    Vorgegebenen Anzahl
  - For Each :

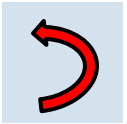
    alle Elemente einer Gruppe
  - Abbruch mit *Exit For*

```
Dim Array_Schleife(4) As Integer

'For Schleife
For i = 0 To 3
    MsgBox ("(For) i = " & i)
    Array_Schleife(i) = i * 5
Next

'For Each-Schleife
For Each i In Array_Schleife
    MsgBox ("(For Each) i= " & i)
Next
```

Schleifen

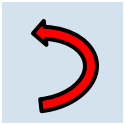


- Do While / Do Until Schleife
  - Bis zur Erfüllung einer Bedingung
  - Do While: Prüfung VOR Schleifendurchlauf
  - Do Until: Prüfung NACH Schleifendurchlauf
  - Abbruch mit *Exit Do*

```
'Do While Schleife
i = 0
Do While i < 3
    MsgBox ("(Do While) i=" & i)
    i = i + 1
Loop

'Do Until Schleife
Do Until i = 3
    MsgBox ("(Do Until) i=" & i)
    i = i + 1
Loop
```

Felder



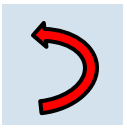
- Felder: gruppierte Variablen

```
Dim ArrayTest_1D() As Variant  'Ein-Dimensional
ReDim ArrayTest_1D(2)
ArrayTest_1D(0) = 4711
ArrayTest_1D(1) = "Wasser"
MsgBox (ArrayTest_1D(1) & " " & ArrayTest_1D(0))
ReDim ArrayTest_1D(100)

Dim ArrayTest_xD(8, 8) As Variant 'Mehr-Dimensional
ArrayTest_xD(0, 0) = "Turm"
ArrayTest_xD(1, 0) = "Springer"
```

- Ein-/mehrdimensional
- Variant für beliebige Typen
- Größe vorab festlegen, ReDim zum Verändern

Sub und Funktionen



- Sub: Makro-Grundfunktion
- Function aus Sub aufrufen
  - *Function FunctName(Aufrufparameter)*
  - *Functname = Rückgabeoparameter* (nur einen)

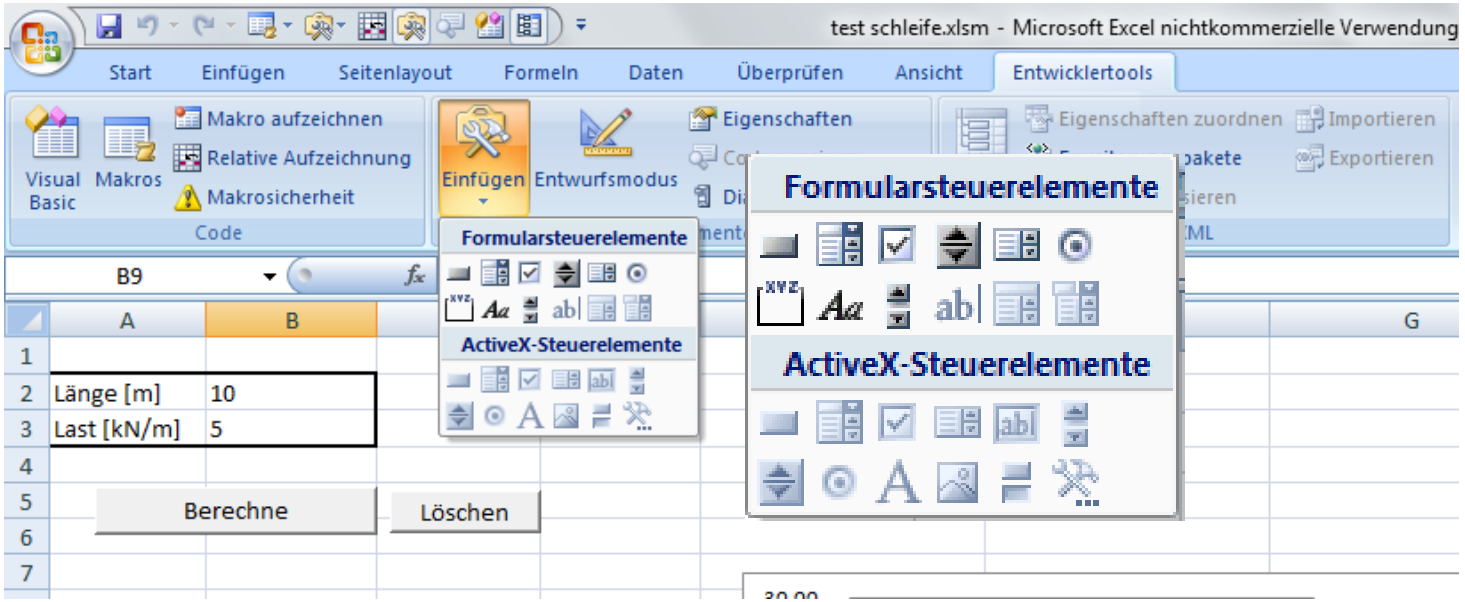
```
laenge = checkInput(Range("B2").Value)

Function checkInput(loc_value)   ' Aufruf (Aufrufparameter)
    checkInput = loc_value       ' Rückgabewert
End Function
```

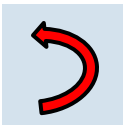
- Sub aus Sub aufrufen
  - Call SubName

Steuerelemente

- Steuerelemente
  - Buttons, Eingabefelder uvm.
  - Für interaktives Arbeiten in Office
  - Zugriff aus VBA, Action-Verweise nach VBA



Prüfung



- Variablen müssen nicht deklariert sein
- Excel-Zellen erlauben bel. Eintrag

→ Prüffunktionen

- IsNumeric(): Zahlenwerte (int, double u.ä.), -
- IsDate(): Datum in allen Formaten
- IsEmpty(): Variable leer / kein Eintrag:
- IsArrray(): Feld

Programmelemente

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

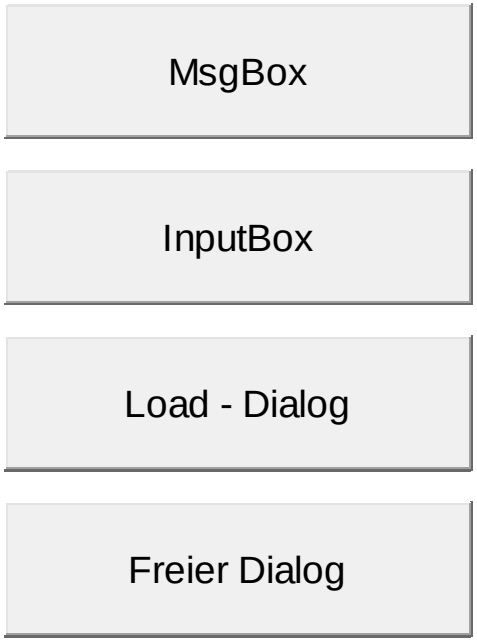
- Operatoren
- Prüffunktionen (neu)
- Textfunktionen
- Verzeichnisse und Dateien
- Zeiten

Gui

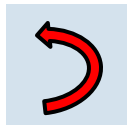
- Steuerelemente & Dialoge

Dialoge

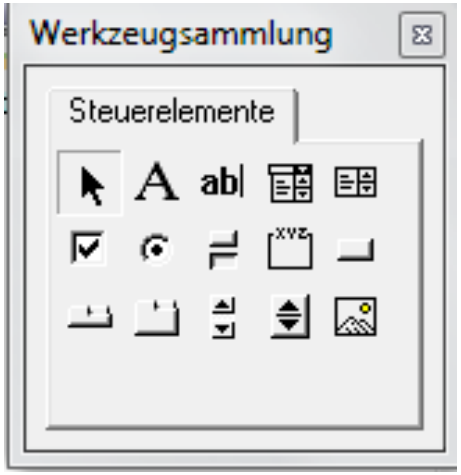
- Dialoge
  - Standard-Dialoge
    - MessageBox
    - InputBox
  - Load/Save-Dialoge
  - Eigene Dialoge mit Steuerelementen



Dialogbox erstellen



- Elemente
- Eigenschaften der Elemente



- Sprung in den Code
  - Shortcut (F7 / Shift F7)
  - Weitere Actions (z.B. MouseMove)

Prüffunktionen Erweiterung

- Inhalt = *VarType(Wert)*
  - Vorsicht:

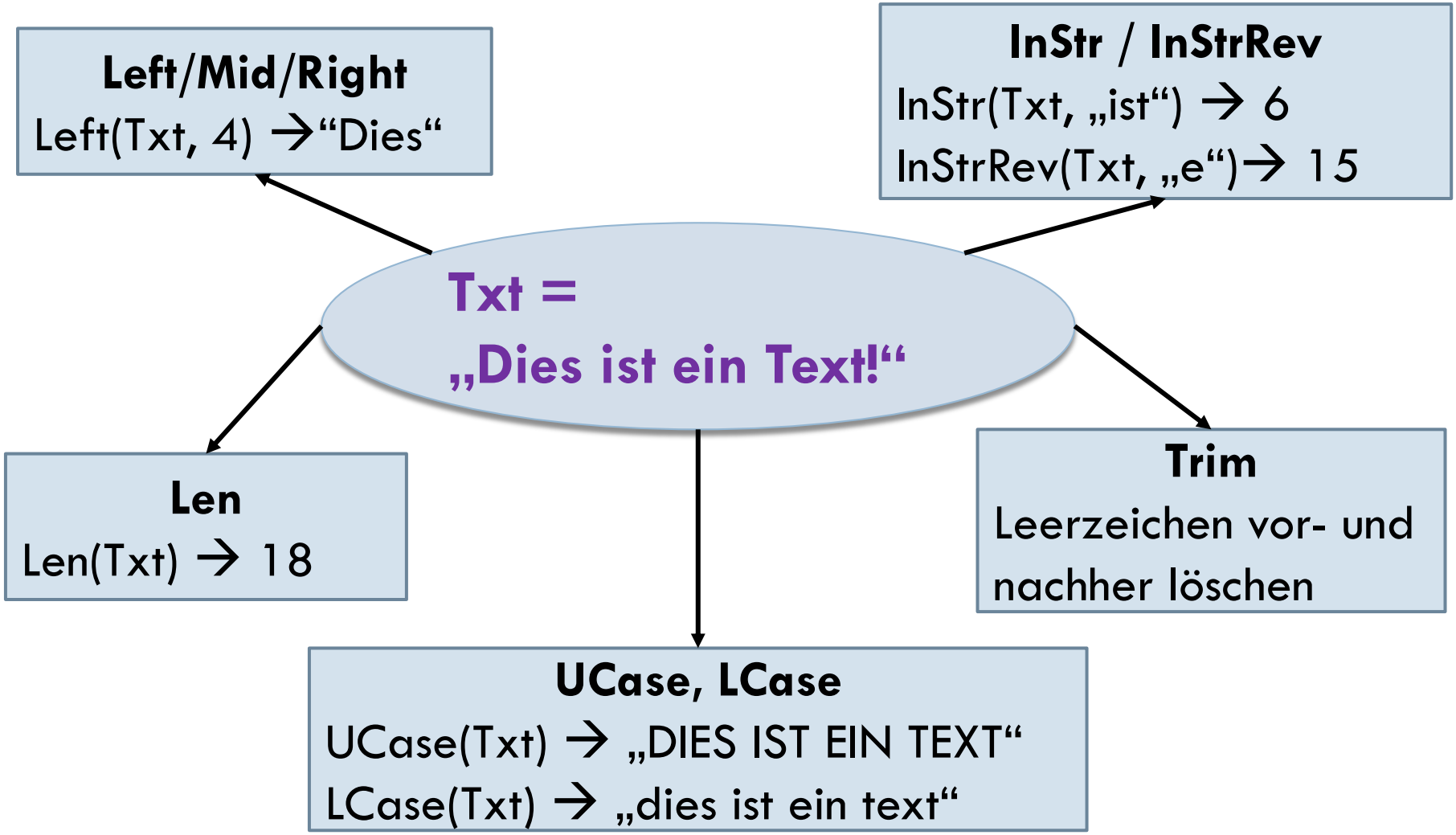
    Eingabe entspricht nicht

    immer erwarteten Wert
  - Z.B. Inputbox → String
  - Z.B. Excel Integer→ Double

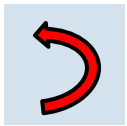
    (je nach –Zellendefinition)

Datentyp	Rückgabewert
Empty	0
Null	1
Integer	2
Long	3
Single	4
Double	5
Currency	6
Date	7
String	8
Object	9
Error	10
Boolean	11

Textfunktionen



Verzeichnisse und Dateien



Excel

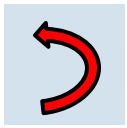
Verzeichnis

- s = Dir(Verz) ← Verzeichnis Inhalt
- MkDir(„C:\Verzeichnis“) ← Verzeichnis erstellen
- RmDir(„C:\Verzeichnis“) ← Verzeichnis löschen

Datei

- Open Datei.txt For Output As #1 ←Datei öffnen
- Print #1, “Text in Datei” ←In Datei schreiben
- Close #1 ← Datei schließen
- Kill (Datei) ← Datei löschen

Zeiten



- Dim Datum As Date ← Typ Datum
- Datum = "24.12.2013"

- Date oder Now ← Aktuelles Datum
- Day(Date) , Month(Date), Year(Date)
- Hour(Now), Minute(Now), Second(Now)

- Zeit = TimeSerial(Stunde, Minute, Sekunde)
- Application.Wait Zeit

Application, Workbook, Sheet, Range Cells

- Application — Anwendung
- Workbook — File
- Sheet — Arbeitsblatt
- Range / Cells — Felder

  - Name — Name
  - Select — Bereich markieren
  - Activate — Ansicht
  - Value — Wert abfragen
  - Cut — Ausschneiden
  - Copy — Kopieren

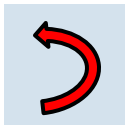
Position in Excel

- ActiveCell
  - Aktives Feld in Excel
  - Details Abfragbar über .Adress .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)



- 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

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_Dateityp()
    Dim person As PersAngaben

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

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

Klassen (OO)

Excel

- VBA ist Objektorientiert (Klassenstruktur)
- Was bedeutet Objektorientiert?
  - Daten und Funktionen im Objekt zusammenzufassen
  - Nach außen kapseln
  - Klasse ist die Form mit der Objekte/Instanzen der Klasse erzeugt werden.