ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΣΥΣΤΗΜΑΤΑ ΓΕΩΓΡΑΦΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ

(8° εξ. Προπτυχιακού εαρινό εξάμηνο 2009-2010)

Διδάσκοντες: Αν. Καθ. Γιάννης Θεοδωρίδης – Δρ. Νίκος Πελέκης

Υπεύθυνος Εργαστηρίων: Δρ. Ηλίας Φρέντζος

Δημιουργία ενός Γεωγραφικού Mashup χρησιμοποιώντας το Microsoft Bing Maps και τη PostgreSQL - PostGIS

Δο. Ηλίας Φοέντζος

1. Γενικά

Σκοπός του εργαστηρίου είναι η εξοικείωση των σπουδαστών την υπηρεσία Bing Maps της Microsoft και του αντίστοιχου Software Development Kit (SDK), σε συνδυασμό με τη βάση δεδομένων ανοικτού PostgreSQL και την επέκταση PostGIS για υποστήριξη χωρικών λειτουργιών. Το εργαστήριο έχει σαν στόχο την παρουσίαση των δυνατοτήτων του Bing Maps SDK, αλλά και την εξοικείωση των σπουδαστών με τις σύγχρονες τάσεις δημοσίευσης χωρικών δεδομένων στο web και δημιουργίας γεωγραφικών mashup. Συγκεκριμένα, θα υλοποιήσουμε μία υβριδική εφαρμογή στο web που χρησιμοποιεί δεδομένα από δύο πηγές – το Bing Maps και μία τοπική βάση δεδομένων - σε μία ολοκληρωμένη εφαρμογή.

Για την εκτέλεση των εφγασιών του εφγαστηφίου είναι απαφαίτητη η εγκατάσταση του παφακάτω λογισμικού στον υπολογιστή σας τα οποία μποφείτε να αποκτήσετε είτε χφησιμοποιώντας το MSDNA λογαφιασμό σας είτε διατίθενται ελεύθεφα για κατέβασμα:

- Microsoft Visual Studio 2008 (MSDNAA)
- Simplovation Web Maps VE (<u>http://simplovation.com/</u>)
- PotgreSQL 8.2 ή ανώτερη (<u>http://www.postgresql.org/</u>
- PostGIS 1.3.1 ή ανώτερο (<u>http://postgis.refractions.net/</u>)
- ODBC Driver για τη PostgreSQL (<u>http://www.postgresql.org/ftp/odbc/versions/msi/</u>)
- ODBC Driver για .NET: <u>http://www.microsoft.com/downloads/details.aspx?FamilyID=6ccd8427-1017-4f33-a062-d165078e32b1&displaylang=en</u>

Λίγα λόγια για το λογισμικό που θα χρησιμοποιηθεί:

Microsoft Visual Studio: Είναι το βασικό ολοκληφωμένο πεφιβάλλον ανάπτυξης λογισμικού (IDE) της Microsoft. Το Visual Studio υποστηφίζει γλώσσες πφογφαμματισμού όπως C/C++, VB.NET, C#, HTML/XHTML, και JavaScript. Ο κώδικας του εφγαστηφίου έχει γφαφτεί σε VB.NET, αλλά μποφεί εύκολα να μεταφφαστεί σε C#.

Microsoft Bing Maps: Όπως περιγράφεται από την Microsoft στην αρχική σελίδα του αντίστοιχου Software Development Kit (http://www.microsoft.com/maps/isdk/ajax/), το Microsoft Bing Maps είναι ένα σύνολο υπηρεσιών βασισμένων στο διαδίκτυο που δίνουν τη δυνατότητα στους χρήστες τους – προγραμματιστές να δώσουν στις εφαρμογές τους ένα ολοκληρωμένο τρόπο θέασης της χωρικής υπόστασης των δεδομένων τους και να αναπτύξουν σελίδες που αλληλεπιδρούν με τους τελικούς χρήστες μέσα από ένα πλούσιο γεω-χωρικό περιβάλλον.

Το βασικό πλεονέκτημα του Microsoft Bing Maps, όπως και του Google Maps, είναι ότι παφέχουν ένα πλήφες μοντέλο δεδομένων που μποφεί να διαχειφιστεί / οπτικοποιήσει χωφικά δεδομένα, κάτω από ένα ενιαίο, πλούσιο χωφικό υπόβαθφο που αποτελείται από διανυσματικά δεδομένα (δφόμοι, ακτογφαμμές, διοικητικά όφια), ψηφιακά μοντέλα εδάφους, και οφθοανηγμένα raster δεδομένα όπως δοφυφοφικοί οφθοφωτοχάφτες (aerial και hybrid view) και αεφοφωτογφαφίες

μεγάλης κλίμακας (bird's eye). Το μοντέλο δεδομένων που παφέχεται μποφεί να χφησιμοποιηθεί για να διαχειφιστεί τον τφόπο θέασης του χάφτη, τον τφόπο που εμφανίζονται τα επιπλέον δεδομένα που «τοποθετεί» ο χφήστης επάνω στο διατιθέμενο υπόβαθφο, καθώς και να εκτελέσει πιο πφοηγμένες λειτουφγίες, όπως γεωκωδικοποίηση (δηλαδή υπολογισμός των συντεταγμένων μίας γεωγφαφικής διεύθυνσης), αντίστφοφη γεωκωδικοποίηση, δφομολόγηση κ.α.

Ένα γρήγορο και απλό παράδειγμα των δυνατοτήτων που παρέχονται από το Microsoft Bing Maps SDK μπορεί να βρεθεί στο URL: <u>http://isl.cs.unipi.gr/courses/gis/lab/bm_lab_data.zip</u> (το αρχείο bing_maps_demo.html). Ο κώδικας τοποθετεί επάνω στο χάρτη τρεις εκκλησίες, και δημιουργεί ένα τρίγωνο με τις κορυφές του επάνω σε αυτές.

Simplavation Web Maps VE: Τα Simplovation Web Maps επιτρέπουν στον προγραμματιστή να αναπτύξει Web εφαρμογές που ενσωματώνουν το Microsoft Bing Maps, χωρίς να γράψει ούτε μία γραμμή κώδικα JavaScript, αλλά να χρησιμοποιήσει το γραφικό περιβάλλον του Visual Studio και τη γλώσσα προγραμματισμού της αρεσκείας του (π.χ. VB.NET ή C#).

PostgreSQL και PostGIS: Βλέπε σημειώσεις εργαστηρίου «Εισαγωγή στη PostgreSQL – PostGIS»

ODBC Driver για τη PostgreSQL και για .NET: Οδηγός ODBC για τη PostgreSQL και το Visual Studio .NET. Το ODBC λειτουργεί ως ενδιάμεσος μεταξύ της γλώσσας προγραμματισμού και της Βάσης Δεδομένων και παρέχει ένα πρότυπο API για τη προσπέλαση δεδομένων που βρίσκονται σε Συστήματα Διαχείρισης Βάσεων Δεδομένων. Το ODBC είναι ανεξάρτητο από τη γλώσσα προγραμματισμού, από το ΣΔΒΔ και το λειτουργικό σύστημα που χρησιμοποιείται.

Αντικείμενο της άσκησης είναι η σύνταξη ενός mashup με δεδομένα από το Bing Maps και από τη τοπική βάση PortgreSQL.

Τα δεδομένα για την εκτέλεση της άσκησης μποφούν αν βφεθούν στο παφακάτω URL: <u>http://isl.cs.unipi.gr/courses/gis/lab/data/bm_lab_data.zip</u>, τα οποία πφέπει να αποσυμπιέσετε στο P:\ (ή σε έναν άλλο τοπικό φάκελο).

1. Δημιουργία της Βάσης Δεδομένων

Ποιν ξεκινήσουμε την υλοποίηση της εφαρμογής, θα πρέπει να δημιουργήσουμε τη βάση δεδομένων των χωρικών δεδομένων που θα περιέχει τις ζώνες (zones), τα οικοδομικά τετράγωνα (parcels), τους άξονες των οδών (roads) και τα σημεία ενδιαφέροντος (landmarks). Οι πίνακες αυτοί θα πρέπει να έχουν ήδη δημιουργηθεί σε περίπτωση που έχουμε εκτελέσει τις ασκήσεις του εργαστηρίου «Εισαγωγή στη PostgreSQL – PostGIS». Σε αντίθετη περίπτωση, ή αν δεν είμαστε σίγουροι ότι η βάση δεδομένων lab_spatial_db περιέχει τα σωστά και ολοκληρωμένα δεδομένα, πρέπει να διαγράψουμε την υπάρχουσα βάση δεδομένων lab_spatial_db και να εκτελέσουμε τα παρακάτω βήματα:

<u>**Βήμα 1**</u>: Θα ξεκινήσουμε από το command prompt για να δημιουργήσουμε μία καινούργια βάση δεδομένων. Επιλέξτε Start > Programs > PostgreSQL 8.2 > command prompt πληκτρολογήστε

createdb -T template_postgis -E UTF8 lab_spatial_db

Εναλλακτικά μπορείτε να δημιουργήσετε τη βάση μέσω sql. Επιλέξτε Start > Programs > PostgreSQL 8.2 > Psql to 'postgres' και πληκτρολογήστε

```
SET ROLE postgres;
CREATE DATABASE lab_spatial_db
TEMPLATE=template_postgis
ENCODING='UTF8'
TABLESPACE=pg_default;
```

<u>Βήμα 2</u>: Δημιουργία του σχήματος της βάσης δεδομένων και εισαγωγή των δεδομένων στους αντίστοιχους πίνακες από ASCII αρχείο sql. Ανοίξτε το αρχείο create_spatial_db.sql με έναν editor (π.χ. notepad) και παρατηρείστε τα περιεχόμενά του. Για να εισάγουμε τα δεδομένα στη PostgrSQL θα χρησιμοποιήσουμε την παρακάτω εντολή που πρέπει να εκτελεστεί από το command prompt:

```
psql -d [database] -f [filename]
```

Επομένως, στην περίπτωσή μας έχουμε:

psql -d lab_spatial_db -f P:\bm_lab_data\create_spatial_db.sql

2. Ρύθμιση του περιβάλλοντος ανάπτυξης

<u>Βήμα 30:</u> Ανοίξτε το Visual Studio 2008 και ανοίξτε το Project LabBM που βρίσκεται μέσα στο φάκελο P:\bm_lab_data\LabBM. Το project περιέχει μόνο τη μορφή της σελίδας με τη θέση των αντικειμένων. Η σελίδα αυτή μπορεί να δημιουργηθεί χωρίς να έχουμε γράψουμε καθόλου κώδικα

<u>Βήμα 4</u>^{**•**}: Θα πρέπει να προσθέσουμε τα αναγκαία References. Αυτό γίνεται από το μενού Project -> AddRefernce και στο παράθυρο που εμφανίζεται θα διαλέξουμε στο Tab .NET, το Microsoft.Data.Odbc το οποίο περιέχει τα απαραίτητα αντικείμενα για την προσπέλαση του odbc από τη γλώσσα προγραμματισμού.

<u>Βήμα 5</u>^ο: Ανοίξτε τον κώδικα της σελίδας Default.aspx (π.χ. δεξί κλικ επάνω στο Default.aspx και επιλογή View Code). Στον κώδικα θα πρέπει καταρχήν να εισάγουμε τα Namespace elements System.Data, Microsoft.Data.Odbc και Simplovation.Web.Maps.VE. Αυτό γίνεται εισάγοντας την εντολή Imports που καθορίζεται στο επίπεδο του αρχείου (δηλαδή πριν από τον καθορισμό της κλάσης)

```
Imports System.Data
Imports Microsoft.Data.Odbc
Imports Simplovation.Web.Maps.VE
Partial Public Class _Default
    Inherits System.Web.UI.Page
...
```

3. Ανάπτυξη κώδικα

<u>Βήμα 6°</u>: Σύνδεση με τη βάση δεδομένων. Αυτό γίνεται με ένα connection και το κατάλληλο Connection String (μεταβλητή szConnect). Προσοχή πρέπει να δοθεί για να μπει το σωστό password.

```
' Το Connection String έχει τη παρακάτω σύνταξη
Dim szConnect As String = "DRIVER={PostgreSQL ODBC " & _
Driver(UNICODE)};database=lab_spatial_db;server=localhost;" & _
"port=5432;uid=postgres;pwd=postgres"
' Δημιουργία του ODBC Connection σύμφωνα με το παραπάνω Connection String
Dim cnDB As OdbcConnection = New OdbcConnection(szConnect)
```

<u>Βήμα 7º:</u> Βασικές έννοιες του Bing Maps Data Model.

Η πρόσβαση στο χάρτη γίνεται μέσω ενός αντικειμένου τύπου Map (Map1 στη δικιά μας περίπτωση). Το αντικείμενο Map παρέχει την πλήρη λειτουργικότητα που απαιτούμε για την εφαρμογή μας.

3

Το αντικείμενο Map μπορεί να περιέχει απευθείας σχήματα (Shapes) τα οποία προστίθενται στο χάρτη με τη μέθοδο AddShape. Τα Shapes μπορεί να είναι σημειακά, γραμμικά ή πολυγωνικά, και κάθε ένα τους συνοδεύεται και από ένα description. Η θέση των Shapes (αλλά και γενικά η θέση του π.χ. κέντρου του χάρτη) δίνεται από αντικείμενα του τύπου LatLong, με λίγα λόγια, από το γεωγραφικό μήκος (Long) και πλάτος (Lat) του κάθε σημείου. Εξ' ου και προκύπτει η ανάγκη οι συντεταγμένες να γράφονται σε μοίρες που αναφέρονται σε γεωγραφικό μήκος και πλάτος (και επομένως στο σύστημα συντεταγμένων WGS84).

Επίσης, ο χάρτης μπορεί να έχει διάφορα Layers, κάθε ένα από τα οποία μπορεί να λειτουργήσει ως συλλογή από Shapes με κοινή ιδιότητα (π.χ. όλοι οι δρόμοι σε ένα Layer). Δεν υπάρχει περιορισμός ως προς το είδος των αντικειμένων που μπορεί να έχει ένα Layer (π.χ. μπορεί να περιέχει συγχρόνως γραμμές και πολύγωνα)

<u>Βήμα 8</u>: Ενέργειες που γίνονται στο φόρτωμα της σελίδας. Στο φόρτωμα της σελίδας, θα πρέπει να ανοίγει η σύνδεση με τη βάση δεδομένων, ο χάρτης να κεντράρεται στην περιοχή ενδιαφέροντος και να προστίθενται τα κατάλληλα Layers. Αυτό θα πρέπει να γίνεται μόνο στο αρχικό φόρτωμα της σελίδας (δηλαδή, όταν δεν γίνεται postback).

```
' έλεγχος αν η σελίδα φορτώνεται από postback

If Not IsPostBack Then

'Άνοιγμα του Connection

cnDB.Open()

'Κεντράρισμα του χάρτη στην περιοχή που μας ενδιαφέρει

SetMapCenter()

'Προσθήκη του Layer των οικοδομικών τετραγώνων

AddParcels()

End If
```

Οι μέθοδοι SetMapCenter και AddParcels θα δείξουμε πως συντάσσονται στα επόμενα βήματα

<u>Βήμα 9</u>: Κεντράρισμα του χάρτη στην περιοχή ενδιαφέροντος. Η ιδέα για το κεντράρισμα του χάρτη είναι να πάμε στη Β.Δ. και να βρω τα όρια των δεδομένων που καλύπτουν την περιοχή που μας ενδιαφέρει. Αυτό θα γίνει εκτελώντας ένα query. Στη συνέχεια θα πρέπει να δηλώσουμε το νέο κέντρο στο χάρτη.

```
Private Sub SetMapCenter()
```

```
' Δημιουργούμε το string που θα περιέχει την SQL εντολή που θα
' τρέξουμε
' Η εντολή ζητάει το κέντρο της συλλογής που περιέχει όλους τους
' δρόμους Η συνάρτηση Transform πραγματοποιεί αλλαγή συντεταγμένων από
' το SRID=2100 (ΕΓΣΑ'87) που βρίσκονται τα δεδομένα στο SRID=4326
' (WGS84) στο οποίο θέλουμε να μεταβούμε
Dim strSQL As String = "SELECT
center(collect(transform(road_geometry,4326))) FROM lab.roads "
' Δημιουργία ενός DataAtapter που θα εκτελέσει την SQL ερώτηση επάνω
' στο Connection
Dim adDB As OdbcDataAdapter = New OdbcDataAdapter(strSQL, cnDB)
' Δημιουργούμε ένα καινούργιο dataset στο οποίο θα εισάγουμε τα
' αποτελέσματα του παραπάνω query
Dim dsDB As DataSet = New DataSet()
' Γεμίζουμε το dataset με τα αποτελέσματα του query
adDB.Fill(dsDB)
' Ξέρουμε ότι το query που εκτελέσαμε επιστρέφει μόνο έναν πίνακα ο
```

```
' οποίος λόγω του Collect περιέχει μόνο μία εγγραφή (λειτουργεί σαν
' aggregation) και μόνο μία στήλη (το κέντρο της συλλογής), η οποία
' έχει τη μορφή (Long,Lat)
' Επομένως δημιουργούμε ένα datarow και σε αυτό αναθέτουμε την πρώτη
' εγγραφή του πρώτου πίνακα του dataset
Dim dtr As DataRow = dsDB.Tables(0).Rows.Item(0)
' Παίρνουμε το string που έχει η πρώτη στήλη της εγγραφής
Dim astr As String = dtr.Item(0).ToString
' διώχνουμε την αριστερή και δεξιά παρένθεση
astr = astr.Replace("(", "")
astr = astr.Replace(")", "")
' σπάμε το string ώστε να πάρουμε το Χ και Υ μόνα τους
Dim vls As String() = astr.Split(",")
' Δίνουμε το κέντρο του χάρτη
Me.Map1.Latitude = Val(vls(1))
Me.Mapl.Longitude = Val(vls(0))
' καθορίζουμε το επίπεδο του Zoom. Όσο μεγαλύτερος ο αριθμός τόσο
' μεγαλύτερο το Zoom
Me.Map1.Zoom = 14
```

End Sub

<u>Βήμα 10</u> : Φόρτωμα των δεδομένων των οικοδομικών τετραγώνων. Η ιδέα πάλι είναι να πάρουμε τα δεδομένα που καλύπτουν την περιοχή που μας ενδιαφέρει εκτελώντας ένα query. Αφού πάρουμε τα δεδομένα, θα πρέπει

- Να δημιουργήσουμε ένα ShapeLayer (δηλαδή, ένα Layer που θα περιέχει διάφορα shapes και επομένως θα λειτουργεί ως συλλογή από Shapes), έστω Parcels
- Να προσθέσουμε στο Layer Parcels ένα shape για κάθε εγγραφή του query που εκτελέσαμε.

Αναλυτικά:

```
Private Sub AddParcels()
```

```
' Δημιουργούμε ένω Layer Roads από τον χάρτη
Parcels = New ShapeLayer
Map1.Layers.Add(Parcels)
   ' Δημιουργώ ένα καινούργιο dataset στο οποίο θα εισάγω τους δρόμους
     της περιοχής ενδιαφεροντος
   Dim dsDB As DataSet = New DataSet()
   ' Δημιουργώ το string που θα περιέχει την SQL εντολή που θα τρέξω
   ' Η εντολή ζητάει το σύνολο των δρόμων που βρίσκονται στη β.δ.
   ' Η εντολή transform πραγματοποιεί αλλαγή συντεταγμένων από το
     SRID=2100 (ΕΓΣΑ'87) που βρίσκονται τα δεδομένα
   ' στο SRID=4326 (WGS84) στο οποίο θέλουμε να μεταβούμε
   Dim strSQL As String = _
  "SELECT *, AsText(Transform(parcel_geometry,4326)) FROM lab.parcels"
   ' Δημιουργία ενός DataAtapter που θα εκτελέσει την SQL ερώτηση _
     επάνω στο Connection
   Dim adDB As OdbcDataAdapter = New OdbcDataAdapter(strSQL, cnDB)
   ' Γεμίζουμε το dataset με τα αποτελέσματα του query
   adDB.Fill(dsDB)
   ' Στη γενική περίπτωση το query θα επιστρέψει έναν πίνακα με _
     πολλές εγγραφές (rows)
   ' οι οποίες περιέχουν τους δρόμους που θέλουμε να εμφανίσουμε _
     στον χάρτη
   ' Άρα θα πρέπει να κάνουμε επαναλήψεις μέσα στις εγγραφές (Rows) _
     του dataset
```

```
For Each dtr As DataRow In dsDB.Tables(0).Rows
    ' Το WKT του parcel βρίσκεται στο 4 πεδίο του dataset
   Dim astr As String = dtr.Item(3).ToString
    ' και έχει τη μορφή π.χ. " MULTIPOLYGON(23.6458233025054 _
     37.9396383673635,23.6462226488666 37.9394232589026,...)"
    ' άρα θα κόψω το " MULTIPOLYGON(" στην αρχή και το ")" _
     στο τέλος
   astr = astr.Substring(15, astr.Length - 18)
    ' παίρνω τα σημεία της γραμμής (χωρίζονται με ",")
   Dim vls As String() = astr.Split(",")
    ' δημιουργώ ένα collection από LatLong το οποίο θα περιέχει _
    ' όλα τα σημεία του MULTIPOLYGON
   Dim ccol As New Generic.List(Of LatLong)
    For i As Long = 0 To vls.Count - 1
        Dim vvls As String() = vls(i).Split(" ")
        Dim VEPoint As LatLong = New LatLong(Val(vvls(1)), _
                                             Val(vvls(0)))
        ccol.Add(VEPoint)
   Next
    ' Δημιουργώ ένα νέο Shape, τύπου polygon, με τις κορυφές του _
    ' MULTIPOLYGON όπως έχουν γεμίσει το collection
   Dim shp As Shape = New Shape(ShapeType.Polyline, ccol)
    ' Κρύβω το εικονίδιο
    shp.ShowIcon = False
    ' Γράφω την περιγραφή του (το όνομά του, το οποίο _
    ' βρίσκεται στο 2ο πεδίο του dataset)
    shp.Description = dtr.Item(1).ToString
    ' προσθέτω το shape στο Layer Parcels
   Parcels.Shapes.Add(shp)
Next
```

End Sub

<u>Βήμα 11°</u>: Εκτέλεση της εφαρμογής. Από το μενού Debug -> Start Debugging. Στην περίπτωση που έχουν πάει όλα καλά στο παράθυρο Output στο κάτω μέρος της οθόνης θα πρέπει να εμφανιστεί ένα μήνυμα όπως το παρακάτω:

```
----- Build started: Project: LabVEM10, Configuration: Debug Any CPU -----
c:\WINDOWS\Microsoft.NET\Framework\v3.5\Vbc.exe /noconfig
LabVEM10 -> C:\Work\_Software\ITA\Lessons\M10\Data\LabVEM10\LabVE.dll
========= Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =========
```

Η εφαφμογή θα εκτελεστεί και θα εμφανίσει το παφακάτω παφάθυφο. Προσοχή, ακόμα δεν έχουμε υλοποιήσει τις λειτουργίες της δεξιά γραμμής εργαλείων.



Αλλάξτε την εμφάνιση του χάρτη σε Aerial / Road /Bird's Eye και πλοηγηθείτε μέσα στην περιοχή ενδιαφέροντος.

<u>Βήμα 12°</u>: Φόρτωμα των δεδομένων των σημείων ενδιαφέροντος. Η ιδέα πάλι είναι να πάρουμε τα δεδομένα που καλύπτουν την περιοχή που μας ενδιαφέρει εκτελώντας ένα query. Αφού πάρουμε τα δεδομένα, θα πρέπει

- Να δημιουργήσουμε ένα ShapeLayer (δηλαδή, ένα Layer που θα περιέχει διάφορα shapes και επομένως θα λειτουργεί ως συλλογή από Shapes), έστω Landmarks
- Να προσθέσουμε στο Layer Landmarks ένα shape για κάθε εγγραφή του query που εκτελέσαμε.

```
Αναλυτικά:
```

```
Private Sub AddLandmarks()
    Landmarks = GetLayer("Landmarks")
    Dim dsDB As DataSet = New DataSet()
    Dim strSQL As String = "SELECT landmark_name, " & _
                 "AsText(Transform(landmark_geometry,4326)) " & _
                 "FROM lab.landmarks"
    Dim adDB As OdbcDataAdapter = New OdbcDataAdapter(strSQL, cnDB)
    adDB.Fill(dsDB)
    For Each dtr As DataRow In dsDB.Tables(0).Rows
        Dim astr As String = dtr.Item(1).ToString
        astr = astr.Substring(6, astr.Length - 7)
        Dim vls As String() = astr.Split(",")
        For i As Long = 0 To vls.Count - 1
            Dim vvls As String() = vls(i).Split(" ")
            Dim VEPoint As LatLong = New _
                LatLong(Val(vvls(1)), Val(vvls(0)))
            Dim shp As Shape = New Shape(VEPoint)
            shp.Description = dtr.Item(0).ToString
            Landmarks.Shapes.Add(shp)
```

	Next
	Next
End	Sub

Εκτελέστε πάλι την εφαρμογή σύμφωνα με τα όσα περιγράφονται στο βήμα 11

<u>Βήμα 13°</u>: Η εκτέλεση του κώδικα για την εισαγωγή νέων πυλώνων φωτισμού, θα πρέπει να εκκινεί όταν ο χρήστης κάνει κλικ επάνω στον χάρτη. Λόγω όμως του γεγονότος ότι, όπως και πριν, με απλό (αριστερό) κλικ, ο χάρτης υλοποιεί άλλη λειτουργικότητα (π.χ. μετακινείται), θα πρέπει να ελέγξουμε το δεξί κλικ.

Επιπλέον, λόγω του γεγονότος ότι υπάρχουν διάφορες λειτουργικότητες που πρέπει να υλοποιηθούν με κλικ επάνω στο χάρτη, θα χρησιμοποιήσουμε το CheckBox με όνομα chkAddLandmarks. Έτσι, όταν ο χρήστης κάνει δεξί κλικ επάνω στο χάρτη Map1 και είναι ενεργοποιημένο το «Προσθήκη Σημείων Ενδιαφέροντος», ένας νέο Landmark θα προστίθεται στη βάση δεδομένων και το χάρτη, στο αντίστοιχο σημείο.

Ο κώδικας για την προσθήκη των πυλώνων φωτισμού θα πρέπει να εκτελείται όταν ο χρήστης κάνει δεξί κλικ επάνω στο χάρτη (με όνομα Map1), χρησιμοποιώντας το αντίστοιχο συμβάν (event) του αντικειμένου. Η περιοχή Map1_click της σελίδας (στην οποία πρέπει να γραφτεί ο αντίστοιχος κώδικας), εμφανίζεται πραγματοποιώντας τις κατάλληλες επιλογές από τις αναπτυσσόμενες λίστες όπως στην παρακάτω οθόνη.



Ο κώδικας που πρέπει να γραφτεί στο συγκεκριμένο συμβάν είναι όπως παρακάτω:

```
Private Sub Map1_Click(ByVal sender As Object, ByVal e As _
Simplovation.Web.Maps.VE.AsyncMapEventArgs) Handles Map1.Click
```

8

```
' ελέγχω αν γίνεται δεξί κλικ επάνω στο χάρτη
' για αυτό το λόγο χρησιμοποιώ το αντικείμενο e
' που περιέχει τα ορίσματα που συνοδεύουν το event
If e.rightMouseButton Then
' παίρνω το αντικείμενο LatLong που συνοδεύει το κλικ
' δηλαδή, το σημείο στο οποίο βρισκόταν το ποντίκι
' τη στιγμή που έγινε το κλικ
Dim LL As LatLong = e.latlong
' αν έχει επιλεχθεί να προστίθεται νέο σημείο
If chkAddLandmarks.Checked Then
' εκτέλεσε την αντίστοιχη μέθοδο
AddLandmarkInDB(LL)
End If
End If
```

End Sub

Η συνάρτηση που προσθέτει τα δεδομένα στη βάση και τον χάρτη, θα χρησιμοποιεί μία εντολή INSERT σε συνδυασμό με την GeomFromText ώστε να δημιουργήσουμε την απαραίτητη γεωμετρία (τη θέση δηλαδή) του πυλώνα φωτισμού.

To SQL Query που θα χρησιμοποιηθεί, μετατρέπει σε WKT τη γεωμετρία ενός αντικειμένου που προκύπτει από τα LatLong του χάρτη. Χρησιμοποιεί επίσης εμφωλιασμένες (nested) sql ερωτήσεις για τον υπολογισμό το μεγαλύτερου τρέχοντος id το οποίο αυξάνει κατά 1 και το χρησιμοποιεί σαν id του νέου αντικειμένου, καθώς και για τον καθορισμό του ονόματός του. Τυπικά, θα έχει την παρακάτω μορφή:

```
INSERT INTO lab.landmarks (landmark_id , landmark_name, landmark_geometry)
SELECT ((SELECT max(landmark_id) FROM lab.landmarks)+1),
'Point Of Interest ' || ((SELECT max(landmark_id) FROM lab. landmarks)+1),
GeomFromText('POINT(23.596658706665 38.4687105113576)',4326)
```

Αφού οριστεί το SQL Query, θα πρέπει να ανοίξουμε το Connection, με τη βάση δεδομένων (μέσω του ODBC), και να εκτελέσουμε την εντολή (η οποία δεν επιστρέφει καμία απάντηση). Τέλος θα πρέπει να προσθέσουμε το νέο πυλώνα φωτισμού στο Layer Lights.

Αναλυτικά, πληκτρολογήστε τον παρακάτω κώδικα, στην περιοχή Declarations της κλάσης _Default, ακριβώς επάνω από την τελευταία γραμμή κώδικα της κλάσης (End Class).

Public Sub AddLightInDB(ByVal LL As LatLong)

- ' Η ιδέα είναι να χρησιμοποιήσουμε μία εντολή INSERT σε συνάρτηση με
- ' την GeomFromText ώστε να δημιουργήσουμε την απαραίτητη γεμετρία

```
' (τη θέση) του πυλώνα φωτισμού
```

```
' Δημιουργώ το string που θα περιέχει την SQL εντολή που θα τρέξω Dim sqlAddLight As String
```

- ' Το SQL Query μετατρέπει σε WKT τη γεωμετρία ενός αντικειμένου που ' προκύπτει από τα LatLong του χάρτη
- ' Χρησιμοποιεί επίσης εμφωλιασμένες (nested) sql ερωτήσεις τον τον
- ' υπολογισμό το μεγαλύτερου τρέχοντος id το οποίο αυξάνει κατά 1 και
- ' το χρησιμοποιεί σαν id του νέου αντικειμένου, καθώς και για τον ' καθορισμό του ονόματός του.

```
sqlAddLight = "INSERT INTO lab.landmarks " & __
"(landmark_id, landmark _name, landmark _geometry) " & __
"SELECT ((SELECT max(landmark _id) FROM lab. landmarks)+1), " & __
"POI ' || ((SELECT max(landmark_id) FROM lights)+1), " & __
"GeomFromText('POINT(" & LL.Longitude.ToString.Replace(",", ".") & __
```

```
" % LL.Latitude.ToString.Replace(",", ".") & ")',4326)"
    ' ανοίγω το connection ώστε να πραγματοποιηθεί η εντολή INSERT
    ' ανοίγω το connection ώστε να πραγματοποιηθεί η εντολή INSERT
    cnDB.Open()
     ορίζω την εντολή
    Dim CmdDb As New OdbcCommand(sqlAddLight, cnDB)
     εκτελώ την εντολή
    CmdDb.ExecuteNonQuery()
     και κλείνω το connection
    cnDB.Close()
     Δημιουργώ ένα νέο Shape, τύπου Pushpin, στις συντεταγμένες
    Dim shp As Shape = New Shape(LL)
    ' Γράφω τον τίτλο του pushpin
    shp.Title = "New Point of Interest"
    ' ανακτώ το Layer Lights
    Landmarks = GetLayer("Landmarks")
    ' και προσθέτω το shape στο Layer Lights
    Landmarks.Shapes.Add(shp)
End Sub
```

Εκτελέστε πάλι την εφαρμογή σύμφωνα με τα όσα περιγράφονται στο βήμα 11

4. Ολοκλήρωση

<u>Βήμα 14°</u>: Ανοίξτε το Project LabBMCompleted που βρίσκεται μέσα στο φάκελο P:\bm_lab_data\LabBM και εκτελέστε το. Το project περιέχει τη τελική εφαρμογή με ολόκληρη τη λειτουργικότητα. Δοκιμάστε να ενεργοποιήσετε / απενεργοποιήσετε τα Layer και να πραγματοποιήσετε αναζητήσεις με το όνομα του δρόμου π.χ. αναζητώντας το 1120 θα βρεθεί ο δρόμος Road1120. Μελετήστε τον κώδικα.



Τέλος