

OUTLOOK: Termine für jeden x-ten Arbeitstag generieren

Dr. Volker Thormählen, Okt./2019

Inhalt

Abbildungen	1
Tabellen.....	1
Quellcode	1
1 Aufgabenstellung	2
2 Globale Konstante und Variablen.....	2
3 Eingabedaten	2
3.1 Benutzerdialoge	2
3.2 Feiertage und Startdatum	3
3.3 Arbeitsfreie Wochentage.....	3
4 Ausgabedaten	4
5 Prozeduren	4
5.1 Hauptprozedur	4
5.2 Starttermin einer Terminserie bestimmen	5
5.3 Feiertage bestimmen.....	5
5.4 Arbeitsfreie Wochentage definieren	5
6 VBA-Quellcode	6
6.1 Globale Konstante und Variable	6
6.2 Hauptprozedur	7
6.3 Funktion 'Workday2'	9
6.4 Subroutine 'GetHolidays'.....	10
6.5 Subroutine 'GetCurrentItem'	11
7 Literaturverzeichnis.....	12

Abbildungen

Abbildung 1: Abstand in Arbeitstagen zwischen den Einzelterminen	2
Abbildung 2: Zahl der Einzeltermine in der Serie	2
Abbildung 3: Feiertage 2019 in NRW und Beginn (22.10.2019 08:00) f. eine Serie von Einzelterminen	3
Abbildung 4: Starttermin (22.10.2019 08:30) mit 5 Folgeterminen im Abstand von 3 Arbeitstagen	4
Abbildung 5: Fehlermeldung bei undefiniertem Anfangstermin	5

Tabellen

Tabelle 1: Schritte bzw. Vorgänge in der Hauptprozedur.....	4
Tabelle 2: Benötigte Subroutinen der Hauptprozedur	5

Quellcode

Listing 1: Globale Konstante und Variablen.....	6
Listing 2: Hauptprozedur 'CreateSeriesOfAppt', Quelle: In Anlehnung an [2]	8
Listing 3: Funktion 'Workday2', Quelle: In Anlehnung an [1]	9
Listing 4: Subroutine 'GetHolidays', Quelle: In Anlehnung an [2]	10
Listing 5: Subroutine 'GetCurrentItem', Quelle: In Anlehnung an [4]	11

1 Aufgabenstellung

Eine Einschränkung bei Outlook-Serienmustern ist, dass es standardmäßig kein Muster für jeden n-ten Arbeitstag (Mo. bis Fr.) gibt. Zwar können Serientermine für bestimmte Wochentage erstellen werden, aber das Muster für jeden n-ten Tag umfasst auch das Wochenende (Sa. und So.).

Im Folgenden wird eine VBA-Prozedur vorgestellt, bei der die oben genannte Einschränkung nicht gilt: Ausgehend von einem im OUTLOOK-Kalender angelegten Starttermin können beliebig viele Folgetermine¹ mit einem ausgewählten Abstand in Arbeitstagen erzeugt werden. Die Prozedur eignet sich also dafür, eine Terminserie für jeden n-ten Arbeitstag zu erstellen. Die Prozedur überspringt beim Zählen der Arbeitstage nicht nur das Wochenende (Sa. und So.), sondern auch sog. Wochenfeiertage. Wenn eine Woche weitere arbeitsfreie Tage aufweist, kann auch dies berücksichtigt werden.

2 Globale Konstante und Variablen

Listing 1 beinhaltet die globale(n) Konstante bzw. Variablen der Anwendung.

3 Eingabedaten

3.1 Benutzerdialoge

Die Anwendung benötigt 2 natürliche Zahlen als Eingaben des Benutzers im Dialog:

- Zahl der Arbeitstage zwischen den Einzelterminen (s. Abbildung 1):

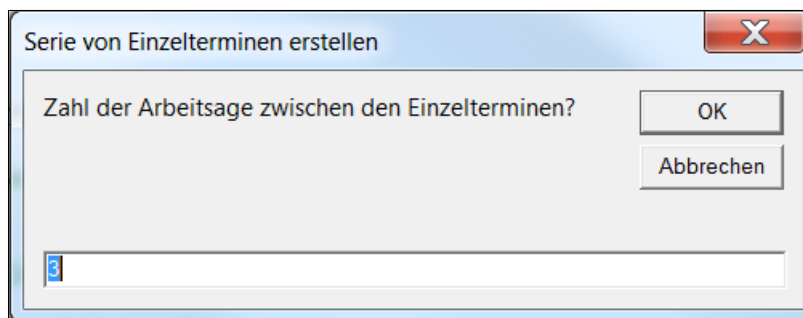
A screenshot of a Microsoft Office dialog box titled "Serie von Einzelterminen erstellen". The dialog has a standard Windows-style title bar with a close button (X) in the top right corner. The main area contains the text "Zahl der Arbeitsage zwischen den Einzelterminen?". Below this text is a text input field containing the number "3". To the right of the input field are two buttons: "OK" and "Abbrechen".

Abbildung 1: Abstand in Arbeitstagen zwischen den Einzelterminen

- Zahl der Einzeltermine in der Serie (s. Abbildung 2):

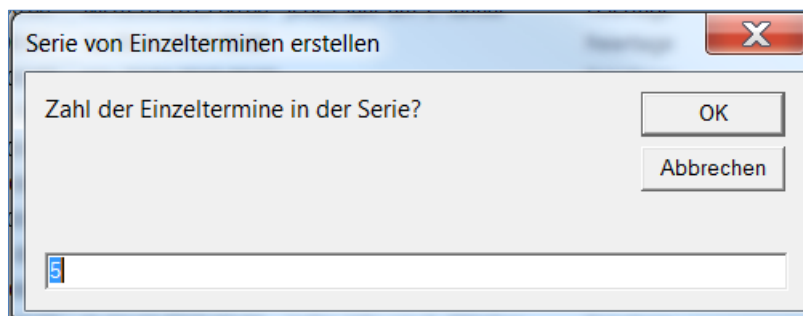
A screenshot of a Microsoft Office dialog box titled "Serie von Einzelterminen erstellen". The dialog has a standard Windows-style title bar with a close button (X) in the top right corner. The main area contains the text "Zahl der Einzeltermine in der Serie?". Below this text is a text input field containing the number "10". To the right of the input field are two buttons: "OK" and "Abbrechen".

Abbildung 2: Zahl der Einzeltermine in der Serie

Mit einem entsprechenden Benutzerformular (engl. *userform*) könnten die beiden Eingaben (s. oben) gemeinsam abgefragt werden.

¹ Das sind jeweils Einzeltermine, also *keine* wiederkehrenden Termine mit einem bestimmten Serienmuster.

3.2 Feiertage und Startdatum

Darüber hinaus benötigt die Anwendung eine Liste der Feiertage² und einen Starttermin für die gewünschte Serie der Einzeltermine (s. Abbildung 3).

Betreff	Termin-/Besprechu...	Beginn	Ende	Serienmuster	Kategorien
Kategorien: Feiertage (13 Elemente)					
Neujahr		Di 01.01.2019 00:00	Mi 02.01.2019 00:00	jedes Jahr am 1. Januar	Feiertage
Karfreitag		Fr 19.04.2019 00:00	Sa 20.04.2019 00:00		Feiertage
Ostersonntag		So 21.04.2019 00:00	Mo 22.04.2019 00:00		Feiertage
Ostermontag		Mo 22.04.2019 00:00	Di 23.04.2019 00:00		Feiertage
Tag d. Arbeit		Mi 01.05.2019 00:00	Do 02.05.2019 00:00	jedes Jahr am 1. Mai	Feiertage
Chr. Himmelfahrt		Do 30.05.2019 00:00	Fr 31.05.2019 00:00		Feiertage
Pfingstsonntag		So 09.06.2019 00:00	Mo 10.06.2019 00:00		Feiertage
Pfingstmontag		Mo 10.06.2019 00:00	Di 11.06.2019 00:00		Feiertage
Fronleichnam		Do 20.06.2019 00:00	Fr 21.06.2019 00:00		Feiertage
Tag der Deutschen Einheit		Do 03.10.2019 00:00	Fr 04.10.2019 00:00	jedes Jahr am 3. Oktober	Feiertage
Allerheiligen		Fr 01.11.2019 00:00	Sa 02.11.2019 00:00	jedes Jahr am 1. November	Feiertage
1. Weihnachtstag		Mi 25.12.2019 00:00	Do 26.12.2019 00:00	jedes Jahr am 25. Dezember	Feiertage
2. Weihnachtstag		Do 26.12.2019 00:00	Fr 27.12.2019 00:00	jedes Jahr am 26. Dezember	Feiertage
Kategorien: Gelbe Kategorie (1 Element)					
Termin	Praxis	Di 22.10.2019 08:00	Di 22.10.2019 08:30		Gelbe Kategorie

Abbildung 3: Feiertage 2019 in NRW und Beginn (22.10.2019 08:00) f. eine Serie von Einzelterminen

3.3 Arbeitsfreie Wochentage

Außerdem muss in der Hauptprozedur festgelegt werden, welche Wochentage als arbeitsfrei gelten. Bei einer 5-Tage-Woche sind das gewöhnlich Sa. und So. Wenn eine Arbeitswoche weitere arbeitsfreie Wochentage (z. B. Mi.) besitzt, muss der Aufruf der Subroutine 'Workday2'³ in der Hauptprozedur⁴ entsprechend geändert werden.

² Quelle: http://www.dr-thormaehlen.de/misc/OUTLOOK_Feiertage.pdf

³ Die in Excel eingebaute Funktion WORKDAY gibt ein Datum zurück, das eine bestimmte Anzahl von Arbeitstagen (ohne Wochenenden und ggf. Wochenfeiertagen) vor/nach einem Startdatum liegt. Sie kann aber nur Samstage und Sonntage auszuschließen. *Andere oder mehr Wochentage können nicht ausgeschlossen werden.*

⁴ In der Hauptprozedur (s. Listing 2) sind die Zeilen 63 und 84 so zu ändern wie in Listing 1 beispielhaft beschrieben.

4 Ausgabedaten

Ausgehend vom markierten *Starttermin* (22.10.2019 08:30) werden 5 *Folgetermine* (Termin 1, ..., Termin 5) im Abstand von jeweils 3 *Arbeitstagen* in den OUTLOOK-Standardkalender eingefügt (s. Abbildung 4):

Betreff	Termin-/Bespr...	Beginn	Ende	Serienmuster	Kategorien
Kategorien: Feiertage (13 Elemente)					
Neujahr		Di 01.01.2019 00:00	Mi 02.01.2019 00:00	jedes Jahr am 1. Januar	Feiertage
Karfreitag		Fr 19.04.2019 00:00	Sa 20.04.2019 00:00		Feiertage
Ostersonntag		So 21.04.2019 00:00	Mo 22.04.2019 00:00		Feiertage
Ostermontag		Mo 22.04.2019 00:00	Di 23.04.2019 00:00		Feiertage
Tag d. Arbeit		Mi 01.05.2019 00:00	Do 02.05.2019 00:00	jedes Jahr am 1. Mai	Feiertage
Chr. Himmelfahrt		Do 30.05.2019 00:00	Fr 31.05.2019 00:00		Feiertage
Pfingstsonntag		So 09.06.2019 00:00	Mo 10.06.2019 00:00		Feiertage
Pfingstmontag		Mo 10.06.2019 00:00	Di 11.06.2019 00:00		Feiertage
Fronleichnam		Do 20.06.2019 00:00	Fr 21.06.2019 00:00		Feiertage
Tag der Deutschen Einheit		Do 03.10.2019 00:00	Fr 04.10.2019 00:00	jedes Jahr am 3. Oktober	Feiertage
Allerheiligen		Fr 01.11.2019 00:00	Sa 02.11.2019 00:00	jedes Jahr am 1. November	Feiertage
1. Weihnachtstag		Mi 25.12.2019 00:00	Do 26.12.2019 00:00	jedes Jahr am 25. Dezember	Feiertage
2. Weihnachtstag		Do 26.12.2019 00:00	Fr 27.12.2019 00:00	jedes Jahr am 26. Dezember	Feiertage
Kategorien: Gelbe Kategorie (6 Elemente)					
Termin	Praxis	Di 22.10.2019 08:30	Di 22.10.2019 09:00		Gelbe Kategorie
Termin 1	Praxis	Fr 25.10.2019 08:30	Fr 25.10.2019 09:00		Gelbe Kategorie
Termin 2	Praxis	Mi 30.10.2019 08:30	Mi 30.10.2019 09:00		Gelbe Kategorie
Termin 3	Praxis	Mo 04.11.2019 08:30	Mo 04.11.2019 09:00		Gelbe Kategorie
Termin 4	Praxis	Do 07.11.2019 08:30	Do 07.11.2019 09:00		Gelbe Kategorie
Termin 5	Praxis	Di 12.11.2019 08:30	Di 12.11.2019 09:00		Gelbe Kategorie

Abbildung 4: Starttermin (22.10.2019 08:30) mit 5 Folgeterminen im Abstand von 3 Arbeitstagen

5 Prozeduren

5.1 Hauptprozedur

Die Hauptprozedur mit dem Namen '*CreateSeriesOfAppt*' hat die Aufgabe, eine OUTLOOK-Terminserie für jeden n-ten Arbeitstag zu erstellen, ausgehend von einem markierten Anfangstermin. Die Prozedur umfasst 6 Schritte bzw. Vorgänge:

Schritt	Vorgang
1	Liste der relevanten Feiertage bestimmen und in der globalen Variablen ' <i>strAllDayOff</i> ' speichern.
2	Einen geöffneten bzw. ausgewählten Termin als Anfangstermin einer Serie von
3	Einzelterminen bestimmen.
4	Abstand (d. h. Zahl der Arbeitstage) zwischen den Einzelterminen im Dialog mit dem Benutzer abfragen (s. Abbildung 1).
5	Zahl der Einzeltermine in der Serie im Dialog mit dem Benutzer abfragen (s. Abbildung 2). Mit einer FOR-NEXT-Schleife die zuvor abgefragte Zahl der Termine mit dem geforderten zeitlichen Abstand erzeugen. Dabei werden Wochenfeiertage und andere arbeitsfreie Wochentage übersprungen (s. Listing 2, Zeilen 69 – 90).
6	Bei gelungener Erstellung der Serie der Einzeltermine wird eine Erfolgsmeldung ausgegeben. Sonst eine Fehlermeldung.

Tabelle 1: Schritte bzw. Vorgänge in der Hauptprozedur

Listing 2 beinhaltet den Quellcode der Hauptprozedur.

Die Hauptprozedur benötigt folgende drei Subroutinen:

Name der Subroutine	Aufgabe
GetCurrentItem	Bestimmt, ob ein OUTLOOK-Kalendereintrag als <i>Beginn</i> einer Serie von Einzelterminen geöffnet oder ausgewählt ist.
GetHolidays	<ul style="list-style-type: none">• Überprüft den OUTLOOK-Standardkalender auf alle als beschäftigt, auswärtig oder vorläufig gekennzeichnete Termine/Ereignisse.• Überspringt wiederkehrende Termine/Ereignisse.• Um eingeschlossen zu werden, müssen Feiertage mit einem Beschäftigungsstatus <i>ungleich</i> 'frei' gekennzeichnet sein.
Workday2	Erlaubt, eine beliebige Zahl von Wochentagen als ' <i>arbeitsfrei</i> ' auszuschließen, also nicht nur Sa. und So.

Tabelle 2: Benötigte Subroutinen der Hauptprozedur

5.2 Startermin einer Terminserie bestimmen

Wie der Name '*GetCurrentItem*' der Subroutine (s. Tabelle 2) bereits besagt, dient sie dazu, den Anfangstermin einer Serie von Einzelterminen zu bestimmen. Wenn kein Anfangstermin geöffnet oder ausgewählt ist, wird die Hauptprozedur mit folgender Fehlermeldung abgebrochen (s. Abbildung 5):

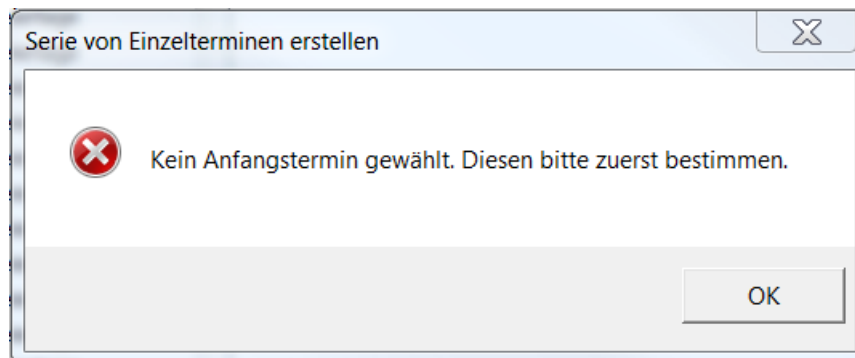


Abbildung 5: Fehlermeldung bei undefiniertem Anfangstermin

5.3 Feiertage bestimmen

Die Subroutine '*GetHolidays*' ermittelt arbeitsfreie Wochentage aufgrund von Feiertagen, die in Deutschland abhängig sind vom jeweiligen Bundesland (s. Abbildung 3).

5.4 Arbeitsfreie Wochentage definieren

Die Subroutine '*Workday2*' erlaubt, auch Wochentage als *arbeitsfrei* zu bestimmen, die *nicht* - wie gewöhnlich - auf einen *Samstag* oder *Sonntag* fallen.

6 VBA-Quellcode

6.1 Globale Konstante und Variable

```
Option Explicit
' 1 globale Konstante definieren:
Const conTask As String = "Serie von Einzelterminen erstellen"
' 2 globale Variablen definieren:
Dim strAllDayOff As String
' Jedem auszuschließenden Wochentag einen Wert zuordnen.
' Zwei Beispiele zu Enum EDaysOfWeek:
'   Um So. und Sa. auszuschließen: 1 + 64 = 65
'   Um So., Mi. und Sa. auszuschließen: 1 + 8 + 64 = 73
Enum EDaysOfWeek ' Enum: Datentyp für eine Enumeration. Quelle: [1]
    Sunday = 1    ' 2 ^ (vbSunday - 1)
    Monday = 2   ' 2 ^ (vbMonday - 1)
    Tuesday = 4  ' 2 ^ (vbTuesday - 1)
    Wednesday = 8 ' 2 ^ (vbWednesday - 1)
    Thursday = 16 ' 2 ^ (vbThursday - 1)
    Friday = 32  ' 2 ^ (vbFriday - 1)
    Saturday = 64 ' 2 ^ (vbSaturday - 1)
End Enum
```

Listing 1: Globale Konstante und Variablen

6.2 Hauptprozedur

```
1 Public Sub CreateSeriesOfAppt()  
2 ' Aufgabe: OUTLOOK-Terminserie für jeden n-ten Arbeitstag erstellen.  
3 ' Anwendung: Einen neuen/bestehenden Termin in OUTLOOK speichern bzw. markieren,  
4 ' dann diese Hauptprozedur ausführen.  
5 Dim objAppt As Object ' Kalendereintrag als Objekt  
6 Dim olAppt2 As Outlook.AppointmentItem  
7 Dim bolOK As Boolean ' boolescher Schalter  
8 Dim apptStartTm As Date ' Startzeit  
9 Dim apptStartDtTm As Date ' Starttermin  
10 Dim dtCurrent As Date ' Startdatum  
11 Dim dtNext As Date ' nächster Termin (Nachfolgetermin)  
12 Dim intLoop As Integer ' Schleifenzähler  
13 Dim lngNumDays As Long ' Zahl der Arbeitstage zwischen Einzelterminen  
14 Dim lngNumAppt As Long ' Zahl der Einzeltermine in einer Serie  
15 Dim strErrMsg As String ' Fehlermeldung  
16 Dim strInput As String ' Eingabewert durch den Benutzer  
17 Dim varReturnVal As Variant ' Rückgabewert der Funktion 'Workday2'  
18 On Error GoTo Error_Point ' Fehlerbehandlung  
19 ' Liste der relevanten Feiertage bestimmen und in der globalen Variablen 'strAllDayOff'  
20 ' speichern.  
21 GetHolidays  
22 ' Geöffneten o. ausgewählten Termin als Anfangstermin bestimmen.  
23 Set objAppt = GetCurrentItem()  
24 If TypeName(objAppt) = "Nothing" Or objAppt.Class <> olAppointmentItem Then  
25 ' Das ausgewählte Objekt ist kein Kalendereintrag!  
26 MsgBox "Kein Anfangstermin gewählt. Diesen bitte zuerst bestimmen.", vbCritical, conTask  
27 Else  
28 ' Erste Eingabe des Benutzers erfassen  
29 bolOK = False  
30 strInput = "3" ' Vorschlagswert  
31 Do  
32 strInput = InputBox("Zahl der Arbeitstage zwischen den Einzelterminen?", conTask,  
33 strInput)  
34 If Len(strInput) > 0 And IsNumeric(strInput) And Val(strInput) > 0 And Val(strInput)  
35 = Int(Val(strInput)) Then  
36 bolOK = True  
37 End If  
38 Loop Until bolOK Or strInput = vbNullString  
39 If strInput = vbNullString Then  
40 MsgBox "Abbruch der Prozedur durch den Benutzer:" & vbCrLf & "Ungültige Eingabe oder  
41 'Abbrechen' gewählt !", vbOKOnly, conTask  
42 GoTo Exit_Point  
43 Else  
44 lngNumDays = Val(strInput)  
45 End If  
46 ' Zweite Eingabe des Benutzers erfassen  
47 bolOK = False  
48 strInput = "5" ' Vorschlagswert  
49 Do  
50 strInput = InputBox("Zahl der Einzeltermine in einer Serie?", conTask, strInput)  
51 If Len(strInput) > 0 And IsNumeric(strInput) And Val(strInput) > 0 And Val(strInput)  
52 = Int(Val(strInput)) Then  
53 bolOK = True  
54 End If  
55 Loop Until bolOK Or strInput = vbNullString  
56 If strInput = vbNullString Then  
57 MsgBox "Abbruch der Prozedur durch den Benutzer:" & vbCrLf & "Ungültige Eingabe oder  
58 'Abbrechen' gewählt!", vbOKOnly, conTask  
59 GoTo Exit_Point  
60 Else  
61 lngNumAppt = Val(strInput)  
62 End If  
63 dtCurrent = Format(objAppt.Start, "mm/dd/yyyy")  
64 apptStartTm = Format(objAppt.Start, "hh:mm:ss AM/PM")  
65 varReturnVal = Workday2(dtCurrent, lngNumDays + 1, 65) ' Sa./So. überspringen  
66 If IsError(varReturnVal) Then  
67 GoTo Exit_Point  
68 Else  
69 dtNext = varReturnVal  
70 End If
```

```

71     For intLoop = 1 To lngNumAppt
72         Set olAppt2 = Session.GetDefaultFolder(olFolderCalendar).Items.Add(olAppointmentItem)
73         apptStartDtTm = dtNext & " " & apptStartTm
74         ' Weitere Felder können ggf. aufgenommen werden.
75         With objAppt
76             olAppt2.Subject = .Subject & " " & intLoop      ' Betreff
77             olAppt2.Location = .Location                    ' Ort
78             olAppt2.Body = .Body                          ' Textkörper
79             olAppt2.Start = apptStartDtTm                 ' Starttermin
80             olAppt2.Duration = .Duration                  ' Dauer
81             olAppt2.Categories = .Categories               ' Kategorie
82             olAppt2.ReminderSet = .ReminderSet            ' Erinnerung
83         End With
84         On Error Resume Next
85         olAppt2.Save
86         varReturnVal = Workday2(dtNext, lngNumDays + 1, 65) ' Sa./So. überspringen
87         If IsError(varReturnVal) Then
88             GoTo Exit_Point
89         Else
90             dtNext = varReturnVal
91         End If
92     Next intLoop
93     MsgBox "Erledigt", vbExclamation, conTask
94 End If
95 Exit_Point:
96     On Error Resume Next
97     Set objAppt = Nothing
98     Set olAppt2 = Nothing
99     Exit Sub ' Prozedur sofort verlassen
100 Error_Point:
101     strErrMsg = "Laufzeitfehler (" & Err.Number & ") " & Err.Description
102     MsgBox strErrMsg, vbCritical, conTask
103     Resume Exit_Point
104 End Sub

```

Listing 2: Hauptprozedur 'CreateSeriesOfAppt', Quelle: In Anlehnung an [2]

6.3 Funktion 'Workday2'

```

Function Workday2(dtStart As Date, _
    lngDaysRequired As Long, _
    ExcludeDoW As EDaysOfWeek, _
    Optional varHolidays As Variant) As Variant
' Aufgabe: Diese Funktion erlaubt, eine beliebige Zahl von Wochentagen auszuschließen.
' Vier Argumente:
'   dtStart           Das Datum, an dem der Zeitraum beginnt.
'   lngDaysRequired   Zahl der benötigten Arbeitstage im Zeitraum.
'   ExcludeDoW        Die Summe der Werte in EDaysOfWeek für alle Wochentage, die
'                   ausgeschlossen werden sollen.
'   varHolidays       Ein Array oder eine Reihe von Terminen, die aus dem Zeitraum
'                   ausgeschlossen werden sollen.
' Ergebnis: Ein Datum, das lngDaysRequired nach dtStart liegt, aber ohne Wochenfeiertage und
'   ausgeschlossene Wochentage.
Dim IsHoliday        As Boolean    ' Ist das Datum dtTest ein Feiertag?
Dim dtTest           As Date      ' Das jeweilig erhöhte Datum
Dim lngDays          As Long      ' Allgemeiner Zähler für Wochentage
Dim lngCnt           As Long      ' Zähler für tatsächliche Arbeitstage
Dim lngLoopCtrl     As Long      ' Obergrenze für Schleife (Verhinderung einer
'                   Endlosschleife)
Dim currDoW         As EDaysOfWeek ' Wochentag von dtTest
Dim varLoop         As Variant    ' Schleifenvariable

varHolidays = Split(strAllDayOff, ",")
If lngDaysRequired = 0 Then
    Workday2 = dtStart
    Exit Function
End If
If ExcludeDoW >= (Sunday + Monday + Tuesday + Wednesday + Thursday + Friday + Saturday) Then
    MsgBox "Fehlerhafte Summe der Werte in EDaysOfWeek!", vbCritical, conTask
    Workday2 = CVErr(2015) ' Benutzerdefinierten Fehler zurückgeben. [3]
    Exit Function
End If
' Endlosschleife verhindern, die unter Umständen möglich ist.
lngLoopCtrl = lngDaysRequired * 10000
' Anfangswerte setzen.
lngDays = lngCnt = 0
' Folgende Schleife wird solange durchlaufen bis die Zahl der tatsächlichen
' Arbeitstage (lngCnt) den benötigten Arbeitstagen (lngDaysRequired) gleicht.
Do Until lngCnt = lngDaysRequired
    lngDays = lngDays + 1
    dtTest = dtStart + lngDays
    currDoW = 2 ^ (Weekday(dtTest) - 1)
    If (currDoW And ExcludeDoW) = 0 Then ' Wochentag ist nicht ausgeschlossen, also weiter.
        IsHoliday = False
        If IsMissing(varHolidays) = False Then
            For Each varLoop In varHolidays
                If varLoop = dtTest Then
                    IsHoliday = True
                    ' Datum nicht zählen
                    Exit For
                End If
            Next varLoop
        End If
        If IsHoliday = False Then
            ' Datum zählen
            lngCnt = lngCnt + 1
        End If
    End If
    If lngDays > lngLoopCtrl Then
        MsgBox "Fehler: Endlosschleife!", vbCritical, conTask
        Workday2 = CVErr(2015) ' Benutzerdefinierten Fehler zurückgeben. [3]
        Exit Function
    End If
Loop
Workday2 = dtStart + lngDays ' Gültiges Ergebnis an die aufrufende Hauptprozedur zurückgeben.
End Function

```

Listing 3: Funktion 'Wordday2', Quelle: In Anlehnung an [1]

6.4 Subroutine 'GetHolidays'

```
Sub GetHolidays()  
    ' Überprüft den Standardkalender auf alle als beschäftigt/auswärtig/vorläufig  
    ' gekennzeichneten Tagesereignisse.  
    ' Überspringt wiederkehrende Termine/Ereignisse.  
    ' Um eingeschlossen zu werden, müssen Feiertage mit einem Beschäftigungsstatus ungleich  
    ' 'frei' gekennzeichnet sein.  
    Dim olFolder As Outlook.Folder ' Verzeichnis  
    Dim olItems As Outlook.Items ' Termine  
    Dim ResItems As Outlook.Items ' gefilterte Termine  
    Dim objItem As Object ' Schleifenvariable  
    Dim strFilter As String ' Filter für Termine  
    ' Verzeichnis für Termine bestimmen.  
    Set olFolder = Session.GetDefaultFolder(olFolderCalendar)  
    Set olItems = olFolder.Items  
    ' Termine nach Startzeit sortieren.  
    olItems.Sort "[Start]"  
    ' Serientermine ausschließen  
    olItems.IncludeRecurrences = False  
    ' Filter definieren.  
    strFilter = "[Start] >= '" & Date & "' And [AllDayEvent] = 'True' And [BusyStatus] <> '0'  
    And [IsRecurring] = 'False'"  
    Set ResItems = olItems.Restrict(strFilter)  
    ' Alle Termine in der Sammlung durchlaufen.  
    For Each objItem In ResItems  
        ' Datenbereich mit relevanten Feiertagsterminen erstellen.  
        strAllDayOff = strAllDayOff & Format(objItem.Start, "mm/dd/yyyy") & ","  
    Next objItem  
    ' Die Zeichenkette des Datenbereichs bereinigen (d. h. letztes Komma abschneiden).  
    strAllDayOff = Left(strAllDayOff, Len(strAllDayOff) - 1)  
    ' Debug.Print strAllDayOff  
    Set ResItems = Nothing  
    Set olItems = Nothing  
    Set olFolder = Nothing  
End Sub
```

Listing 4: Subroutine 'GetHolidays', Quelle: In Anlehnung an [2]

6.5 Funktion 'GetCurrentItem'

```
Function GetCurrentItem() As Object
' Aufgabe: Bestimmt, ob ein OUTLOOK-Element geöffnet oder ausgewählt ist.
Dim objApp As Outlook.Application
Set objApp = Outlook.Application
' Den ausgewählten Termin bestimmen.
On Error Resume Next
Select Case TypeName(objApp.ActiveWindow)
    Case "Explorer"
        Set GetCurrentItem = objApp.ActiveExplorer.Selection.item(1)
    Case "Inspector"
        Set GetCurrentItem = objApp.ActiveInspector.CurrentItem
End Select
Set objApp = Nothing
End Function
```

Listing 5: Funktion 'GetCurrentItem', Quelle: In Anlehnung an [4]

7 Literaturverzeichnis

- [1] C. H. Pearson, „A Better WORKDAY Function,“ 05 06 2018. [Online]. Available: <http://www.cpearson.com/excel/BetterWorkday.aspx>. [Zugriff am 22 10 2019].
- [2] D. Poremsky, „Create Outlook appointments for every nn workday,“ 03 05 2018. [Online]. Available: <https://www.slipstick.com/developer/create-outlook-appointments-for-every-xx-weekday-using-vba/>. [Zugriff am 22 10 2019].
- [3] C. H. Pearson, „Returning Errors From User Defined Functions In VBA,“ 09 10 2019. [Online]. Available: <http://www.cpearson.com/excel/ReturningErrors.aspx>. [Zugriff am 22 10 2019].
- [4] D. Poremsky, „Outlook VBA: Work with Open Item or Selected Item,“ 04 12 2018. [Online]. Available: <https://www.slipstick.com/developer/outlook-vba-work-with-open-item-or-select-item/#getcurrentitem>. [Zugriff am 22 10 2019].
- [5] V. Thormählen, „OUTLOOK: Feiertage mit VBA-Skript erstellen/entfernen,“ 28 09 2019. [Online]. Available: http://www.dr-thormaehlen.de/misc/OUTLOOK_Feiertage.pdf. [Zugriff am 03 11 2019].