ingenieur
wissenschaften
htw saar

Digital Skills für Ingenieure

6. Vorlesungstag

„Einführung in Makros“

ingenieur
wissenschaften
htw saar

Einführung in Makros

Makro – Immer wiederkehrende Schritte automatisieren

Ein Makro stellt eine vordefinierte Abfolge von Befehlen oder Tastatureingaben dar, die für spätere Verwendungen gespeichert wird.

Die LibreOffice-Makrosprache zeichnet sich durch hohe Flexibilität aus und ermöglicht die Automatisierung von sowohl einfachen als auch komplexen Aufgaben. Makros erweisen sich als besonders nützlich, wenn Sie eine Aufgabe wiederholt auf die gleiche Weise ausführen müssen oder wenn Sie eine Aufgabe mit nur einem Tastendruck erledigen möchten, obwohl sie normalerweise mehrere Schritte erfordern würde.

ingenieur
wissenschaften
htw saar

Einführung in Makros

Makro – Menüpunkte

- Ausführen von Makro:
  Extras → Makros → Makro ausführen

- Bearbeiten von Makros:
  Extras → Makros → Makros bearbeiten

- Verwalten von Makros:
  Extras → Makros → Makros verwalten → Basic

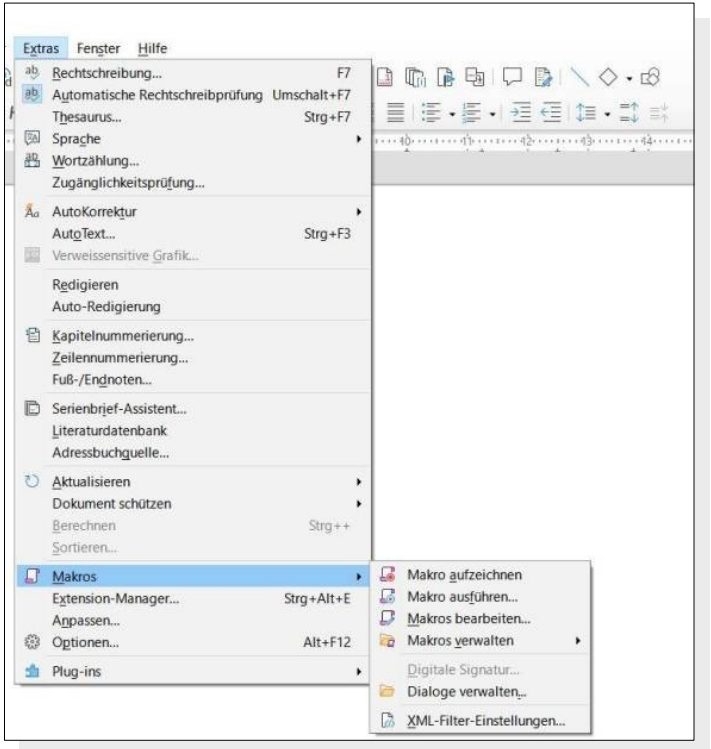


Abbildung 1: Makro Menüpunkte

Einführung in Makros

Libre Office Basic

Die Makrosprache von LibreOffice nutzt als Grundlage die Programmiersprache StarOffice Basic, die oft als "LibreOffice Basic" oder einfach "Basic" bezeichnet wird.

```
Sub Main
	print "Hello World!"
End Sub
```

Code Listing 1: hello_world.bas

ingenieur
wissenschaften
htw saar

Einführung in Makros

Libre Office Basic IDE

- Wenn Sie Ihre Makros bearbeiten bzw. verwalten möchten, steht Ihnen dazu eine von Libre Office integrierte Entwicklungsumgebung zur Verfügung.

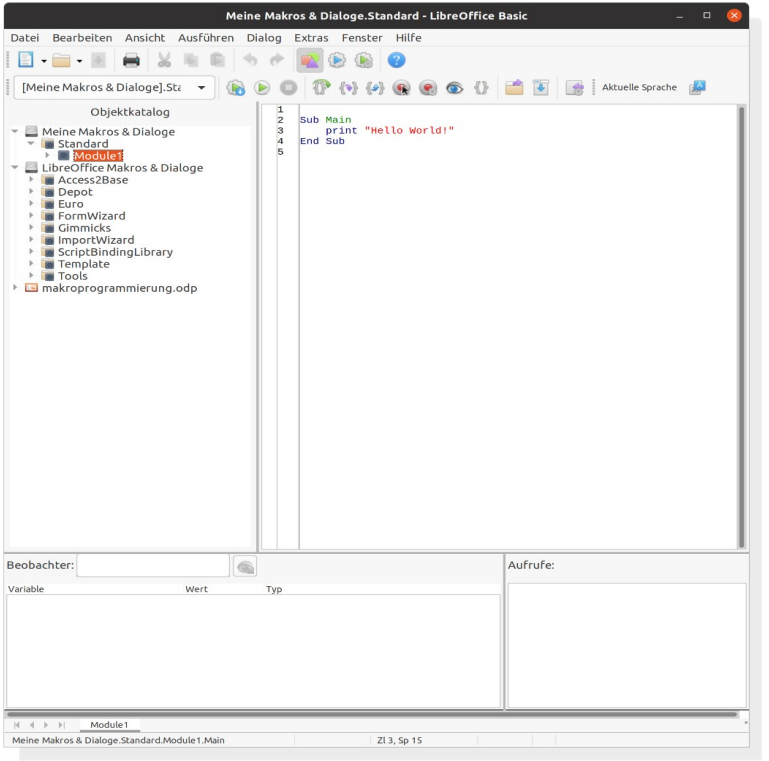


Abbildung 3: Libre Office Basic

ingenieur
wissenschaften
htw saar

Einführung in Makros

Libre Office Basic IDE

- Wenn Sie Ihre Makros bearbeiten bzw. verwalten möchten, steht Ihnen dazu eine von Libre Office integrierte Entwicklungsumgebung zur Verfügung.

- In dem oberen Teil auf der linken Seite (Objektkatalog), finden Sie Ihre selbstgeschriebenen Bibliotheken und Module.

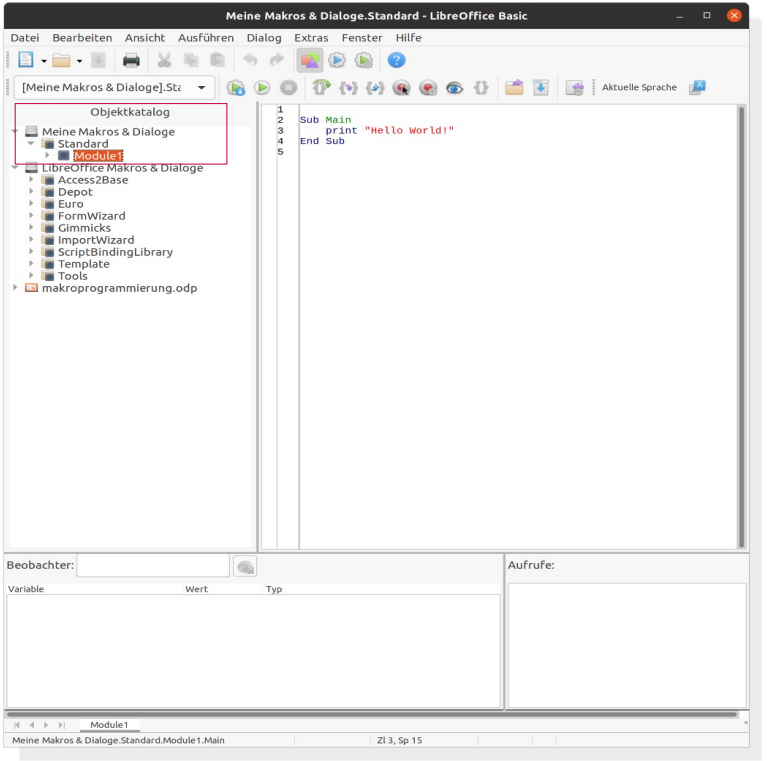


Abbildung 4: Libre Office Basic Objektkatalog – eigene Bibliotheken/Module

ingenieur
wissenschaften
htw saar

Einführung in Makros

Libre Office Basic IDE

- Wenn Sie Ihre Makros bearbeiten bzw. verwalten möchten, steht Ihnen dazu eine von Libre Office integrierte Entwicklungsumgebung zur Verfügung.
- In dem oberen Teil auf der linken Seite (Objektkatalog), finden Sie Ihre selbstgeschriebenen Bibliotheken und Module.
- In dem unteren Teil sind schon standardmäßig implementierte Bibliotheken vorhanden (z.B. Access2Base für Datenbankzugriffe)

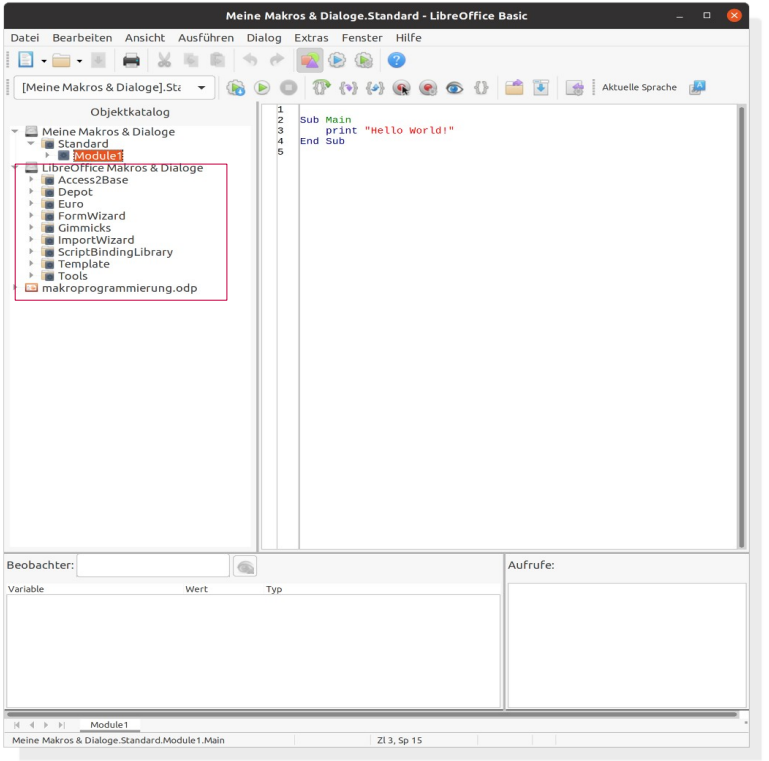


Abbildung 5: Libre Office Basic Standardbibliotheken

ingenieur
wissenschaften
htw saar

Einführung in Makros

Libre Office Basic IDE

- Auf der rechten Seite ist der Editor zu sehen, der es ermöglicht das aktuell ausgewählte Modul zu bearbeiten (in diesem Fall *Module1* mit unserer hello_world.bas)

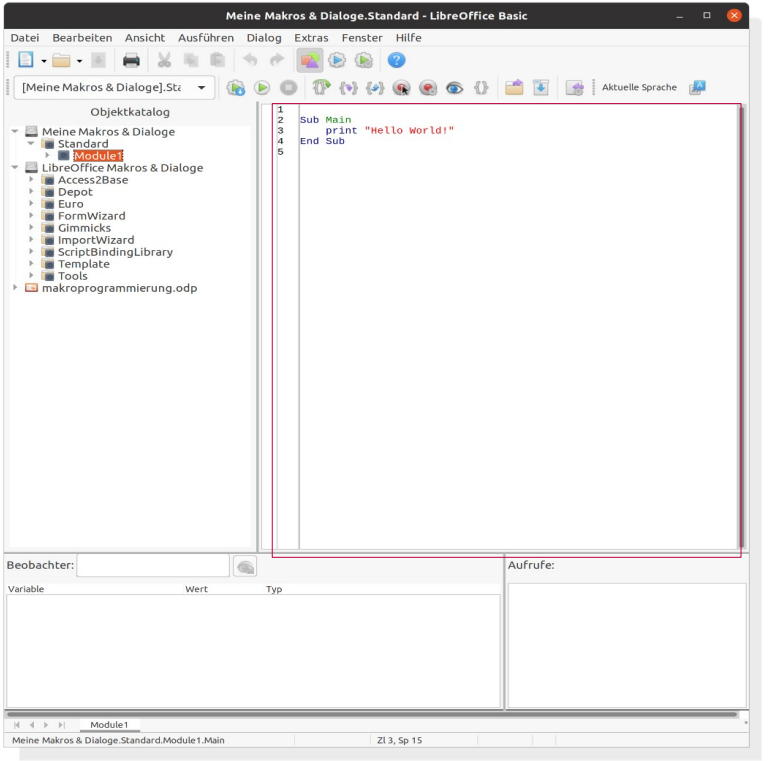


Abbildung 6: Editor für Modulbearbeitung

ingenieur
wissenschaften
htw saar

Einführung in Makros

Libre Office Basic IDE

- Auf der rechten Seite ist der Editor zu sehen, der es ermöglicht das aktuell ausgewählte Modul zu bearbeiten (in diesem Fall *Module1* mit unserer hello_world.bas)

- Mit der Play Taste können Sie das aktuell ausgewählte Modul/Makro ausführen.

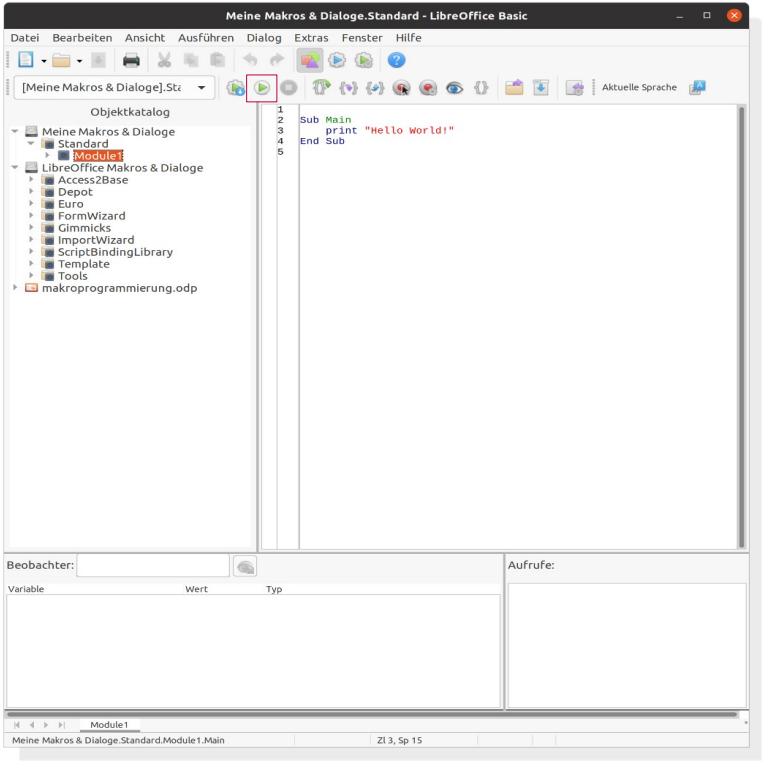


Abbildung 7: Ausführung des aktuell ausgewählten Moduls

ingenieur
wissenschaften
htw saar

Einführung in Makros

Libre Office Basic IDE

- Auf der rechten Seite ist der Editor zu sehen, der es ermöglicht das aktuell ausgewählte Modul zu bearbeiten (in diesem Fall *Module1* mit unserer hello_world.bas)
- Mit der Play Taste können Sie das aktuell ausgewählte Modul/Makro ausführen.
- Debug Button Gruppe ermöglicht es Prozedurschritte, Einzelschritte und Rücksprünge zu ermöglichen.

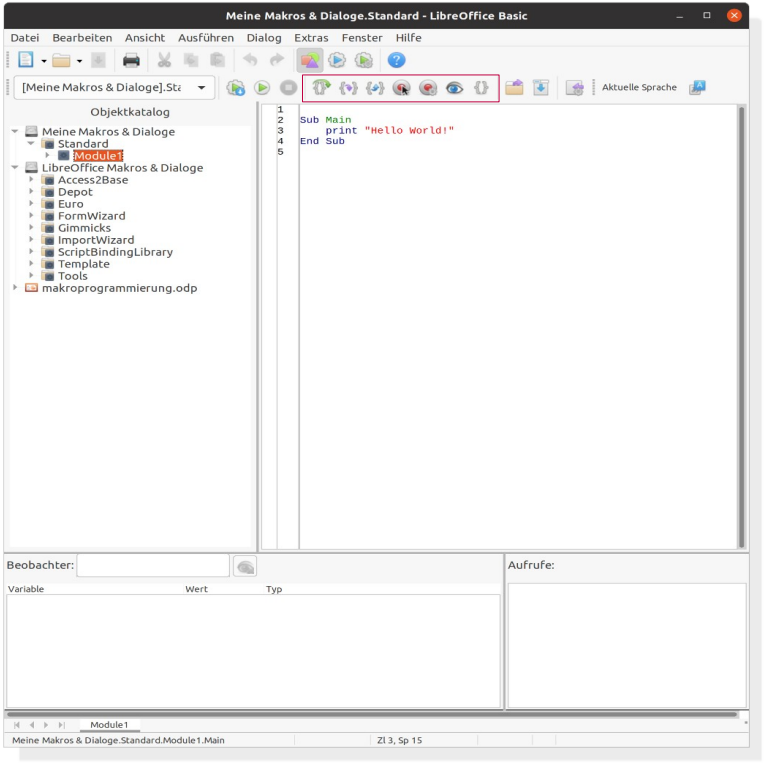


Abbildung 7: Ausführung des aktuell ausgewählten Moduls

ingenieur
wissenschaften
htw saar

Einführung in Makros

Sprachstrukturen – Variablennamen

- Offiziell darf ein Variablenname nicht mehr als 255 Zeichen enthalten.
- Das erste Zeichen eines Variablennamens muss ein Buchstabe sein: A-Z oder a-z.
- Variablennamen dürfen die Ziffern 0-9 und den Unterstrich (_) enthalten, dürfen jedoch nicht mit diesen starten. Ein Unterstrich am Ende eines Variablennamens wird nicht als Zeilenverbinder interpretiert.
- Bei Variablennamen spielt die Groß- und Kleinschreibung keine Rolle. Sowohl „AlTer“ als auch „alTER“ verweisen auf dieselbe Variable.

ingenieur wissenschaften htw saar

Einführung in Makros

Verfügbare Variablentypen und ihre Anfangswerte

Typ	*Annex*	*Anfangswert*	*Bytes*	*Konvertierung*	*Beschreibung*
Boolean		False	1	CBool	True/False
Currency	@	0.0000	8	CCur	Währung mit 4 Dezimalstellen
Date		00:00:00	8	CDate	Datum & Uhrzeit
Double	#	0.0	8	CDbl	Dezimalzahl im Bereich +/- 1,79769313486232 x 10E308
Integer	%	0	2	CInt	Ganzzahl im Bereich von -32768 bis 32767

ingenieur
wissenschaften
htw saar

Einführung in Makros

Verfügbare Variablentypen und ihre Anfangswerte

Typ	*Annex*	*Anfangswert*	*Bytes*	*Konvertierung*	*Beschreibung*
Long	&	0	4	CLng	Ganzzahl im Bereich von -2.147.483.648 bis 2.147.483.647
Object		Null	wechselt		Objekt
Single	!	0.0	4	CSng	Dezimalzahl im Bereich von -/+ 3,402823 x 10E38
String	$	“”	wechselt	CStr	Text mit knapp ~2GB
Variant		Empty	wechselt	CVar	Kann Daten jeden Typs enthalten

ingenieur
wissenschaften
htw saar

Einführung in Makros

Deklaration einfacher Variablen

Deklarierung	*Beschreibung*
Dim Name	Name hat den Typ Variant, weil kein Typ deklariert ist.
Dim Name As String	Name ist als Typ String deklariert.
Dim Name$	Name ist als Typ String deklariert, da am Ende des Bezeichners der Annex $ genutzt wurde.
Dim Name As String, Gewicht As Single	Name ist vom Typ String und Gewicht hat den Typ Single.

Einführung in Makros

Deklaration einfacher Variablen

Deklarierung	*Beschreibung*
Dim Breite, Laenge	Breite und Laenge haben beide den Typ Variant.
Dim Gewicht, Groesse als Single	Gewicht hat den Typ Variant und Groesse den Typ Single.

Einführung in Makros

Sprachstrukturen – Ungarische Notation [1]

- geht auf den ungarisch-amerikanischen Informatiker Charles Simonyi zurück.
- Hierbei setzt sich der Variablennamen aus Präfix, Datentyp und Bezeichner zusammen.
- Kern der ungarischen Notation ist es, die Aufgabe und den Typ (Apps Hungarian) bzw. nur Typ (Systems Hungarian) einer Variable (oder Methode) in deren Namen zu verdeutlichen.

Abbildung 8: Charles Simonyi

[1]: Vgl. https://de.wikipedia.org/wiki/Ungarische_Notation

Einführung in Makros

Sprachstrukturen – Ungarische Notation [1]

Beispiele für die Deklaration von Variablen in ungarischer Notation

```
Sub Deklaration
    Dim oBrutto as Object
    Dim daDatum as Date
    Dim doZahl as Double
    Dim siZahl as Single
    Dim stText as String
End Sub
```

Code Listing 2: deklaration.bas

[1]: Vgl. https://de.wikipedia.org/wiki/Ungarische_Notation

Einführung in Makros

Sprachstrukturen – Ungarische Notation [1]

Beispiele für die Deklaration von Variablen in ungarischer Notation

```
Sub Deklaration
    Dim oBrutto as Object
    Dim daDatum as Date
    Dim doZahl as Double
    Dim siZahl as Single
    Dim stText as String
End Sub
```

Code Listing 2: deklaration.bas

[1]: Vgl. https://de.wikipedia.org/wiki/Ungarische_Notation

Einführung in Makros

Notation

1.) Deklaration von Routinen Namen → UpperCamelCase

2.) Deklaration von Variablen Namen → UpperCamelCase

```
Sub DeklarationEinerRoutine
    Dim EineZahl as Double
    Dim EinText as String
End Sub
```

Code Listing 3: notation_in_vorlesung.bas

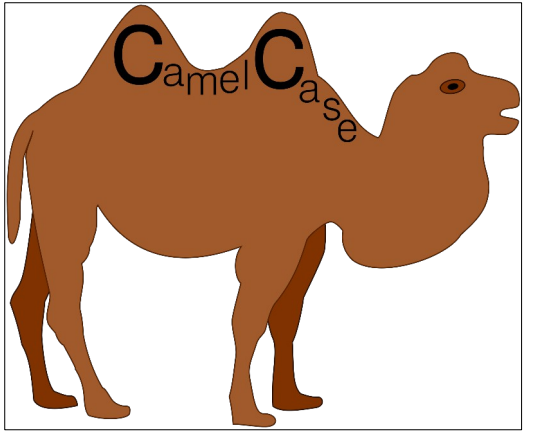


Abbildung 9: CamelCase Kamel ;-)

ingenieur
wissenschaften
htw saar

Einführung in Makros

Subroutinen

- Subroutinen sind Codezeilen, die zu sinnvollen Arbeitsabschnitten gruppiert sind.

Einführung in Makros

Subroutinen

- Subroutinen sind Codezeilen, die zu sinnvollen Arbeitsabschnitten gruppiert sind.

```
Sub EineSubroutine
    Print "Führe EineSubroutine aus"
End Sub
```

Code Listing 4: eine_subroutine.bas

Einführung in Makros

Subroutinen

- Subroutinen sind Codezeilen, die zu sinnvollen Arbeitsabschnitten gruppiert sind.

```
Sub EineSubroutine
    Print "Führe EineSubroutine aus"
End Sub
```

Code Listing 4: eine_subroutine.bas

```
Sub Main
    Call EineSubroutine
    EineSubroutine
End Sub
```

Code Listing 5: eine_subroutine_aufrufen.bas

Einführung in Makros

Funktionen

- Eine Funktion ist eine Subroutine, die einen Wert zurückgibt.
- Mit dem Schlüsselwort Function wird eine Funktion deklariert, die wie eine Variable ihren Rückgabetyp definieren kann. Ohne deklarierten Typ wird der Standardtyp Variant zurückgegeben.

Einführung in Makros

Funktionen

- Eine Funktion ist eine Subroutine, die einen Wert zurückgibt.
- Mit dem Schlüsselwort Function wird eine Funktion deklariert, die wie eine Variable ihren Rückgabetyp definieren kann. Ohne deklarierten Typ wird der Standardtyp Variant zurückgegeben.

```
Function HalloFunktion As String
    HalloFunktion = "Hallo"
End Function
```

Code Listing 6: hallo_funktion.bas

ingenieur
wissenschaften
htw saar

Einführung in Makros

Funktionen

- Eine Funktion ist eine Subroutine, die einen Wert zurückgibt.
- Mit dem Schlüsselwort Function wird eine Funktion deklariert, die wie eine Variable ihren Rückgabetyp definieren kann. Ohne deklarierten Typ wird der Standardtyp Variant zurückgegeben.

```
Function HalloFunktion As String
    HalloFunktion = "Hallo"
End Function
```

Code Listing 6: hallo_funktion.bas

```
Sub FunktionsAufruf
    Print "Die Funktion
    EineFunktion gibt "
    + HalloFunktion + "
    zurück."
End Sub
```

Code Listing 7: funktionsaufruf.bas

Einführung in Makros

Parameterübergabe

- Eine Variable, die einer Routine übergeben wird, nennt man Argument.
- Argumente müssen deklariert werden. Dann spricht man auch von Parametern.
- Für die Deklarierung von Parametern mit Namen und Typen gelten dieselben Regeln wie für Variablen.

Einführung in Makros

Parameterübergabe

- Eine Variable, die einer Routine übergeben wird, nennt man Argument.
- Argumente müssen deklariert werden. Dann spricht man auch von Parametern.
- Für die Deklarierung von Parametern mit Namen und Typen gelten dieselben Regeln wie für Variablen.

```
Sub PrintString(s1 as String)
        print s1
End Sub
```

Code Listing 8: print_string.bas

Einführung in Makros

Parameterübergabe

- Eine Variable, die einer Routine übergeben wird, nennt man Argument.
- Argumente müssen deklariert werden. Dann spricht man auch von Parametern.
- Für die Deklarierung von Parametern mit Namen und Typen gelten dieselben Regeln wie für Variablen.

```
Sub PrintString(s1 as String)
    print s1
End Sub
```

Code Listing 8: print_string.bas

```
Sub CallStringParameter
    Dim EinString as String
    EinString = "Ein String"
    PrintString(EinString)
End Sub
```

Code Listing 9: call_print_string.bas

Einführung in Makros

Parameterübergabe (Call by Reference/Call by Value)

- Standardmässig werden Argumente als Referenz übergeben und nicht als Wert.
- Anders gesagt: wenn die aufgerufene Subroutine ein Argument verändert, sieht die aufrufende Subroutine die Änderung.
- Man kann mit dem Schlüsselwort ByVal dieses Verhalten dahin abwandeln, dass eine Kopie des Arguments (statt einer Referenz auf das Argument) versendet wird.

Einführung in Makros

Parameterübergabe (Call by Reference/Call by Value)

- Zwei Additionsroutinen mit Parameterübergabe als Referenz bzw. als Value.

```
Sub AdditionByReference(a, b)
    a = a + b
End Sub
```

Code Listing 10: addition_by_reference.bas

```
Sub AdditionByVal(ByVal a, ByVal b)
    a = a + b
End Sub
```

Code Listing 11: addition_by_value.bas

ingenieur wissenschaften htw saar

Einführung in Makros

Parameterübergabe (Call by Reference/Call by Value)

- Zwei Additionsroutinen mit Parameterübergabe als Referenz bzw. als Value.

```
Sub AdditionByReference(a, b)
    a = a + b
End Sub
```

Code Listing 10: addition_by_reference.bas

```
Sub AdditionByVal(ByVal a, ByVal b)
    a = a + b
End Sub
```

Code Listing 11: addition_by_value.bas

```
Sub CallAddition
    Dim Zahl1 as Integer
    Dim Zahl2 as Integer
    Zahl1 = 2
    Zahl2 = 4
    AdditionByReference(Zahl1,
    Zahl2)
    print "Zahl1: " + Zahl1
End Sub
```

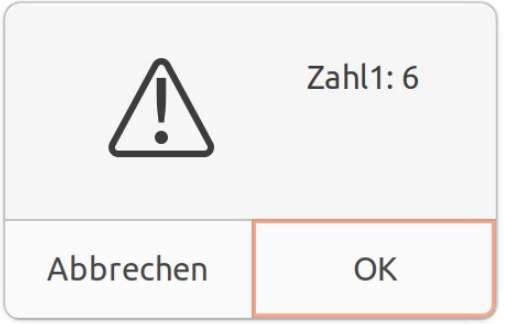
Code Listing 12: addition_aufruf_ref.bas

Einführung in Makros

Parameterübergabe (Call by Reference/Call by Value)

Einführung in Makros

Parameterübergabe (Call by Reference/Call by Value)



ingenieur wissenschaften htw saar

Einführung in Makros

Parameterübergabe (Call by Reference/Call by Value)

- Zwei Additionsroutinen mit Parameterübergabe als Referenz bzw. als Value.

```
Sub AdditionByReference(a, b)
    a = a + b
End Sub
```

Code Listing 10: addition_by_reference.bas

```
Sub AdditionByVal(ByVal a, ByVal b)
    a = a + b
End Sub
```

Code Listing 11: addition_by_value.bas

```
Sub CallAddition
    Dim Zahl1 as Integer
    Dim Zahl2 as Integer
    Zahl1 = 2
    Zahl2 = 4
    AdditionByVal(Zahl1,
    Zahl2)
    print "Zahl1: " + Zahl1
End Sub
```

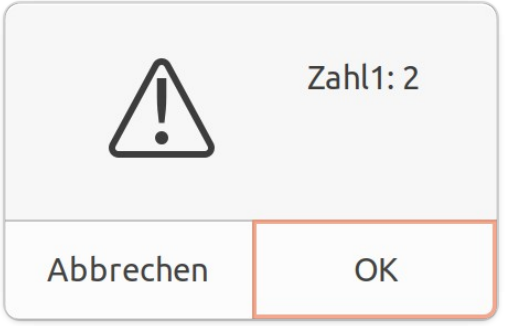
Code Listing 13: addition_aufruf_ref.bas

Einführung in Makros

Parameterübergabe (Call by Reference/Call by Value)

Einführung in Makros

Parameterübergabe (Call by Reference/Call by Value)



Literatur

[1]: Vgl. https://de.wikipedia.org/wiki/Ungarische_Notation