

# OUTLOOK: Verfügbare Zeitfenster finden

Dr. Volker Thormählem, 14.11.2019

---

## Inhalt

1	Aufgabenstellung .....	2
2	Ausgangssituation .....	2
3	Schrittfolge im Benutzerformular .....	2
3.1	Erster Schritt: Verfügbare Zeitfenster finden .....	3
3.2	Zweiter Schritt: Verfügbares Zeitfenster auswählen .....	4
3.3	Dritter Schritt: Einmaligen Termin erstellen .....	4
4	Gestaltung des Benutzerformulars .....	6
4.1	Steuerelemente .....	6
4.2	Einzelheiten .....	6
5	Quellcode der Anwendung .....	7
5.1	Prozedurtypen .....	7
5.2	Listing mit Zeilennummern .....	7
6	Erweiterung der Anwendung für mehrere Benutzer .....	7
7	Literaturverzeichnis .....	13

## Abbildungen

Abbildung 1:	Drei vorhandene Einzeltermine am 22.10.2019 .....	2
Abbildung 2:	Vorbelegtes Benutzerformular zum Finden verfügbarer Zeitfenster .....	3
Abbildung 3:	Benutzerformular mit verfügbaren Zeitfenstern .....	4
Abbildung 4:	Erfolgsmeldung nach Erstellung eines einmaligen Termins .....	5
Abbildung 5:	'Neuer Termin' in OUTLOOK-Kalender am 22.10.2019 um 10:00 Uhr .....	5
Abbildung 6:	Drei vorhandene Einzeltermine und ein 'Neuer Termin' im OUTLOOK-Kalender am 22.10.2019 .....	5

## Listings

Listing 1:	Gesamter VBA-Quellcode der Anwendung .....	12
------------	--	----

## Tabellen

Tabelle 1:	Vier verfügbare Zeitfenster am 22.10.2019 .....	2
Tabelle 2:	Steuerelemente des Benutzerformulars in alphabetischer Reihenfolge .....	6
Tabelle 3:	Variablenamen für das Benutzerformular .....	6
Tabelle 4:	Prozedurtypen im Quellcode der Anwendung .....	7

# 1 Aufgabenstellung

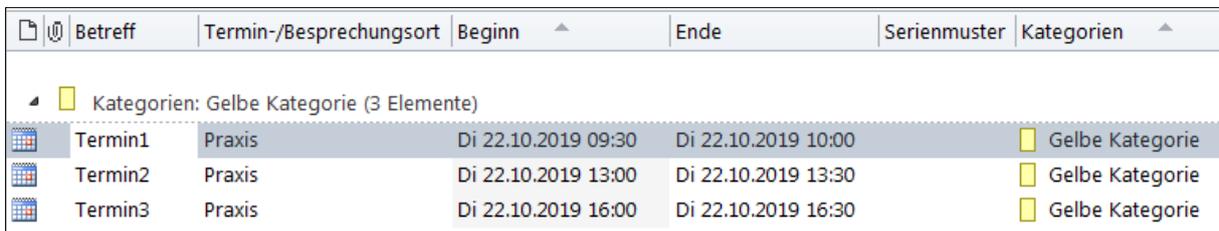
In diesem Beitrag wird beschrieben, wie ein einmaliger Termin im OUTLOOK-Kalender angelegt werden kann, ohne mit den evtl. schon vorhandenen Terminen des relevanten Tages zu kollidieren. Zur Ermittlung der/des verfügbaren Zeitfenster(s) wird ein Benutzerformular eingesetzt, das auf folgenden Annahmen beruht: Ein verfügbares Zeitfenster muss ...

- größer oder gleich der normalen Dauer (in ganzen Minuten) eines Termins sein,
- zwischen dem normalen Beginn und Ende der Arbeitszeit eines Tages liegen.

Wenn ein oder mehr verfügbare Zeitfenster gefunden werden, kann der Benutzer des Formulars auswählen, welches davon zur automatischen Erstellung eines neuen Einzeltermins im OUTLOOK-Kalender verwandt werden soll.

# 2 Ausgangssituation

Die angenommene Ausgangssituation ist in Abbildung 1 dargestellt. Sie beinhaltet drei vorhandene Einzeltermine für den 22.10.2019:



Betreff	Termin-/Besprechungsort	Beginn	Ende	Serienmuster	Kategorien
Kategorien: Gelbe Kategorie (3 Elemente)					
Termin1	Praxis	Di 22.10.2019 09:30	Di 22.10.2019 10:00		Gelbe Kategorie
Termin2	Praxis	Di 22.10.2019 13:00	Di 22.10.2019 13:30		Gelbe Kategorie
Termin3	Praxis	Di 22.10.2019 16:00	Di 22.10.2019 16:30		Gelbe Kategorie

Abbildung 1: Drei vorhandene Einzeltermine am 22.10.2019

Wenn ein Termin jeweils 30 Minuten dauert und wenn die normale Arbeitszeit am genannten Tag um 08.30 Uhr beginnt und um 19:00 Uhr endet, dann existieren 4 verfügbare Zeitfenster zur automatischen Erstellung eines neuen Einzeltermins (s. Tabelle 1):

Zeitfenster	
Beginn	Ende
08:30	09:30
10:00	13:00
13:30	16:00
16:30	19:00

Tabelle 1: Vier verfügbare Zeitfenster am 22.10.2019

# 3 Schrittfolge im Benutzerformular

Das im Folgenden präsentierte Benutzerformular (s. Abbildung 2) umfasst 3 Schritte:

1. Verfügbare Zeitfenster für einen Einzeltermin an einem bestimmten Datum ermitteln.
2. Eines davon auswählen.
3. Mit dem gewählten Zeitfenster einen Einzeltermin im OUTLOOK-Kalender erstellen.

### 3.1 Erster Schritt: Verfügbare Zeitfenster finden

Zur Bestimmung der verfügbaren Zeitfenster sind vier Parameter erforderlich (s. Abbildung 2, Rahmen 'Parameter ...'). Diese Parameter sind änderbar. Bei Änderung wird ihre Plausibilität durch programmierte Kontrollen<sup>1</sup> sichergestellt.

The screenshot shows a window titled 'frmZeitfenster' with a close button in the top right corner. The window is divided into three main sections:

- 1. Schritt : Verfügbare Zeitfenster finden.**
  - Parameter (ggf. ändern)
  - Datum: 22.10.2019 (Format: dd.mm.yyyy)
  - Normale Dauer eines Termins: 30 (in ganzen Minuten)
  - Frühester Beginn eines Termins um: 08:30 (Format: hh:nn)
  - Spätestes Ende eines Termins um: 19:00 (Format: hh:nn)
  - Button: Verfügbare Zeitfenster finden
- 2. Schritt : Verfügbares Zeitfenster auswählen.**
  - Verfügbarkeit: Beginn | Ende
  - Bitte ein verfügbares Zeitfenster auswählen: [Dropdown menu]
- 3. Schritt : Einmaligen Termin erstellen.**
  - Eigenschaften des einmaligen Termins (ggf. ändern)
  - Betreff: [Text field]
  - Ort: [Text field]
  - Start am/um: [Text field] (nicht änderbar, siehe gewähltes Zeitfenster)
  - Dauer: [Text field] (in ganzen Minuten, nicht änderbar, siehe oben)
  - Erinnerung:  Ja/Nein
  - Textkörper: [Text area]
  - Button: Einmaligen Termin erstellen

At the bottom of the window is a button labeled 'Abbruch/Ende'.

Abbildung 2: Vorbelegtes Benutzerformular zum Finden verfügbarer Zeitfenster

Die Formularfelder im Rahmen 'Eigenschaften des einmaligen Termins ...' sind während der Suche nach evtl. verfügbaren Zeitfenstern deaktiviert (erkennbar an der jeweils grauen Hintergrundfarbe).

Durch Betätigung der Schaltfläche 'Verfügbare Zeitfenster finden' (s. Abbildung 2) werden diese mit einer VBA-Funktion ermittelt und in dem Kombinationsfeld<sup>2</sup> mit der Beschriftung 'Bitte ein verfügbares Zeitfenster wählen' angezeigt (s. Abbildung 3).

<sup>1</sup> Durch sog. Exit-Ereignisprozeduren (s. Tabelle 4).

<sup>2</sup> Dieses Steuerelement ist eine Mischung aus einem Text- und einem Listenfeld, ohne dafür viel Platz im Benutzerformular zu beanspruchen.

### 3.2 Zweiter Schritt: Verfügbares Zeitfenster auswählen

Der Benutzer kann nun wählen, ob er ein verfügbares Zeitfenster akzeptiert und anschließend mit der Schaltfläche 'Einmaligen Termin erstellen' einen solchen Termin automatisch generiert oder die VBA-Prozedur mit der Schaltfläche 'Abbruch/Ende' sofort verlässt.

The screenshot shows a Windows-style dialog box titled 'frmZeitfenster'. It is divided into three main sections:

- 1. Schritt : Verfügbare Zeitfenster finden.** This section contains four input fields for search parameters: 'Datum:' (22.10.2019), 'Normale Dauer eines Termins:' (30), 'Frühester Beginn eines Termins um:' (08:30), and 'Spätestes Ende eines Termins um:' (19:00). A button 'Verfügbare Zeitfenster finden' is located below these fields.
- 2. Schritt : Verfügbares Zeitfenster auswählen.** This section has a 'Verfügbarkeit' header and a table with columns 'Beginn' and 'Ende'. Below the table is a dropdown menu 'Bitte ein verfügbares Zeitfenster auswählen:' with a list of time slots: 08:30|09:30, 10:00|13:00 (highlighted), 13:30|16:00, and 16:30|19:00.
- 3. Schritt : Einmaligen Termin erstellen.** This section contains several input fields: 'Betreff:' (Neuer Termin), 'Ort:' (Praxis), 'Start am/um:' (22.10.2019 10:00), 'Dauer:' (30), 'Erinnerung:' (checkbox for Ja/Nein), and 'Textkörper:' (Kommentar). A button 'Einmaligen Termin erstellen' is at the bottom of this section.

At the very bottom of the dialog box is a button labeled 'Abbruch/Ende'.

Abbildung 3: Benutzerformular mit verfügbaren Zeitfenstern

### 3.3 Dritter Schritt: Einmaligen Termin erstellen

Bevor der Benutzer die Schaltfläche 'Einmaligen Termin erstellen' betätigt, hat er die Möglichkeit, vier Vorschlagswerte im Rahmen 'Eigenschaften des einmaligen Termins' zu ändern:

- Betreff
- Ort
- Erinnerung
- Textkörper

Die grau eingefärbten Formularfelder (Start am/um, Dauer) sind aus logischen Gründen nicht änderbar.

Bei Betätigung der oben genannten Schaltfläche, wird eine Erfolgsmeldung (s. Abbildung 4) angezeigt und ein entsprechender Termin mit dem Betreff 'Neuer Termin' im OUTLOOK-Kalender angelegt (s. Abbildung 5 und Abbildung 6).

Die vier Textfelder im Rahmen 'Parameter ...' sind bei Erstellung eines einmaligen Termins aus logischen Gründen deaktiviert (d.h. die jeweilige Hintergrundfarbe ist grau, s. Abbildung 3).

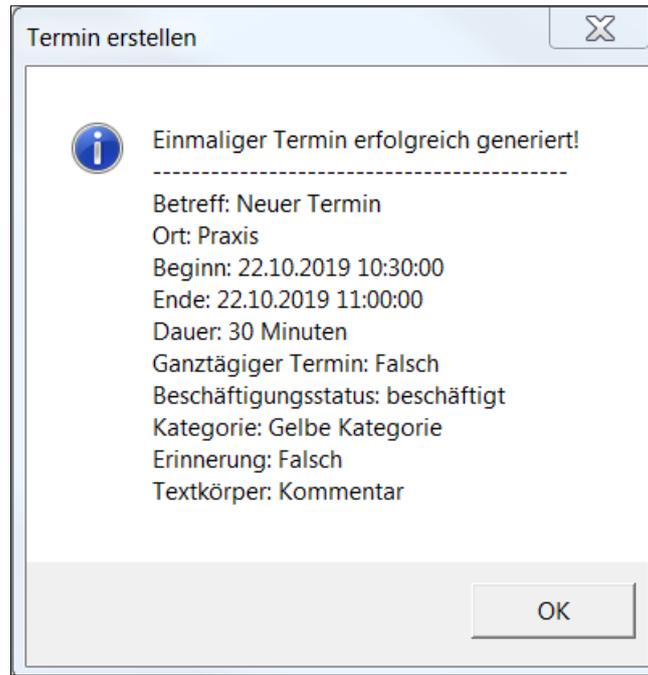


Abbildung 4: Erfolgsmeldung nach Erstellung eines einmaligen Termins

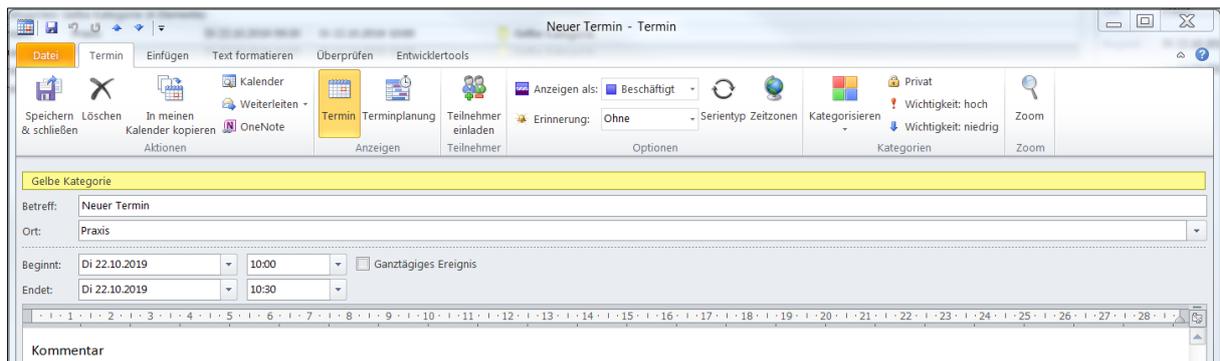


Abbildung 5: 'Neuer Termin' in OUTLOOK-Kalender am 22.10.2019 um 10:00 Uhr

Betreff	Termin-/Besprechungsort	Beginn	Ende	Serienmuster	Kategorien
Kategorien: Gelbe Kategorie (4 Elemente)					
Termin1	Praxis	Di 22.10.2019 09:30	Di 22.10.2019 10:00		Gelbe Kategorie
Neuer Termin	Praxis	Di 22.10.2019 10:00	Di 22.10.2019 10:30		Gelbe Kategorie
Termin2	Praxis	Di 22.10.2019 13:00	Di 22.10.2019 13:30		Gelbe Kategorie
Termin3	Praxis	Di 22.10.2019 16:00	Di 22.10.2019 16:30		Gelbe Kategorie

Abbildung 6: Drei vorhandene Einzeltermine und ein 'Neuer Termin' im OUTLOOK-Kalender am 22.10.2019

## 4 Gestaltung des Benutzerformulars

### 4.1 Steuerelemente

Das schon mehrmals gezeigte Benutzerformular mit dem Namen 'frmZeitfenster' enthält 51 Steuerelemente<sup>3</sup> (s. Tabelle 2). Alle Steuerelemente sind mit einem Präfix versehen, der ihren jeweiligen Typ eindeutig angibt.

Steuerelemente des Benutzerformulars		
Typ	Präfix	Anzahl
Befehlsschaltfläche	cmd	3
Bezeichnungsfeld	lbl	28
Kombinationsfeld	cbo	1
Kontrollkästchen	chk	1
Rahmen	fra	4
Textfeld	txt	14
Summe		51

Tabelle 2: Steuerelemente des Benutzerformulars in alphabetischer Reihenfolge

### 4.2 Einzelheiten

Tabelle 3 enthält die Variablennamen des Benutzerformulars. Sie dienen zum Verständnis des dazugehörigen VBA-Quellcodes (s. Listing 1):

Feldbezeichnung	Feldtyp	Variablenname
<b>... im Rahmen 'Parameter'</b>		
Datum	Textfeld	txtDate
Normale Dauer eines Termins	Textfeld	txtDuration
Frühester Beginn eines Termins	Textfeld	txtStartTime
Spätestes Ende eines Termins	Textfeld	txtEndTime
(Format: dd.mm.yyy)	Bezeichnungsfeld	
(in Minuten)	Bezeichnungsfeld	
(Format hh:nn)	Bezeichnungsfeld	
(Format hh:nn)	Bezeichnungsfeld	
<b>... im Rahmen 'Verfügbarkeit'</b>		
Bitte ein verfügbares Zeitfenster wählen	Kombinationsfeld	cboTimeSlots
<b>... im Rahmen 'Eigenschaften des einmaligen Termins'</b>		
Betreff	Textfeld	txtTopic
Ort	Textfeld	txtWhere
Start am/um	Textfeld	txtBegin
Dauer	Textfeld	txtLength
Erinnerung	Kontrollkästchen	chkRemind
Textkörper	Textfeld	txtComment
(nicht änderbar, s. gewähltes Zeitfenster)	Bezeichnungsfeld	
(in Minuten, nicht änderbar, siehe oben)	Bezeichnungsfeld	
Ja/Nein	Bezeichnungsfeld	
<b>Sonstige Formularfelder</b>		
Verfügbare Zeitfenster finden	Befehlsschaltfläche	cmdFind
Einmaligen Termin erstellen	Befehlsschaltfläche	cmdCreate
Abbruch/Ende	Befehlsschaltfläche	cmdCancel

Tabelle 3: Variablennamen für das Benutzerformular

<sup>3</sup> Direktfenster: ? Userform2.Controls.Count ergibt 51

## 5 Quellcode der Anwendung

### 5.1 Prozedurtypen

Im VBA-Quellcode der Anwendung befinden sich 17 *Private*<sup>4</sup> Prozeduren (s. Tabelle 4):

VBA-Prozeduren	
Anzahl	Typ
1	<b>AfterUpdate</b> -Ereignisprozedur
1	<b>Change</b> -Ereignisprozedur
3	<b>Click</b> -Ereignisprozedur
6	<b>Exit</b> -Ereignisprozedur
4	<b>Function</b> -Prozedur
5	<b>Userform</b> -Prozedur

Tabelle 4: Prozedurtypen im Quellcode der Anwendung

### 5.2 Listing mit Zeilennummern

Das Listing des gesamten VBA-Quellcodes der Anwendung umfasst 355 Befehlszeilen (s. Listing 1). Im Listing ist der jeweilige Beginn der 17 Prozeduren (s. Tabelle 4) *gelb* hervorgeben. Die Hintergrundfarbe der sog. Inline-Kommentare ist *grau*.

Kern der Anwendung ist die Funktionsprozedur mit dem Namen '*FindAvailableTimeSlots*' (s. Listing 1, Zeile 50 – 140). Sie ermittelt verfügbare Zeitfenster im OUTLOOK-Kalender an einem vorgegebenen Datum.

## 6 Erweiterung der Anwendung für mehrere Benutzer

Die weiter oben beschriebene Anwendung zur Ermittlung verfügbarer Zeitfenster eignet sich *nicht* für mehrere Benutzer. In [1] wird eine VBA-Funktion und präsentiert, die auf separaten, freigegebenen Kalendern aller einbezogenen Mitarbeiter beruht. Die Funktionalität dieser Funktion wird dort wie folgt beschrieben (übersetzt aus dem Englischen vom Verf. mit '*Google Übersetzer*'):

*“Mit dieser Funktion können Sie verfügbare Zeitfenster einer bestimmten Dauer für alle Mitarbeiter in einer Liste suchen. Sie übergeben das Datum und ein Array mit den Mitarbeiternamen an die Funktion und suchen nach freien Zeitfenstern mit einer Dauer von mindestens 2 Stunden zwischen einer festgelegten Start- und Endzeit an diesem Tag. Die Dauer und die Start-/Endzeiten sind Konstanten. Sie können diese Werte also entsprechend ändern. Die Funktion gibt eine Zeichenfolge zurück, die dann zum Auffüllen eines Listenfelds verwendet wird. Hier werden alle verfügbaren Zeiträume aufgelistet, und der Benutzer kann sich dann dafür entscheiden, einen Termin in einem dieser Zeiträume zu erstellen.”*

---

<sup>4</sup> Alle VBA-Prozeduren (ausgenommen Ereignisprozeduren) sind standardmäßig '*Public*', also öffentlich. Wenn zu Beginn einer Prozedur das Schlüsselwort '*Private*' deklariert wird, können nur die im selben VBA-Modul gespeicherten Prozeduren darauf zugreifen.

```

1 Option Explicit
2 ' Globale Variable: Ein nullbasiertes, eindimensionales Datenfeld
3 Dim varTm As Variant
4
5 Private Sub UserForm_Initialize()
6     With UserForm2
7         ' Voreinstellungen
8         .txtDate.Value = CDate("22. Okt. 2019") ' Datum
9         .txtDuration.Value = 30 ' normale Dauer eines Termins in Minuten
10        .txtStartTime.Value = "08:30" ' Startzeit
11        .txtEndTime.Value = "19:00" ' Endzeit ( > Startzeit)
12        .cmdCreate.Enabled = False ' Befehlsschaltfläche deaktivieren
13        With .cboTimeSlots ' Kombinationsfeld f. Zeitfenster
14            .List = Array()
15            .Enabled = False
16            .BackColor = RGB(211, 211, 211)
17        End With
18        With .txtTopic ' Betreff
19            .Value = ""
20            .Enabled = False
21            .BackColor = RGB(211, 211, 211)
22        End With
23        With .txtWhere ' Ort
24            .Value = ""
25            .Enabled = False
26            .BackColor = RGB(211, 211, 211)
27        End With
28        With .txtBegin ' Beginn
29            .Value = ""
30            .Enabled = False
31            .BackColor = RGB(211, 211, 211)
32        End With
33        With .txtLength ' Dauer in Minuten
34            .Value = ""
35            .Enabled = False
36            .BackColor = RGB(211, 211, 211)
37        End With
38        With .chkRemind ' Erinnerung (Ja/Nein)
39            .Enabled = False
40            .BackColor = RGB(211, 211, 211)
41        End With
42        With .txtComment ' Textkörper
43            .Value = ""
44            .Enabled = False
45            .BackColor = RGB(211, 211, 211)
46        End With
47    End With
48 End Sub
49
50 Private Function FindAvailableTimeSlots(dtmAppt As Date, iDuration As Integer, tmFirstAppt As
51 Date, tmLastAppt As Date) As String
52 ' Aufgabe: Am übergebenen Tag (dtmDate) alle verfügbaren Zeitfenster zwischen den schon
53 vorhandenen Terminen finden.
54 ' Argumente: dtmAppt Das zu durchsuchende Datum im Outlook-Kalender
55 ' iDuration Normale Dauer eines Termine in Minuten
56 ' tmFirstAppt Beginn des ersten Termins
57 ' tmLastAppt Ende des letzten Termins
58 ' Annahmen: Das freie Zeitfenster muss ...
59 ' * größer oder gleich der normalen Dauer eines Termins sein.
60 ' * zwischen dem normalen Beginn- und Ende der Arbeitszeit eines Tages liegen.
61 Const conFctName As String = "FindAvailableTimeSlots" ' Name der Funktionsprozedur
62 Dim objOL As New Outlook.Application ' Outlook-Anwendung
63 Dim olNS As Outlook.Namespace ' Basis aller Ordner d. jeweiligen
64 Namespace
65 Dim olFldr As Outlook.MAPIFolder ' Verzeichnis
66 Dim olItems As Outlook.items ' Sammlung von Terminen
67 Dim olAppt As Object ' einzelner Termin
68 Dim dtmNext As Date ' Nächster verfügbarer Termin
69 Dim intDuration As Integer ' Dauer der freien Lücke zwischen Terminen
70 Dim strFullDay As String ' Voller Tag
71 Dim strList As String ' Liste der verfügbaren Zeitfenster
72 On Error GoTo Error_Point
73 ' Anfangswert setzen
74 strList = ""
75 ' Den vollen Zeitraum des ausgewählten Tages bestimmen
76 strFullDay = "[Start] >= ' " & dtmAppt & "' and " & "[Start] < ' " & dtmAppt & " 11:59 pm'"
77 Set olNS = objOL.GetNamespace("MAPI")
78 On Error GoTo Error_Point

```

```

79 Set olFldr = olNS.GetDefaultFolder(olFolderCalendar)
80 Set olItems = olFldr.items
81 dtmNext = tmFirstAppt
82 ' Termine sortieren (erforderlich wg. "IncludeRecurrences", s. u.)
83 olItems.Sort "[Start]"
84 ' Wiederkehrende Termine einbeziehen
85 olItems.IncludeRecurrences = True
86 ' Die Termine so sortieren, dass nur alle Termine des relevanten Tages berücksichtigt werden.
87 Set olItems = olItems.Restrict(strFullDay)
88 ' Die Termine erneut sortieren , um wiederkehrende Termine in die richtige Reihenfolge zu bringen.
89 olItems.Sort "[Start]"
90 With olItems
91     ' Jeweils Betreff, Starttermin und Dauer ermitteln
92     Set olAppt = .GetFirst
93     Do While TypeName(olAppt) <> "Nothing"
94         ' Verfügbare Zeitfenster finden
95         Select Case DateValue(dtmAppt) ' Zeichenfolge in einen gültigen Datumswert umwandeln.
96             Case DateValue(Format(olAppt.Start, "dd.mm.yyyy"))
97                 If Format(dtmNext, "Hh:Nn") < Format(olAppt.Start, "Hh:Nn") Then
98                     ' Die zeitliche Lücke bestimmen bevor der nächste Termin beginnt.
99                     If Format(olAppt.Start, "Hh:Nn") < Format(tmLastAppt, "Hh:Nn") Then
100                         intDuration = DateDiff("n", dtmNext, Format(olAppt.Start, "Hh:Nn"))
101                     Else
102                         intDuration = DateDiff("n", dtmNext, Format(tmLastAppt, "Hh:Nn"))
103                     End If
104                     ' Passt ein Termin in das Zeitfenster?
105                     If intDuration >= iDuration Then
106                         strList = strList & Format(dtmNext, "hh:mm") & "|" & _
107                             & Format(DateAdd("n",intDuration,dtmNext), "hh:mm") & ";"
108                     End If
109                 End If
110                 ' Das erste verfügbare Zeitfenster nach dem Termin finden.
111                 dtmNext = DateAdd("n", olAppt.Duration + intDuration, dtmNext)
112                 ' Nicht über den letztmöglichen Termin hinausgehen!
113                 If dtmNext > tmLastAppt Then
114                     Exit Do
115                 End If
116             End Select
117             intDuration = 0
118             Set olAppt = .GetNext
119         Loop
120     End With
121     ' Den Rest des Tages bestimmen.
122     intDuration = DateDiff("n", dtmNext, Format(tmLastAppt, "Hh:Nn"))
123     If intDuration >= iDuration Then
124         strList = strList & Format(dtmNext, "hh:mm") & _
125             & Format(DateAdd("n", intDuration, dtmNext), "hh:mm") & ";"
126     End If
127     FindAvailableTimeSlots = strList
128     Me.cmdFind.Enabled = False
129 Exit_Point:
130 On Error Resume Next
131 Set olAppt = Nothing
132 Set olItems = Nothing
133 Set olFldr = Nothing
134 Set olNS = Nothing
135 Set objOL = Nothing
136 Exit Function
137 Error_Point:
138 MsgBox Err.Number & ": " & conFctName & vbCrLf & Err.Description
139 Resume Exit_Point
140 End Function
141 Private Function GetBusyStatus(intBusyStatus As Integer) As String
142     ' Verfügbarkeit ermitteln.
143     Select Case intBusyStatus
144         Case olFree
145             GetBusyStatus = "verfügbar"
146         Case olTentative
147             GetBusyStatus = "mit Vorbehalt"
148         Case olBusy
149             GetBusyStatus = "beschäftigt"
150         Case olOutOfOffice
151             GetBusyStatus = "abwesend"
152         Case Else
153             GetBusyStatus = "unbekannt (" & intBusyStatus & ")"
154     End Select
155 End Function
156

```

```

157 Private Function Check_Integer(Tbox As MSForms.TextBox) As Boolean
158     ' Minuten prüfen.
159     Dim strMsg As String
160     With Tbox
161         If .Text = "" Then
162             strMsg = "Normale Dauer ist Pflichtfeld!"
163         ElseIf Not IsNumeric(.Text) Then
164             strMsg = "Eingabe ist nicht nummerisch!"
165         ElseIf CInt(.Text) <> .Text Then
166             strMsg = "Ganze Minuten sind erforderlich!"
167         End If
168     End With
169     If Len(strMsg) > 0 Then
170         MsgBox strMsg, vbExclamation, "Eingabefehler"
171         Check_Integer = True ' --> springt wieder in das Eingabefeld zurück
172     End If
173 End Function
174 Private Function Check_Time(Tbox As MSForms.TextBox) As Boolean
175     ' Zeitangabe prüfen.
176     Dim strMsg As String
177     With Tbox
178         If .Text = vbNullString Then
179             strMsg = "Zeitabgabe ist Pflichtfeld!"
180         ElseIf Len(Trim(.Text)) <> 5 Then
181             strMsg = "Länge der Zeitabgabe ungleich 5!"
182         ElseIf InStr(3, .Text, ":", vbTextCompare) = 0 Then
183             strMsg = "Zeitabgabe enthält keinen Doppelpunkt!"
184         End If
185     End With
186     If Len(strMsg) > 0 Then
187         MsgBox strMsg, vbExclamation, "Eingabefehler"
188         Check_Time = True ' --> springt wieder in das Eingabefeld zurück
189     End If
190 End Function
191 Private Sub cmdFind_Click()
192     ' Verfügbare Zeitfenster bestimmen
193     Dim strResult As String
194     strResult = FindAvailableTimeSlots(Me.txtDate, Me.txtDuration, Me.txtStartTime, Me.txtEndTime)
195     Dim varResult As Variant
196     varResult = Split(strResult, ";")
197     If IsArray(varResult) Then
198         With Me.cboTimeSlots
199             .BackColor = RGB(255, 255, 255)
200             .Enabled = True
201             .List = varResult
202             .ListIndex = -1
203             .ListIndex = 0 ' 1. Eintrag d. Kombinationsfelds anzeigen
204         End With
205         With Me
206             .Hide
207             With .txtDate ' Datum
208                 .Enabled = False
209                 .BackColor = RGB(211, 211, 211)
210             End With
211             With .txtDuration ' Dauer
212                 .Enabled = False
213                 .BackColor = RGB(211, 211, 211)
214             End With
215             With .txtStartTime ' Beginn
216                 .Enabled = False
217                 .BackColor = RGB(211, 211, 211)
218             End With
219             With .txtEndTime ' Ende
220                 .Enabled = False
221                 .BackColor = RGB(211, 211, 211)
222             End With
223             .Show
224         End With
225     End If
226     Unload Me
227 End Sub
228

```

```

229 Private Sub cmdCreate_Click()
230     Dim newAppt As Outlook.AppointmentItem
231     Set newAppt = Application.CreateItem(olAppointmentItem)
232     With newAppt
233         .Subject = Me.txtTopic.Value           ' Betreff
234         .Location = Me.txtWhere.Value         ' Ort
235         .AllDayEvent = False                  ' Kein ganztägiger Termin
236         .BusyStatus = olBusy                 ' Termin besetzt
237         .Categories = "Gelbe Kategorie"      ' Kategorie
238         .Start = Me.txtDate & " " & Left(varTm(0), 5) ' Beginn
239         .ReminderSet = Me.chkRemind          ' Erinnerung (Ja/Nein)
240         .Duration = Me.txtDuration.Value     ' Dauer in Minuten
241         .Body = Me.txtComment                ' Kommentar (unformatierter Text)
242         .Save                               ' Termin speichern
243     End With
244     With Me
245         .cmdCreate.Enabled = False
246         .cmdFind.Enabled = False
247     End With
248     Dim strMsg As String
249     With newAppt
250         strMsg = "Einmaliger Termin erfolgreich generiert!" & vbCrLf & _
251             "-----" & vbCrLf & _
252             "Betreff: " & .Subject & vbCrLf & _
253             "Ort: " & .Location & vbCrLf & _
254             "Beginn: " & .Start & vbCrLf & _
255             "Ende: " & .End & vbCrLf & _
256             "Dauer: " & .Duration & " Minuten" & vbCrLf & _
257             "Ganztägiger Termin: " & .AllDayEvent & vbCrLf & _
258             "Beschäftigungsstatus: " & GetBusyStatus(.BusyStatus) & vbCrLf & _
259             "Kategorie: " & .Categories & vbCrLf & _
260             "Erinnerung: " & .ReminderSet & vbCrLf & _
261             "Textkörper: " & .Body & vbCrLf
262     End With
263     MsgBox strMsg, vbInformation + vbOKOnly, "Termin erstellen"
264     Set newAppt = Nothing
265 End Sub
266 Private Sub cmdCancel_Click()
267     Me.txtTopic = ""           ' Betreff
268     Me.txtWhere = ""          ' Ort
269     Me.txtBegin = ""         ' Beginn
270     Me.txtLength = ""        ' Dauer
271     Me.chkRemind = False     ' Erinnerung
272     Me.txtComment = ""       ' Kommentar
273     Unload Me                ' Benutzerformular aus Speicher entfernen
274 End Sub
275 Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
276     ' Verhindern, dass das Benutzerformular geschlossen wird, wenn auf das obere rechte X geklickt wird.
277     If CloseMode = vbFormControlMenu Then
278         MsgBox "Bitte das Formular mit der Schaltfläche 'Abbrechen' schließen!"
279         Cancel = True
280     End If
281 End Sub
282 Private Sub txtDate_Exit(ByVal Cancel As MSForms.ReturnBoolean)
283     With Me.txtDate
284         If Not IsDate(.Text) And Not .Text = vbNullString Then
285             .Text = ""
286             MsgBox "Bitte ein gültiges Datum eingeben!", vbExclamation, "Eingabefehler"
287             Cancel = True ' ---> springt wieder in das Eingabefeld zurück
288         Else
289             On Error Resume Next
290             .Text = CDate(.Text)
291         End If
292     End With
293 End Sub
294 Private Sub txtStartTime_Exit(ByVal Cancel As MSForms.ReturnBoolean)
295     Cancel = Check_Time(txtStartTime) ' True --> springt wieder in das Eingabefeld zurück
296 End Sub
297 Private Sub txtEndTime_Exit(ByVal Cancel As MSForms.ReturnBoolean)
298     Cancel = Check_Time(txtEndTime) ' True --> springt wieder in das Eingabefeld zurück
299 End Sub
300 Private Sub txtTopic_Exit(ByVal Cancel As MSForms.ReturnBoolean)
301     If Len(Trim(Me.txtTopic.Value)) = 0 Then
302         MsgBox "Betreff ist Pflichtfeld!", vbExclamation, "Eingabefehler"
303         Cancel = True
304     End If
305 End Sub
306 Private Sub txtWhere_Exit(ByVal Cancel As MSForms.ReturnBoolean)

```

```

307     If Len(Trim(Me.txtWhere.Value)) = 0 Then
308         MsgBox "Ort ist Pflichtfeld!", vbExclamation, "Eingabefehler"
309         Cancel = True
310     End If
311 End Sub
312 Private Sub txtDuration_Exit(ByVal Cancel As MSForms.ReturnBoolean)
313     Cancel = Check_Integer(txtDuration)
314 End Sub
315 Private Sub txtEndTime_AfterUpdate()
316     On Error Resume Next
317     If CDate(Me.txtEndTime.Text) <= CDate(Me.txtStartTime.Text) Then
318         MsgBox "Ende vor/gleich Beginn!", vbExclamation, "Eingabefehler"
319     End If
320 End Sub
321 Private Sub cboTimeSlots_Change()
322     ' Zeitfenster ändert sich.
323     If Me.cboTimeSlots.ListIndex = -1 Then
324         MsgBox "Nichts ausgewählt!", vbExclamation, "Zeitfenster"
325     Else
326         ' Untergeordnete Zeichenfolgen im Kombinationsfeld f. Zeitfenster bestimmen.
327         varTm = Split(Me.cboTimeSlots.Value, "|")
328         ' Formularfelder aufbereiten
329         With Me
330             .cmdCreate.Enabled = True
331             With .txtTopic
332                 .Enabled = True
333                 .BackColor = RGB(255, 255, 255)
334                 .Value = "Neuer Termin"
335             End With
336             With .txtWhere
337                 .Enabled = True
338                 .BackColor = RGB(255, 255, 255)
339                 .Value = "Praxis"
340             End With
341             With .txtComment
342                 .Enabled = True
343                 .BackColor = RGB(255, 255, 255)
344                 .Value = "Kommentar"
345             End With
346             With .chkRemind
347                 .Enabled = True
348                 .BackColor = RGB(255, 255, 255)
349                 .Value = False
350             End With
351             .txtBegin.Value = Me.txtDate & " " & Left(varTm(0), 5)
352             .txtLength.Value = .txtDuration.Value
353         End With
354     End If
355 End Sub

```

Listing 1: Gesamter VBA-Quellcode der Anwendung

## 7 Literaturverzeichnis

- [1] o.V., „Search through a series of shared Calendars in Outlook for a free timeslot,“ o. J. [Online]. Available: [http://www.vbaexpress.com/kb/getarticle.php?kb\\_id=671](http://www.vbaexpress.com/kb/getarticle.php?kb_id=671). [Zugriff am 16 11 2019].