Mathematische Formeln in ein gegliedertes Word-Dokument einfügen und je Kapitel automatisch benummern

Tabellenbasierte und tabellenlose Methode

In diesem Beitrag werden zwei Methoden beschrieben, wie mathematische Formeln bequem in ein gegliedertes WORD-Dokument eingefügt und kapitelweise automatisch benummert werden können. Die erste Methode ist tabellenbasiert, die zweite tabellenlos. Die zur Automatisierung der beiden Methoden erforderlichen Prozeduren werden beschrieben und der jeweils dazugehörige VBA-Quellcode wird gezeigt.

15.10.2016, Dr. Volker Thormählen

Inhaltsverzeichnis

1 Aufgabenstellung 1
1.1 Verwendung eingebauter Formatvorlagen 1
1.2 Liste mit mehreren Ebenen 1
2 Tabellenbasierte Methode 2
2.1 Ziel der tabellenbasierten Methode 2
2.2 Nummernschema 3
2.2.1 Kapitelnummer 3
2.2.2 Gliederungszeichen 3
2.2.3 Zählnummer 3
2.3 Auswahl im Schnellbaustein-Katalog speichern 4
2.4 Autotext-Baustein aufrufen und einfügen 5
2.5 Automatische Erstellung eines AutoText-Bausteins 5
2.5.1 Hauptprogramm 5
2.5.2 Funktionen 6
2.5.2.1 CheckChapterNbr 6
2.5.2.2 CheckAutoTextExists 7
2.5.3 Unterprogramme 7
2.5.3.1 CreateTable 7
2.5.3.2 PopulateCells 8
2.5.3.3 DeleteAutoText 9
2.5.3.4 CreateAutoText 10
2.6 Aufruf des AutoText-Bausteins 10
3 Tabellenlose Methode 11
3.1 Ziel der tabellenlosen Methode 11
3.2 Nummernschema 11
3.3 Hauptprogramm 12
3.4 Funktionen 13
3.4.1 CheckChapterNbr 13
3.4.2 IsStyle 14
3.5 Unterprogramme 14
3.5.1 CreateEquationStyle 14
3.5.2 CheckCaptionLabels 14
3.5.3 CreateNewBookmark 15
3.6 Aufruf des Hauptprogramm 18

Abbildungsverzeichnis

Abb. 2.1: Einen neuen AutoText-Baustein namens „InsEq“ erstellen 2
Abb. 2.2: Kapitelnummer definieren 3
Abb. 2.3: Zählnummer erzeugen 4
Abb. 2.4: AutoText-Baustein namens „InsEq“ im Organizer für Schnellbausteine 4
Abb. 2.5: Eingefügter AutoText-Baustein namens „InsEq“ vor Aufruf des Formel-Editors 5
Abb. 2.6: Eingefügter AutoText-Baustein namens „InsEq“ nach Eingabe einer Formel 5
Abb. 3.1: Formel beschriften 11
Abb. 3.2: Kapitelnummer und Gliederungszeichen 11
Abb. 3.3 Neue Beschriftungskategorie „Formel“ erstellen 12
Abb. 3.4: Textmarke für die erste Formel im sechsten Kapitel 17
Abb. 3.5: Querverweis auf eine benummerte Formel mit definierter Textmarke 17
Abb. 3.6: Aufruf des Hauptprogramms via „*Symbolleiste für den Schnellzugriff*“ 18
Abb. 3.7: Symbol zum Aufruf des Hauptprogramms in der „*Symbolleiste für den Schnellzugriff*" 18

Listings

Listing 2.1: Hauptprogramms für die tabellenbasierte Methode 6
Listing 2.2: Funktion „*CheckChapterNbr*“: Nummerische Kapitelnummer prüfen 7
Listing 2.3: Funktion „*CheckAutoTextExists*“: AutoText-Baustein prüfen 7
Listing 2.4: Leere Tabelle am Ende des aktuellen Dokuments einfügen 7
Listing 2.5: Rahmenlinien der übergebenen Tabelle entfernen 8
Listing 2.6: Die zweite und dritte Zelle der übergebenen Tabelle belegen 8
Listing 2.7: Kapitelnummer für Formeln einfügen 8
Listing 2.8: Zählnummer für Formeln einfügen 8
Listing 2.9: Benannten AutoText-Baustein entfernen, falls vorhanden 9
Listing 2.10: Benannten AutoText-Baustein erstellen 10
Listing 3.1: Hauptprogramm für die tabellenlose Methode 13
Listing 3.2: Existenz der Formatvorlage „MatheFormel“ prüfen 14
Listing 3.3: Benutzerdefinierte Formatvorlage für Formeln und deren Nummerung erstellen 14
Listing 3.4: Existenz der Beschriftungskategorie "*Formel*" prüfen und ggf. neu anlegen 15
Listing 3.5: Textmarke für Querverweise auf Formeln und deren Nummerung erzeugen 16
Listing 3.6: Voreinstellung für den Namen einer Textmarke erstellen 16
Listing 3.7: Gültigkeitsprüfung des Namens einer Textmarke 16

1 Aufgabenstellung

In diesem Beitrag werden zwei Methoden beschrieben, wie mathematische Formeln bequem in ein gegliedertes WORD-Dokument eingefügt und je Kapitel automatisch benummert werden können. Die erste Methode ist tabellenbasiert, die zweite tabellenlos. Zur automatischen Nummerung der mathem. Formeln wird jeweils folgendes Nummernschema benutzt:

1. Nummernteil: Nummerische Kapitelnummer des entsprechend gegliederten WORD[1]-Dokuments
2. Nummernteil: Gliederungszeichen, gewöhnlich ein Punkt
3. Nummernteil: Zählnummer für die Formeln in einem Kapitel

Die Automatisierung beider Methoden mittels VBA[2]-Prozeduren wird demonstriert. Beide Methoden beruhen auf gleichen Voraussetzungen:

1.1 Verwendung eingebauter Formatvorlagen

Das betreffende WORD-Dokument muss mit den eingebauten Formatvorlagen für die benutzten Überschriftsebenen gegliedert sein. Die eingebauten Formatvorlagen *Überschrift 1*, *Überschrift 2* und so weiter werden also verwendet. Navigation: Start ▶ Formatvorlagen

1.2 Liste mit mehreren Ebenen

Aus den Listenvorlagen muss der Listentyp „*Gliederung*“ (*wdOutlineNumberGallery*) gewählt werden. Navigation: *Start* ▶ *Absatz* ▶ *Liste mit mehreren Ebenen*. Mit diesem Listentyp lässt sich eine mehrstufige Aufzählungsliste auswählen und mit den dazugehörigen Formatvorlagen verbinden. Unter „Listenbibliothek“ wird dort gewöhnlich die 5. Liste ausgewählt. Sie enthält als ersten Eintrag „*1 Überschrift 1*“, dann „*1.1 Überschrift 2*“ und so weiter.

[1] WORD ist ein weit verbreitetes Textverarbeitungsprogramm der Firma *Microsoft*

[2] Das Kürzel VBA bedeutet: **V**isual **B**asic for **A**pplications. Das ist die Skriptsprache der sog. Office-Suite von *Microsoft*.

© Copyright Dr. V. Thormählen, 2016

2 Tabellenbasierte Methode

Dabei wird eine einzeilige WORD-Tabelle mit drei Spalten eingesetzt. Die aus drei Nummernteilen zusammengesetzte Nummer[3] einer mathem. Formel wird automatisch in runden Klammern in die 3. Spalte der Tabelle eingesetzt:

- Die 1. Spalte bleibt leer.
- Die 2. Spalte enthält die von Benutzer erfasste mathem. Formel.
- Die 3. Spalte beinhaltet das Nummernschema[4] für die jeweilige mathem. Formel.

Die Rahmenlinien der WORD-Tabelle werden ausgeblendet.

2.1 Ziel der tabellenbasierten Methode

Das Ziel der Tabellen-Methode für mathematische Formeln besteht darin, einen AutoText-Baustein mit einem bestimmten Namen zu erzeugen und diesen an der gewünschten Stelle im aktuellen Dokument immer wieder neu einzufügen. Dazu sind folgende vier Schritte erforderlich:

1. Prüfen, ob die Tabellen-Methode angewandt werden kann. Das ist nur möglich, wenn das betreffende WORD-Dokument mit den eingebauten Formatvorlagen für Überschriften gegliedert ist und auf der obersten Gliederungsebene nummerische Kapitelnummern zugeordnet sind. Wenn das zutrifft, dann weiter mit Schritt 2, sonst Ende.
2. Eine WORD-Tabelle mit 1 Zeile und 3 Spalten ist zu erstellen und die Breite der Tabellenspalten ist festzulegen. Daher stammt die Bezeichnung „Tabellen-Methode“.
3. Die erste Zelle der eingefügten WORD-Tabelle bleibt frei. Die zweite Zelle wird mit einem Platzhalter für den in WORD eingebauten Formel-Editor belegt. Die dritte Zelle nimmt die dazugehörige Nummer auf, zusammensetzt aus Kapitelnummer, Gliederungszeichen und Zählnummer[5].
4. Die so korrekt belegte WORD-Tabelle ist als Ganzes zu markieren und diese Auswahl ist im sog. Schnellbaustein-Katalog als neuer AutoText-Baustein unter einem vom Benutzer bestimmten Namen zu speichern, s. Listing 2.1. Dieser Name dient später dazu, um den erzeugten AutoText-Baustein an der gewählten Stelle in das aktuelle WORD-Dokument einzufügen.

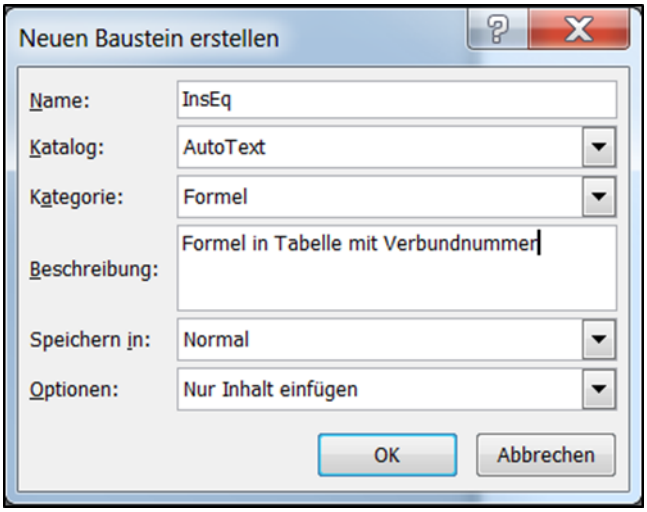


Abb. 2.1: Einen neuen AutoText-Baustein namens „InsEq“ erstellen

[3] Sie besteht aus einem klassifizierenden und einem zählenden Nummernteil, wobei der zählende (Zählnummer) von dem klassifizierenden (Kapitelnummer) abhängt.
[4] Darstellung des Aufbaus einer Nummer.
[5] Sie wird in WORD Folge- oder Sequenznummer genannt.

© Copyright Dr. V. Thormählen, 2016

2.2 Nummernschema

2.2.1 Kapitelnummer

Als Kapitelnummer wird die Absatznummer der Formatvorlage „*Überschrift 1*“ benutzt, s. Abb. 2.2. Dafür wird das Feld „*StyleRef*“ herangezogen.

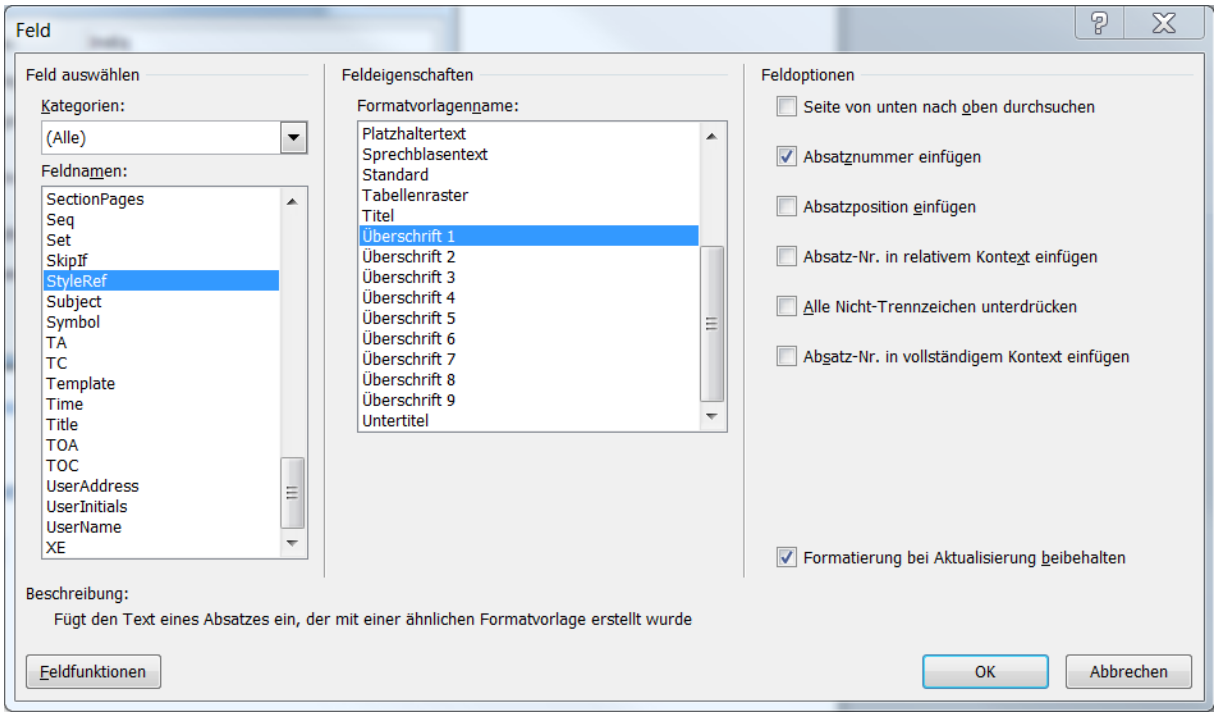


Abb. 2.2: Kapitelnummer definieren

2.2.2 Gliederungszeichen

Als Gliederungszeichen wird ein Punkt verwandt. Er dient dazu, Nummern übersichtlicher und leichter lesbar zu machen.

2.2.3 Zählnummer

Das Feld „Seq“ mit dem gewählten Erkennungszeichen „*Eq*“ und dem speziellen Feldschalter „\n“ wird dazu genutzt, um eine Zählnummer für die jeweilige Formel in einem Kapitel zu erzeugen, s. Abb. 2.3.

© Copyright Dr. V. Thormählen, 2016

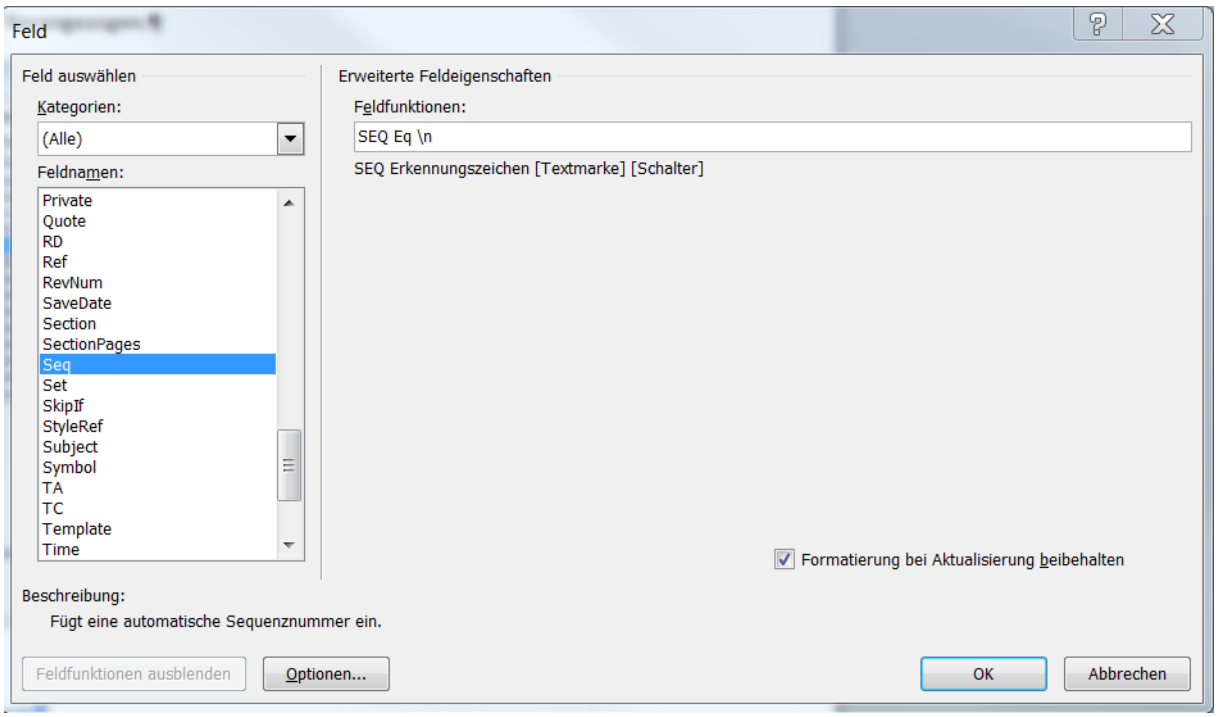


Abb. 2.3: Zählnummer erzeugen

2.3 Auswahl im Schnellbaustein-Katalog speichern

Die korrekt belegte WORD-Tabelle ist als Ganzes zu markieren, damit ein AutoText-Baustein erzeugt werden kann, der beliebig oft und an beliebiger Stelle im aktuellen WORD-Dokument eingefügt werden kann, vgl. Abb. 2.1und Abb. 2.4.

Mit dem Befehl „*Felder aktualisieren*“ in sog. Kontextmenü kann der Inhalt aller Felder ggf. auf den neusten Stand gebracht werden. Das Kontextmenü öffnet sich durch einen Klick mit der rechten Maustaste an der entsprechenden Stelle und enthält alle hier anwendbaren Befehle.

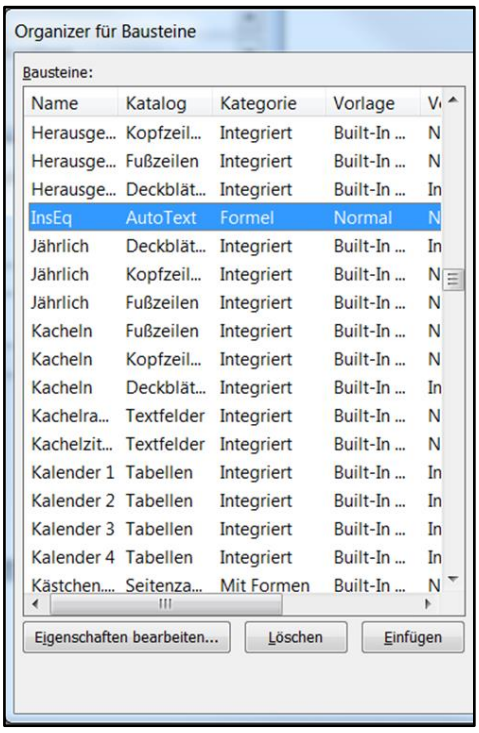


Abb. 2.4: AutoText-Baustein namens „InsEq“ im Organizer für Schnellbausteine

© Copyright Dr. V. Thormählen, 2016

2.4 Autotext-Baustein aufrufen und einfügen

Wird der Name eines AutoText-Baustein an beliebiger Stelle im aktuellen WORD-Dokument eingetippt, erscheint dort automatisch folgende Aufforderung: *„InsEq (Eingabetaste drucken, um einzufügen)“*. *„InsEq“* ist dabei der gewählte Name des relevanten AutoText-Bausteins, s. Abb. 2.1. Alternativ kann nach dem Schreiben des gewählten Namens die Funktionstaste *„F3“* gedrückt werden. Abb. 2.5 zeigt das jeweilige Ergebnis:

¤	Geben Sie hier eine Formel ein.¤	(6.1)¤

Abb. 2.5: Eingefügter AutoText-Baustein namens „InsEq“ vor Aufruf des Formel-Editors

Nach der Eingabe einer mathematischen Formel mit dem eingebauten Editor-Editor ergibt sich beispielsweise (s. Abb. 2.6):

¤	$\sqrt{a^2 + b^2}$¤	(6.1)¤

Abb. 2.6: Eingefügter AutoText-Baustein namens „InsEq“ nach Eingabe einer Formel

Auf die Bedienung des Formel-Editors wird an dieser Stelle nicht weiter eingegangen. Sie ist ausführlich beschrieben in [1] .

2.5 Automatische Erstellung eines AutoText-Bausteins

2.5.1 Hauptprogramm

Aufgabe der Prozedur *„CreateEquationTable“* ist, die sog. Tabellen-Methode für das Einfügen mathematischer Formeln in ein Word-Dokument zu automatisieren. Sie benötigt dafür zwei Funktionen (türkis hervorgehoben) und vier Unterprogramme (gelb hervorgehoben). Ihre jeweilige Aufgabe ist im VBA-Quellcode dokumentiert, s. Listing 2.1.

```
Sub CreateEquationTable()
   ' Tabellenbasierte Methode für mathematische Formeln anwenden.
   Const conClassificationSign As String = "." ' Gliederungszeichen: Punkt
   Dim objCell As Cell     ' Tabellenzelle
   Dim objRng  As Range    ' Bereich
   Dim objTbl  As Table    ' Tabelle
   Dim objDoc  As Document ' Dokument
   On Error GoTo Err_Point ' Fehlerroutine
   Set objDoc = ActiveDocument
   ' Prüfen, ob die Tabellen-Methode angewandt werden kann.
   If Not CheckChapterNbr(objDoc) Then
      MsgBox "Abbruch: Das aktuelle Dokument enthält keine nummerierten Listen.", _
         vbCritical, "Tabellen-Methode für mathematische Formeln"
      Exit Sub
   End If
   ' Am Ende des aktuellen Dokuments eine Tabelle mit 1 Zeile und 3 Spalten erstellen.
   Call CreateTable(objDoc)
   Set objTbl = objDoc.Tables(objDoc.Tables.Count)
   ' Die zweite und dritte Zelle der eingefügten Tabelle belegen.
   Call PopulateCells(objTbl)
   ' Eingefügte Tabelle markieren
   objTbl.Range.Select
   ' AutoText-Baustein mit dem Namen "InsEq" entfernen, falls bereits vorhanden.
   If CheckAutoTextExists("InsEq") Then
      Call DeleteAutoText("InsEq")
   End If
   ' AutoText-Baustein mit dem Namen "InsEq" neu erzeugen.
```

© Copyright Dr. V. Thormählen, 2016

```
   Call CreateAutoText("InsEq")
   ' Erfolgsmeldung ausgeben
   MsgBox "Der AutoText-Baustein 'InsEq' wurde erfolgreich erstellt!" & vbCrLf & vbCrLf & _
         "(InsEq tippen und dann die Eingabetaste drücken, um einzufügen.)", _
         vbInformation, "AutoText-Baustein 'InsEq' einfügen."
Exit_Point:
   Set objCell = Nothing
   Set objRng = Nothing
   Set objTbl = Nothing
   Set objDoc = Nothing
   Exit Sub
Err_Point:
   MsgBox Err.Description, vbCritical, "AutoText-Baustein 'InsEq' erstellen."
   Resume Exit_Point
End Sub
```

Listing 2.1: Hauptprogramms für die tabellenbasierte Methode

2.5.2 Funktionen

2.5.2.1 CheckChapterNb*r*

Die Funktion „*CheckChapterNbr*“ dient im Hauptprogramm zur Prüfung, ob die Voraussetzungen für die Anwendung der sog. Tabellen-Methode gegeben sind, vgl. dazu Seite 1. Der zugehörige Quellcode ist in Listing 2.2 zu finden. Wenn diese Funktion den Wert „*False*“ zurückgibt, ist die Prüfung negativ ausgefallen und das Hauptprogramm wird mit *Exit Sub* abgebrochen.

```
Function CheckChapterNbr(objDoc As Document) As Boolean
   ' Nummerische Kapitelnummer auf oberster Gliederungsebene prüfen.
   Dim strStyleNm As String   ' Name der Formatvorlage für Überschrift 1
   Dim strListVal As Long     ' Listenwert
   On Error GoTo Err_Point
   CheckChapterNbr = False    ' Anfangswert setzen: Kapitelnummer ist nicht nummerisch
   If objDoc.Lists.Count > 0 Then
      strStyleNm = objDoc.Styles(wdStyleHeading1).NameLocal
      With Selection
         .HomeKey Unit:=wdStory, Extend:=wdMove
         ' Ersten Absatz finden mit der Formatvorlage für Überschrift 1
         With .Find
            .ClearFormatting
            .Forward = True
            .Wrap = wdFindStop
            .Style = strStyleNm
            .Execute
            If .Found Then
               With Selection
                  ' Gliederungsebene prüfen
                  If .Paragraphs(1).OutlineLevel = wdOutlineLevel1 Then
                     strListVal = .Range.ListFormat.ListString
                     ' Listenwert prüfen
                     If IsNumeric(strListVal) Then
                        CheckChapterNbr = True  ' Kapitelnummer ist nummerisch
                     End If
                  End If
               End With
            End If
         End With
      End With
   End If
Exit_Point:
   Exit Function
```

```
Err_Point:
   MsgBox "Laufzeitfehler: " & Err.Number & ", " & Err.Description, _
      vbCritical, "Test Kapitelnummer"
   Resume Exit_Point
End Function
```

Listing 2.2: Funktion „*CheckChapterNbr*“: Nummerische Kapitelnummer prüfen

2.5.2.2 CheckAutoTextExists

Die Funktion „*CheckAutoTextExists*“ prüft im Hauptprogramm (s. Listing 2.1), ob bereits ein AutoText-Baustein mit dem Namen „*InsEq*“ existiert. Die Funktion erhält diesen Namen in der Variablen *strEntryNm* übergeben, s. Listing 2.3. Wenn „*True*“, wird er entfernt, um unmittelbar danach neu angelegt zu werden.

```
Function CheckAutoTextExists(strEntryNm As String) As Boolean
   Dim entry As AutoTextEntry
   CheckAutoTextExists = False
   For Each entry In NormalTemplate.AutoTextEntries
      If entry.Name = strEntryNm Then
         CheckAutoTextExists = True
         Exit Function
      End If
   Next entry
End Function
```

Listing 2.3: Funktion „*CheckAutoTextExists*“: AutoText-Baustein prüfen

2.5.3 Unterprogramme

2.5.3.1 CreateTable

```
Sub CreateTable(objDoc As Document)
   ' Am Ende des aktuellen Dokuments eine Tabelle mit 1 Zeile und 3 Spalten erstellen.
   Dim objTbl As Table     ' Tabelle
   With Selection
      .EndKey Unit:=wdStory, Extend:=wdMove
      .InsertAfter Text:=vbCrLf
      .MoveDown Unit:=wdLine, Count:=1
   End With
   With objDoc
      .Tables.Add _
         Range:=Selection.Range, _
         NumRows:=1, NumColumns:=3, _
         DefaultTableBehavior:=wdWord9TableBehavior, _
         AutoFitBehavior:=wdAutoFitFixed
      Set objTbl = .Tables(.Tables.Count)
      With objTbl
         .AllowAutoFit = False
         ' Breite in Prozent für die angegebene Spalte bestimmen.
         .Columns(1).PreferredWidthType = wdPreferredWidthPercent
         .Columns(1).PreferredWidth = 15
         .Columns(2).PreferredWidthType = wdPreferredWidthPercent
         .Columns(2).PreferredWidth = 70
      End With
      Call HideBorderLines(objTbl) ' Rahmenlinien der Tabelle entfernen
   End With
End Sub
```

Listing 2.4: Leere Tabelle am Ende des aktuellen Dokuments einfügen

© Copyright Dr. V. Thormählen, 2016

```
Sub HideBorderLines(objTbl As Table)
   ' Rahmenlinien der übergebenen Tabelle entfernen
   With objTbl.Borders
      .InsideLineStyle = wdLineStyleNone
      .OutsideLineStyle = wdLineStyleNone
   End With
End Sub
```

Listing 2.5: Rahmenlinien der übergebenen Tabelle entfernen

2.5.3.2 PopulateCells

```
Sub PopulateCells(objTbl As Table)
   ' Die zweite und dritte Zelle der übergebenen Tabelle belegen.
   Const conClassificationSign As String = "."  ' Gliederungszeichen: Punkt
   Dim objCell As Cell                          ' Tabellenzelle
   With objTbl
       For Each objCell In .Columns(2).Cells
         With objCell
            .Range.ParagraphFormat.Alignment = wdAlignParagraphCenter
            .VerticalAlignment = wdCellAlignVerticalCenter
            .Select
            With Selection
               .Collapse
               .OMaths.Add Range:=Selection.Range     ' Formel-Editor einfügen
               .MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdMove
            End With
         End With
      Next objCell
      For Each objCell In .Columns(3).Cells
         With objCell
            .Range.ParagraphFormat.Alignment = wdAlignParagraphRight
            .VerticalAlignment = wdCellAlignVerticalCenter
            .Select
            With Selection
               .TypeText Text:="()"                  ' Runde Klammern einfügen
               .MoveLeft Unit:=wdCharacter, Count:=1
               InsertChapterNumber                   ' Kapitelnummer einfügen
               .TypeText Text:=conClassificationSign  ' Gliederungszeichen einfügen
               InsertSequenceNumber                  ' Zählnummer für Formeln einfügen
            End With
         End With
      Next objCell
   End With
End Sub
```

Listing 2.6: Die zweite und dritte Zelle der übergebenen Tabelle belegen

```
Sub InsertChapterNumber()
   ' Kapitelnummer für Formeln einfügen
   Selection.Fields.Add Range:=Selection.Range, Type:=wdFieldEmpty, Text:= _
      "STYLEREF  ""Überschrift 1"" \n ", PreserveFormatting:=True
End Sub
```

Listing 2.7: Kapitelnummer für Formeln einfügen

```
Sub InsertSequenceNumber()
   ' Zählnummer für Formeln einfügen
   Selection.Fields.Add Range:=Selection.Range, Type:=wdFieldEmpty, Text:= _
      "SEQ Eq\* Arabic\n ", PreserveFormatting:=True
End Sub
```

Listing 2.8: Zählnummer für Formeln einfügen

© Copyright Dr. V. Thormählen, 2016

2.5.3.3 DeleteAutoText

```
Sub DeleteAutoText(strAutoTextNm As String)
   ' Benannten AutoText-Baustein entfernen, falls vorhanden.
   Dim objBB      As BuildingBlock
   Dim objTmp     As Template
   Dim strTitle   As String
   Dim strPrompt  As String
   On Error GoTo Err_Point
   strPrompt = "Abbruch: AutoText-Baustein '" & strAutoTextNm & "' nicht gefunden."
   strTitle = "AutoText-Baustein '" & strAutoTextNm & "' entfernen."
   Templates.LoadBuildingBlocks
   For Each objTmp In Templates
      If objTmp.Name = "Normal.dotm" Then Exit For
   Next objTmp
   If Not objTmp Is Nothing Then
      On Error Resume Next
      ' Bausteintyp, Kategorie und Name des AutoText-Bausteins bestimmen
      Set objBB = objTmp.BuildingBlockTypes(wdTypeAutoText). _
         Categories("Formel").BuildingBlocks(strAutoTextNm)
      If Err.Number = 0 Then
         objBB.Delete   ' AutoText-Baustein entfernen
      Else
         MsgBox strPrompt, vbCritical, strTitle
      End If
   Else
      MsgBox strPrompt, vbCritical, strTitle
   End If
Exit_Point:
   Set objBB = Nothing
   Set objTmp = Nothing
   Exit Sub
Err_Point:
   MsgBox Err.Description, vbExclamation, strTitle
   Resume Exit_Point
End Sub
```

Listing 2.9: Benannten AutoText-Baustein entfernen, falls vorhanden

© Copyright Dr. V. Thormählen, 2016

2.5.3.4 CreateAutoText

```
Sub CreateAutoText(strAutoTextNm As String)
   ' Benannten AutoText-Baustein erstellen
   Dim objRng  As Word.Range
   Dim objTmp  As Template
   On Error GoTo Err_Point
   Templates.LoadBuildingBlocks
   For Each objTmp In Templates
      If objTmp.Name = "Normal.dotm" Then Exit For
   Next objTmp
   If Not objTmp Is Nothing And Selection.Type = wdSelectionRow Then
      Set objRng = Selection.Range
      objTmp.BuildingBlockEntries.Add _
         Name:=strAutoTextNm, _
         Type:=wdTypeAutoText, _
         Category:="Formel", _
         Range:=objRng, _
         InsertOptions:=wdInsertContent
   Else Else
      MsgBox "Keine Tabelle mit eingefügter Formel markiert", _
         vbExclamation, "AutoText-Baustein 'InsEq' erstellen."
   End If
Exit_Point:
   Set objRng = Nothing
   Set objTmp = Nothing
   Exit Sub
Err_Point:
   MsgBox Err.Description, vbCritical, "AutoText-Baustein 'InsEq' erstellen."
   Resume Exit_Point
End Sub
```

Listing 2.10: Benannten AutoText-Baustein erstellen

2.6 Aufruf des AutoText-Bausteins

Der Schnellbaustein durch die Eingabe seines Namens „InsEq“ aufgerufen, s. dazu Abschnitt 2.4

© Copyright Dr. V. Thormählen, 2016

3 Tabellenlose Methode

Dabei wird – wie die Überschrift besagt - keine WORD-Tabelle benutzt, sondern mit zwei Tabstopps gearbeitet:

1. Der erste Tabstopp wird zentriert. Hier wird die mathem. Formel platziert.
2. Der zweite Tabstopp wird rechtsbündig angeordnet. Hier wird die aus den drei Nummernteilen zusammengesetzte Nummer für die jeweilige Formel in runden Klammern platziert.

3.1 Ziel der tabellenlosen Methode

Das Ziel der tabellenlosen Methode ist das Einsetzen und das Benummern von mathem. Formeln in ein WORD-Dokument *ohne* Benutzung von WORD-Tabellen.

Dazu sind folgende Schritte erforderlich:

1. Prüfen, ob die tabellenlose Methode angewandt werden kann. Wie bei der tabellenbasierten Methode ist das nur möglich, wenn das betreffende WORD-Dokument mit den eingebauten Formatvorlagen für Überschriften gegliedert ist und auf der obersten Gliederungsebene nummerische Kapitelnummern zugeordnet sind. Wenn zutreffend, dann weiter mit Schritt 2, sonst Ende.
2. Ein zentrierter und ein rechtsbündiger Tabstopp wird in der gewünschten Zeile hinzufügt. Eine WORD-Tabelle wird also nicht benötigt.
3. An der Stelle des zentrierten Tabstopps wird ein Platzhalter für den in WORD eingebauten Formel-Editor eingefügt. Rechtsbündig erscheint die dazugehörige Nummer in runden Klammern, zusammensetzt aus Kapitelnummer, Gliederungszeichen und Zählnummer.
4. Um Querverweise auf die jeweilige Nummer zu ermöglichen, wird automatisch eine Textmarke erzeugt. Textmarken haben in WORD die Aufgabe, das schnelle Springen an eine bestimmte Stelle (hier: mathem. Formel) zu gestatten.

3.2 Nummernschema

In WORD gibt es eine eingebaute Funktionalität namens *„Beschriftung einfügen“* (Navigation: *Verweise* ▶ *Beschriftung einfügen*) mit der Objekte wie Abbildungen, Tabellen und auch Formeln beschriftet und benummert werden können. Eine Beschriftung besteht aus einer auswählbaren und anpassbaren Kategorie (wie *„Abbildung*", *„Tabelle*", *„Listing“* oder *„Formel*") gefolgt von einem wählbaren Nummernschema.

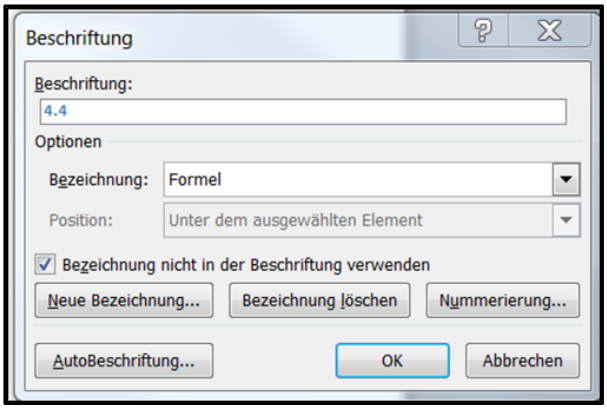


Abb. 3.1: Formel beschriften

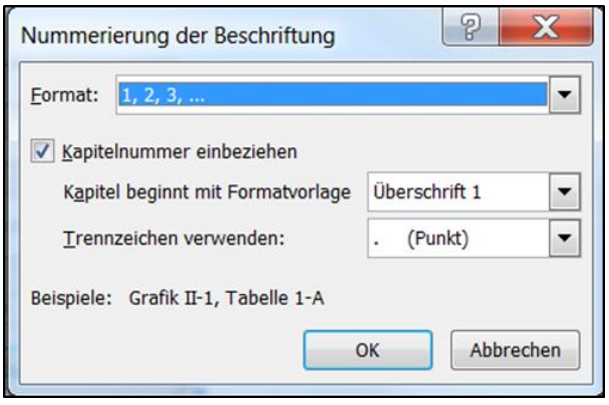


Abb. 3.2: Kapitelnummer und Gliederungszeichen

© Copyright Dr. V. Thormählen, 2016

Mit der Schaltfläche „*Neue Bezeichnung*“ (s. Abb. 3.1) kann eine neue Beschriftungskategorie eingerichtet werden, s. Abb. 3.3.

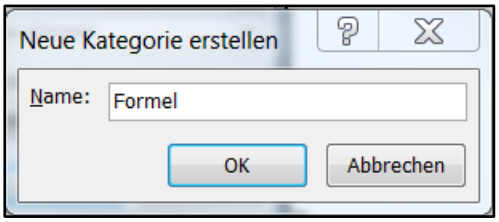


Abb. 3.3 Neue Beschriftungskategorie „Formel“ erstellen

Mit der Schaltfläche „*Nummerierung*“ (s. Abb. 3.1) kann das Nummernschema für die Beschriftungskategorie „*Formel*“ festgelegt werden, s. Abb. 3.2. Es besteht aus nummerischer Kapitelnummer, Gliederungszeichen (Punkt) und einer nummerischen Nummer (Zählnummer), genau in dieser Reihenfolge.

3.3 Hauptprogramm

Aufgabe der Prozedur „*InsertNumberedEquation*“ ist die Anwendung der tabellenlosen Methode für das Einfügen mathematischer Formeln in ein gegliedertes WORD-Dokument. Sie benötigt dafür zwei Funktionen (türkis hervorgehoben) und nur ein Unterprogramm (gelb hervorgehoben). Ihre jeweilige Aufgabe ist im VBA-Quellcode dokumentiert, s. Listing 3.1. Hinzu kommt noch ein weiteres Unterprogramm für das Anlegen von Textmarken für Querverweise auf neu eingefügte Formeln und deren Nummerung, s. Listing 3.5.

```
Sub InsertNumberedEquation()
   Const conCaptionNm As String = "Formel"                ' Beschriftungskategorie
   Const conStyleNm As String = "MatheFormel"             ' Formatvorlage
   Const conProcNm  As String = "InsertNumberedEquation" ' Prozedurname
   Dim objDoc  As Document                                ' Dokument
   On Error GoTo Err_Point                                ' Fehlerbehandlung
   Set objDoc = ActiveDocument
   ' Prüfen, ob das aktuelle Dokument eine nummerierten Liste enthält.
   If Not CheckChapterNbr(objDoc) Then
      MsgBox "Abbruch: Das aktuelle Dokument enthält keine nummerierten Listen!", _
         vbCritical, conProcNm
      GoTo Exit_Point
   End If
   ' Prüfen, ob die Formatvorlage "MatheFormel" existiert, sonst anlegen.
   If Not IsStyle(conStyleNm) Then
      Call CreateEquationStyle(objDoc, conStyleNm)
   End If
   ' Prüfen, ob die Beschriftungskategorie "Formel" definiert ist,
   ' sonst anlegen.
   Call CheckCaptionLabels(conCaptionNm)
   ' Zum Ende des aktuellen Dokuments springen
   Selection.EndKey Unit:=wdStory, Extend:=wdMove
   ' Absatzformatierung setzen
   With Selection.ParagraphFormat
       .SpaceBeforeAuto = False
       .SpaceAfterAuto = False
       .FirstLineIndent = CentimetersToPoints(0)
   End With
   ' Zentrierten und rechtsbündigen Tabstopp setzen
   ' für die Formel und ihre Beschriftung (d. h. Nummerung)
   With objDoc.PageSetup
```

© Copyright Dr. V. Thormählen, 2016

```
      ' Benutzerdefinierten Tabstopps im aktiven Dokument entfernen
      Selection.ParagraphFormat.TabStops.ClearAll
      ' Zentralen Tabstopp setzen
      Selection.ParagraphFormat.TabStops.Add _
         Position:=(.PageWidth - .LeftMargin - .RightMargin) / 2, _
         Alignment:=wdAlignTabCenter, Leader:=wdTabLeaderSpaces
      ' Rechtsbündigen Tabstopp setzen
      Selection.ParagraphFormat.TabStops.Add _
         Position:=(.PageWidth - .LeftMargin - .RightMargin), _
         Alignment:=wdAlignTabRight, Leader:=wdTabLeaderSpaces
   End With
   With Selection
      ' Zentralen Tabstopp einfügen
      .TypeText Text:=vbTab
      .Collapse Direction:=wdCollapseEnd
      ' Platzhalter für Formel-Editor einfügen
      .OMaths.Add Range:=Selection.Range
      .MoveRight Unit:=wdCharacter, Count:=1, Extend:=wdMove
      Set objRange = .Range
      ' Rechtsbündigen Tabstopp einfügen
      .TypeText Text:=vbTab & "("
      ' Formel beschriften (d. h. benummern)
      .InsertCaption _
         Label:=conCaptionNm, _
         TitleAutoText:="X", _
         ExcludeLabel:=True
      .TypeText Text:=")"
      ' Formatvorlage auf Formel u. zugehörige Beschriftung anwenden
      .Style = objDoc.Styles(conStyleNm)
      objRange.SetRange Start:=objRange.Start, End:=Selection.End
      .InsertAfter Text:=vbCrLf
      .Collapse Direction:=wdCollapseEnd
   End With
   If Len(objRange.Text) > 5 And InStr(objRange.Text, ".") > 0 Then
      ' Textmarke anlegen für Querverweise auf die neu eingefügte Formel
      Call CreateNewBookmark(objRange)
   End If
Exit_Point:
   On Error Resume Next
   Set objDoc = Nothing
   Set objRange = Nothing
   Exit Sub
Err_Point:
   MsgBox "Laufzeitfehler: " & Err.Description, vbCritical, conProcNm
   Resume Exit_Point
End Sub
```

Listing 3.1: Hauptprogramm für die tabellenlose Methode

3.4 Funktionen

3.4.1 CheckChapterNbr

Die Funktion „*CheckChapterNbr*“ dient im Hauptprogramm zur Prüfung, ob die Voraussetzungen für die Anwendung der tabellenlosen Methode gegeben sind. Der zugehörige Quellcode ist wiederum in Listing 2.2 zu finden. Wenn diese Funktion den Wert „*False*“ zurückgibt, ist die Prüfung negativ ausgefallen und das Hauptprogramm wird abgebrochen.

© Copyright Dr. V. Thormählen, 2016

3.4.2 IsStyle

Die Funktion „*IsStyle*“ prüft, ob die benutzerdefinierte Formatvorlage mit dem Namen „*MatheFormel*“ existiert. Falls nein, wird sie angelegt, s. Listing 3.3.

```
Function IsStyle(strStyleNm As String) As Boolean
   ' Prüfen, ob die Formatvorlage "MatheFormel" existiert
   Dim objStyle As Style
   IsStyle = False
   For Each objStyle In ActiveDocument.Styles
      If objStyle.NameLocal = strStyleNm Then
         IsStyle = True
         Exit For
      End If
   Next objStyle
End Function
```

Listing 3.2: Existenz der Formatvorlage „MatheFormel“ prüfen

3.5 Unterprogramme

3.5.1 CreateEquationStyle

Dieses Unterprogramm erzeugt die benutzerdefinierte Formatvorlage mit dem Namen „*MatheFormel*“, falls diese noch nicht existiert. Diese Routine definiert die Positionen für einen zentralen (*lngCenter*) und einen rechtsbündigen (*lngRight*) Tabstopp, s. Listing 3.3. Diese Formatvorlage sorgt dafür, dass die jeweilige mathem. Formel und deren Nummerung an der dafür vorgesehenen Stelle eingefügt werden.

```
Sub CreateEquationStyle(objDoc As Document, strStyleNm As String)
   ' Formatvorlage "MatheFormel" erstellen
   Dim lngCenter  As Long
   Dim lngRight   As Long
   With objDoc
      With .PageSetup
         lngCenter = CLng((.PageWidth - .LeftMargin - .RightMargin) / 2)
         lngRight = CLng(.PageWidth - (.LeftMargin + .RightMargin))
      End With
      .Styles.Add Name:=strStyleNm, Type:=wdStyleTypeParagraph
      With .Styles(strStyleNm)
         With .ParagraphFormat.TabStops
            .ClearAll
            .Add Position:=lngCenter, Alignment:=wdAlignTabCenter, Leader:=wdTabLeaderSpaces
            .Add Position:=lngRight, Alignment:=wdAlignTabRight, Leader:=wdTabLeaderSpaces
         End With
         .AutomaticallyUpdate = False
         .BaseStyle = "Standard"
         .NextParagraphStyle = "Standard"
         .LanguageID = wdGerman
         .NoProofing = True   ' Keine Rechtschreibprüfung
         .Frame.Delete        ' Rahmenformatierung entfernen
      End With
   End With
End Sub
```

Listing 3.3: Benutzerdefinierte Formatvorlage für Formeln und deren Nummerung erstellen

3.5.2 CheckCaptionLabels

Zum Benummern mathematischer Formeln wird die Existenz der Beschriftungskategorie „*Formel*“ geprüft. Falls die Prüfung negativ ausfällt, wird diese Beschriftungskategorie neu angelegt.

© Copyright Dr. V. Thormählen, 2016

Die Beschriftungskategorie „*Formel*“ besitzt folgende Eigenschaften, s. Listing 3.4 in Verbindung mit Abb. 3.2:

- *NumberStyle*: Nummerische Nummer
- *IncludeChapterNumber*: Kapitelnummer wird einbezogen
- *ChapterStyleLevel*: Kapitelnummer entspricht der obersten Gliederungsebene
- *Separator*: Gliederungszeichen ist ein Punkt

```
Sub CheckCaptionLabels(ByVal strCaptionNm As String)
   ' Prüfen, ob die Beschriftungskategorie "Formel" definiert ist,
   ' sonst neu anlegen
   Dim lngLabel   As Long     ' Schleifenzähler für Beschriftungskategorien
   Dim bolLabel   As Boolean  ' Schalter für die Beschriftungskategorie "Formel"
   bolLabel = False
   For lngLabel = 1 To Application.CaptionLabels.Count
      If Application.CaptionLabels(lngLabel).Name = strCaptionNm Then
         bolLabel = True
         Exit For
      End If
   Next lngLabel
   If bolLabel Then Exit Sub  ' ... gefunden
   On Error Resume Next
   ' ... nicht gefunden, also hinzufügen
   Application.CaptionLabels.Add (strCaptionNm)
   If Err.Number > 0 Then
      MsgBox "Laufzeitfehler " & Err.Description, vbCritical, "CheckCaptionLabels"
      Exit Sub
   Else
      On Error GoTo 0
      With Application.CaptionLabels(strCaptionNm)
         .NumberStyle = wdCaptionNumberStyleArabic
         .IncludeChapterNumber = True
         .ChapterStyleLevel = 1
         .Separator = wdSeparatorPeriod
      End With
   End If
End Sub
```

Listing 3.4: Existenz der Beschriftungskategorie "*Formel*" prüfen und ggf. neu anlegen

3.5.3 CreateNewBookmark

Dieses abschließende Unterprogramm erzeugt eine gültige Textmarke, um Querweise auf eine Formel und deren Nummerung zu ermöglichen, s. Listing 3.5.

```
Sub CreateNewBookmark(ByVal objRng As Range)
   ' Textmarke für Querverweise auf die eingefügte Formel anlegen
   Dim strReference  As String   ' Textmarke
   Dim strRefDefault As String   ' Voreinstellung
   ' Voreinstellung für den Namen der Textmarke erstellen
   strRefDefault = GetDefaultBmkName(objRng)
   ' Textmarke erstellen für mögliche Querverweise auf die Formel
   strReference = InputBox( _
      Prompt:="Name der Textmarke für Querverweise", _
      Title:="Textmarke für Querverweis erstellen", _
      Default:=strRefDefault)
   ' Wenn der Name der eingegebenen Textmarke formal richtig ist ...
   If IsValidBookmark(strReference, "^(?!\d)\w{1,39}$") Then
      With ActiveDocument
         ' Wenn die Textmarke bereits vorhanden ist ...
         If .Bookmarks.Exists(strReference) Then
            ' Textmarke löschen
```

© Copyright Dr. V. Thormählen, 2016

```
            .Bookmarks.Item(strReference).Delete
         End If
         With .Bookmarks
            ' Textmarke hinzufügen
            .Add Range:=objRng, Name:=strReference
            .DefaultSorting = wdSortByName
            .ShowHidden = False
         End With
      End With
   Else
      MsgBox Prompt:="Ungültige Textmarke für Querweise.", _
         Buttons:=vbInformation, Title:="Ungültige Textmarke"
   End If
End Sub
```

Listing 3.5: Textmarke für Querverweise auf Formeln und deren Nummerung erzeugen

Das Unterprogramm „*CreateNewBookmark*“, s. Listing 3.5, benötigt wiederum zwei Funktionen:

- GetDefaultBmkName, s. Listing 3.6.
- IsValidBookmark, s. Listing 3.7.

Die zuerst genannte Funktion erzeugt die Voreinstellung für die Erfassung des Namens einer Textmarke mit einem Eingabedialog (sog. *InputBox*). Die Voreinstellung kann vom Benutzer überschrieben werden. Die zweite Funktion prüft den gewählten Namen für die Textmarke auf Gültigkeit.

```
Function GetDefaultBmkName(objRng As Range) As String
   ' Voreinstellung für den Name einer zu erstellenden Textmarke festlegen
   Dim strRefDefault As String
   ' Tabstopp und rechte Klammer vor Nummer entfernen
   strRefDefault = Mid(Trim(objRng.Text), 3)
   ' Rechte Klammer hinter Nummer entfernen
   strRefDefault = Left(strRefDefault, Len(strRefDefault) - 1)
   ' Gliederungszeichen (Punkt) in Nummer durch Unterstrich ersetzen
   strRefDefault = Replace(strRefDefault, ".", "_")
   strRefDefault = "Formel_" & strRefDefault
   GetDefaultBmkName = strRefDefault ' Rückgabewert der Funktion
End Function
```

Listing 3.6: Voreinstellung für den Namen einer Textmarke erstellen

Bei der Gültigkeitsprüfung wird ein regulärer Ausdruck (engl. *regular expression*) angewandt. Dafür muss in WORD ein entsprechender Verweis gesetzt werden (Navigation in WORD 2010: *Entwicklertools* ▶ *Visual Basic* ▶ *Extras* ▶ *Verweise*). An die Funktion *IsValidBookmark* werden zwei Argumente übergeben, s. Listing 3.7*:*

- Der gewählte Name (*strBmkNm*) der Textmarke.
- Das Prüfmuster (*strPattern*) des regulären Ausdrucks.

```
Function IsValidBookmark(ByVal strBmkNm As String, ByVal strPattern As String) As Boolean
    ' Mit dem Muster (strPattern) prüfen, ob der übergebene Name (strBmkNm) einer Textmarke
      gültig ist.
    ' Benötigt den Verweis auf 'Microsoft VBScript Regular Expressions 5.5"
    With New RegExp
        .MultiLine = False
        .IgnoreCase = True
        .Global = False
        .Pattern = strPattern
        IsValidBookmark = .Test(strBmkNm)
    End With
End Function
```

Listing 3.7: Gültigkeitsprüfung des Namens einer Textmarke

© Copyright Dr. V. Thormählen, 2016

Wie sieht beispielsweise der Name einer Textmarke aus, wenn die automatisch erzeugte Voreinstellung für die InputBox (s. Listing 3.5) vom Benutzer unverändert übernommen wird? Ausgehend von der Nummerung einer Formel (wie in Abb. 2.6 gezeigt), ergibt sich dann die Textmarke **Formel_6_1**, s. Abb. 3.4.

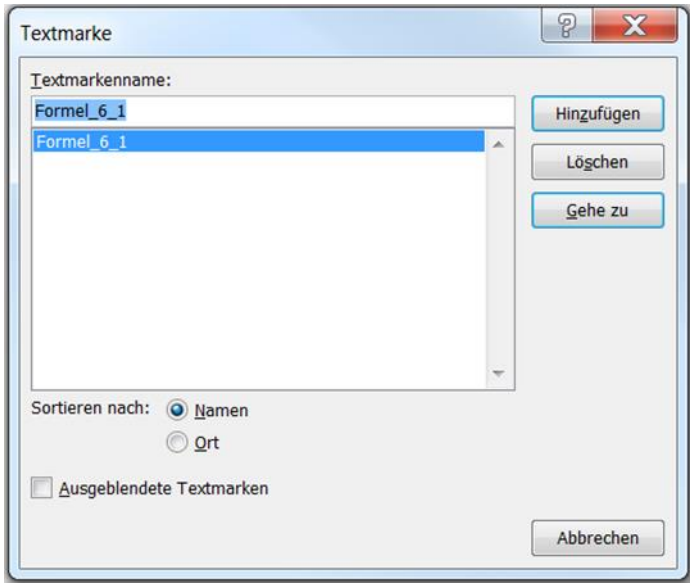


Abb. 3.4: Textmarke für die erste Formel im sechsten Kapitel

Mit der Navigation *Verweise* ▶ *Beschriftungen* ▶ *Querverweis* kann dann der Verweistyp *„Textmarke“* gewählt werden, um die erste Formel im sechsten Kapitel zu referenzieren, also (6.1). Vgl. dazu Abb. 3.5.

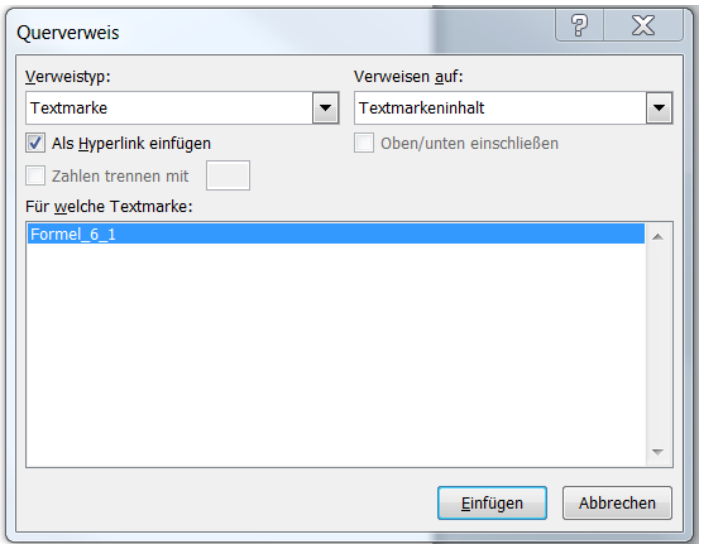


Abb. 3.5: Querverweis auf eine benummerte Formel mit definierter Textmarke

© Copyright Dr. V. Thormählen, 2016

3.6 Aufruf des Hauptprogramm

Um das Hauptprogramm (s. Listing 3.1) bequem starten zu können, kann eine Schaltfläche in die *„Symbolleiste für den Schnellzugriff“* eingebaut werden, s. Abb. 3.6.

- Navigation: *Datei* ▶ *Optionen* ▶ *Symbolleiste für den Schnellzugriff*
- 1. Spalte, oben: "*Befehle auswählen*": Makros
- 1. Spalte, unten: Name des aufzurufenden Hauptprogramms (*InsertNumberedEquations*) auswählen.
- Auf die Schaltfläche *„Hinzufügen“* klicken.
- 2. Spalte, oben: "*Symbolleiste für den Schnellzugriff anpassen*": Für alle Dokumente (Standard)
- Dann ggf. mit der Schaltfläche „*Ändern*" ein passenderes Symbol auswählen.

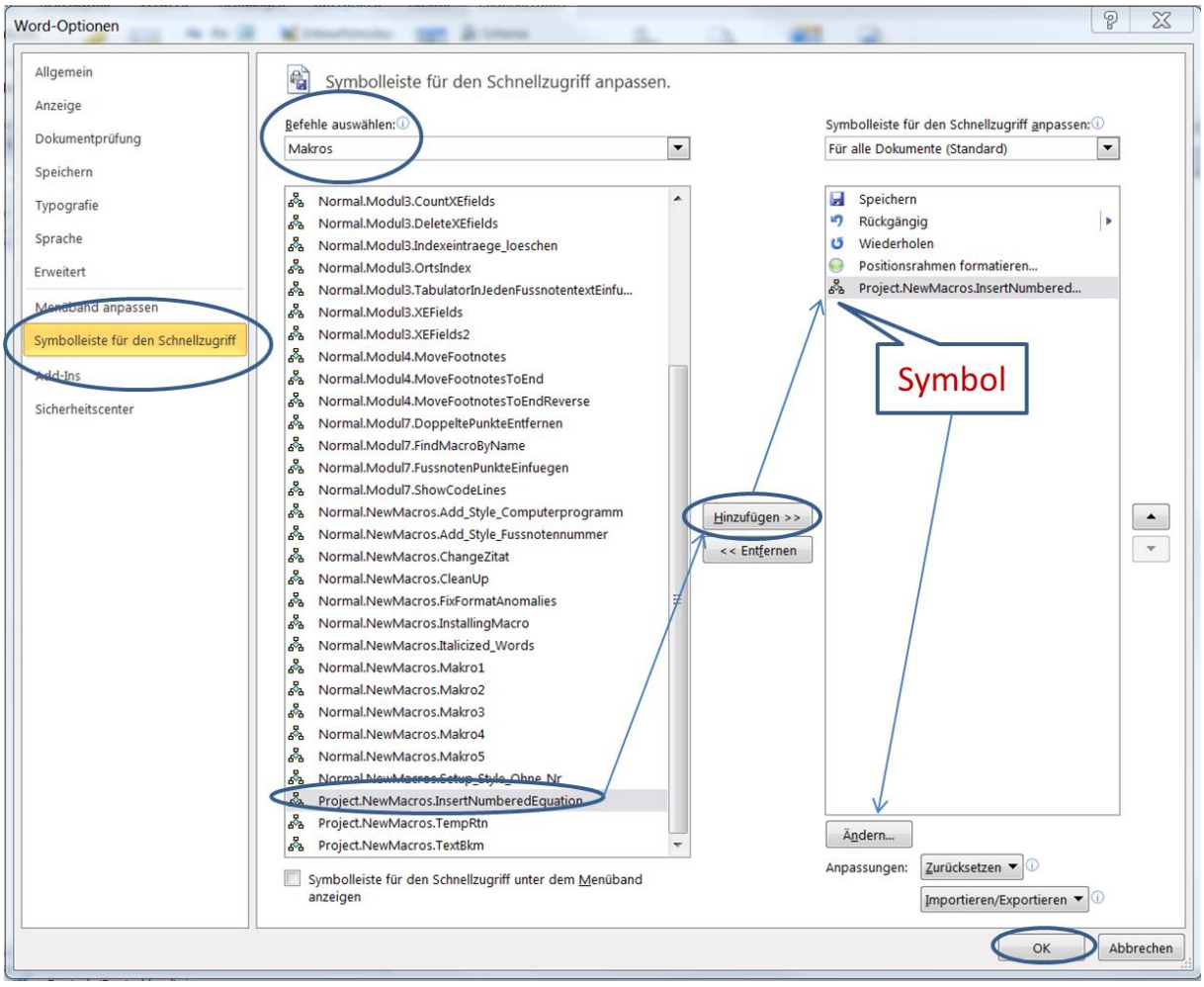


Abb. 3.6: Aufruf des Hauptprogramms via *„Symbolleiste für den Schnellzugriff“*

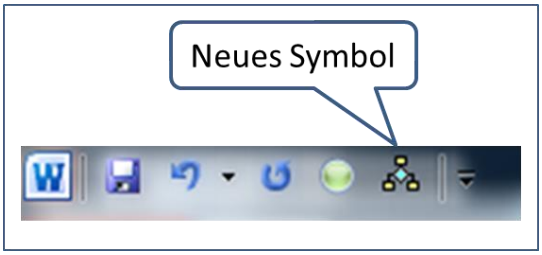


Abb. 3.7: Symbol zum Aufruf des Hauptprogramms in der *„Symbolleiste für den Schnellzugriff“*

© Copyright Dr. V. Thormählen, 2016

Literaturverzeichnis

[1] V. Thormählen, „Mathematische Formeln in Word. Der Formel-Editor in Word 2010.," 24 03 2014. [Online]. Available: http://dr-thormaehlen.de/WORD/Formel_Editor.pdf. [Zugriff am 15 10 2016].

[2] V. Thormählen, „Makros erstellen und aufrufen," 11 11 2014. [Online]. Available: http://dr-thormaehlen.de/WORD/Makros_erstellen_und_aufrufen.pdf. [Zugriff am 15 10 2016].

[3] S. Franklin, „How to Insert Equations Numbers in Word 2016," 22 02 2016. [Online]. Available: https://www.youtube.com/watch?v=9YGTH4WrY_8. [Zugriff am 14 10 2016].

© Copyright Dr. V. Thormählen, 2016