

12. ArcView-Anwendertreffen 2010

Workshop „Programmierung in ArcGIS“

Daniel Fuchs



Wo kann eigene Programmierung in ArcGIS verwendet werden?

- 1) Makros für die Automatisierung einzelner Arbeitsschritte im Kartenfenster von ArcGIS – Programmiersprache ist hier VBA («Visual Basic für Anwendungen»).
- 2) Skripte als Bestandteil von Werkzeugen in der Toolbox – mehrere mögliche Programmiersprachen: Python (ArcPy), VBScript, Perl, JScript.
- 3) Programmcode für die Berechnung von Feldwerten in Attributtabelle – Programmiersprache VBA.
- 4) berechnete Ausdrücke für komplexere Beschriftungen – Programmiersprache VBScript oder JScript.

Workshop „Programmierung in ArcGIS“



Visual Basic für Anwendungen – Überblick

VBA ist eine von Microsoft entwickelte Programmiersprache für die Anpassung von Anwendungen. Am bekanntesten ist sie vielen Nutzern vielleicht über die Aufzeichnung von Makros in den Office-Programmen (Word, Excel, Access ...).

Ein Vorteil von VBA ist die Verbreitung in zahlreichen verschiedenen Anwendungen: neben Office und ArcGIS z. B. auch AutoCAD oder CorelDraw. Das bedeutet:

- einzelne Funktionen können zwischen verschiedenen Anwendungen kopiert werden,
- Anwendungen können über vordefinierte Schnittstellen miteinander kommunizieren: z. B. können Datenbankabfragen in Access aus ArcGIS heraus aufgerufen werden.

Workshop „Programmierung in ArcGIS“



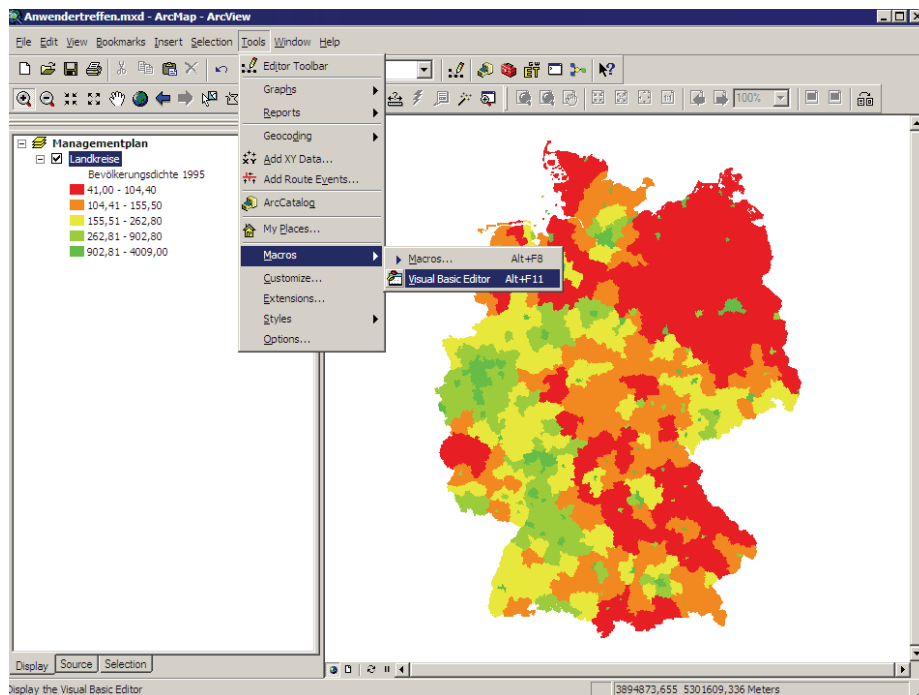
VBA in ArcGIS – die Programmierumgebung

Als Programmierumgebung steht in ArcGIS der »Visual Basic Editor« zur Verfügung.

Eigene Programmierung wird im Visual Basic Editor als Text eingegeben. Dieser Text enthält Befehle, die vom Programm in Handlungen umgesetzt werden. Eine Gruppe von Befehlen, die zusammen ausgeführt werden sollen, werden als »Prozedur« oder »Funktion« unter einem Namen zusammengefasst. VBA kennzeichnet solche Programmblöcke durch die Kennwörter »Sub« und »Function«. Im Unterschied zu Prozeduren können Funktionen einen errechneten Wert an andere Funktionen oder Prozeduren weitergeben.

Workshop „Programmierung in ArcGIS“





VBA in ArcGIS – die Programmierumgebung

Für die Programmierarbeit stellt der VB-Editor Hilfsmittel zur Verfügung:

- Kontexthilfe zu VBA über die Funktionstaste F1,
- Ausführung des Programms innerhalb des Editors (Funktionstaste F5), ggf. auch schrittweise (Funktionstaste F8)
- Setzen von Unterbrechungen, an denen die Ausführung angehalten wird und ggf. auf Fehler untersucht werden kann.
- Automatische Vervollständigung von Schlüsselwörtern und Variablen (Leertaste).

VBA – eingebaute Funktionen

Für die Programmierarbeit stellt VBA eine Reihe von eingebauten Funktionen zur Verfügung, die als Bestandteile des Codes verwendet werden können (komplette Liste: VBA Hilfe – Inhalt – Functions).

- Kommunikation mit dem Nutzer: MsgBox und InputBox.
- Datum und Zeit: Date, Time, Now, Day, Year ...
- Berechnungen von Zahlen: +, -, *, /, Exp, Sin, Log, Sgn, Sqr ...
- Verarbeitung von Text: &, Left, Right, Mid, Format, InStr, Len ...
- Prüfung von Werten: IsDate, IsNumeric, IsNull ...
- Umwandlung von Werten: Val, Str, Format.

VBA – eingebaute Funktionen

Die meisten Funktionen müssen beim Aufruf einen oder mehrere Parameter erhalten, die von der Funktion verarbeitet werden:

```
MsgBox "Hallo"  
Format(49.6)
```

Oft kommen zu den verarbeiteten Parametern weitere (optionale), die das Ergebnis der Funktion weiter verändern:

```
MsgBox "Hallo", vbYesNo, "Ausgabe"  
Format(49.6, "0.00 %")
```

VBA – Arbeiten mit Variablen

Variablen dienen dazu, Werte zu speichern, die während des Programmablaufs verarbeitet werden sollen. Sie müssen zunächst eingerichtet oder definiert werden, bevor ihnen ein Wert zugewiesen werden kann. Dies geschieht mit dem Schlüsselwort »Dim ... As«.

```
Dim MeineZahl As Double
Dim MeinErgebnis As Double
Dim MeinText As String
MeineZahl = 45.9
MeinErgebnis = MeineZahl * 2
MeinText = "Das Ergebnis ist "
MsgBox MeinText & MeinErgebnis
```

VBA – Arbeiten mit Variablen

VBA verwendet verschiedene Datentypen, die den Variablen zugewiesen werden können:

- **String**: Text, gekennzeichnet durch Anführungszeichen "".
- **Integer**: Ganze Zahlen
- **(Single und) Double**: Dezimalzahlen
- **Date**: Datums- und Zeitangaben, gekennzeichnet durch Nummernzeichen # #.
- **Boolean**: Wahr-Falsch (in Zahlen: -1 und 0).
- **Var**: Variant, d. h. alles mögliche

VBA – Arbeiten mit Variablen

```
Dim MeinDatum As Date
Datum = #3/9/2010#

Dim MeineZahl As Double
MeineZahl = 45.6

Dim MeineZweiteZahl As Integer
MeineZweiteZahl = 2

Dim MeinText As String
MeinText = "Mein Ergebnis ist "

Dim JaNein As Boolean
JaNein = True
```

VBA – den Programmablauf strukturieren

Befehle und Funktionen werden normalerweise nacheinander ausgeführt. Oft ist es aber sinnvoll oder notwendig, diesen Ablauf zu strukturieren. Dafür gibt es »Programmschleifen«:

1. Prüfung einer Bedingung und je nach Ergebnis verschiedene Abläufe: If ... then ... else ... end if.

```
If MeineZahl < 50 Then
    MeineZweiteZahl = MeineZahl * 2
Else
    MeineZweiteZahl = MeineZahl * 1.5
End If
```

VBA – den Programmablauf strukturieren

Befehle und Funktionen werden normalerweise nacheinander ausgeführt. Oft ist es aber sinnvoll oder notwendig, diesen Ablauf zu strukturieren. Dafür gibt es »Programmschleifen«:

2. Mehrmaliges Durchführen einer Anweisung, bis eine Bedingung erfüllt ist: Do ... until ... loop.

```
Do Until MeineZahl > 50
    MeineZahl = MeineZahl + 1
Loop

MsgBox "Meine Zahl ist jetzt " & MeineZahl
```

VBA – den Programmablauf strukturieren

Befehle und Funktionen werden normalerweise nacheinander ausgeführt. Oft ist es aber sinnvoll oder notwendig, diesen Ablauf zu strukturieren. Dafür gibt es »Programmschleifen«:

3. Abarbeiten einer festen Anzahl von Elementen: For ... next.

```
For MeineZahl = 1 To 10
    MsgBox "Meine Zahl ist " & MeineZahl
Next MeineZahl
```

Zusammenfassung VBA allgemein

Variablen

```
Dim MeineZahl As Double
Dim MeineZweiteZahl As Integer
Dim MeinText As String
```

Funktionen

```
MeineZahl = InputBox("Bitte Zahl eingeben:")
If MeineZahl > 50 Then
    MeineZweiteZahl = MeineZahl / 2
Else
    MeineZweiteZahl = MeineZahl * 2
End If
```

Programmschleife

```
MeinText = "Ergebnis: " & Format(MeineZweiteZahl, "0.00")
MsgBox MeinText
```

Aufgabe 1: Grundfunktionen VBA

Erstellen Sie eine Prozedur, die prüft, ob eine vom Benutzer eingegebene Zahl kleiner oder größer als 1 ist, und diese Zahl je nachdem als Prozentwert oder als gerundete Ganzzahl wieder ausgibt.

Verwenden Sie dabei die Funktionen **Format()** und **Round()** und prüfen Sie die notwendigen und optionalen Parameter für jede Funktion mit der VBA-Hilfe.

Verwenden Sie außerdem die Konstruktion **If ... then ... else ... end if**.

Prüfen Sie Ihre Funktion, indem Sie sie als Makro aus ArcGIS heraus aufrufen.

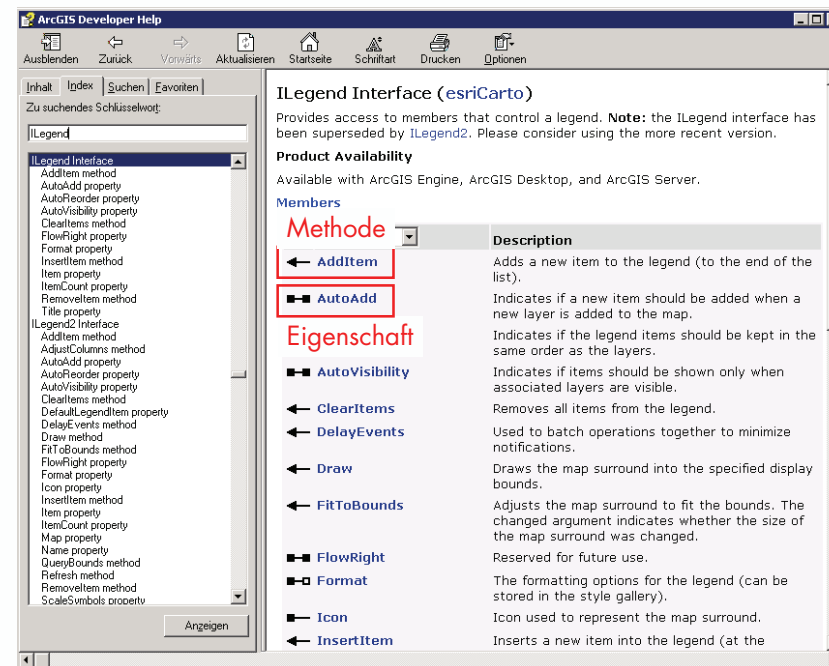
VBA in ArcGIS: Objekte, Eigenschaften, Methoden

Den eigentlichen Ansatzpunkt für die Programmierung in ArcGIS liefern die »Objekte«. Objekte sind Elemente der GIS-Anwendung, die in der Programmierung verwendet werden können.

Datenrahmen, Layer, Legenden, Grafiken, Beschriftungen – all das sind Objekte. Jedes Objekt in einer Programmierumgebung hat dabei »Eigenschaften« und »Methoden«.

Das Objekt Map (Datenrahmen) hat z. B. die Eigenschaften MapScale, MapUnits oder Name und die Methoden SelectByShape, Deletelayer oder AddLayer.

Das Objekt Legend (Legende) hat z. B. die Eigenschaften Map, Name und AutoAdd und die Methoden AddItem, ClearItems und Refresh.



VBA in ArcGIS: Objekte, Eigenschaften, Methoden

```
Dim pMxDocument As IMxDocument  
Set pMxDocument = Application.Document
```

```
Dim pLayer As ILayer  
Set pLayer = pMxDocument.SelectedLayer
```

```
If Not TypeOf pLayer Is IGeoFeatureLayer Then  
    MsgBox "Kein Featurelayer ausgewählt!", vbCritical,  
    "Abbruch"  
    Exit Sub  
End If
```

VBA in ArcGIS: Objekte, Eigenschaften, Methoden

Set: wird verwendet, wenn einer Variablen nicht ein Wert, sondern die Referenz (der Bezug) auf ein Objekt zugewiesen wird.

Type Of: wird verwendet, um die Art einer Variablen zu prüfen (z. B. ob ein Layer im Datenrahmen ein Feature- oder ein Rasterlayer ist).

Exit Sub: wird verwendet, um eine Prozedur vor dem vollständigen Ablauf abubrechen, z. B. weil ein erwarteter Wert nicht übergeben wurden (ähnlich: **Exit For**, um eine Programmschleife abubrechen).

VBA in ArcGIS: Objekte, Eigenschaften, Methoden

```
Dim pMxDocument As IMxDocument
Set pMxDocument = Application.Document

Dim pLayer As ILayer
Set pLayer = pMxDocument.SelectedLayer

If Not TypeOf pLayer Is IFeatureLayer Then
    MsgBox "Kein Featurelayer ausgewählt!", vbCritical,
    "Abbruch"
    Exit Sub
End If

Dim pFDef As IFeatureLayerDefinition
Set pFDef = pLayer
pFDef.DefinitionExpression = "DENSIT_J95 > 902.91"
pMxDocument.ActiveView.Refresh
```

Aufgabe 2

Ändern Sie die Prozedur »FeatureDefinition« so ab, dass nicht die Bevölkerungsdichte, sondern die Anzahl der Motorräder als Grundlage verwendet wird, und dass außerdem der Nutzer eingeben soll, welcher Grenzwert für die FeatureDefinition verwendet werden soll.