

GIS 程序设计课程 设计报告

图书馆查询 GIS 信息系统程序设计开发

学院: *****
指导老师: *****
班级: *****
学号: *****
姓名: Gnat TANG

一、课程设计的目标和内容

为了进一步理解课堂上讲述的有关 GIS 开发与设计的内容,掌握 Engine 组件进行开发的过程,加强实际动手操作和编程能力,更进一步学习程序的调试与编译过程。

使用在 VS2010 开发平台之上利用 Engine 组件构造一个 GIS 应用系统,完成基本的地图载入、显示、漫游、定位、图书查询及其它功能。在对基于组件设计开发基本了解的情况下,独立进行 GIS 开发,要求多查看有关开发环境和组件 Engine 的帮助文档,完成 GIS 程序的设计与开发。

二、采用的开发平台和技术

本次实习使用微软的 Windows 操作系统, .Net 服务框架,开发了一个基于组建的可扩充易扩展的图书信息系统,借助 ESRI 公司的 ArcGIS 集成了图书馆地理信息查询等地理信息系统功能。

[Windows 系统](#) 是一个采用了图形化模式 GUI,比起从前的 DOS 需要键入指令使用的方式更为人性化,而且使用方便的操作系统。Windows 是全世界最流行的操作系统之一,用这个平台开发,是最终用户的所熟悉和期盼的。另一方面,Windows 有自己的一套规范,只要遵守,易于日后的维护和更新。开发人员无需担心无聊的系统更新和兼容性问题。

[.NET](#) 是 Microsoft XML Web services 平台。XML Web services 允许应用程序通过 Internet 进行通讯和共享数据,而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建 XML Web services 并将这些服务集成在一起之所需。对个人用户的好处是无缝的、吸引人的体验。从用户的角度来看,.Net 是一个能省去很多不必要交互麻烦的强大框架,在这个体系内,数据传输的消息传递都异常简单通用快捷,方便敏捷开发。另一方面,接口规范的 .Net 框架让我们可以方便的使用 ArcEngine 来实现地理信息功能。

[ArcGIS Engine](#) 是用于构建定制应用的一个完整的嵌入式的 GIS 组件库。利用 ArcGIS Engine,开发者能将 ArcGIS 功能集成到一些应用软件。正是这一特点,我们才选择 ArcEngine 来开发这个图书馆信息系统。

三、功能设计

让我们从整个系统运行的流程来说明其功能。对于用户,即使没有读者账户,他也可以做到查询图书和图书馆,这些功能是开放。如果他需要借书,那他需要注册账户,在用户注册界面填入相应信息,系统验证了就会添加这个用户。然后本用户就可以用这个账户登陆。登陆的账户除了基本的查询服务(查询服务包含图书、作者、图书馆地理信息的查询),还拥有图书借阅、评价、打分等功能。额外的,注册用户有自己的读者主页,方便读者们写书评,分享读书笔记,以及读书会等线下交流活动。这些是用户层面可以看到的。

另一方面,用户看不到的是管理员的工作:图书入库,更新,维护作者信息、图书馆信息等等。由于这是个地理信息系统,所以还要负责地图更新。



四、开发流程

1. 需求分析

- 1) 图书查询服务
- 2) 读者注册,借阅服务
- 3) 图书馆地理信息查询服务

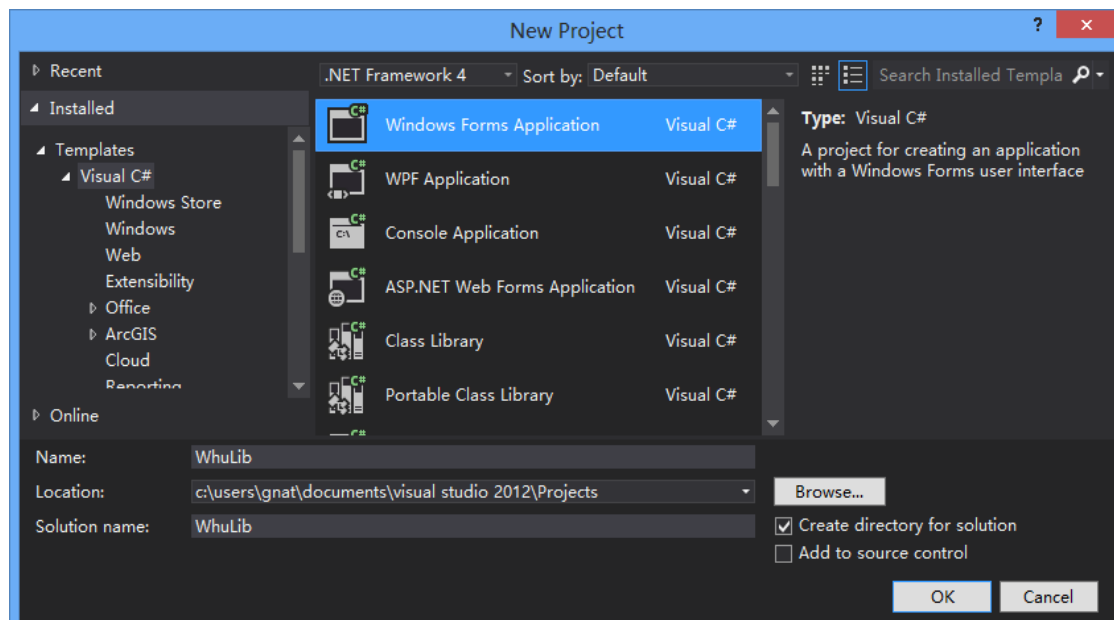
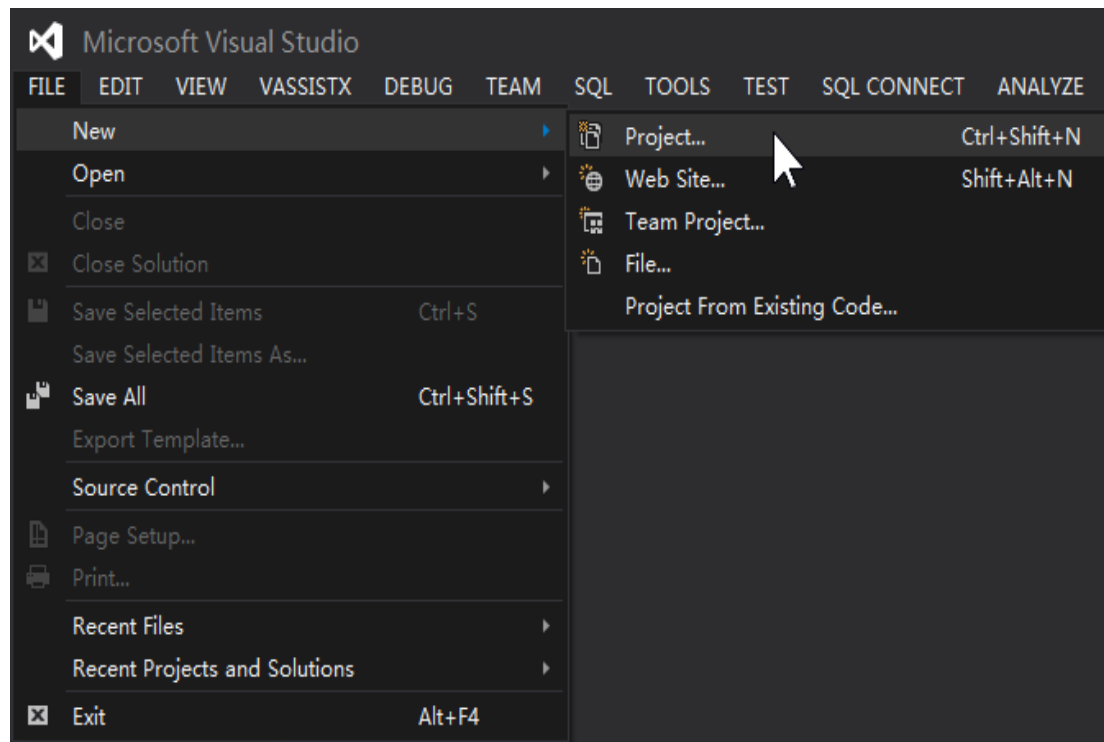
2. 首先准备需要用到的软件

- 1) 安装 Windows 系统
- 2) 安装 Visual Studio 2012
- 3) 安装 ArcEngine
- 4) 安装 Access/Oracle 数据库

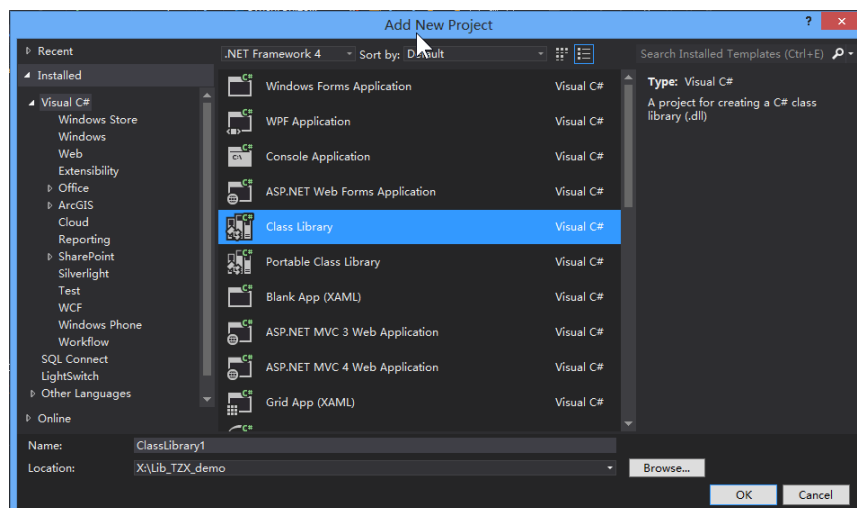
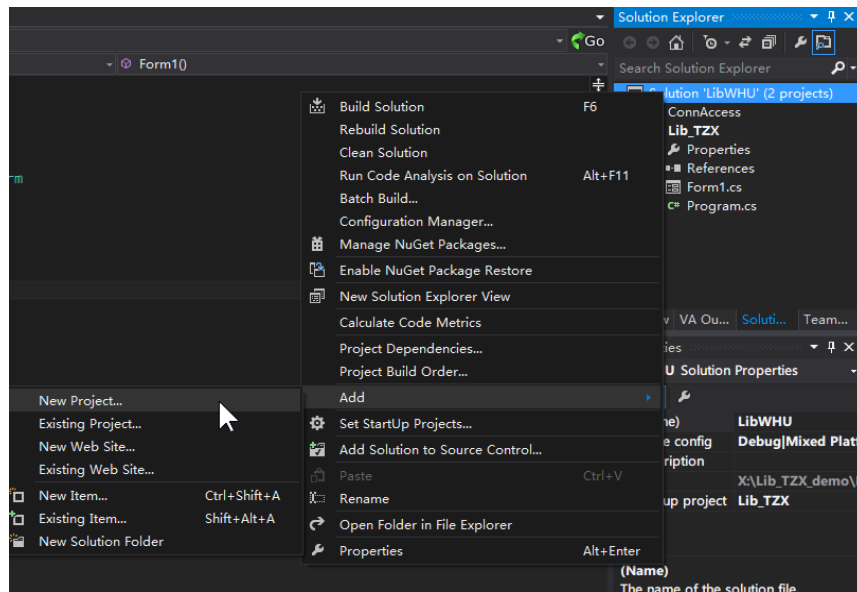


3. 工程建立

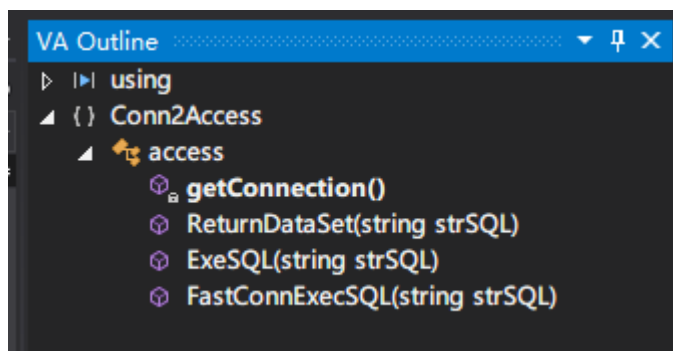
1) 打开 VS2012, 新建工程 LibTZX



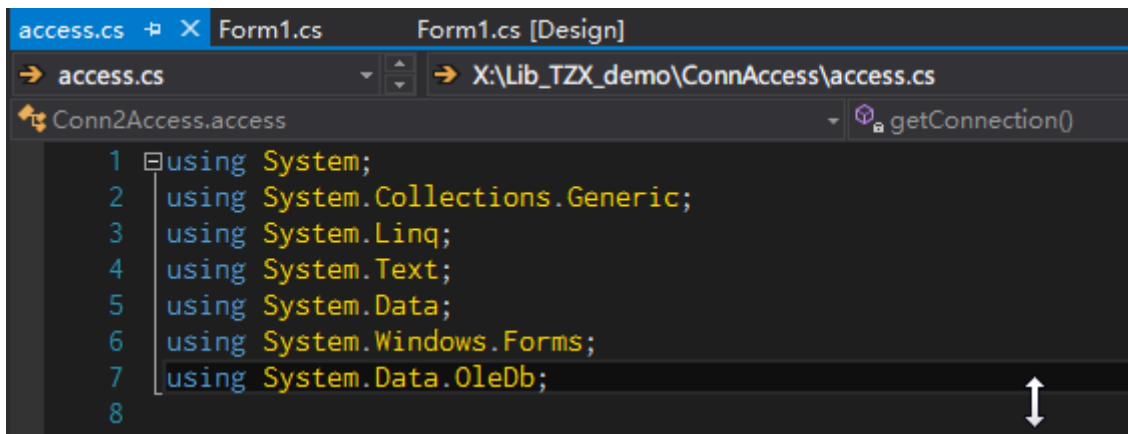
2) 新建 Library Class: Conn2Access (做数据库连接)



这是 Conn2Access 所要实现的方法:



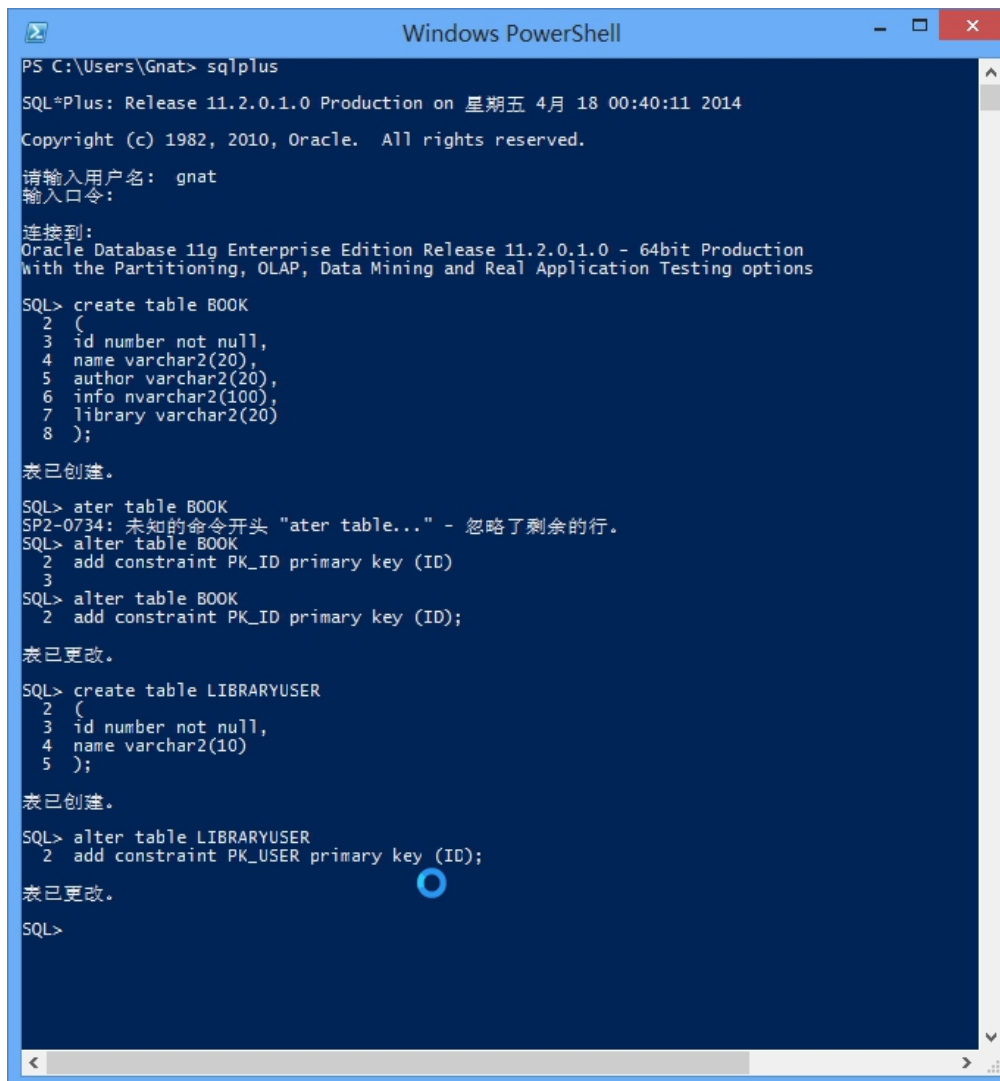
为了实现连接,先加入头文件 System.Data.OleDb



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Data;
6 using System.Windows.Forms;
7 using System.Data.OleDb;
8
```

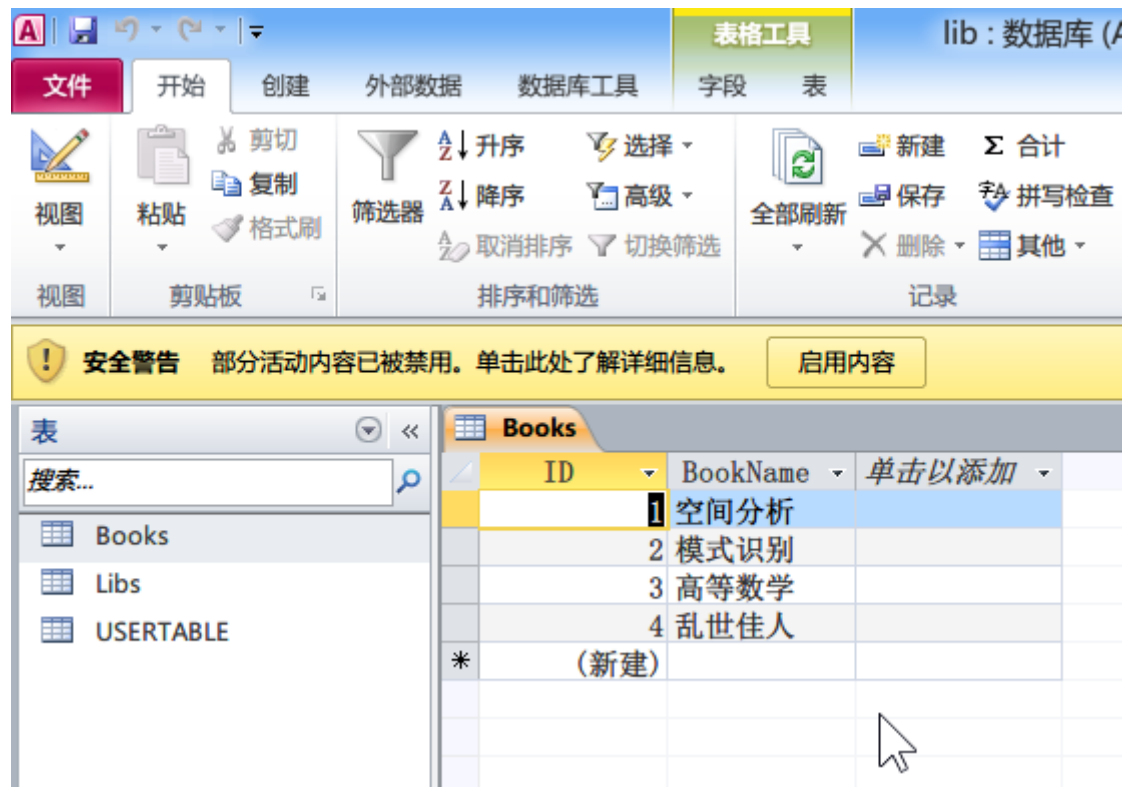
然后实现刚才的四个方法,并编译成 dll。

2) 新建 Access 数据库, 新建 Books, Libs 等表



```
PS C:\Users\Gnat> sqlplus
SQL*Plus: Release 11.2.0.1.0 Production on 星期五 4月 18 00:40:11 2014
Copyright (c) 1982, 2010, Oracle. All rights reserved.
请输入用户名: gnat
输入口令:
连接到:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> create table BOOK
2 (
3 id number not null,
4 name varchar2(20),
5 author varchar2(20),
6 info nvarchar2(100),
7 library varchar2(20)
8 );
表已创建。
SQL> ater table BOOK
SP2-0734: 未知的命令开头 "ater table..." - 忽略了剩余的行。
SQL> alter table BOOK
2 add constraint PK_ID primary key (ID)
3
SQL> alter table BOOK
2 add constraint PK_ID primary key (ID);
表已更改。
SQL> create table LIBRARYUSER
2 (
3 id number not null,
4 name varchar2(10)
5 );
表已创建。
SQL> alter table LIBRARYUSER
2 add constraint PK_USER primary key (ID);
表已更改。
SQL>
```

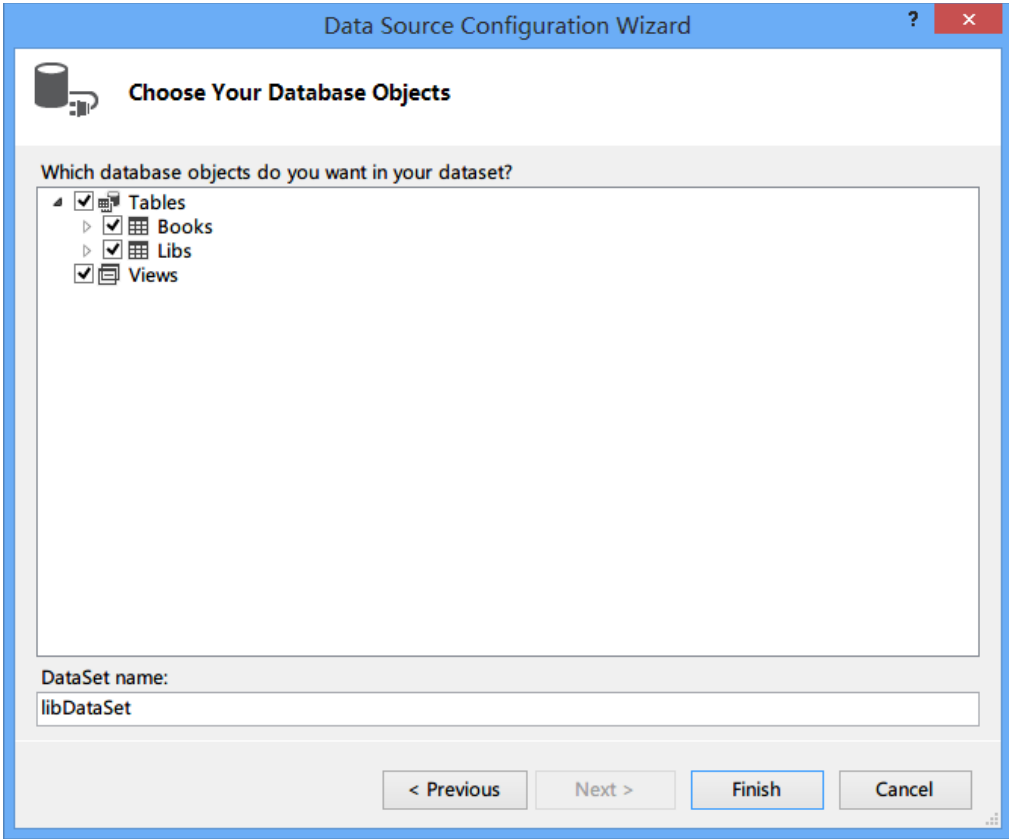
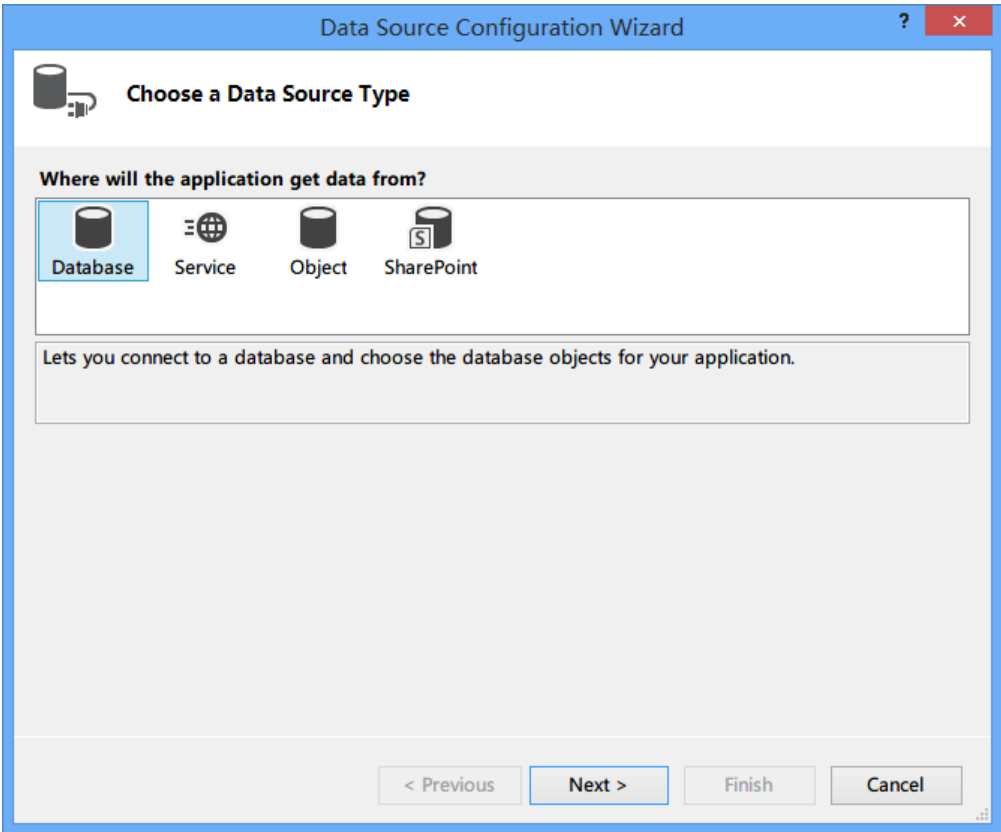
简单地加入一点测试数据



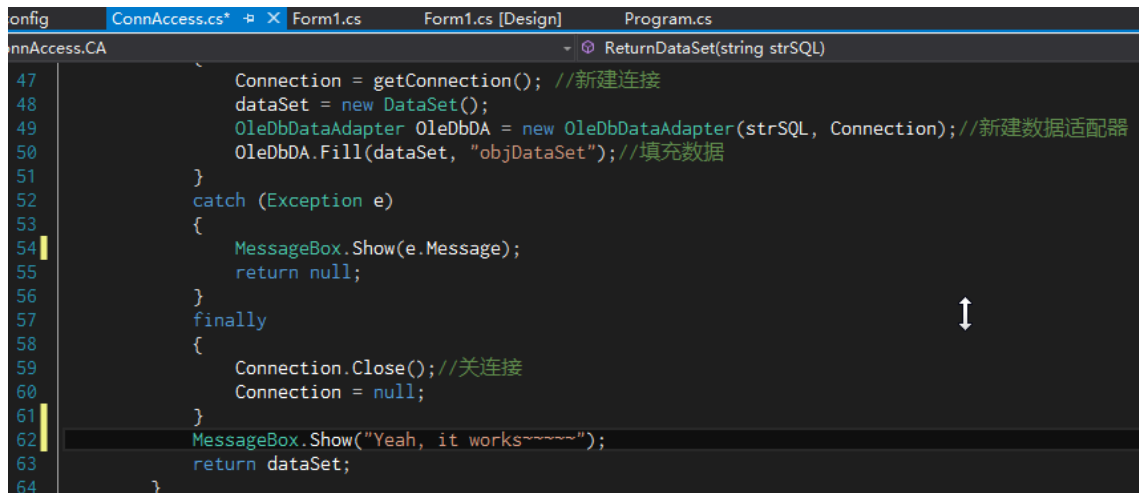
3) 在 Visual Studio 里连接数据库



已经测试链接成功,现加入需要的表单

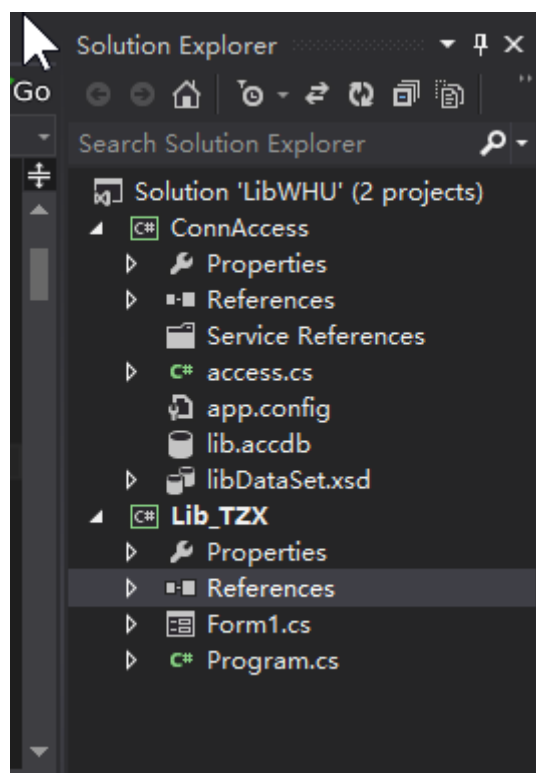


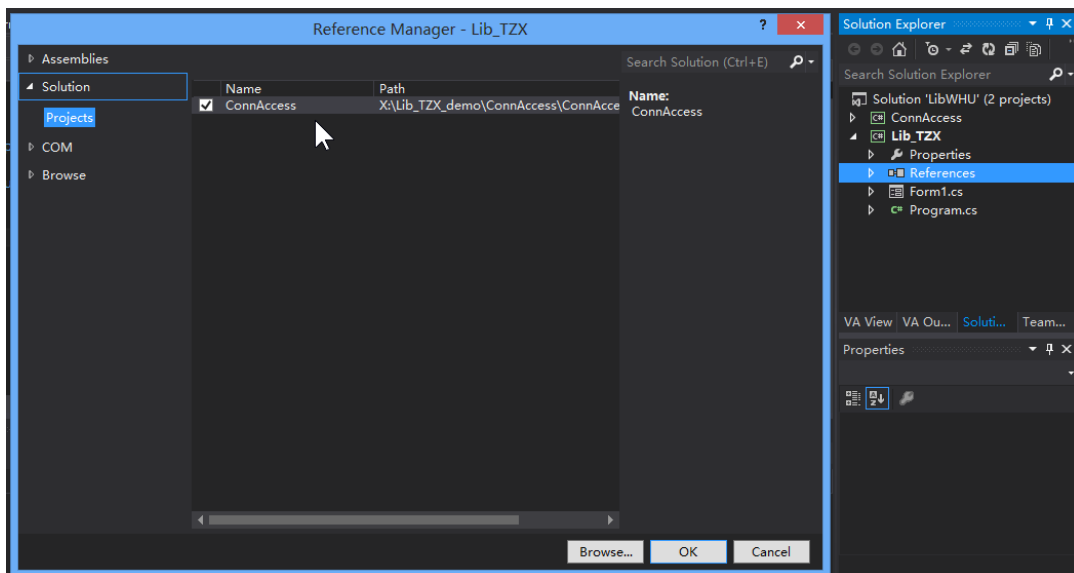
用简单的界面先测试数据库连接可用性



```
47         Connection = getConnection(); //新建连接
48         dataSet = new DataSet();
49         OleDbDataAdapter OleDbDA = new OleDbDataAdapter(strSQL, Connection); //新建数据适配器
50         OleDbDA.Fill(dataSet, "objDataSet"); //填充数据
51     }
52     catch (Exception e)
53     {
54         MessageBox.Show(e.Message);
55         return null;
56     }
57     finally
58     {
59         Connection.Close(); //关连接
60         Connection = null;
61     }
62     MessageBox.Show("Yeah, it works~~~~~");
63     return dataSet;
64 }
```

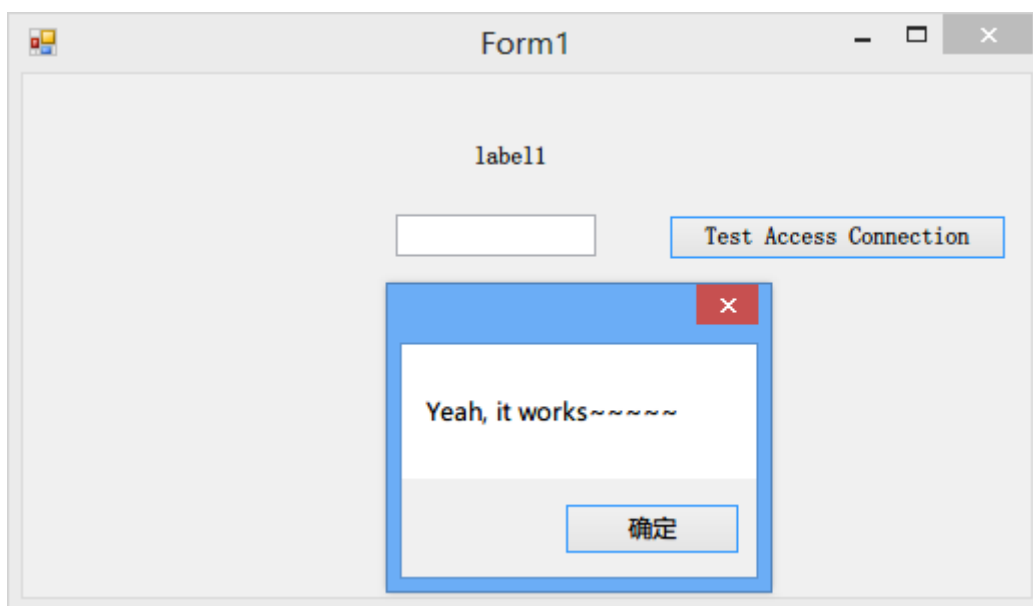
在图书馆系统主界面程序里添加数据库连接的 references





添加完后,再加上相应调用语句, 运行。

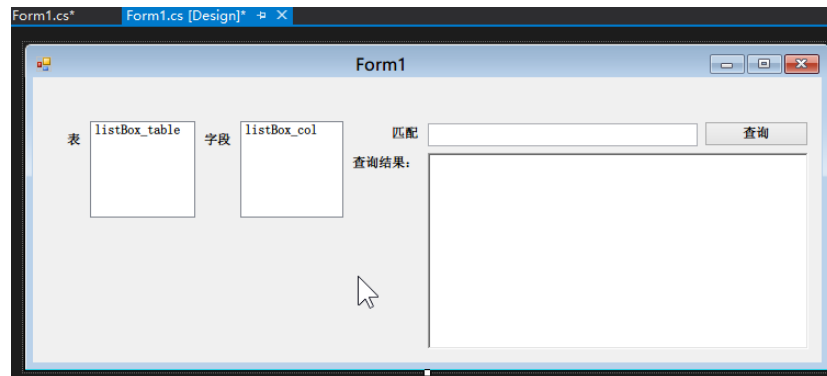
```
11 namespace Access
12 {
13     public partial class Form1 : Form
14     {
15         private access ac;
16         private DataSet ds;
17         private String sql;
18     }
19 }
```



成功。可以放心地进行下一步用户界面的设计和数据库完善了。

4) 按照需求设计用户界面

从一个简单的 Implementation, 逐步细化完善



加入 namespace

```
Program.cs  access.cs  Form1.cs  Form1.cs [Design]
→ Form1.cs  → X:\Lib_TZX_demo\Access\Form1.cs
Access.Form1
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Windows.Forms;
9 using Conn2Access;
```

在 Initialization 里调用一些自己的函数做一些界面的预处理
(诸如设置空间可见行,可读写,数据项等等)

```
19 public Form1()
20 {
21     InitializeComponent();
22 }
```

数据项的添加由于我不太熟悉如何获得数据库的表和列, 先硬编码写好, 日后再完善。

```
25 // set controls behaviours
26 void initControls()
27 {
28     this.richTextBox_out.ReadOnly = true;
29     setListTables();
30     setListCols();
31 }
32
33 // Set List Items
34 void setListTables()
35 {
36     this.listBox_table.Items.Add("table 1");
37     this.listBox_table.Items.Add("table 2");
38     this.listBox_table.Items.Add("Books");
39     this.listBox_table.Items.Add("table 3");
40     this.listBox_table.Items.Remove("table 3");
41     this.listBox_table.SelectedIndex = 2;
42 }
43 void setListCols()
```

查询做灵活的处理, 从界面的控件获取信息生成查询语句, 分成了三个部分:

1. qrySelectClause
2. qryFromClause
3. qryWhereClause

同样, 细节也没有太在意, 只是搭了个框架, 硬编码。

```
56 // Get SQL Params from Controls
57 string qrySelectClause()
58 {
59     return "select *";
60 }
61 string qryFromClause()
62 {
63     return " from " + this.listBox_table.SelectedItem.ToString();
64 }
65 string qryWhereClause()
66 {
67     if (this.textBox_match.Text.ToString() != "")
68     {
69         return " where BookName='" + this.textBox_match.Text.ToString() + "
70     }
71     else
72     {
73         return "";
74     }
75 }
```

然后就是 查询语句的执行和结果的输出

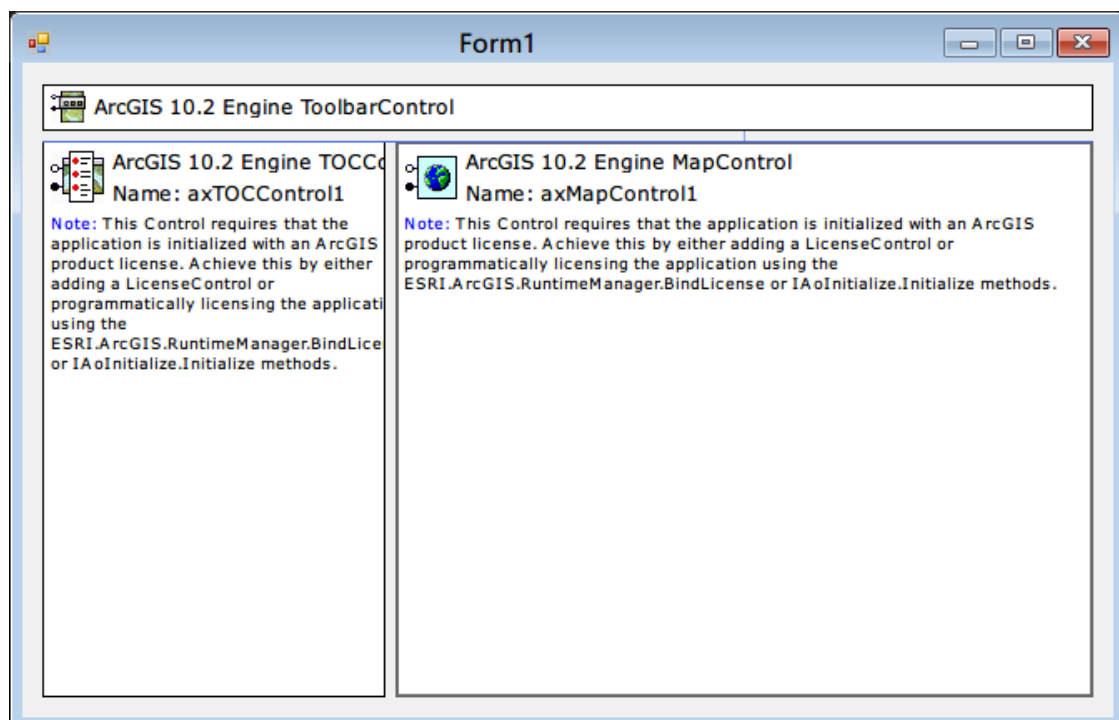
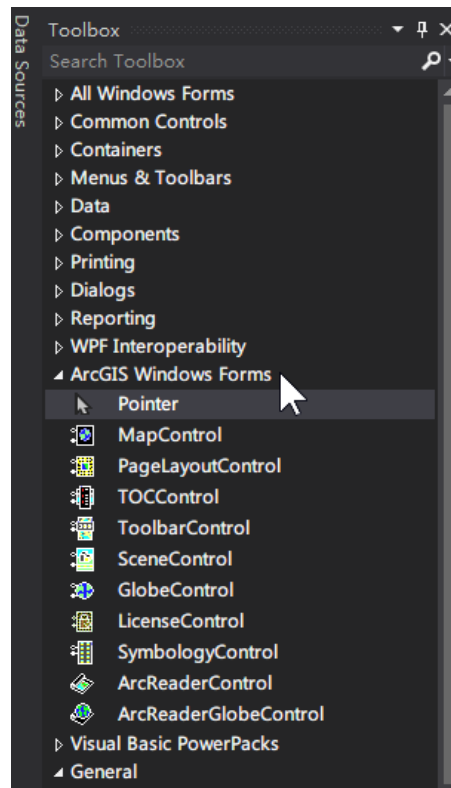
```
59 public bool ExeSQL(string strSQL)
60 {
61     OleDbConnection Connection = null;
62     bool resultState = false;
63
64     Connection = getConnection();
65     OleDbTransaction myTrans = Connection.BeginTransaction();
66     OleDbCommand command = new OleDbCommand(strSQL, Connection, myTrans);
67
68     try
69     {
70         command.ExecuteNonQuery();
71         myTrans.Commit();
72         resultState = true;
73     }
74     catch(Exception e)
```

```
77 String dataset2string(DataSet ds)
78 {
79     String output = "";
80     for (int i = 0; i < ds.Tables.Count; i++)
81     {
82         DataTable table = ds.Tables[i];
83         for (int j = 0; j < ds.Tables[i].Rows.Count; j++)
84         {
85             DataRow row = ds.Tables[i].Rows[j];
86             string id = "ID: " + row["ID"].ToString();
87             string name = "\tBookName: " + row["BookName"].ToString();
88             output += id;
89             output += name;
90             output += '\n';
91         }
92     }
93     return output;
94 }
95
```

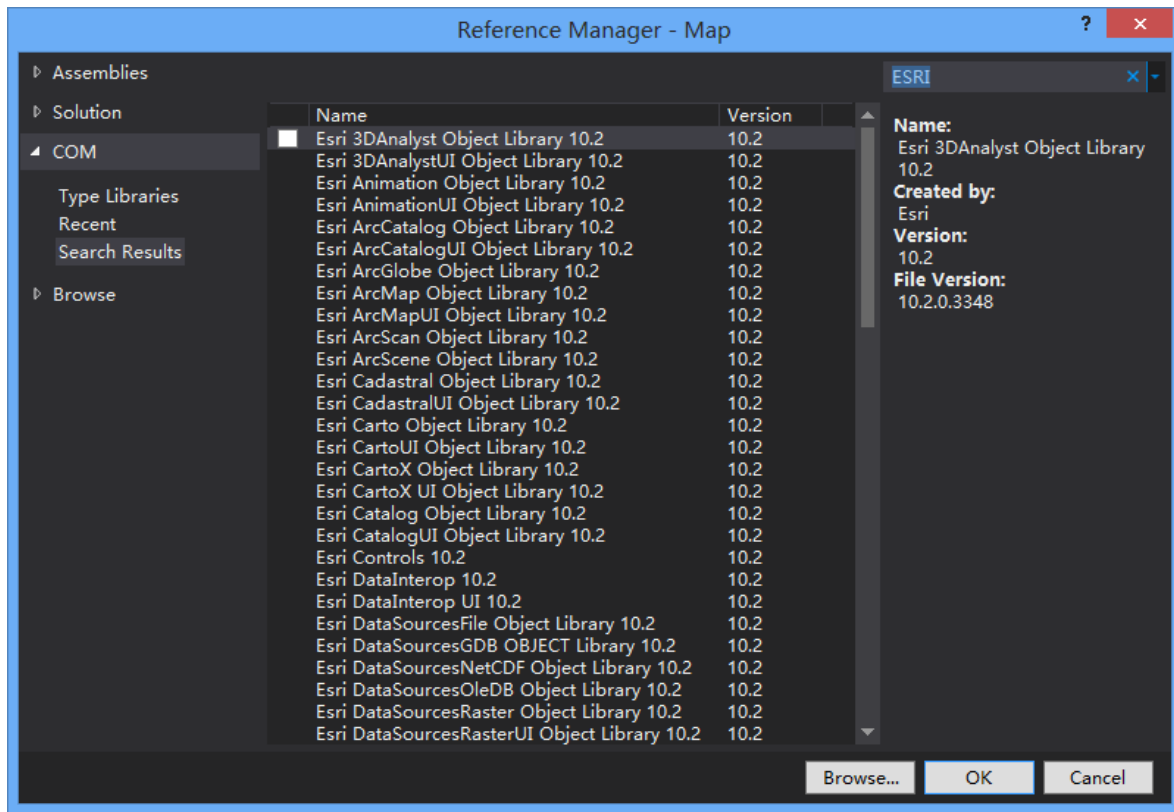
处理界面的消息响应

```
97 private void button_search_Click(object sender, EventArgs e)
98 {
99     sql = qrySelectClause() +
100         qryFromClause() +
101         qryWhereClause();
102
103     ds = ac.ReturnDataSet(sql);
104     string output = "Query Result: \n\n\n";
105     if (ds == null && ds.Tables.Count <= 0)
106     {
107         MessageBox.Show("No Matching Items");
108         return;
109     }
110     output += dataset2string(ds);
111     this.richTextBox_out.Text = output;
112 }
```

添加地理信息系统的地图功能,新建一个 Form, 由于安装了 ArcEngine, ToolBox 里有很多方便的控件可以直接使用,用这些可以方便地创建一个地图查询显示界面。



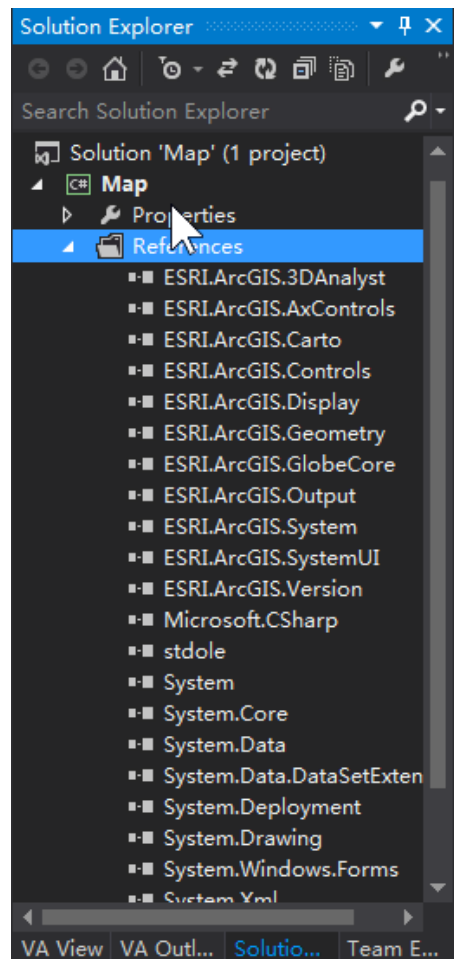
界面有了,但还是不能运行的,需要做些添加引用等额外处理。



```
84     private ESRI.ArcGIS.Controls.AxMapControl axMapControl1;  
85     private ESRI.ArcGIS.Controls.AxTOCControl axTOCControl1;  
86     private ESRI.ArcGIS.Controls.AxToolbarControl axToolbarControl1;
```

```
25     /// <summary>  
26     /// Required method for Designer support - do not modify  
27     /// the contents of this method with the code editor.  
28     /// </summary>  
29     private void InitializeComponent()  
30     {  
31         System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(  
32             this.axMapControl1 = new ESRI.ArcGIS.Controls.AxMapControl();  
33             this.axTOCControl1 = new ESRI.ArcGIS.Controls.AxTOCControl();  
34             this.axToolbarControl1 = new ESRI.ArcGIS.Controls.AxToolbarControl();  
35             ((System.ComponentModel.ISupportInitialize)(this.axMapControl1)).BeginInit();  
36             ((System.ComponentModel.ISupportInitialize)(this.axTOCControl1)).BeginInit();  
37             ((System.ComponentModel.ISupportInitialize)(this.axToolbarControl1)).BeginInit();  
38             this.SuspendLayout();  
            ↑
```

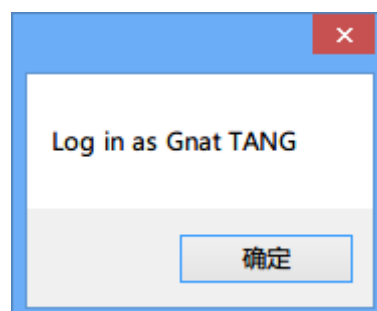
最后, Map 界面的引用如下:



加上简单的登陆,就大功告成。运行。



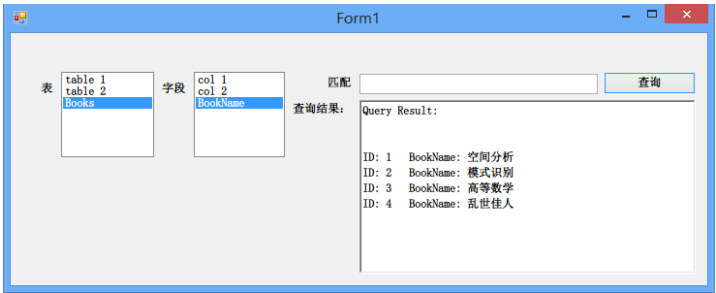
登陆后弹出提示信息



主界面



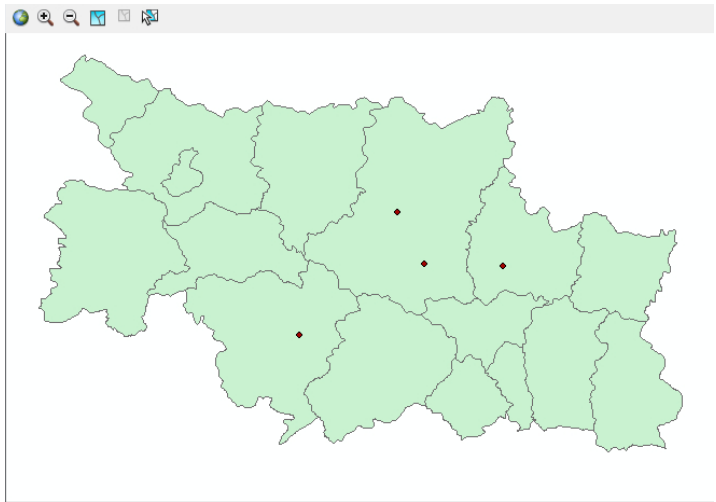
检索全部



检索某一本书



图书馆查询



别看界面略丑,但这也是一个带有地理信息功能的图书馆系统。

五、软件(程序)介绍

本信息系统能实现登录、注册,图书查询、作者查询、图书馆地理信息查询,还兼有注册用户的书评分享,图书交友等功能。

系统现阶段虽简陋粗旷,但整体架构灵活,基于插件的体系易于扩展,模块化的系统分工也让其维护升级更新更容易。

六、本次 GIS 程序设计与开发的总结

本次实习学习了用 C#和 .Net、ArcEngine 开发信息系统的基本流程。

巩固了数据库的建立,学习了数据库的连接。

额外地,还学习了 Svg 图的制作,文档撰写也越来越得心应手。

七、存在的问题及进一步改进的建议

1. 一个系统的建立应当从简单的实现开始,最好有一个作参考的最小案例。而本次实习居然没有一个好的实习指导书, 给定了一个方向,但大多数人都会茫然没有头绪。而且没有时间表 (或者说,没有明确的基准点。“今天干,明天收工”, 这种计划, 和没有没有区别。)

2. 数据库的选择也应当从小巧考虑。选择 Oracle 除了给自己添麻烦, 并没有什么实际意义,不会让你对数据库有更深刻的理解。而且,安装所带来的所有问题不是一个数据库使用者应当去处理的, 那是系统管理员、数据库管理员的工作。应当注意应用开发者在一个体系中的角色,不应该一人任务独担。总之,安装 Oracle 是个力气活,完全没有任何意义,不会让人学习到任何东西。一个人装了几十几百遍遍 Windows 系统,除了手熟, 以及浪费几十遍的装机时间,没有其他。

3. 开源软件更利于学习。数据库可以选择 PostgresQL, 地理信息系统可选择 Grass, Q GIS。界面还可以用浏览器, Web 框架用 BootStrap + d3, 这些都是在 Github 开源的, 十分流行且成熟的方案,而且不要钱, 不存在版权的问题。开源的趋势, 比几年人们所能想象的大的多,就连微软和 NASA 都渐渐开放了。另外,在地理信息系统中, Python 比 .Net 更通用。 Ruby on Rails 作为最快捷的 Web 框架, 比其 .Net 也是优势明显。未来是互联网时代, 作为一个有眼光的人,我觉得, flex 注定被淘汰,而且会很快, flash 也是; .Net 就算再强, 受到世界上最大的两家科技公司(Apple 和 Google) 以及开源框架的围攻, 注定失利。UI 会逐渐等同于 HTML5。

4. 我记得有次编程作业, 然后怎么就没有了。