

IBAN erfassen und formal prüfen

Volker Thormählen, 17. Juni 2021

Inhaltsverzeichnis

1	Problem.....	2
2	Lösung.....	2
1.1	VBA-Prozeduren.....	2
1.2	Ablauf.....	2
1.3	Einzelheiten.....	2
2	Quellcode der Anwendung.....	4
3	Alternativer Quellcode der Anwendung.....	6

Abbildungen

Abbildung 1:	InputBox zur Erfassung einer deutschen IBAN in 4er-Blöcken.....	3
Abbildung 2:	Meldung bei erfolgreicher Erfassung einer deutschen IBAN.....	3
Abbildung 3:	Meldung bei erfolgreicher Überprüfung der erfassten IBAN.....	3
Abbildung 4:	Ausgabe der Laufzeit der Anwendung.....	3

Listings

Listing 1:	Subprozedur IBAN_Pruefen.....	4
Listing 2:	Subprozedur IBAN_Inhalt_Pruefen.....	4
Listing 3:	Funktionsprozedur verifyIBAN.....	5
Listing 4:	Funktionsprozedur Modulo.....	6
Listing 5:	Funktionsprozedur ConvertToNumber.....	6
Listing 6:	Subprozedur ElapsedTime.....	6
Listing 7:	Subprozedur ToString.....	6
Listing 8:	Treiber-Subprozedur der Anwendung.....	7
Listing 9:	IBAN formal auf Gültigkeit prüfen.....	7
Listing 10:	MOD-Operation bei großen Zahlen durchführen.....	8

Tabellen

Tabelle 1:	Typ, Name und Aufgabe der Prozeduren der Anwendung.....	2
Tabelle 2:	Typ, Name und Aufgabe des alternativen Quellcodes der Anwendung.....	6

1 Problem

In diesem Beitrag wird ausführlich beschrieben, wie eine internationale Bankkontonummer (Kürzel: IBAN) in Rahmen einer Visual-Basic-Anwendung

- von Benutzer manuell erfasst und
- automatisch auf Gültigkeit überprüft werden kann.

Zur Abwicklung des Zahlungsverkehrs wird gewöhnlich die jeweils zutreffende internationale Bankkontonummer benötigt.

Eine *deutsche* IBAN ist genau 22 Stellen lang und besteht aus folgenden Nummernteilen:

- Ländercode (2 Großbuchstaben: DE)
- Prüfziffer (2 Ziffern)
- Bankleitzahl (8 Ziffern)
- Kontonummer (10 Ziffern). Kürzere werden linksseitig mit Nullen ausgefüllt.

2 Lösung

1.1 VBA-Prozeduren

Die Anwendung benötigt drei Subprozeduren und vier Funktionsprozeduren (s. Tabelle 1):

VBA-Prozedur		
Typ	Name	Aufgabe
Subprozedur	IBAN_Pruefen	Erfassung der IBAN durch den Benutzer prüfen.
Subprozedur	IBAN_Inhalt_Pruefen	IBAN auf Gültigkeit prüfen.
Funktionsprozedur	verifyIBAN	Übergebene IBAN auf Gültigkeit prüfen.
Funktionsprozedur	Modulo	Restwert für lange Zahlen bestimmen.
Funktionsprozedur	ConvertToNumber	Buchstaben in Zahlen umwandeln.
Subprozedur	ElapsedTime	Laufzeit der Anwendung ermitteln und ausgeben.
Funktionsprozedur	ToString	1-stellige Zahl mit führender Null zurückgeben.

Tabelle 1: Typ, Name und Aufgabe der Prozeduren der Anwendung

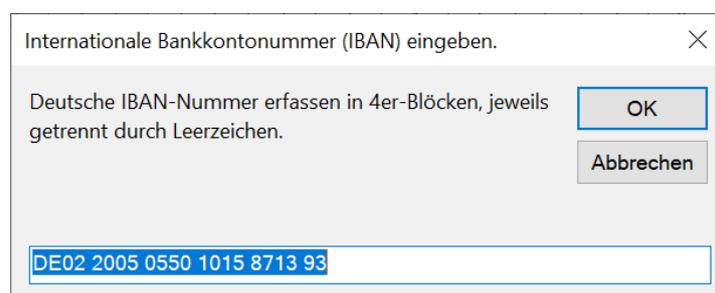
1.2 Ablauf

Die Anwendung unterstützt zwei wesentliche Aufgaben:

- Die Erfassung einer IBAN durch den Benutzer
- Die Überprüfung der erfassten IBAN auf Gültigkeit

1.3 Einzelheiten

- Die Erfassung einer deutschen IBAN erfolgt mit der in VBA eingebauten InputBox-Funktion (s. Abbildung 1):



The screenshot shows a standard VBA InputBox dialog. The title bar reads 'Internationale Bankkontonummer (IBAN) eingeben.' with a close button (X) on the right. The main text area contains the instruction: 'Deutsche IBAN-Nummer erfassen in 4er-Blöcken, jeweils getrennt durch Leerzeichen.' Below this text is a text input field containing the example IBAN 'DE02 2005 0550 1015 8713 93'. At the bottom right of the dialog are two buttons: 'OK' and 'Abbrechen'.

Abbildung 1: InputBox zur Erfassung einer deutschen IBAN in 4er-Blöcken

- Die erfolgreiche Erfassung einer IBAN wird mit dieser Meldung (s. Abbildung 2) durch die Anwendung quittiert:

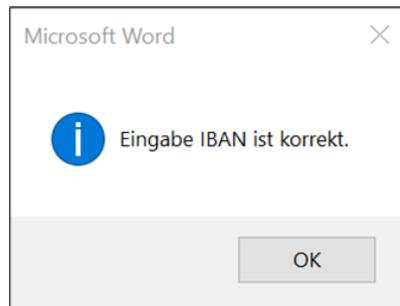


Abbildung 2: Meldung bei erfolgreicher Erfassung einer deutschen IBAN

- Die erfolgreiche Überprüfung der erfassten IBAN wird von der Anwendung wie folgt gemeldet (s. Abbildung 3):

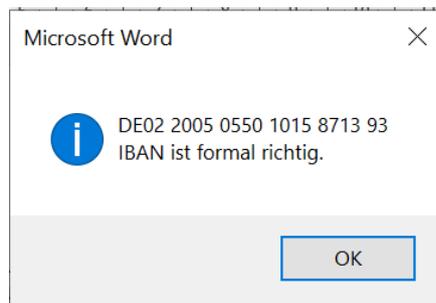


Abbildung 3: Meldung bei erfolgreicher Überprüfung der erfassten IBAN

- Schließlich wird die Laufzeit der Anwendung in ganzen Sekunden angezeigt.

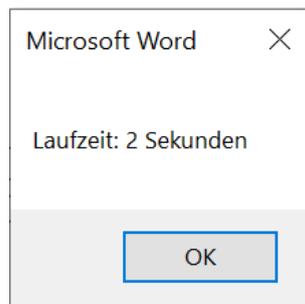


Abbildung 4: Ausgabe der Laufzeit der Anwendung

- Buchstaben in Zahlen umwandeln.

2 Quellcode der Anwendung

```
Private Sub IBAN_Pruefen()  
    ' Aufgabe: Erfassung der IBAN durch den Benutzer prüfen.  
    Dim isOK As Boolean ' boolescher Fehlerindikator  
    Dim strPrompt As String ' Meldung  
    Dim strTitle As String ' Überschrift  
    strPrompt = "Deutsche IBAN erfassen in 4er-Blöcken von links nach rechts," & vbCrLf & _  
        " jeweils getrennt durch 1 Leerzeichen."  
    strDefault = "DE07 1234 1234 1234 12"  
    strTitle = "Internationale Bankkontonummer (IBAN) eingeben."  
    isOK = False  
    Do While isOK = False  
        varInhalt = InputBox(strPrompt, strTitle, strDefault)  
        If varInhalt Like "[A-X][A-X]## #### #### #### #### ##" Then  
            MsgBox "Eingabe IBAN ist korrekt.", vbInformation  
            isOK = True  
            ' Erfasste IBAN ist OK. Jetzt ihren Inhalt auf formale Gültigkeit prüfen  
            IBAN_Inhalt_Pruefen varInhalt  
        Else  
            If varInhalt = vbNullString Then  
                MsgBox "Eingabe ist falsch oder abgebrochen durch Benutzer!", vbExclamation  
                Exit Sub  
            Else  
                MsgBox "Eingabe IBAN ist nicht korrekt!", vbExclamation  
            End If  
        End If  
    Loop  
End Sub
```

Listing 1: Subprozedur IBAN_Pruefen

```
Sub IBAN_Inhalt_Pruefen(ByVal strIBAN As Variant)  
    ' Aufgabe: Internationale Bankkontonummer (IBAN) auf Gültigkeit prüfen.  
    Dim dtmStart As Date ' Startzeit d. Prozedur  
    Dim dtmEnd As Date ' Endezeit d. Prozedur  
    ' Startzeit d. Prozedur  
    dtmStart = Time  
    ' IBAN auf Gültigkeit prüfen  
    verifyIBAN strIBAN  
    ' Endezeit d. Prozedur  
    dtmEnd = Time  
    ' Laufzeit ausgeben  
    ElapsedTime dtmStart, dtmEnd  
End Sub
```

Listing 2: Subprozedur IBAN_Inhalt_Pruefen

```
Function verifyIBAN(ByVal strIBAN As String) As Boolean  
    ' Aufgabe: Übergebene IBAN auf Gültigkeit überprüfen.  
    Dim lngLen As Long  
    Dim lngLoop As Long  
    Dim strChr As String  
    Dim strTmp As String  
  
    On Error GoTo ERR_HANDLER  
  
    ' Schritt 1: Übergebene IBAN ggf. bereinigen  
    If Left(strIBAN, 6) = "IBAN:" Then  
        strTmp = Mid(strIBAN, 7)  
    ElseIf Left(strIBAN, 5) = "IBAN:" Then  
        strTmp = Mid(strIBAN, 6)  
    ElseIf Left(strIBAN, 4) = "IBAN" Then  
        strTmp = Mid(strIBAN, 5)  
    Else  
        strTmp = LTrim(strIBAN)  
    End If  
  
    ' Schritt 2: Alle Leerzeichen in strTmp entfernen  
    strTmp = Replace(strTmp, " ", "")  
    ' Minimale (maximale) Länge der IBAN beträgt 5 (34) Zeichen.  
    If Len(strTmp) < 5 Or Len(strTmp) > 34 Then
```

```

    verifyIBAN = False
    GoTo ERR_EXIT
End If

' Schritt 3: 2-stelliges alphabetisches Länderkürzel auswerten
Select Case Left(strTmp, 2)
    Case "NO"
        lngLen = 15
    Case "BE", "NO"
        lngLen = 16
    Case "DK", "FI", "NL"
        lngLen = 18
    Case "SI"
        lngLen = 19
    Case "AT", "EE", "LT", "LU"
        lngLen = 20
    Case "CH", "LV"
        lngLen = 21
    Case "DE", "IE", "GB"
        lngLen = 22
    Case "GI"
        lngLen = 23
    Case "AD", "CZ", "ES", "SE", "SK"
        lngLen = 24
    Case "PT"
        lngLen = 25
    Case "IS"
        lngLen = 26
    Case "FR", "IT", "GR"
        lngLen = 27
    Case "CY", "HU", "PL"
        lngLen = 28
    Case Else
        verifyIBAN = False
        GoTo ERR_EXIT
End Select

' Schritt 4: strTmp rechts mit Nullen auffüllen, um IBAN-Länge gemäß Länderkürzel zu erhalten.
strTmp = strTmp + String(lngLen - Len(strTmp), "0")

' Schritt 5: Länderkürzel und Prüfziffer zusammenfassen
strTemp = Mid(strTmp, 5) & Left(strTmp, 4)

' Schritt 6: Buchstaben in Zahlen umwandeln.
strTemp = ConvertToNumber(strTemp)

' Schritt 7: strTemp mit Mod 97 prüfen.
' Wenn das Ergebnis gleich 1 ist, dann ist die IBAN richtig, sonst falsch.
If Modulo(strTemp, 97) = 1 Then
    verifyIBAN = True
    MsgBox strIBAN & vbCrLf & "IBAN ist formal richtig.", vbInformation
Else
    verifyIBAN = False
    MsgBox strIBAN & vbCrLf & "IBAN ist formal falsch!", vbExclamation
End If
ERR_EXIT:
Exit Function
ERR_HANDLER:
verifyIBAN = False
MsgBox Err.Description
GoTo ERR_EXIT
End Function

```

Listing 3: Funktionsprozedur verifyIBAN

```

Private Function Modulo(strModulus As String, intDivisor As Integer) As String
    ' Aufgabe: Restwert für lange Zahlen in 9-Blöcken bestimmen.
    Dim intLoop As Integer ' Schleifenvariable
    Dim lngDividend As Long
    Dim lngRest As Long ' Divisionsrest
    Dim strDividend As String
    Dim strDivRest As String
    Dim strRest As String
    strRest = ""
    strDivRest = strModulus
    For intLoop = 0 To 4

```

```

    strDividend = Left(strRest + strDivRest, 9)
    lngDividend = CLng(Val(strDividend))
    lngRest = lngDividend Mod intDivisor
    strDivRest = Mid(strRest + strDivRest, 10)
    strRest = ToString(CStr(lngRest))
    If strDivRest = "" Then Exit For
Next intLoop
Modulo = strRest
End Function

```

Listing 4: Funktionsprozedur Modulo

```

Private Function ConvertToNumber(ByVal strTmp As String) As String
    ' Aufgabe: Buchstaben in Zahlen umwandeln.
    Dim lngLoop As Long ' Schleifenvariable
    Dim strChr As String ' alphabetisches Zeichen (Buchstabe)
    Dim strConverted As String ' Numerisches Zeichen
    strConverted = ""
    For lngLoop = 1 To Len(strTmp)
        strChr = Mid(strTmp, lngLoop, 1)
        If strChr >= "A" And strChr <= "Z" Then
            strChr = CStr(Asc(strChr) - 55)
        End If
        strConverted = strConverted & strChr
    Next lngLoop
    ConvertToNumber = strConverted
End Function

```

Listing 5: Funktionsprozedur ConvertToNumber

```

Sub ElapsedTime(dtmStart As Date, dtmEnd As Date)
    ' Aufgabe: Laufzeit der Anwendung in Sekunden ermitteln und ausgeben.
    Dim dtmInterval As Date
    dtmInterval = dtmEnd - dtmStart
    MsgBox "Laufzeit: " & Int(CSng(dtmInterval * 24 * 3600)) & " Sekunden"
End Sub

```

Listing 6: Subprozedur ElapsedTime

```

Public Function ToString(ByVal lngPart As Long) As String
    ' Aufgabe: 1-stellige Zahl mit führender Null zurückgeben.
    ToString = Format(CStr(lngPart), "00")
End Function

```

Listing 7: Funktionsprozedur ToString

3 Alternativer Quellcode der Anwendung

VBA-Prozedur		
Typ	Name	Aufgabe
Subprozedur	Driver	Treiber-Prozedur der Anwendung
Funktionsprozedur	IBANOK	IBAN auf Gültigkeit prüfen.
Funktionsprozedur	ModuloBig	MOD-Operation bei großen Zahlen durchführen.

Tabelle 2: Typ, Name und Aufgabe der alternativen Prozeduren der Anwendung

```

Sub Driver()
    ' Aufgabe: Gültigkeit einer internationalen Bankkontonummer (IBAN) prüfen.
    Const conResult As String = "Die IBAN ist "
    Dim blnResult As Boolean ' Boolesches Ergebnis der Gültigkeitsprüfung
    Dim strIBAN As String ' Internationale Bankkontonummer (IBAN)
    strIBAN = "DE29 3345 0000 1040 9743 78"
    blnResult = IBANOK(strIBAN) ' Gültigkeitsprüfung der IBAN mit Funktionsprozedur.
    ' Das Ergebnis muss 'wahr' sein, sonst ist die IBAN falsch.
    If blnResult = True Then
        MsgBox conResult & "richtig!", vbExclamation
    Else
        MsgBox conResult & "falsch!", vbExclamation
    End If
End Sub

```

Listing 8: Treiber-Subprozedur der Anwendung

```

Public Function IBANOK(ByVal strIBAN As String) As Boolean
    ' Aufgabe: Gültigkeit der übergebenen IBAN prüfen.
    Dim lngLoop As Long ' Schleifenvariable
    ' 1. Schritt: Alle Leerzeichen aus IBAN entfernen.
    strIBAN = Replace(strIBAN, " ", "")
    ' 2. Schritt: Die minimale Länge der IBAN prüfen.
    If Len(strIBAN) < 5 Then
        IBANOK = False
        Exit Function
    End If
    ' 3. Schritt: IBAN umstellen: Ländercode u. Prüfziffern an das Ende der IBAN
    ' verschieben.
    strIBAN = Mid(strIBAN, 5) & Left(strIBAN, 4)
    ' 4. Schritt: Alle Großbuchstaben (A, B, ..., Z) durch eine Zahl ersetzen, die ihre
    ' Position in der ASCII-Tabelle repräsentiert, plus 9
    For lngLoop = 65 To 90
        strIBAN = Replace(strIBAN, Chr(lngLoop), lngLoop - 64 + 9, , , vbTextCompare)
    Next lngLoop
    ' 5. Schritt: Restwert berechnen, der sich bei der ganzzahligen Division von
    ' strIBAN durch 97 ergibt (d. h. Modulo 97). Das Ergebnis muss 1 sein,
    ' sonst ist die IBAN falsch.
    IBANOK = ModuloBig(strIBAN, 97) = 1
End Function

```

Listing 9: IBAN formal auf Gültigkeit prüfen

```

Function ModuloBig(strNumber As String, lngDiv As Long) As String
    ' Aufgabe: MOD-Operation bei großen Zahlen durchführen.
    Dim lngStart As Long ' Beginn
    Dim lngEnd As Long ' Ende
    Dim lngNumb As Long ' große Zahl
    Dim lngErg As Long ' Ergebnis
    Dim lngRest As Long ' Rest
    Dim strErg As String ' Ergebnis
    Dim strPZ As String ' Prüfziffern
    lngStart = 1: lngEnd = 1
    Do While lngEnd <= Len(strNumber)
        lngNumb = Val(lngRest & Mid(strNumber, lngStart, lngEnd - lngStart + 1))
        If lngNumb >= lngDiv Then
            lngErg = lngNumb \ lngDiv ' ganzzahlige Division
            lngRest = lngNumb - lngErg * lngDiv
            strErg = strErg & CStr(lngErg)
            lngStart = lngEnd + 1
            lngEnd = lngStart
        Else
            If strErg <> "" Then
                strErg = strErg & "0"
            End If
            lngEnd = lngEnd + 1
        End If
    Loop
    If lngStart <= Len(strNumber) Then
        lngRest = Val(lngRest & Mid(strNumber, lngStart))
    End If
    strPZ = ToString(lngRest)
    ModuloBig = strPZ
End Function

```

Listing 10: MOD-Operation bei großen Zahlen durchführen