

# Internationale Standardbuchnummer (ISBN) überprüfen

Volker Thormählen, Sep. 2021

## Inhalt

1	Aufgabenstellung .....	2
2	Teile einer ISBN-13 Buchnummer.....	2
3	Prüfziffernberechnung .....	2
4	VBA-Prozeduren .....	3
5	Quellcode der VBA-Prozeduren.....	3
6	Erweiterung der Anwendung .....	3

## Listings

Listing 1:	Subprozedur 'ISBN_Checker' .....	4
Listing 2:	Funktionsprozedur 'fctGetCheckDigit' .....	5

## Tabellen

Tabelle 1:	Zahlenbeispiel zur Prüfziffernberechnung .....	2
------------	--	---

## 1 Aufgabenstellung

Die Richtigkeit einer *Internationalen Standardbuchnummer* (ISBN) soll mithilfe von zwei VBA-Prozeduren überprüft werden.

Zur Identifizierung von Büchern wird die *Internationale Standardbuchnummer* (ISBN) verwendet. Sie war früher 10-stellig. Wegen des knapper werdenden Nummernbereichs wurde seit 1. Januar 2007 die Angabe einer 13-stelligen ISBN zur Pflicht.

## 2 Teile einer ISBN-13 Buchnummer

Eine ISBN-13 Nummer enthält stets 5 Nummernteile:

1. Präfix
2. Gruppennummer
3. Verlagsnummer
4. Band-/Titelnummer
5. Prüfziffer

Diese fünf Nummernteile werden gewöhnlich mit je einem Bindestrich gegliedert. Eine ISBN-13-Buchnummer ist also stets 13-stellig.

## 3 Prüfziffernberechnung

Gegeben sei folgende ISBN Buchnummer:

*Wesp, Sandra: Probleme beim wissenschaftlichen Schreiben. München: GRIN Verlag, 2003, ISBN 978-3-656-16673-3*

Die Prüfziffer dieser ISBN Buchnummer wird mit den ersten 12 Datenstellen bestimmt (siehe Tabelle 1):

Datenstelle	Inhalt	Gewicht	Produkt	Produktsumme
(1)	(2)	(3)	(4) = (2) x (3)	(5) = (4) kumuliert
1	9	1	9	9
2	7	3	21	20
3	8	1	8	38
4	3	3	9	47
5	6	1	6	53
6	5	3	15	68
7	6	1	6	74
8	1	3	3	77
9	6	1	6	83
10	6	3	18	101
11	7	1	7	108
12	3	3	9	117
117 Mod 10 = 7; <b>Prüfziffer: 10 – 7 = 3</b>				

Tabelle 1: Zahlenbeispiel zur Prüfziffernberechnung

1. Die Datenstellen werden von links nach rechts mit 1 bis 12 benummert.
2. Der Inhalt der geraden/ungeraden Datenstellen wird abwechselnd mit 1 bzw. 3 multipliziert.
3. Das jeweilige mathematische Produkt steht in Sp. 4.
4. Die Produkte werden addiert.
5. Auf die Produktsumme (*hier 117*) wird die Modulo 10 Funktion angewandt.

6. Der ganzzahlige Divisionsrest (*hier 7*) wird von 10 subtrahiert. Die Differenz ergibt die Prüfziffer (*hier 3*).
7. Falls die Differenz 10 ist, dann ist die Prüfziffer 0.

## 4 VBA-Prozeduren

Die Subprozedur 'ISBN\_Checker()' steuert die Anwendung (sog. Treiberprozedur), s. Listing 1.

Die Funktionsprozedur 'fctGetCheckDigit' führt die Prüfziffernberechnung aus, s. Listing 2.

## 5 Quellcode der VBA-Prozeduren

Siehe weiter unten, Listing 1 u. Listing 2.

## 6 Erweiterung der Anwendung

Die beschriebene Anwendung kann mit etwas Arbeitsaufwand so erweitert werden, dass im Literatur- und/oder Quellenverzeichnis einer wiss. Arbeit (sog. *Zieldokument*) alle ISBN-17 Buchnummern gefunden und anschließend auf Richtigkeit überprüft werden. Die erweiterte Anwendung umfasst folgende drei Lösungsschritte:

1. Das WORD-*Zieldokument* im Dialog mit dem Benutzer auswählen. Dafür eignet sich z. B. folgende Codesequenz:

```
With Application.FileDialog(msoFileDialogFilePicker) ... End With
```

2. Das ausgewählte WORD-*Zieldokument* nach *ISBN-17 Buchnummern* durchsuchen. Dafür eignet sich der in WORD standardmäßig eingebaute Find-Befehl:

```
With rng.Find
    .Text = "[ISBN ][0-9-]{17}" ' => 13 Datenstellen + 4 Gliederungszeichen
    .MatchWildcards = True
    ...
    Do While .Execute(Forward:=True)
        MsgBox fctGetCheckDigit(rng.Text)
    Loop
End With
```

3. Jede gefundene ISBN-17 Buchnummer mit der Funktionsprozedur *fctGetCheckDigit* auf Gültigkeit prüfen, s. oben, 2. Schritt.

```

1 Private Sub ISBN_Checker()
2   ' Bei EAN-13 ist das Präfix je nach Buchtitel 978 oder 979.
3   Application.ScreenUpdating = False
4   Dim strBookNum As String
5   Dim strISBN As String
6   Dim strPZ As String * 1
7   ' 1. Beispiel
8   strISBN = "978-3-8218-1252-4" ' Buchtitel : Sag's mit Witz
9   strBookNum = strISBN
10  strPZ = Right(strISBN, 1) ' Prüfziffer
11  If fctGetCheckDigit(strISBN) = strPZ Then
12    MsgBox "ISBN ist richtig.", vbExclamation, "ISBN " & strBookNum
13  Else
14    MsgBox "ISBN ist falsch.", vbExclamation, "ISBN " & strBookNum
15  End If
16  ' 2. Beispiel
17  strISBN = "978-3-499-13599-X" ' Buchtitel: Fräulein Smillas Gespür für Schnee
18  strBookNum = strISBN
19  strPZ = Right(strISBN, 1) ' Prüfziffer
20  If fctGetCheckDigit(strISBN) = strPZ Then
21    MsgBox "ISBN ist richtig.", vbExclamation, "ISBN " & strBookNum
22  Else
23    MsgBox "ISBN ist falsch.", vbExclamation, "ISBN " & strBookNum
24  End If
25  ' 3. Beispiel
26  strISBN = "978-3-640-73061-5" ' Buchtitel: Schreiben wissenschaftlicher Texte
27  strBookNum = strISBN
28  strPZ = Right(strISBN, 1) ' Prüfziffer
29  If fctGetCheckDigit(strISBN) = strPZ Then
30    MsgBox "ISBN ist richtig.", vbExclamation, "ISBN " & strBookNum
31  Else
32    MsgBox "ISBN ist falsch.", vbExclamation, "ISBN " & strBookNum
33  End If
34  Application.ScreenUpdating = True
35 End Sub

```

*Listing 1: Subprozedur 'ISBN\_Checker'*

```

1 Private Function fctGetCheckDigit(strISBN As String) As String
2     Dim intLoop As Integer, lngSum As Long
3     strISBN = Replace(strISBN, "-", "") ' Gliederungszeichen (Bindestriche) entfernen
4     If Len(strISBN) > 10 Then
5         strISBN = Mid(strISBN, 4)
6     End If
7     If Len(strISBN) = 10 Then
8         For intLoop = 1 To Len(strISBN) - 1
9             lngSum = lngSum + (11 - intLoop) * CLng(Mid(strISBN, intLoop, 1))
10        Next intLoop
11        fctGetCheckDigit = (11 - (lngSum Mod 11)) Mod 11
12        If fctGetCheckDigit = 10 Then fctGetCheckDigit = "X"
13    End If
14 End Function

```

*Listing 2: Funktionsprozedur 'fctGetCheckDigit'*