

# Unpaarige Klammern in einem WORD-Dokument finden

Volker Thormählen, 31. Mai 2021

## Inhaltsverzeichnis

1	Problem: Unpaarige Klammern im Zieldokument .....	2
2	Lösung .....	2
2.1	Zieldokument auswählen .....	2
2.2	Unpaarige Klammern im Zieldokument kennzeichnen .....	2
2.3	Mögliche Fehlermeldungen im Zieldokument entfernen .....	2
3	VBA-Quellcode der Prozeduren.....	2
3.1	Subprozedur „ <b>Driver</b> “ .....	2
3.2	Funktionsprozedur „ <b>GetTargetDocument</b> “ .....	3
3.3	Subprozedur „ <b>CheckPairedBrackets</b> “ .....	4
3.4	Funktionsprozedur „ <b>CountChars</b> “ .....	5
3.5	Subprozedur „ <b>RemoveErrorHints</b> “ .....	6
4	Quellen.....	6

# 1 Problem: Unpaarige Klammern im Zieldokument

Für drei Typen von Klammern (runde, eckige oder geschweifte) soll in einem WORD-Dokument jeweils vermerkt werden, ob sie in einem bestimmten Absatz des betreffenden WORD-Dokuments paarig vorkommen oder nicht. Bei Unpaarigkeit soll über dem jeweiligen Absatz folgender Fehlerhinweis eingefügt werden:

- > Fehler: Unpaarige runde Klammern im nächsten Absatz gefunden! <---
- > Fehler: Unpaarige eckige Klammern im nächsten Absatz gefunden! <---
- > Fehler: Unpaarige geschweifte Klammern im nächsten Absatz gefunden! <---

## 2 Lösung

### 2.1 Zieldokument auswählen

Zur interaktiven Auswahl des betreffenden WORD-Dokuments (sog. *Zieldokument*) dienen die VBA-Prozeduren mit dem Namen **Driver** bzw. **GetTargetDocument**.

### 2.2 Unpaarige Klammern im Zieldokument kennzeichnen

Das Problem kann bequem mit zwei VBA-Prozeduren gelöst werden:

- Subprozedur mit dem Namen **CheckPairedBrackets**
- Funktionsprozedur mit dem Namen **CountChars**

### 2.3 Mögliche Fehlermeldungen im Zieldokument entfernen

Zum automatischen Entfernen möglicher Fehlermeldungen im betreffenden WORD-Dokument kann die Subprozedur mit dem Namen **RemoveErrorHints** eingesetzt werden.

## 3 VBA-Quellcode der Prozeduren

### 3.1 Subprozedur „Driver“

```
Sub Driver()  
    ' Treiber-Prozedur  
    Dim docTgt As Document  
    Dim strFullPath As String  
    On Error GoTo ErrorHandler  
    strFullPath = GetTargetDocument  
    If Len(strFullPath) > 0 Then  
        Set docTgt = Documents.Open(FileName:=strFullPath)  
        CheckPairedBrackets  
  
        docTgt.Close SaveChanges:=wdSaveChanges, OriginalFormat:=wdOriginalDocumentFormat  
    Else  
        MsgBox "Abbruch durch den Benutzer. Nichts ausgewählt!", vbExclamation, "Driver"  
    End If  
ExitPoint:  
    On Error Resume Next  
    Exit Sub  
ErrorHandler:  
    If Err.Number = 4198 Then  
        MsgBox "Dokument wurde NICHT geschlossen!"  
    Else  
        MsgBox "Laufzeitfehler: " & Err.Number & ", ", Err.Description, "Zieldokument auswählen."  
    End If  
    Resume ExitPoint  
End Sub
```

### 3.2 Funktionsprozedur „GetTargetDocument“

```
Function GetTargetDocument() As String
    ' Aufgabe: Zieldokument via Benutzerdialog auswählen.
    Const conMsg As String = "WORD-Dokument auswählen!"
    Dim fd As Office.FileDialog
    Dim varSelected As Variant
    Set fd = Application.FileDialog(msoFileDialogFilePicker)
    With fd
        .Title = conMsg
        .Filters.Clear
        .Filters.Add "Documents", "*.doc; *.docx; *.docm", 1
        .AllowMultiSelect = False
        .InitialFileName = "C:\Users\...\OneDrive\Dokumente"
        If .Show = True Then
            varSelected = .SelectedItems(1)
        Else
            varSelected = ""
        End If
    End With
    GetTargetDocument = varSelected
    Set fd = Nothing
End Function
```

### 3.3 Subprozedur „CheckPairedBrackets“

```
Sub CheckPairedBrackets()
    In Anlehnung an: [1]
    ' Lokale Variablen definieren
    Dim bolCheckP() As Boolean
    Dim intParaCnt As Integer, intPara As Integer, intLeft As Integer, intRight As Integer, intHits As Integer
    Dim strOpenP As String, strCloseP As String, strPara As String, strReport As String, strBracket As String
    ' Inputbox definieren
    Dim strMsg As String, strTitle As String, strDefault As String, strBracketType As String
    strTitle = "Klammerpaare prüfen." & vbCrLf & "Bitte den Klammertyp (1, 2 oder 3) erfassen:"
    strMsg = "1: rund" & vbCrLf & "2: eckig" & vbCrLf & "3: geschweift"
    strDefault = "1"
    NeuerVersuch:
    strBracketType = InputBox(strMsg, strTitle, strDefault)
    If IsNumeric(strBracketType) And Len(strBracketType) > 0 Then
        Select Case strBracketType
            Case "1"
                strOpenP = "(" ' öffnende runde Klammer
                strCloseP = ")" ' schließende runde Klammer
                strReport = "----> Fehler: Unpaarige runde Klammern im nächsten Absatz gefunden! <---"
            Case "2"
                strOpenP = "[" ' öffnende eckige Klammer
                strCloseP = "]" ' schließende eckige Klammer
                strReport = "----> Fehler: Unpaarige eckige Klammern im nächsten Absatz gefunden! <---"
            Case "3"
                strOpenP = "{" ' öffnende geschweifte Klammer
                strCloseP = "}" ' schließende geschweifte Klammer
                strReport = "----> Fehler: Unpaarige geschweifte Klammern im nächsten Absatz gefunden! <---"
            Case Else
                MsgBox "Falsche Eingabe: Nur Klammertyp 1, 2 o. 3", vbExclamation
                If MsgBox("Neuer Versuch?", vbYesNo) = vbYes Then
                    GoTo NeuerVersuch
                Else
                    MsgBox "Abbruch durch den Benutzer!", vbExclamation, "Klammertyp erfassen."
                    Exit Sub
                End If
            End Select
        Else
            MsgBox "'Abbrechen' gewählt d. Benutzer!", vbExclamation, "Klammertyp erfassen."
            Exit Sub
        End If
        intParaCnt = ActiveDocument.Paragraphs.Count
        ReDim bolCheckP(intParaCnt)
        For intPara = 1 To intParaCnt
            bolCheckP(intPara) = False
            strPara = ActiveDocument.Paragraphs(intPara).Range.Text
            If Len(strPara) <> 0 Then
                intLeft = CountChars(strPara, strOpenP)
                intRight = CountChars(strPara, strCloseP)
                If intLeft <> intRight Then bolCheckP(intPara) = True
            End If
        Next intPara
        For intPara = intParaCnt To 1 Step -1
            If bolCheckP(intPara) Then
                With Selection
                    .HomeKey Unit:=wdStory, Extend:=wdMove
                    If intPara > 1 Then
                        .MoveDown Unit:=wdParagraph, Count:=(intPara - 1), Extend:=wdMove
                    End If
                    ' Fehlerhinweis vor Absatz einfügen
                    .InsertParagraphBefore
                    .MoveLeft Unit:=wdCharacter, Count:=1
                    .Style = ActiveDocument.Styles(wdStyleNormal)
                    .Font.Color = wdColorRed
                    .TypeText Text:=strReport
                    intHits = intHits + 1
                End With
            End If
        Next intPara
        Select Case strBracketType
            Case "1": strBracket = "runde Klammern"
            Case "2": strBracket = "eckige Klammern"
            Case "3": strBracket = "geschweifte Klammern"
```

```
End Select
MsgBox "Fertig! " & CStr(intHits) & " Vorkommen. (Klammertyp = " & strBracket & ")", vbExclamation,
"Klammerpaare prüfen."
End Sub
```

### 3.4 Funktionsprozedur „CountChars“

```
Private Function CountChars(strA As String, strC As String) As Integer
Quelle: [1]
Dim intCnt As Integer, intFound As Integer
intCnt = 0
intFound = InStr(strA, strC)
Do While intFound <> 0
    intCnt = intCnt + 1
    intFound = InStr(intFound + 1, strA, strC)
Loop
CountChars = intCnt
End Function
```

### 3.5 Subprozedur „RemoveErrorHints“

```
Sub RemoveErrorHints()  
    ' Aufgabe: Alle roten Fehlerhinweise automatisch aus dem Zieldokument entfernen.  
    Dim lngCnt As Long          ' Fehlerzähler  
    Dim para As Paragraph      ' Absatz im aktuellen Dokument (Zieldokument)  
    For Each para In ActiveDocument.Paragraphs  
        With para  
            If .Range.Sentences.Count = 1 Then  
                If .Range.Sentences(1).Font.Color = vbRed And _  
                    Left(.Range.Sentences(1), 7) = "Fehler:" Then  
                    lngCnt = lngCnt + 1  
                    .Range.Delete  
                End If  
            End If  
        End With  
    Next para  
    MsgBox CStr(lngCnt) & " rote(n) Fehlerhinweis(e) entfernt!", _  
        vbInformation, "Rote Fehlerhinweise entfernen."  
End Sub
```

## 4 Quellen

- [1] A. Wyatt, „Überprüfen auf passende Klammern“, 2019 11 2019. [Online]. Available: [https://word.tips.net/T001308\\_Checking\\_for\\_Matching\\_Parentheses.html](https://word.tips.net/T001308_Checking_for_Matching_Parentheses.html),. [Zugriff am 25 05 2021].