

24.8.2 Geo

Die Klasse `Geo` (`gb.map`) stellt Werkzeuge zum Bearbeiten von Karten-Projektionen und zur Formatierung von geografischen Daten bereit.

24.8.2.1 Methoden

Die Klasse `Geo` verfügt über diese Methoden:

Methoden	Rückgabotyp	Beschreibung
<code>DecToSex(Value As Float, Typ As Integer)</code>	String	Die Funktion formatiert den Wert in einen sexagesimalen Wert. Für den Typ gilt: Latitude = 1 und Longitude = 2.
<code>SexToDec(Value as String) As Float</code>	Float	Die Funktion berechnet einen Breiten- oder Längenwert aus einem geeignet formatierten sexagesimalen (String-)Wert.
<code>MapPointToPixel(hMapPoint as MapPoint, Zoom As Integer)</code>	Point	Die Funktion konvertiert die Koordinaten eines geographischen Punktes (Breite/Länge) in einen Bildpunkt zu einer bestimmten Zoom-Stufe.
<code>MapPointToTile(hMapPoint as MapPoint, Zoom As Integer)</code>	Point	Die Funktion konvertiert die Koordinaten eines geographischen Punktes (Breite/Länge) in Karten-Koordinaten (xTile/yTile) zu einer bestimmten Zoomstufe.
<code>PixelToMapPoint(hPoint As Point, Zoom As Integer)</code>	MapPoint	Die Funktion wandelt die Koordinaten eines Punktes in Pixel-Koordinaten bei einem bestimmten Zoom-Faktor in die geografischen Koordinaten Breite und Länge.
<code>PixelToTile(hPoint As Point)</code>	Point	Die Funktion konvertiert Pixel-Koordinaten in Karten-Koordinaten.
<code>TileBounds (X As Integer, y As Integer, Zoom As Integer)</code>	MapBounds	Gibt einen <code>MapBounds</code> -Objekt zu einer bestimmten Kachel (xTile / yTile) mit gegebenem Zoom-Faktor zurück.
<code>TileToMapPoint (X As Integer, y As Integer, Zoom As Integer)</code>	MapPoint	Gibt den linken oberen <code>MapPoint</code> zu einer bestimmten Kachel (xTile / yTile) mit gegebenem Zoom-Faktor zurück.

Tabelle 24.8.2.1.1 : Methoden der Klasse `Geo`

24.8.3 Map


Die Klasse `Map` enthält die Routinen, um die Karten zu zeichnen. Nach der Initialisierung haben Sie die Möglichkeit den Kartenausschnitt zu ändern sowie den Zoom-Faktor zu verändern. Hier ein Beispiel dafür, wie Sie eine Karte initialisieren und anzeigen:

```
Public Sub ShowMap()
    Dim iZoom As Integer
    Dim fLatitude, fLongitude As Float
    Dim sCacheName, sTileName, sTilePattern As String
    Dim cArguments As New Collection

    ' Initialisierung
    iZoom = 13
    sTileName = "OpenStreetMap"
    sTilePattern = "85.30.190.241/{z}/{x}/{y}.png"
    cArguments = Null
    sCacheName = Null
    fLatitude = 52.7905 ' °Breite Osterburg
    fLongitude = 11.7531 ' °Länge Osterburg

    MapView1.Map.AddTile(sTileName, sTilePattern, cArguments, sCacheName)
    MapView1.Map[sTileName].Copyright = " © OpenStreetMap"
    MapView1.Map[sTileName].Visible = True ' optional; Standard-Einstellung ist TRUE
    MapView1.Map.AddShape("P1")
    MapView1.Map!P1.AddPoint("Osterburg", MapPoint(fLatitude, fLongitude))
    MapView1.Map.Center = MapPoint(fLatitude, fLongitude)
End Sub
```

```
' MapView1.Map.Center = MapView1.Map!P1!Osterburg.Points ' Alternative Zentrierung auf die Marke P1
  MapView1.Map.Zoom = iZoom
  MapView1.AllowEffect = True
End ' ShowMap()
```

Mit diesen beiden Zeilen aus dem o.a. Quelltext-Ausschnitt fügen Sie eine markante Markierung  in die angezeigte Karte ein:

```
MapView1.Map.AddShape("P1")
MapView1.Map!P1.AddPoint("Osterburg", MapPoint(fLatitude, fLongitude))
```