How to calculate distance between two landmarks

This example demonstrates:

- how to chose landmark from database
- how to calculate the distance between two landmarks, using own algorithm

There are used following header files:

```
#include <clmklandmarkselectordlg.h>
#include <tlmkitemiddbcombiinfo.h>
#include <epos_cposlandmarkdatabase.h>

#include <e32math.h> // class Math
```

MMP-file must include following libraries:

```
LIBRARY lmkcommonui.lib
LIBRARY eposlandmarks.lib
LIBRARY lbs.lib
```

Class TLocality contains own methods (TLocality::Distance) for calculating distance between two landmarks - it is preferable to use own methods. But in some situations, You may need own algorithm to calculate distance. The following formula can be used for calculating the distance between two points on a sphere surface.

Haversinus formula:

$$angle = \arctan\left(\frac{\sqrt{\left(\cos\alpha_2 * \sin\Delta\lambda\right)^2 + \left(\cos\alpha_1 * \sin\alpha_2 - \sin\alpha_1 * \cos\alpha_2 * \cos\Delta\lambda\right)^2}}{\sin\alpha_1 * \sin\alpha_2 + \cos\alpha_1 * \cos\alpha_2 * \cos\Delta\lambda}\right)$$

$$\Delta\lambda = \lambda_2 - \lambda_1$$

$$\alpha_1, \lambda_1 - latitude\ \ and\ \ longitude\ \ of\ \ the\ \ landmark\ \mathbf{1}$$

$$\alpha_2, \lambda_2 - latitude\ \ and\ \ longitude\ \ of\ \ the\ \ landmark\ \mathbf{2}$$

The final distance can be calculated after multiplying angle to the radius of the sphere (radius of the Earth).

The calculation of the distance according to the formula given above:

```
const TInt KEarthRadius = 6372795; // in meters
TReal64 YourClassName :: CalcDistance( TReal64 aLat1, TReal64 aLong1,
                                       TReal64 aLat2, TReal64 aLong2 )
{
   TReal64 result = -1,
           lat1Sin, lat2Sin, lat1Cos, lat2Cos, deltaLongCos, deltaLongSin;

   if( Math :: Sin( lat1Sin,    aLat1 * KPi / 180) == KErrNone &&
       Math :: Sin( lat2Sin,    aLat2 * KPi / 180) == KErrNone &&
       Math :: Cos( lat1Cos,    aLat1 * KPi / 180) == KErrNone &&
       Math :: Cos( lat2Cos,    aLat2 * KPi / 180) == KErrNone &&
       Math :: Cos( deltaLongCos, ( aLong2 - aLong1 ) * KPi / 180 ) == KErrNone &&
       Math :: Sin( deltaLongSin, ( aLong2 - aLong1 ) * KPi / 180 ) == KErrNone )
   {
           TReal64 a = lat2Cos * deltaLongSin,
            b = lat1Cos * lat2Sin - lat1Sin * lat2Cos * deltaLongCos,
            c = lat1Sin * lat2Sin,
            d = lat1Cos * lat2Cos * deltaLongCos,
            e = a * a + b * b,
            f;
    if( Math :: Sqrt( f, e  ) == KErrNone )
    {
     e = f / ( c + d );
     if( Math :: ATan( f, e ) == KErrNone )
     {
        while( f < 0 )
           f += KPi;
        result = f * KEarthRadius;
     }
    }
   }
   return result;
}
```

The following function shows how to select one landmark:

```
TInt YourClassName::SelectOneLandmark(TLmkItemIdDbCombiInfo& aItem)
{
   // this dialog allows to select landmark
   CLmkLandmarkSelectorDlg* selectLandmarkDlg = CLmkLandmarkSelectorDlg::NewL();
   // dlg is deleted automatically
   return selectLandmarkDlg->ExecuteLD(aItem);
```


(C) Copyright Nokia 2013. All rights reserved.

```
}
```

The following code fragment shows how to select two landmarks and calculate distance between them:

```
TLmkItemIdDbCombiInfo item, item2;
if( SelectOneLandmark( item ) != 0 && SelectOneLandmark( item2 ) != 0 )
{
   // get item ID
   TPosLmItemId itemId  = item.GetItemId(),
                item2Id = item2.GetItemId();

   // get item info
   CPosLandmark *lmItemPos  = item.GetLmDb()->ReadLandmarkLC( itemId ),
                *lmItem2Pos = item2.GetLmDb()->ReadLandmarkLC( item2Id );

   // reading item position
   TLocality pos1, pos2;
   if( lmItemPos->GetPosition(pos1) == KErrNone &&
       lmItem2Pos->GetPosition(pos2) == KErrNone )
   {
   // calc. distance
      TReal64 dist = CalcDistance( pos1.Latitude(), pos1.Longitude(),
                                   pos2.Latitude(), pos2.Longitude() );
      if( dist != -1 )
      {
         TRealFormat format( 16,4 );
         TBuf<64> buf;
         _LIT( KDist, "Distance is:\n" );
         buf.Append( KDist );
         buf.AppendNum(dist, format);

         CAknInformationNote* note = new (ELeave) CAknInformationNote;
         note->ExecuteLD(buf);
      }
   }
}
delete item.GetLmDb();
delete item2.GetLmDb();
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.