

Kapitel I

Kalenderblatt erstellen

In diesem Kapitel:

Die Aufgabenstellung	2
Diskussion zum Code	4
Zusammenfassung	9

In Kapitel 5 wurden die Grundlagen der Arbeit mit dem Word-Objektmodell vorgestellt. Die vorliegende Lösung veranschaulicht die unten stehenden Objekte, Eigenschaften und Methoden, wobei der Schwerpunkt auf der Arbeit mit Tabellen liegt. Zudem wird deutlich, wie flexibel es sich mit Datumsangaben in VBA arbeiten lässt, im Gegensatz zu den Feldfunktionen von Word (siehe den Abschnitt »Feldfunktionen« in Kapitel 7).

Im Blickpunkt:**VB-Funktionen**

`CVDate`

`DateAdd, DateDiff`

`Format`

Word-Objekte

`Application.CentimetersToPoints`

`Document.PageSetup.Zoom`

`FormFields.Add, FormField.Name`

`Table.Add, Table.Borders, Table.Shading, Table.Style, Table.Condition`

`Row.HeightRule, Row.Height`

`Cell.Merge, Cell.VerticalAlignment`

Die Aufgabenstellung

Ein oft geäußelter Wunsch ist ein Werkzeug, um Monats-Kalenderblätter zu erstellen. Als Anregung stellen wir hier eine Lösung vor, die auf einer Tabelle basiert. Das Blatt aus Abbildung I.1 eignet sich sowohl für den Online-Gebrauch wie auch für den Ausdruck auf Papier. Für die Dateneingabe online (in Word) enthält jede nicht dunkel schattierte Zelle ein Formularfeld, um zu gewährleisten, dass der Text an der richtigen Stelle eingegeben wird. Der Dokumentschutz stellt sicher, dass die Kalenderstrukturen nicht unbeabsichtigt geändert werden. Für die Ausgabe auf einen Drucker kann die Feldschattierung ausgeschaltet werden.

Als Benutzerschnittstelle für die Eingabe der Monats- und Jahresangaben dient entweder eine `InputBox` oder eine `UserForm` mit Kalender-Steuerelement, wie in Abbildung I.2 dargestellt. Nach Bestätigung durch den Benutzer wird die Eingabe auf Gültigkeit geprüft. Weiter rechnet der Code aus, an welchem Wochentag der Monat beginnt und wie viele Tage er hat. Darauf basierend wird eine Tabelle erstellt und mit Tagesnamen und -daten gefüllt.

Abbildg. I.1 Ein Kalenderblatt mit Textformularfeldern zur Eingabe von Terminen und Notizen

Abbildg. I.2 Eine InputBox oder eine UserForm mit Kalender-Steuerelement nimmt die Benutzereingaben auf.

Leere Zellen am Monatsanfang oder -ende werden verbunden, um eine Fläche für allgemeine Notizen bereitzustellen. Zuletzt werden die Schrift-, Rahmen und Schattierungsformatierungen zugewiesen.

HINWEIS

Das in diesem Beispiel verwendete Kalender-Steuerelement, *MSCAL.ocx*, gehört zum Lieferumfang von Microsoft Access aus Office Professional und muss explizit installiert werden. Sie dürfen es nur dort einsetzen, wo Office Professional installiert ist; es darf nicht frei weitergereicht werden. Wo dieses Steuerelement nicht vorhanden ist, müssen Sie eine InputBox oder andere Methode verwenden.

Diskussion zum Code

Die Prozedur *KalenderErstellen* blendet die Schnittstelle für die Benutzereingabe ein. Die Zeichenkette aus dem Eingabefeld (InputBox) wird mit der Funktion *CVDate* in ein Datum umgewandelt (Date ist ein Datentyp wie Long oder String), woraus der Wochentag für den ersten Tag des Monats ermittelt wird. Dabei entspricht »1« dem Sonntag – erster Tag bzw. erste Spalte im Kalenderblatt. Eine Kombination der Funktionen *DateDiff* und *DateAdd* berechnet die Anzahl der Tage eines Monats. *DateDiff* kalkuliert die Anzahl der Tage (»d«), die zwischen dem eingegebenen Datum und diesem Datum plus einem Monat liegen.

Danach wird für den Kalender ein neues Dokument erstellt und im DIN A4-Querformat mit ein Zentimeter breiten Seitenrändern eingerichtet. Sie können auch ein beliebiges anderes Papierformat wählen. Im Beispielmakro passt sich die Tabelle dem Papier an. Wird sie dadurch für das von Ihnen gewählte Papier zu klein, müssen Sie unter Umständen die Schriftgröße der ersten Zeile etwas verringern.

Die Funktion *KalenderTabelleEinfügen* wird nun aufgerufen, um die Grundtabelle samt Wochentagen einzufügen. Sie gibt diese Tabelle als Objektvariable zurück; die Formatierung wird damit getrennt vorgenommen. In *KalenderTabelleEinfügen* sehen Sie, wie die angesprochene Anpassung an die Papiergröße erfolgt. Der Parameter *DefaultTableBehavior:=wdWord8TableBehavior* ist unabdingbar, um automatisch gleich breite Spalten über die ganze Seitenbreite zu erhalten, die sich in der Breite dem Inhalt nicht anpassen. Um die passende Zeilenhöhe zu bekommen, wird die Papierhöhe minus der Randbreite durch die Anzahl der Zeilen dividiert. Achten Sie auf die *HeightRule*-Eigenschaft für Tabellenzeilen: Sie müssen diese auf *wdRowHeightExactly* oder *wdRowHeightAtLeast* setzen, um die Einstellung für die Zeilenhöhe wirksam zu machen.

HINWEIS

Die Wochentage werden automatisch in der Standardsprache von Windows in den Kalender eingefügt.

Zurück in *KalenderErstellen*, werden die Monattage nummeriert. Die Prozedur fängt mit dem Wochentag an, der dem ersten des Monats entspricht, und schleift für jeden Tag im Monat durch die Tabellenzellen. In jeder Zelle wird die Tageszahl fett formatiert, gefolgt von einem Formularfeld in einer neuen Zeile. Das Formularfeld wird umbenannt, so dass es mit dem Datum bezeichnet ist (MMM_DD_YYYY). Dies erleichtert die Datenübertragung zwischen Word und anderen Anwendungen wie Outlook, falls Sie die Funktionalität ausbauen möchten.

Wenn alle Monate 28 Tage hätten, wäre die Erstellung eines Kalenders direkter: Alles würde immer in fünf Zeilen passen. Es gibt aber immer Ausreißer, die uns das Leben schwer machen: Monate mit 30 oder 31 Tagen, die an einem Freitag oder Samstag beginnen. Um in solchen Fällen Platz in der letzten Zeile zu schaffen, wird die Notizfläche in die zweite Tabellenzeile versetzt. Egal wo sie sich befindet, werden vier Zellen für diese Fläche verbunden – ebenso für den Monatsnamen, unten

rechts. Außerdem werden Textmarken gesetzt, damit das Makro später auf diese Zellen zurückgreifen kann.

Jetzt kommt die Formatierung dran. Wenn die Tabellenformatvorlage in Ihrer Dokumentvorlage bereits vorhanden wäre, könnten Sie diese Aufgabe mit einer einzigen Befehlszeile erledigen. Sie müssten lediglich die Anweisung

```
tbl.Style = TabelleFormatvorlage(doc)
```

mit

```
tbl.Style = "Name der Tabellenformatvorlage"
```

ersetzen. Nun soll dieses Beispiel auch die Erstellung einer Tabellenformatvorlage veranschaulichen, also ...

Die Tabellenformatvorlage wird von der Prozedur *TabelleFormatvorlage* erstellt. Um eine neue Formatvorlage des Typs »Tabelle« zu bestimmen, genügt es, dem Parameter *Type* der *Add*-Methode den Konstantenwert *wdStyleTypeTable* zu übergeben. Die allgemeinen Attribute wie Schriftart und -größe werden wie für jede Formatvorlage oder wie für einen Bereich festgelegt. Auch die Tabelleneigenschaften, wie Rahmen oder Schattierungen, werden nicht anders gehandhabt als für die Formatierung einer Tabelle.

Unterschiedlich ist jedoch die Definition der AutoFormat-spezifischen Teile: oberste und unterste Zeile (*wdFirstRow* bzw. *wdLastRow*), Eckzellen (*wdCell1SE*) und Streifen (*wdEvenRowBanding*). Diese werden durch die neue Eigenschaft *Condition* bestimmt.

WICHTIG

Die Streifen kommen in VBA erst zum Vorschein, wenn Sie einen Wert für die Eigenschaft *RowStripe* bzw. *ColumnStripe* setzen. In der Benutzeroberfläche übernimmt Word diese Aufgabe automatisch, sobald eine Formatierung für ungerade oder gerade Zeilen oder Spalten festgelegt wurde.

Die Festlegung der Rahmenlinien ist der heikelste Teil der Formatvorlagendefinition, egal ob sie in der Benutzeroberfläche oder über VBA erfolgt. Fangen Sie unbedingt mit den Einstellungen für die ganze Tabelle an, gefolgt von den Randreihen und -spalten und schließlich den Eckzellen. Sobald ein Element angesprochen wird, überschreiben seine Attribute die geerbten der »höheren« Stufe. Deshalb werden alle *.Borders* für jeden Tabellenteil einzeln und ausdrücklich gesetzt.

Formatierungen des Layouts können nicht über eine Tabellenformatvorlage bestimmt werden, weshalb wir sie anschließend in *KalenderErstellen* vornehmen. Zum Schluss folgen Feinarbeiten am eigentlichen Dokument. Die letzte Absatzmarke, die einen Seitenumbruch verursacht, kann nicht gelöscht werden, weil sie seit Word 2000 Informationen über die Tabellen-Positionierung speichert. Formatieren wir sie jedoch mit der Schriftgröße »1 Punkt«, bleibt sie auf der gleichen Seite mit der Tabelle.

Falls Sie es vorziehen, keine Formularfelder in die Tabelle einzufügen, können Sie die vorletzte Zeile des Makros löschen, da der Dokumentschutz nicht benötigt wird.

Listing I.1 Den Kalender erstellen

```

Option Explicit

Sub KalenderErstellen()
    Dim doc As Word.Document
    Dim tbl As Word.Table
    Dim ffld As Word.FormField
    Dim rng As Word.Range
    Dim cel As Word.Cell
    Dim strDatum As String
    Dim lWochenTag As Long
    Dim lAnzTage As Long
    Dim dat As Date
    Dim sngCM1 As Single
    Dim lZaehler As Long
    Dim frm As frmKalender

    sngCM1 = CentimetersToPoints(1)
    Application.ScreenUpdating = False

    'Mit UserForm und Kalender-Steuerelement
    ' Set frm = New frmKalender
    ' frm.Show
    ' If frm.Tag = "Abbrechen" Then
    '     Unload frm
    '     Exit Sub
    ' End If
    ' strDatum = CStr(frm.cal.Value)
    ' Debug.Print strDatum
    ' Unload frm

    'Schleifen, bis die Eingabe ein gültiges Datum ist.
    Do
        strDatum = InputBox("Datum im Format 'Monat Jahr' eingeben:")
        If strDatum = "" Then Exit Sub
    Loop While Not IsDate(strDatum)

    dat = CDate(strDatum)
    lWochenTag = Weekday(dat)
    lAnzTage = DateDiff("d", dat, DateAdd("m", 1, dat))

    'Dokument einfügen und formatieren.
    Set doc = Documents.Add
    With doc.PageSetup
        .PaperSize = wdPaperA4
        .Orientation = wdOrientLandscape
        .RightMargin = sngCM1
        .LeftMargin = sngCM1
        .TopMargin = sngCM1
        .BottomMargin = sngCM1
    End With

    Set tbl = KalenderTabelleEinfügen(doc)
    'In der Zelle anfangen, die dem Wochentag des ersten Tags des Monats entspricht.
    tbl.Rows(2).Cells(lWochenTag).Select
    'Die Tage nummerieren und ein Formularfeld für die Benutzereingabe einfügen.
    For lZaehler = 1 To lAnzTage

```

Listing I.1 Den Kalender erstellen (Fortsetzung)

```

Selection.Collapse wdCollapseStart
Selection.Font.Bold = True
Selection.TypeText lZaehler
Selection.Font.Bold = False
Selection.TypeText Chr$(11)
Set ffld = Selection.FormFields.Add(
    Range:=Selection.Range, Type:=wdFieldFormTextInput)
'Formularfeld mit dem Datum benennen.
ffld.Name = Format(dat, "MMM_DD YYYY")
Selection.Cells(1).Range.Next(wdCell, 1).Select
Next lZaehler

'Wenn ein Monat (außer Feb) an einem Freitag oder Samstag beginnt,
'stehen die Notizen oben statt unten links.
If (lWochenTag = 6 Or lWochenTag = 7) And lAnzTage >= 30 Then
    'Zelle verbinden, um Platz für Notizen zu schaffen.
    Set cel = tbl.Cell(2, 1)
    cel.Merge MergeTo:=tbl.Cell(2, 5)
    cel.Range.Bookmarks.Add Name:="Notizen", Range:=cel.Range
    'Monatsbezeichnung steht unten rechts.
    Set cel = tbl.Cell(tbl.Rows.Count, 5)
    cel.Merge MergeTo:=tbl.Cell(tbl.Rows.Count, 7)
    doc.Bookmarks.Add Name:="Monat", Range:=cel.Range
Else
    'Zelle verbinden, um Platz für Notizen zu schaffen.
    Set cel = tbl.Cell(tbl.Rows.Count, 1)
    cel.Merge MergeTo:=tbl.Cell(tbl.Rows.Count, 4)
    cel.Range.Bookmarks.Add Name:="Notizen", Range:=cel.Range
    'Monatsbezeichnung steht unten rechts.
    Set cel = tbl.Cell(tbl.Rows.Count, 2)
    cel.Merge MergeTo:=tbl.Cell(tbl.Rows.Count, 4)
    doc.Bookmarks.Add Name:="Monat", Range:=cel.Range
End If

'Formatvorlage im Dokument erstellen und der Tabelle zuweisen.
tbl.Style = TabelleFormatvorlage(doc)
'Monat und Jahr in die Zelle unten rechts einfügen.
With doc.Range.Bookmarks("Monat").Range
    .Text = Format(dat, "MMMM") & " " & Format(dat, "yyyy")
    'Die Formatvorlage kann die Eckzellen-Formatierung nicht anzeigen.
    'Weil die erste und letzte Spalte keine besondere Formatierung hat,
    'müssen wir die Zelle ausdrücklich formatieren.
    .Cells(1).Shading.BackgroundPatternColor = wdColorGray70
    .Font.Size = 24
    .Font.ColorIndex = wdWhite
    .ParagraphFormat.Alignment = wdAlignParagraphRight
    .Cells.VerticalAlignment = wdCellAlignVerticalBottom
End With

'Notizenzeile bezeichnen und Formularfeld einfügen.
Set rng = doc.Bookmarks("Notizen").Range
rng.Text = "Notizen: "
rng.Collapse wdCollapseEnd
rng.MoveEnd wdCharacter, -1
doc.FormFields.Add Range:=rng, Type:=wdFieldFormTextInput

```

Listing I.1 Den Kalender erstellen *(Fortsetzung)*

```

'Erste Reihe mit den Tagesnamen formatieren.
'Über einer Formatvorlage können wir die vertikale Ausrichtung nicht bestimmen.
tbl.Rows(1).Cells.VerticalAlignment = wdCellAlignVerticalCenter
'Die Höhe etwas verringern, sodass die Tabelle nicht über die Seite bricht.
tbl.Rows(1).Height = tbl.Rows(1).Height - 10

'Aus dem gleichen Grunde die letzte Absatzmarke klein formatieren.
doc.Paragraphs.Last.Range.Font.Size = 1
'Die ganze Seite anzeigen.
doc.ActiveWindow.Panes(1).Zooms(wdPrintView).PageFit = wdPageFitFullPage
'An Dokumentanfang springen.
Selection.HomeKey wdStory
'Das Dokument schützen, sodass die Formularfelder aktiv sind.
doc.Protect wdAllowOnlyFormFields
End Sub

Function KalenderTabelleEinfügen(doc As Word.Document) As Word.Table
    Dim tbl As Word.Table
    Dim cel As Word.Cell
    Dim lAnzZeilen As Long
    Dim lAnzSpalten As Long
    Dim lZaehler As Long

    lAnzZeilen = 7
    lAnzSpalten = 7
    Set tbl = doc.Tables.Add(Range:=doc.Range, NumRows:=lAnzZeilen, _
        NumColumns:=lAnzSpalten, DefaultTableBehavior:=wdWord8TableBehavior)
    tbl.Rows.HeightRule = wdRowHeightExactly
    With doc.PageSetup
        tbl.Rows.Height = (.PageHeight - .TopMargin - .BottomMargin) / lAnzZeilen
    End With
    For Each cel In tbl.Rows(1).Cells
        lZaehler = lZaehler + 1
        cel.Range.Text = Format(Weekday(lZaehler), "dddd")
    Next cel
    Set KalenderTabelleEinfügen = tbl
End Function

Function TabelleFormatvorlage(doc As Word.Document) As String
    Dim sty As Word.Style
    Dim strTFV As String, strFVFont

    strTFV = "KalenderFV"
    strFVFont = "Arial"
    Set sty = doc.Styles.Add(Name:=strTFV, Type:=wdStyleTypeTable)
    sty.Font.Name = strFVFont
    sty.Font.Size = 9
    With sty.Table
        .Borders.OutsideLineStyle = wdLineStyleDouble
        .Borders.OutsideLineStyle = wdLineStyleSingle
        With .Condition(wdEvenRowBanding)
            .Shading.Texture = wdTextureNone
            .Shading.BackgroundPatternColor = wdColorGray10
            .Borders(wdBorderLeft).LineStyle = wdLineStyleDouble
            .Borders(wdBorderRight).LineStyle = wdLineStyleDouble
        End With
    End With
End Function

```


Listing I.1 Den Kalender erstellen (Fortsetzung)

```

        .Borders(wdBorderBottom).LineStyle = wdLineStyleSingle
        .Borders(wdBorderTop).LineStyle = wdLineStyleSingle
    End With
    With .Condition(wdFirstRow)
        .Shading.BackgroundPatternColor = wdColorGray70
        .Font.Size = 16
        .Font.ColorIndex = wdWhite
        .Font.Bold = True
        .Borders(wdBorderBottom).LineStyle = wdLineStyleSingle
        .Borders(wdBorderLeft).LineStyle = wdLineStyleDouble
        .Borders(wdBorderRight).LineStyle = wdLineStyleDouble
        .Borders(wdBorderTop).LineStyle = wdLineStyleDouble
    End With
    With .Condition(wdFirstColumn)
        .Font.Size = 9
    End With
    With .Condition(wdLastColumn)
    End With
    With .Condition(wdLastRow)
        .Shading.BackgroundPatternColor = wdColorAutomatic
        .Borders(wdBorderBottom).LineStyle = wdLineStyleDouble
    End With
    With .Condition(wdSECell)
        .Shading.BackgroundPatternColor = wdColorGray70
        .Font.Size = 24
        .Font.ColorIndex = wdWhite
        .ParagraphFormat.Alignment = wdAlignParagraphRight
        .Borders(wdBorderBottom).LineStyle = wdLineStyleDouble
        .Borders(wdBorderRight).LineStyle = wdLineStyleDouble
    End With
    .RowStripe = 1
End With

TabelleFormatvorlage = strTFV
End Function

```



Die Beispieldatei *Bspl_01.dot* finden Sie auf der CD-ROM zum Buch im Ordner *\Beispiele\Kapl.*

Zusammenfassung

Anhand des Lösungsvorschlags, wie ein Online-Kalendarblatt zu erstellen ist, wurden einige Aspekte des Word-Objektmodells näher vorgestellt. Unter anderen wurden hervorgehoben:

- Tabellen und Tabellenformatvorlagen
- Der Umgang mit einigen Date-Funktionen der VB-Sprache
- Formularfelder und der Dokumentschutz

