Fortschrittsbalken im Benutzerformular dargestellt am Beispiel der Füllwort-Analyse eines WORD-Dokuments

Inhaltsverzeichnis

1 Aufgabe 2

2 Aufbau 2

3 Befehlsschaltfläche zum Starten 3

4 Benutzerformular (UserForm) 3

4.1 Steuerelemente im Benutzerformular 4

4.1.1 Rahmen (fraProgress) 4

4.1.2 Bezeichnungsfeld (lblProgress) 4

4.1.3 Bezeichnungsfeld (lblStatus) 4

4.2 Ereignisprozeduren 5

5 Standardprozeduren 6

5.1 Hauptprogramm 6

5.2 Existenz einer WORD-Datei prüfen 9

5.3 Inhalt einer Tabellenzelle bestimmen 9

5.4 Horizontalen Fortschrittbalken aktualisieren 10

5.5 Output-Tabelle formatieren und beschriften 11

5.6 Benutzerformular abschließend aktualisieren 12

6 Output-Datei 12

7 Resümee 12

8 Abbildungen 13

9 Listings 13

10 Tabellen 13

11 Literatur 13

1 Aufgabe

Im Folgenden wird eine WORD-Anwendung beschrieben, die tabellarisch vorgegebene Füllwörter[1] in einem WORD-Dokument findet, markiert, zählt und zusätzlich auflistet. Ein Benutzerformular mit einem horizontalen Fortschrittsbalken (engl. progress bar) zeigt dabei den Stand der Verarbeitungsschritte an. Das ist nützlich und benutzerfreundlich bei lange laufenden VBA[2]-Prozeduren (sog. VBA-Makros[3]).

Die Anwendung benötigt vier WORD-Dateien:

1. *Input*: Die vorliegende Datei mit allen benötigten VBA-Prozeduren und einem Benutzerformular mit einem horizontalen Fortschrittsbalken: **ProgessBar.docm**
2. *Input*: Eine Datei mit der Tabelle der Füllwörter: **FillerTable.docx**
3. *Update*: Eine Datei mit dem zu untersuchenden Inhalt: **SampleText.docx**
4. *Output*: Eine Datei mit den Untersuchungsergebnissen: **FillerResult.docx**

In der Datei **SampleText.docx** werden die im Text gefundenen Füllwörter türkis hervorgehoben. Sonst wird nichts verändert.

Zusätzlich werden die Untersuchungsergebnisse in der Datei **FillerResult.docx** tabellarisch in 3 Spalten dargestellt:

1. Gefundenes Füllwort
2. Häufigkeit des Vorkommens im untersuchten Text
3. Liste der Fundstelle(n) im analysierten Text mit Seitenangabe(n)

Die Anwendung wird mit dieser Befehlsschaltfläche gestartet:

2 Aufbau

Die Anwendung benötigt die ersten drei der oben genannten WORD-Dateien. Die vierte wird während des Programmablaufs neu erstellt, falls sie nicht bereits vorhanden ist. Benötigt wird außerdem ein Benutzerformular („*Userform1*“) und ein Modul („*Modul1*“) (siehe Abb. 1).

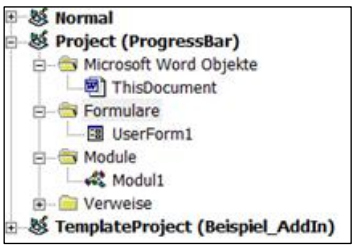

Abb. 1: Aufbau des Projekts 'ProgressBar'

In Abb. 1 repräsentiert „*ThisDocument*“ das vorliegende Word-Dokument.

---

[1] Füllwörter sind überflüssig, wenn sie nichts zum Verständnis eines Textes beitragen.

[2] VBA ist ein Kürzel für ***V**isual **B**asic for **A**pplications*, eine zu den Microsoft-Office-Programmen gehörende Skriptsprache.

[3] Ein VBA-Makro enthält eine Folge von Anweisungen und Deklarationen, die mit einem einfachen Aufruf ausgeführt werden können.

3 Befehlsschaltfläche zum Starten

Mit der Navigation[4] *Entwicklertools* ▶ *Steuerelemente* wird in *ThisDocument* (siehe Abb. 1) die bereits gezeigte Befehlsschaltfläche mit der Beschriftung *„Start Füllwörter“* erstellt. Beim Anklicken wird folgende VBA-Prozedur ausgeführt (siehe Listing 1).

```
Option Explicit

Private Sub cmdRun_Click()
    ' Benutzerformular laden
    Load UserForm1
    ' Benutzerformular anzeigen
    UserForm1.Show
End Sub
```

Listing 1: VBA-Code zum Laden und Anzeigen des Benutzerformulars mit dem Name *UserForm1*

4 Benutzerformular (UserForm)

Ausgehend von einem neuen Projekt (d. h. der Erstellung eines neuen WORD-Dokuments) kann mit der Navigation *Entwicklertools* ▶ Visual Basic ▶ *Einfügen* ▶ *Userform* ein neues leeres Benutzerformular erzeugt werden. Es wird dort in der Titelleiste *Fortschrittsbalken* genannt und mit drei Steuerelementen bestückt (siehe

Abb. 2).

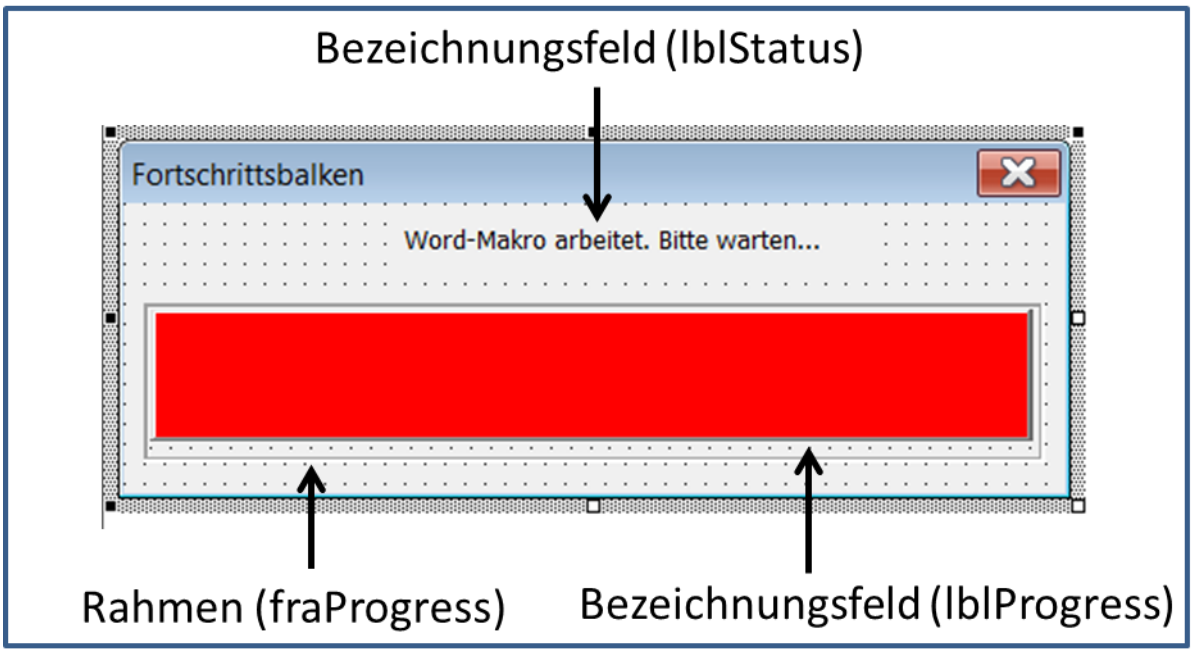

Abb. 2: Steuerelemente im Benutzerformular mit dem Namen *UserForm1*

[4] In WORD kann der Navigationsbereich (d. h. die sog. Multifunktionsleiste) eingeblendet und genutzt werden, um sich schneller und einfacher innerhalb eines Dokuments zu bewegen.

4.1 Steuerelemente im Benutzerformular

4.1.1 Rahmen (fraProgress)

Die Eigenschaften des Rahmens lauten:

- Caption - leer
- Height = 48
- Left = 6
- Top = 30
- Width = 270

4.1.2 Bezeichnungsfeld (lblProgress)

Die Eigenschaften sind wie folgt festgelegt:

- Caption – leer
- BackColor – rot, ggf. blau
- Height = 40
- Left = 0
- Top = 0
- Width = 265

4.1.3 Bezeichnungsfeld (lblStatus)

- Caption = “Word-Makro arbeitet. Bitte warten …“ (anfänglich, später Meldung des Programmendes)
- Height = 20
- Left =85
- Top =6
- Width = 145

4.2 Ereignisprozeduren

Zum Benutzerformular gehören die folgenden drei Ereignisprozeduren (siehe Listing 2).

```
Option Explicit

Private Sub UserForm_Initialize()
    ' Die Breite des Fortschrittsbalkens zurücksetzen
    Me.lblProgress.Width = 0
End Sub

Private Sub UserForm_Activate()
    ' Das Hauptprogramm aufrufen
    Call Fuellwoerter
    ' Das Benutzerformular entladen
    Unload Me
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    ' Das Schließen des Benutzerformulars mit dem roten Kreuz verhindern
    If CloseMode = vbFormControlMenu Then Cancel = True
End Sub
```

Listing 2: Ereignisprozeduren des Benutzerformulars

Die Ereignisprozedur *UserForm_Activate* ruft das Hauptprogramm *Fuellwoerter* auf, das sich im *Modul1 befindet* (siehe Abb. 1). Es enthält auch alle anderen VBA-Standardprozeduren (siehe Seite 6 ff).

5 Standardprozeduren

5.1 Hauptprogramm

Das Makros mit dem Namen *Fuellwoerter* hat die Aufgabe, alle Füllwörter in einem vorbestimmten WORD-Dokument zu finden, zu markieren, zu zählen und in einem weiteren Dokument aufzulisten (siehe Listing 3). In diesem Listing sind Aufrufe von Routinen gelb markier, Kommentare grau.

```
Sub Fuellwoerter()
    ' Aufgabe: Füllwörter in einem Word-Dokument finden, markieren, zählen
    ' und zusätzlich auflisten.
    ' Aufruf durch: UserForm_Activate
    ' Ruft folgende Routinen auf:
    ' IsWordDoc             - Existenzprüfung einer Word-Datei
    ' UpdateProgressBar     - Fortschrittsbalken im Benutzerformular aktalisieren
    ' FormatOutputTable     - Output-Tabelle formatieren und beschriften
    ' TerminateProgressbar - Benutzerformular abschließend aktualisieren
    ' Word-Dokumente:
    Dim docThis   As Document  ' aktuelles Dokument (mit VBA-Code)
    Dim docFiller As Document  ' Dokument mit Füllwörtern in 2-spaltiger Tabelle
    Dim docSample As Document  ' Dokument mit Beispieltext, der analysiert werden soll
    Dim docTgt    As Document  ' Dokument mit den Ergebnissen der Analyse
    ' Lokale Variablen:
    Dim lngRow    As Integer   ' Zeilenzähler für Tabelle 'tblInp'
    Dim lngRows   As Long      ' Zahl der Zeilen in Tabelle 'tblInp'
    Dim lngFiller As Long      ' Zahl der Füllwörter in Tabelle 'tblInp'
    Dim objCell   As Cell      ' Zelle in Ziel-Tabelle
    Dim strCell   As String    ' Inhalt einer Zelle in Tabelle 'tblInp'
    Dim strFileNm As String    ' Name einer Word-Datei
    Dim strMsg    As String    ' Meldung
    Dim strPgNbr  As String    ' Seitennummer
    Dim tblInp    As Table     ' Tabelle in 'docFiller'
    Dim tblTgt    As Table     ' Tabelle in 'docTgt'

    On Error GoTo Err_Point
    Application.ScreenUpdating = False
    Set docThis = ActiveDocument  ' aktuelles Dokument

    ' Vorhandene Word-Datei mit der Tabelle der Füllwörter öffnen
    strFileNm = "FillerTable.docx"
    If IsWordDoc(strFileNm) Then
       Set docFiller = Documents.Open(FileName:=strFileNm, _
          AddToRecentFiles:=False, ReadOnly:=True, Visible:=False)
       With docFiller
          If .Tables.Count > 0 Then
             Set tblInp = .Tables(1)
          Else
             MsgBox "Keine Tabelle mit Füllwörtern im Dokument '" & strFileNm & _
                "' gefunden.", vbCritical, "Fuellwoerter"
             GoTo Exit_Point
          End If
       End With
    Else
       MsgBox "Die Word-Datei '" & strFileNm & "' wurde nicht gefunden.", _
          vbCritical, "Fuellwoerter"
       GoTo Exit_Point
    End If

    ' Vorhandene Word-Datei mit Beispiel-Text öffnen, der analysiert werden soll.
```

```
strFileNm = "SampleText.docx"
If IsWordDoc(strFileNm) Then
   Set docSample = Documents.Open(FileName:=strFileNm, _
      AddToRecentFiles:=False, Visible:=True)
   If docSample.Characters.Count < 2 Then
      MsgBox "Die Word-Datei '" & strFileNm & "' ist leer.", vbCritical, "Fuellwoerter"
      GoTo Exit_Point
   End If
   ' Hervorhebungen in diesem Dokument entfernen, falls bereits vorhanden
   docSample.Range.HighlightColorIndex = wdNoHighlight
Else
   MsgBox "Die Word-Datei '" & strFileNm & "' wurde nicht gefunden.", _
      vbCritical, "Fuellwoerter"
   GoTo Exit_Point
End If

strFileNm = "FillerResult.docx"
If IsWordDoc(strFileNm) Then
   Set docTgt = Documents.Open(FileName:=strFileNm, AddToRecentFiles:=False, _
      Visible:=True)
   docTgt.Content.Delete   ' alles löschen
Else
   Set docTgt = Documents.Add(NewTemplate:=False, DocumentType:=wdNewBlankDocument)
   docTgt.SaveAs FileName:=strFileNm
End If

If Documents.Count <> 4 Then
   MsgBox Prompt:="Genau vier Word-Dokumente müssen geöffnet sein!", _
      Buttons:=vbCritical, Title:="Laufzeitfehler"
   GoTo Exit_Point
End If

docFiller.Activate
lngRows = tblInp.Rows.Count

' Füllwörter in 'docSample' suchen und zählen, die in 'tblInp' enthalten sind.
For lngRow = 1 To lngRows
   ' Zellinhalt bestimmen
   strCell = GetCellContent(tblInp.Cell(lngRow, 1))
   lngFiller = 0
   strPgNbr = ""
   docSample.StoryRanges(wdMainTextStory).Select
   With Selection.Find
      Do While .Execute(FindText:=strCell, Forward:=True, MatchWholeWord:=True, _
           MatchWildcards:=False, Wrap:=wdFindStop) = True
         lngFiller = lngFiller + 1
         Selection.Range.HighlightColorIndex = wdTurquoise
         strPgNbr = strPgNbr & CStr(Selection.Information(wdActiveEndPageNumber)) & ","
         Selection.Collapse Direction:=wdCollapseEnd
      Loop
   End With

   ' Ausgabe der Ergebnisse
   If lngFiller > 0 Then
      If Len(strPgNbr) > 0 Then
         strPgNbr = Left(strPgNbr, Len(strPgNbr) - 1)
      End If
      strMsg = strCell & vbTab & CStr(lngFiller) & vbTab & strPgNbr & vbCr
      docTgt.Select
      With Selection
```

```
            .EndKey unit:=wdStory, Extend:=wdMove
            .InsertAfter Text:=strMsg
         End With
      End If
      docFiller.Activate
      ' Fortschrittsbalken im Benutzerformular aktalisieren
      Call UpdateProgressBar(lngRow, lngRows)
   Next lngRow

   ' Untersuchungs-Ergebnisse in Tabelle umwandeln
   docTgt.Select
   With Selection
      .Collapse Direction:=wdCollapseStart
      .InsertAfter Text:="Füllwort" & vbTab & "Häufigkeit" & vbTab & "Seite(n)" & vbCr
      .WholeStory
      .Range.ConvertToTable _
         Separator:=wdSeparateByTabs, _
         NumRows:=Selection.Paragraphs.Count, _
         NumColumns:=3, _
         Format:=wdTableFormatNone, AutoFit:=True, _
         ApplyBorders:=True
   End With
   ' Output-Tabelle formatieren und beschriften
   Call FormatOutputTable(docTgt)
   docTgt.Windows(1).View.Type = wdPrintView
   ' Füllwörter: Änderungen nicht speichern
   docFiller.Close SaveChanges:=wdDoNotSaveChanges
   ' Beispieltext: Änderungen speichern
   docSample.Close SaveChanges:=wdSaveChanges
   ' Benutzerformular abschließend aktualisieren
   Call TerminateProgressbar
   MsgBox "Erledit", vbExclamation, "Füllwörter"
Exit_Point:
   On Error Resume Next
   Application.ScreenUpdating = True
   Set docFiller = Nothing
   Set docSample = Nothing
   docThis.Close SaveChanges:=wdSaveChanges  ' Änderungen speichern
   Set docThis = Nothing
   Exit Sub
Err_Point:
   Resume Exit_Point
   MsgBox "Laufzeitfehler # " & Err.Number & ", " & Err.Description & ", " & "Fuellwoerter"
End Sub
```

Listing 3: Hauptprogramm (Aufrufe von anderen Makros sind gelb markiert)

5.2 Existenz einer WORD-Datei prüfen

Im Hauptprogramm werden 3 WORD-Dateien geöffnet. Die jeweilige Existenz wird vorher mit folgender Funktion geprüft (siehe Listing 4).

```
Function IsWordDoc(strFileNm As String) As Boolean
   ' Aufgabe: Existenzprüfung einer Word-Datei
   ' Argument: strFileNm - Name einer Word-Datei mit dem Zusatz 'docx'
   Dim strFullpath As String  ' vollständiger Dateipfad
   Dim strPath     As String  ' aktueller Dateipfad
   strPath = ThisDocument.Path
   strFullpath = strPath & Application.PathSeparator & strFileNm
   If Len(Dir(strFullpath)) > 0 And Right(strFileNm, 4) = "docx" Then
      IsWordDoc = True
   Else
      IsWordDoc = False
   End If
End Function
```

Listing 4: Existenzprüfung einer WORD-Datei mit dem Dateizusatz „docx“ prüfen

5.3 Inhalt einer Tabellenzelle bestimmen

Die folgende Funktion bestimmt den Inhalt der jeweils aktuellen Zelle in der Tabelle der vorgebebenen Füllwörter (siehe Listing 5).

```
Function GetCellContent(ByRef objCell As Cell) As String
   ' Aufgabe: Zellinhalt bestimmen
   Dim objRng As Range
   Set objRng = objCell.Range
   objRng.End = objRng.End - 1
   GetCellContent = objRng.Text
   Set objRng = Nothing
End Function
```

Listing 5: Funktion zur Bestimmung des Inhalts der übergebenen Tabellenzelle

5.4 Horizontalen Fortschrittbalken aktualisieren

Im Hauptprogramm wird die Prozedur *UpdateProgressBar* aufgerufen (siehe Listing 6). Sie bewirkt:

- Die Verbreiterung des horizontalen Fortschrittbalkens im Benutzerformular
- Die zahlenmäßige Aktualisierung der Titelleiste des Benutzerformulars
- Die prozentuale Anzeige des aktuellen Fortschritts im Bezeichnungsfeld des Rahmen-Steuerelements im Benutzerformular.

Diese Prozedur ist allgemeingültig und kann deshalb auch von anderen einschlägigen Hauptprogrammen unverändert aufgerufen werden.

```
Sub UpdateProgressBar(ByVal lngRow As Long, ByVal lngRows As Long)
   ' Aufgabe: Fortschrittsbalken im Benutzerformular aktualiseren
   ' Argumente:   lngRow   - Scheifenzähler im Hauptprogramm
   '              lngRows  - Obergrenze des Scheifenzählers im Hauptprogramm
   Dim sngPctDone As Single   ' Fertigstellung in Prozent
   sngPctDone = lngRow / lngRows
   With UserForm1
      .Repaint
      ' Die Titelleiste der Benutzerformulars aktualisieren
      .Caption = "Fortschrittsbalken. Füllwort # " & CStr(lngRow) & " von " & CStr(lngRows)
      ' Die Breite des Bezeichnungsfeld-Steuerelements vergrößern
      .lblProgress.Width = sngPctDone * (.fraProgress.Width - 10)
      With .fraProgress
         ' Die Titelleiste des Rahmen-Steuerelements aktualisieren
         .Caption = Format(sngPctDone, "0%")
         ' Den Fortschrittsbalken aktualisieren
         .Repaint
      End With
   End With
End Sub
```

Listing 6: Benutzerformular schrittweise aktualisieren

5.5 Output-Tabelle formatieren und beschriften

Die Formatierung und Beschriftung der Tabelle mit den gefundenen Füllwörtern in der Outputdatei erfolgt mit folgender VBA-Prozedur (siehe Listing 7).

```
Sub FormatOutputTable(ByRef docTgt AsDocument)
   ' Aufgabe: Output-Tabelle formatieren und beschriften
   ' Argument: tblTgt - Output-Datei
   Dim tblNew  As Table ' Output-Tabelle
   Dim objCell As Cell  ' Zelle
   Dim lngRow  As Long  ' Zeilenzähler
   Set tblNew = docTgt.Tables(1)
   With tblNew
      ' Tabellenzeilen zentrieren
      .Rows.Alignment = wdAlignRowCenter
      ' Formatvorlage für die angegebene Tabelle festlegen
      If .Style <> "Tabellenraster" Then
         .Style = "Tabellenraster"
      End If
      ' Die erste Tabellenzeile ...
      With .Rows(1)
         .HeadingFormat = True                          ' als Kopfzeile formatieren
         With .Range
            With .Shading
               .BackgroundPatternColor = wdColorGray25   ' hellgrau schattieren
            End With
            .Font.Bold = True                           ' fett drucken
         End With
      End With
      ' Schriftgrad der restlichen Tabellenzeilen festlegen
      For lngRow = 2 To .Rows.Count
         For Each objCell In .Rows(lngRow).Cells
            objCell.Range.Font.Size = 10                ' 10 Punkte
         Next objCell
      Next lngRow
      ' Den Inhalt der zweiten Tabellenspalte horizontal und vertikal zentrieren
      For Each objCell In .Columns(2).Cells
         With objCell
            .Range.ParagraphFormat.Alignment = wdAlignParagraphCenter
            .VerticalAlignment = wdCellAlignVerticalCenter
         End With
      Next objCell
      ' Breite der drei Tabellenspalten prozentual festlegen
      .Columns.PreferredWidthType = wdPreferredWidthPercent
      .Columns(1).PreferredWidth = 20
      .Columns(2).PreferredWidth = 15
      .Columns(3).PreferredWidth = 65
      ' Output-Tabelle beschriften
      .Range.Cells(.Range.Cells.Count).Select
      With Selection
         .EndKey Unit:=wdStory
         .InsertRowsAbove
         .Cells.Merge
         .ParagraphFormat.Alignment = wdAlignParagraphCenter
         .TypeText Text:="Vollständiger Dateiname: " & docTgt.FullName
      End With
   End With
End Sub
```

Listing 7: Tabelle in der Outputdatei formatieren und beschriften

5.6 Benutzerformular abschließend aktualisieren

Am Ende des Hauptprograms (siehe Listing 3) wird das Benutzerformular mit folgender Prozedur abgeschlossen (siehe Listing 8):

```
Sub TerminateProgressbar()
   ' Aufgabe: Benutzerformular abschließend aktualisieren
   With UserForm1
      .Caption = "Füllwörter"
      .lblStatus.Caption = "Normales Programmende!"
   End With
End Sub
```

Listing 8: Benutzerformular abschließend aktualisieren

6 Output-Datei

Die im Rahmen der Prozedur *Fuellwoerter* (siehe Listing 3) erstellte Output-Tabelle in der WORD-Datei mit dem Namen *FillerResult.docx* (vgl. Seite 2) sieht beispielsweise so aus (siehe Tabelle 1).

Füllwort	Häufigkeit	Seite(n)
abermals	2	12,12
auch	3	5,10,12
außerdem	2	2,12
bereits	4	2,3,7,12
dabei	2	2,12
dafür	2	12,12
deshalb	1	10
so	2	12,12
sonst	2	2,12
Vollständiger Dateiname: C:\Users\volker\Documents\Word_Dokumente\FillerResult.docx		

Tabelle 1: Beispiel für gefundene Füllwörter

Die Fundstellen aller Füllwörter in ersten Spalte der Output-Tabelle werden im zu untersuchenden WORD-Dokument (hier: *SampleText.docx*) türkis hervorgeben. Voraussetzung dafür ist, dass u. a. „*abermals*“ in der Tabelle der Füllwörter enthalten ist. Das entsprechende Input-Dokument (hier: *FillerTable.docx*) enthält zurzeit insgesamt 220 Tabelleneinträge. Es kann beliebig ergänzt werden.

Die Anzeige eines Fortschrittsbalkens bietet sich an bei erwarteter langer Laufzeit verursacht durch …

- die große Zahl der tabellarisch vorgegebenen Füllwörter,
- die Größe der zu untersuchenden WORD-Datei.

7 Resümee

Die in diesem Beitrag gezeigte Lösung für einen horizontalen Fortschrittsbalken lässt sich wegen des strukturierten, modularen und ausführlich kommentierten Absatzes ohne großen Aufwand auf andere sog. „*Langläufer*“ übertragen.

Zum Schluss noch ein Hinweis: Zur Aktualisierung des horizontalen Fortschrittsbalkens kann statt der Anweisung „*Repaint*“ (siehe Listing 6) die Anweisung „*DoEvents*“ benutzt werden, siehe beispielsweise [1]. In [2] wird aber davon abgeraten, weil „*DoEvents*“ angeblich sehr ressourcenhungrig ist.

8 Abbildungen

Abb. 1: Aufbau des Projekts 'ProgressBar' 2
Abb. 2: Steuerelemente im Benutzerformular mit dem Namen *UserForm1* 3

9 Listings

Listing 1: VBA-Code zum Laden und Anzeigen des Benutzerformulars mit dem Name *UserForm1* 3
Listing 2: Ereignisprozeduren des Benutzerformulars 5
Listing 3: Hauptprogramm (Aufrufe von anderen Makros sind gelb markiert) 8
Listing 4: Existenzprüfung einer WORD-Datei mit dem Dateizusatz „docx“ prüfen 9
Listing 5: Funktion zur Bestimmung des Inhalts der übergebenen Tabellenzelle 9
Listing 6: Benutzerformular schrittweise aktualisieren 10
Listing 7: Tabelle in der Outputdatei formatieren und beschriften 11
Listing 8: Benutzerformular abschließend aktualisieren 12

10 Tabellen

Tabelle 1: Beispiel für gefundene Füllwörter 12

11 Literatur

[1] o.V., „How to display a progress bar with user form in Excel,“ 01 02 2012. [Online]. Available: https://support.microsoft.com/en-us/kb/211736. [Zugriff am 05 01 2017].

[2] Ibby, „Implementing a Progress Bar in Word VBA,“ 23 07 2011. [Online]. Available: http://word.mvps.org/Faqs/Userforms/CreateAProgressBar.htm. [Zugriff am 03 01 2017].

[3] B. Hardley, „Füllwörter markieren mit WORD,“ 01 05 2011. [Online]. Available: http://www.bod.de/autorenpool/anleitung-fuellwoerter-markieren-mit-word-t12885.html. [Zugriff am 12 31 2016].