

### 24.8.3 MapPoint

Die Klasse *MapPoint* definiert geografische Punkte (Länge, Breite) sowie Methoden, um mit den geografischen Punkten im 'World Geodetic System 1984' (WGS84 oder EPSG:4326) zu arbeiten.

Wenn Sie noch nie mit geografischen Koordinaten gearbeitet haben, dann hilft ein Blick auf diese Webseiten weiter:

- <http://spatialreference.org/ref/epsg/4326/>
- <http://epsg.io/4326>
- <http://www.kompf.de/gps/distcalc.html>

#### 24.8.3.1 Geografische Koordinaten

- Die geografischen Koordinaten eines Ortes auf der fiktiven Erd-Kugel sind die Breite (Latitude) und die Länge (Longitude).
- Die geografische Breite und die geografische Länge sind Winkelwerte. Die Angabe der Winkel erfolgt entweder in Dezimal-Schreibweise oder in Sexagesimal-Schreibweise (Basis 60), wobei 1 Grad 60 (Bogen-)Minuten hat und eine Minute 60 (Bogen-)Sekunden.
- Der Koordinaten-Ursprung für die geografische Breite ist der Äquator. Die geografische Breite nimmt somit Werte von  $-90^\circ < \text{Breite} \leq +90^\circ$  oder  $90^\circ\text{S} < \text{Breite} \leq +90^\circ\text{N}$  an. Besonderheiten werden in einem Artikel unter [http://en.wikipedia.org/wiki/Mercator\\_projection](http://en.wikipedia.org/wiki/Mercator_projection) sehr anschaulich beschrieben.
- Der Koordinaten-Ursprung für die geografische Länge ist der Null-Meridian. Die geografische Länge nimmt somit Werte von  $-180^\circ < \text{Länge} \leq +180^\circ$  oder  $180^\circ\text{W} < \text{Länge} \leq 180^\circ\text{E}$  an.

Folgende Startwerte sind in den Gambas-Projekten erfolgreich erprobt worden:

- Weltkarte                      Latitude =  $8^\circ$ , Longitude =  $-8^\circ$  und Zoom = 1
- Deutschland                Latitude =  $51,4^\circ$ , Longitude =  $10,7^\circ$  und Zoom = 6
- Osterburg                    Latitude =  $52.78979^\circ$ , Longitude =  $11.75280^\circ$  und Zoom = 13

#### 24.8.3.2 Eigenschaften

Die Klasse *MapPoint* verfügt über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Lat	Float°	Gibt die geografische Breite eines Karten-Punktes (MapPoint) als dezimalen Grad-Wert zurück.
Lon	Float°	Gibt die geografische Länge eines Karten-Punktes als dezimalen Grad-Wert zurück.

Tabelle 24.8.3.2.1 : Eigenschaften der Klasse MapPoint

Die Funktion *MapPoint(Latitude As Float, Longitude As Float) As MapPoint* initialisiert ein Karten-Objekt mit einem geografischen Start-Punkt (Breite°, Länge°):

```
Dim myMapPoint = MapPoint(52.78631, 11.73872) ' Ort Osterburg
```

#### 24.8.3.3 Methoden

Die Klasse *MapPoint* besitzt die folgenden drei Methoden:

Methode	Rückgabotyp	Beschreibung
Distance ( PointA As MapPoint, PointB As MapPoint )	Float	Die Funktion berechnet den (direkten) Abstand ( <i>Distanz in Metern!</i> ) zwischen zwei geografischen Punkten, von denen die geografische Breite und Länge als dezimaler Grad-Wert bekannt sind.
Bearing(From As MapPoint, To As MapPoint)	Float	Berechnet die (Start-)Peilung in Grad zwischen zwei geografischen Punkten, von denen die Breite und Länge als dezimaler Grad-Wert bekannt sind.

Methode	Rückgabtyp	Beschreibung
From(MapPoint as MapPoint, Bearing As Float, Distance As Float)	MapPoint	-

Tabelle 24.8.3.3.1 : Methoden der Klasse MapPoint

### 24.8.3.4 Beispiel

Das Beispiel bezieht sich auf die Berechnungen der *Entfernung* und der *Start-Peilung* für die beiden Städte Osterburg und Berlin:

```
Dim mpOsterburg, mpBerlin, mpDummy As MapPoint

mpDummy = New MapPoint
mpOsterburg = New MapPoint(52.78631, 11.73872)
mpBerlin = New MapPoint(52.516250, 13.378870)

Print "Geografische Breite 'Osterburg' = "; mpOsterburg.Lat; "°"
Print "Geografische Länge 'Osterburg' = "; mpOsterburg.Lon; "°"
Print "Entfernung Osterburg-Berlin = "; Round(mpDummy.Distance(mpOsterburg, mpBerlin) / 1000, -3); " km"
Print "Start-Peilung = "; Round(mpDummy.Bearing(mpOsterburg, mpBerlin), -3); "°"

Geografische Breite 'Osterburg' = 52,78631°
Geografische Länge 'Osterburg' = 11,73872°
Entfernung Osterburg-Berlin = 114,994 km
Start-Peilung = 104,532°
```

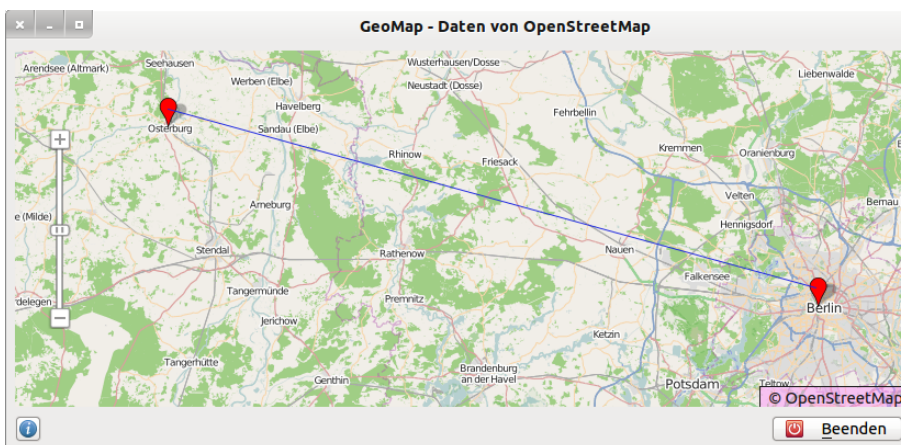


Abbildung 24.8.3.4.1: Entfernung Osterburg – Berlin (Luftlinie)

Auch das Konsolen-Programm *geod* liefert sehr gute Ergebnisse, da es – wie auch die Komponente *gb.map* – das Modell WGS84-Ellipsoid zur Berechnung verwendet. Es ist schnell installiert:

```
hans@linux:~$ sudo apt-get install proj-bin
```

und liefert mit dem folgenden Aufruf und den danach einzugebenden geografischen **Koordinaten** für zwei Orte (Osterburg und Berlin (Mitte)) adäquate Ergebnisse (Angabe der Entfernung in Kilometern) wie sie auch die *MapPoint-Distance-Methode* der Klasse *MapPoint* liefert:

```
hans@linux:~$ geod +ellps=WGS84 +units=km -I
52.78631 11.73872 52.516250 13.378870
104d29'46.077" -74d11'59.931" 114.994
```