**《GIS设计与开发》报告**

**姓 名 武成龙**

**学 号 07192329**

**班 级 地理信息科学19-1班**

**环境与测绘学院**

**2022.6**

**上机实验1 基于组件的常用GIS功能实现**

**【上机目的】**

掌握基于ArcGIS Engine的GIS基本功能的开发方法。

**【上机内容】**

在入门实验的基础上用C#（或其他高级程序设计语言）结合ArcGIS Engine开发一个包含GIS基本操作的应用程序，实现GIS基本操作（放大、缩小、漫游、全图显示等）、打开/保存Mxd地图文档、点选查询、距离量算/面积量算、图形和属性数据互查[可选]、缓冲区查询/分析[可选]。（说明：可选内容二选一。）

**【上机要求】**

1．完成代码的编写工作，基于ArcGIS Engine实现GIS基础功能。

2．了解利用ComGIS进行应用型地理信息系统建设工作的基本思路。

3．独立实验，认真、按时完成上机实验报告。

**【上机步骤】**

**1.窗口设计**

添加基础控件，设计主界面窗口，功能加入如菜单栏、工具条（放大、缩小、漫游、全图显示等）、显示界面、属性界面等。

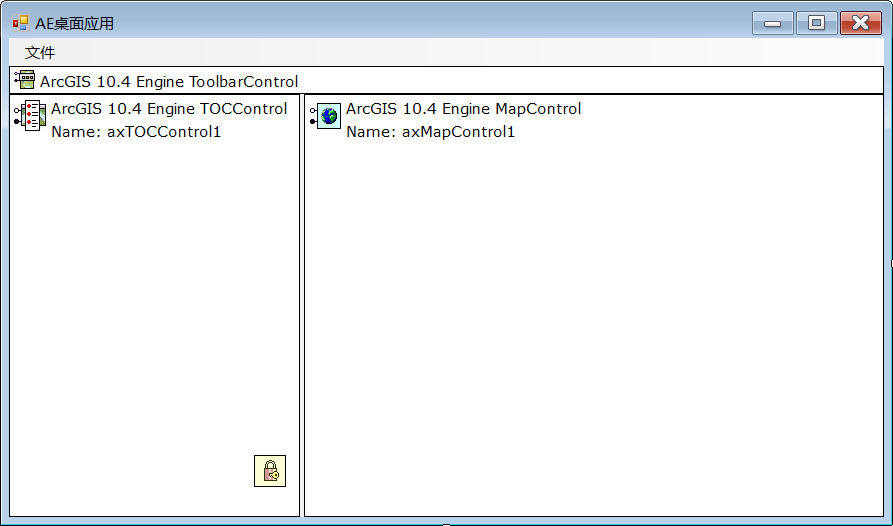


图1 窗口设计

**2.添加地图文档并设置工具条和目录控件与地图控件的关联**

设置完毕后进入系统。

默认打开“0617上机4.mxd”。

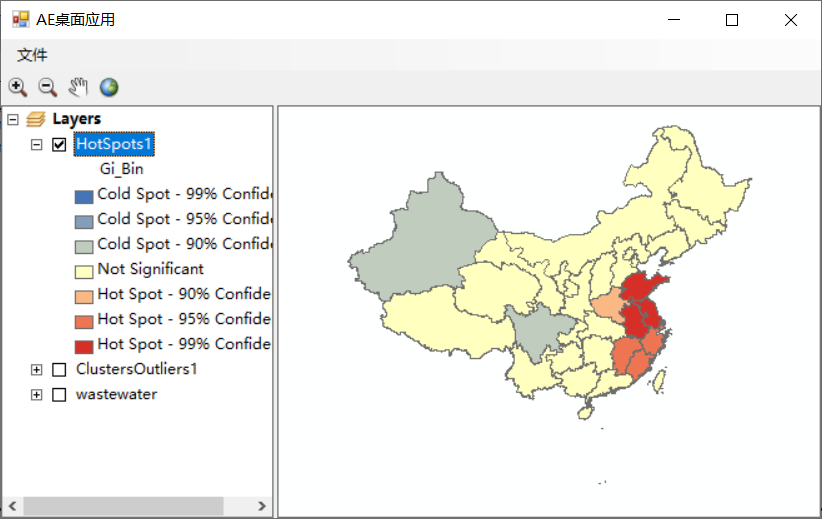


图2 添加地图文档

**3.具体功能实现**

（1）打开文档、保存文档、添加数据

添加其他的栅格数据，如图。

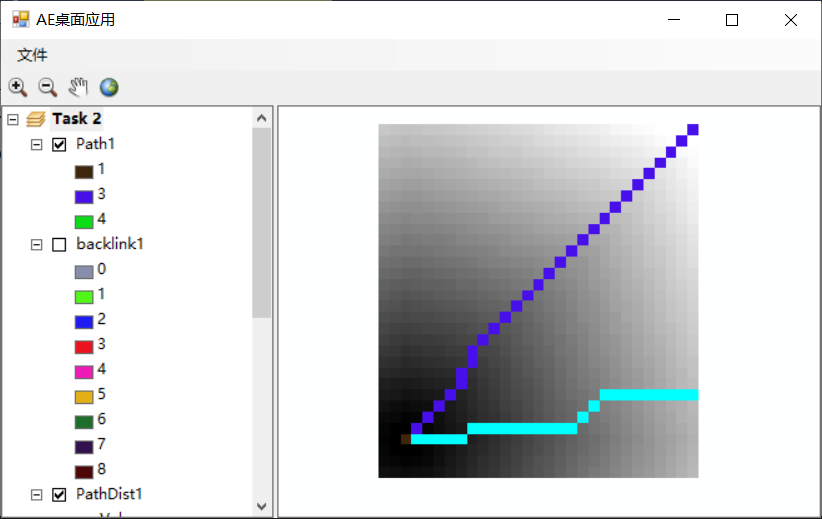


图3 添加数据

保存地图的主要代码如下：

private void 保存地图ToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog pSaveFileDialog = new SaveFileDialog();

pSaveFileDialog.Title = "请选择保存路径";

pSaveFileDialog.OverwritePrompt = true;

pSaveFileDialog.Filter = "ArcMap文档(\*.mxd)|\*.mxd|ArcMap模板(\*.mxt)|\*.mxt";

pSaveFileDialog.RestoreDirectory = true;

if (pSaveFileDialog.ShowDialog() == DialogResult.OK)

{

sMxdFileName = pSaveFileDialog.FileName;

}

else

{

return;

}

}

pMapDocument.New(sMxdFileName);

pMapDocument.ReplaceContents(axMapControl1.Map as IMxdContents);

pMapDocument.Save(pMapDocument.UsesRelativePaths,true);

pMapDocument.Close();

MessageBox.Show("保存地图文档成功！");

}

（2）距离量算/面积量算

主要代码：

// 用于判断何种工具

string mTool;

public int mQueryModel;

public int mLayerIndex;

private void 距离量算ToolStripMenuItem\_Click(object sender, EventArgs e)

{

mTool = "DistanceMearsure";

}

private void 面积量算ToolStripMenuItem\_Click(object sender, EventArgs e)

{

mTool = "AreaMearsure";

}

实现效果：

图4 距离量算/面积量算

（3）空间查询

主要代码：

private void axMapControl1\_OnMouseDown(object sender, ESRI.ArcGIS.Controls.IMapControlEvents2\_OnMouseDownEvent e)

{

this.axMapControl1.Map.ClearSelection();

if (mTool == "DistanceMearsure")

{

IPolyline polyline = DistanceMeasure();

MessageBox.Show("距离为" + Convert.ToInt64(polyline.Length).ToString());

}

else if (mTool == "AreaMearsure")

{

IPolygon polygon = AreaMeasure();

IArea pArea = polygon as IArea;

MessageBox.Show("面积为" + Convert.ToInt64(Math.Abs(pArea.Area)).ToString());

}

else if(mTool == "SpatialQuery")

{

this.axMapControl1.Map.ClearSelection();

ESRI.ArcGIS.Carto.IActiveView qActiveView = this.axMapControl1.ActiveView;

ESRI.ArcGIS.Geometry.IPoint pPoint = qActiveView.ScreenDisplay.DisplayTransformation.ToMapPoint(e.x, e.y);

ESRI.ArcGIS.Geometry.IGeometry pGeometry = null;

switch (this.mQueryModel)

{

case 0:

pGeometry = this.axMapControl1.TrackRectangle();

break;

case 1:

pGeometry = this.axMapControl1.TrackCircle();

break;

case 2:

pGeometry = AreaMeasure();

axMapControl1.ActiveView.GraphicsContainer.DeleteAllElements();

axMapControl1.Refresh();

break;

}

ESRI.ArcGIS.Carto.IFeatureLayer pFeatureLayer = this.axMapControl1.Map.get\_Layer(this.mLayerIndex) as ESRI.ArcGIS.Carto.IFeatureLayer;

DataTable pDataTable = this.LoadQueryResult(axMapControl1, pFeatureLayer, pGeometry);

panel1.Visible = true;

this.dataGridView1.DataSource = pDataTable.DefaultView;

this.dataGridView1.Refresh();

}

}

实现效果：

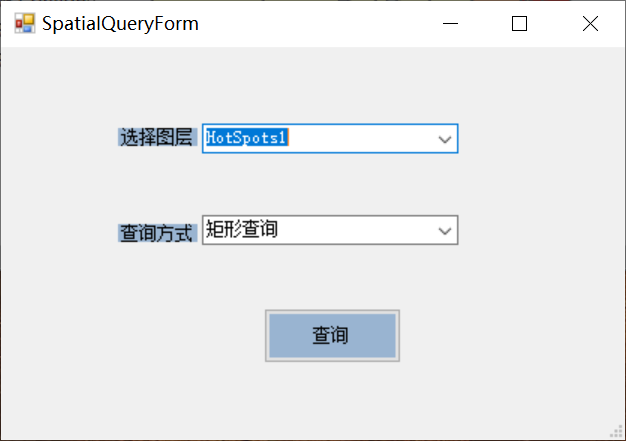
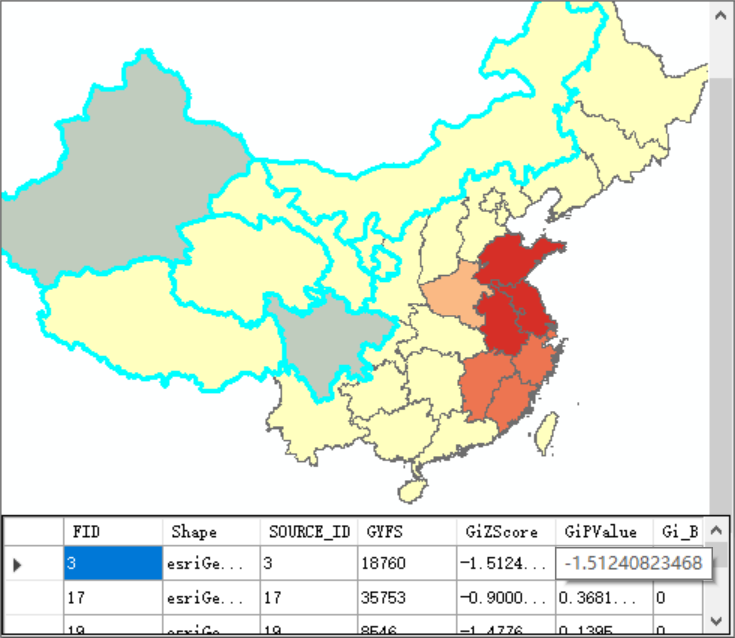
 

图5 查询窗体与查询结果

（4）右键菜单

由于右键功能异常，故写成了点击按钮触发方式，显示属性表和删除图层效果相同。

主要代码：

label1.Text = pFeatureLayer.Name.ToString() + "的属性表";

name = pFeatureLayer.Name.ToString();

IFields pFields;

pFields = pFeatureLayer.FeatureClass.Fields;

dataGridView1.ColumnCount = pFields.FieldCount;

for (int i = 0; i < pFields.FieldCount; i++)

{

string fldName = pFields.get\_Field(i).Name;

dataGridView1.Columns[i].Name = fldName;

dataGridView1.Columns[i].ValueType = System.Type.GetType(ParseFieldType(pFields.get\_Field(1).Type));

}

IFeatureCursor pFeatureCursor;

pFeatureCursor = pFeatureLayer.FeatureClass.Search(null, false);

IFeature pFeature;

pFeature = pFeatureCursor.NextFeature();

while (pFeature != null)

{

string[] fldValue = new string[pFields.FieldCount];

for (int i = 0; i < pFields.FieldCount; i++)

{

string fldName;

fldName = pFields.get\_Field(i).Name;

if (fldName == pFeatureLayer.FeatureClass.ShapeFieldName)

{

fldValue[i] = Convert.ToString(pFeature.Shape.GeometryType);

}

else

{

fldValue[i] = Convert.ToString(pFeature.get\_Value(i));

}

}

dataGridView1.Rows.Add(fldValue);

pFeature = pFeatureCursor.NextFeature();

}

实现效果：

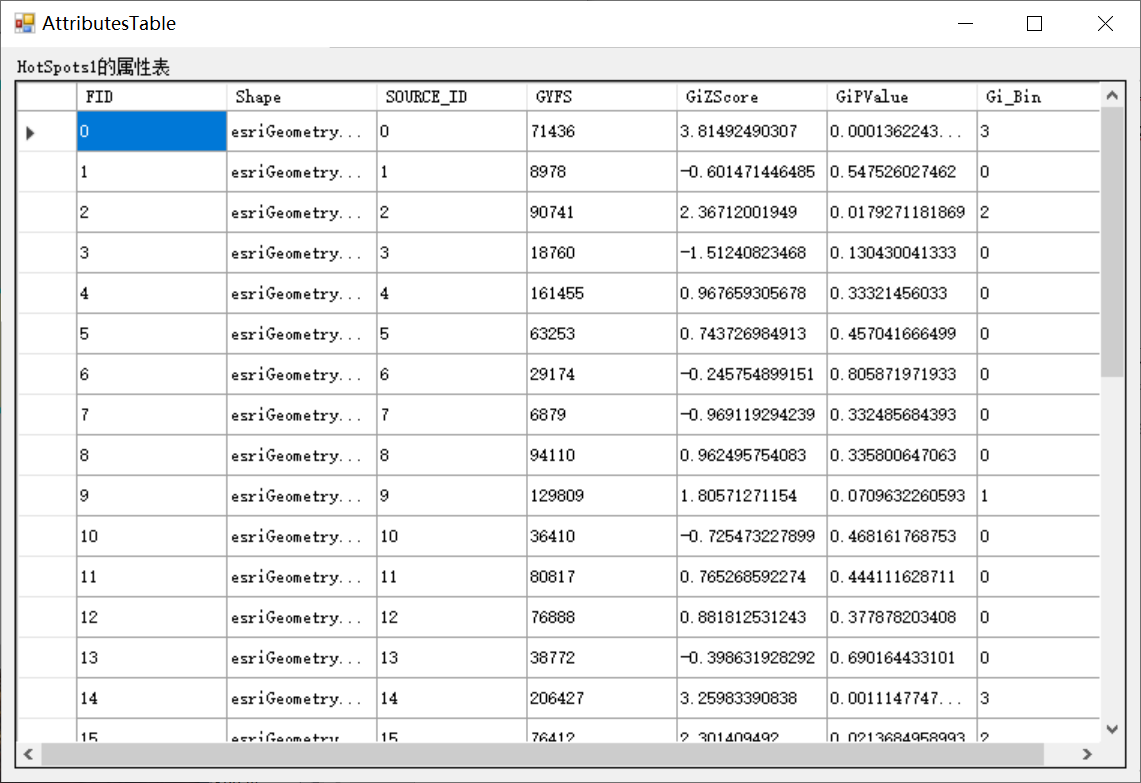


图6 查看属性表

**【上机体会】**

本次实验主要是提升AE开发技能，针对不同的功能需要不同的代码来实现，但编写代码本身并不是全部的难点，代码与控件之间的组织方式也是需要仔细分析与处理的，在代码出现问题的时候也需要有一定的调试能力。

最有特点的是用if-else if语句判断鼠标点击事件的逻辑关系，即按钮改变一特点变量值，这个值决定了执行何种功能（包括空间量算和空间查询）。

**上机实验2 基于ArcSDE的地理空间数据库的建立**

**【上机目的】**

了解空间数据存储的发展历程，进一步加深对空间数据库的理解，掌握空间数据的存储结构、空间数据库的简单建立及访问方法。

**【上机内容】**

利用ESRI公司的ArcSDE技术结合一种常用的关系数据库管理系统（如SQL Server等）建立地理空间数据库。对该数据库中的空间数据存储结构进行分析，从而进一步认识空间数据的存储方式。用代码对存放在空间数据库中的空间数据进行访问，并实现GIS的基本操作功能。

**【上机要求】**

1．独立实验，认真、按时完成实验并提交上机实验报告。

**【上机方法】**

1. 启动数据库服务器。
2. 在ArcCatalog中建立与数据库服务器的连接。
3. 将空间数据导入数据库。
4. 用代码对空间数据库中的空间数据进行访问，并实现GIS基本操作。
5. 进行实验小结。

**【上机步骤、主要代码及界面】**

**1.数据库部分**

启动 ArcCatalog ，连接 SQL“云南”数据库。

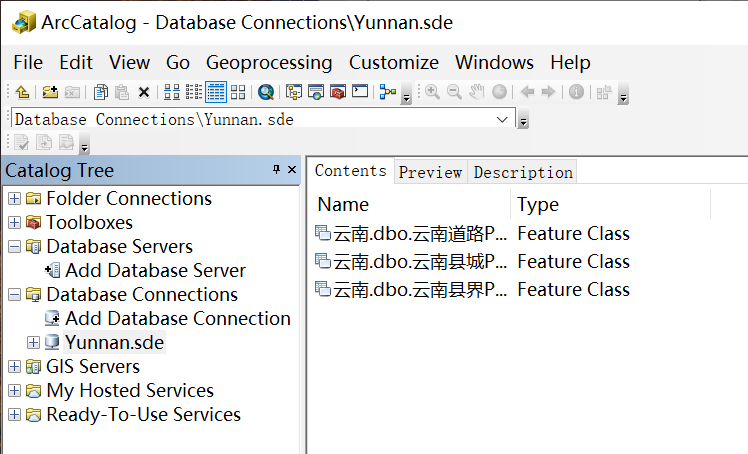
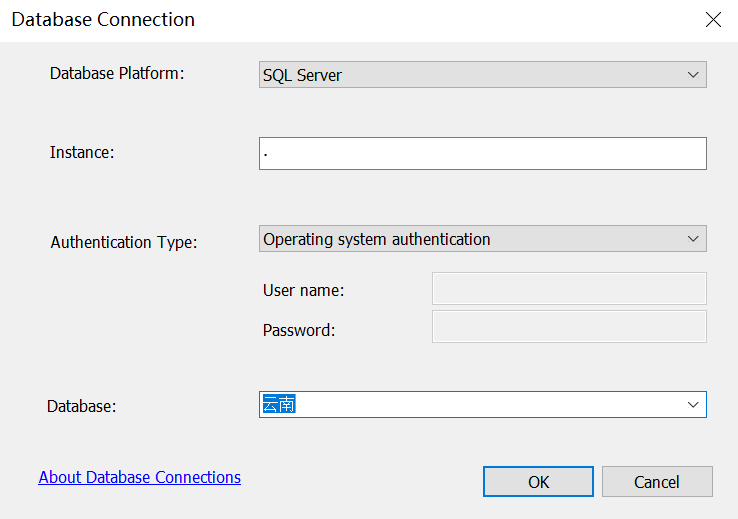


图1 连接数据库与结果

**2.程序设计部分**

添加“连接数据库”按钮，添加代码。

主要代码：

private void 连接数据库ToolStripMenuItem\_Click(object sender, EventArgs e)

{

IWorkspace pWkspace = null;

Type factoryType = Type.GetTypeFromProgID("esriDataSourcesGDB.SdeWorkspaceFactory");

IWorkspaceFactory workspaceFactory = (IWorkspaceFactory)Activator.CreateInstance(factoryType);

pWkspace = workspaceFactory.OpenFromFile(@"C:\Users\10599\AppData\Roaming\ESRI\Desktop10.4\ArcCatalog\Yunnan.sde", 0);

//添加数据

IEnumDataset enumDataSet = pWkspace.get\_Datasets(esriDatasetType.esriDTAny);

enumDataSet.Reset();

IDataset dataSet = enumDataSet.Next();

while (dataSet != null)

{

if (dataSet is IFeatureDataset)

{

IFeatureWorkspace featureWorkspace = pWkspace as IFeatureWorkspace;

IFeatureDataset featureDataset = featureWorkspace.OpenFeatureDataset(dataSet.Name);

IEnumDataset pEnumDataSet1 = featureDataset.Subsets;

pEnumDataSet1.Reset();

IDataset pDataset1 = pEnumDataSet1.Next();

if (pDataset1 is IFeatureClass)

{

IFeatureLayer pfeatureLayer = new FeatureLayerClass();

pfeatureLayer.FeatureClass = featureWorkspace.OpenFeatureClass(pDataset1.Name);

pfeatureLayer.Name = pfeatureLayer.FeatureClass.AliasName;

axMapControl1.Map.AddLayer(pfeatureLayer);

axMapControl1.ActiveView.Refresh();

}

}

else if (dataSet is IFeatureClass)

{

IFeatureWorkspace featureWorkspace = pWkspace as IFeatureWorkspace;

IFeatureClass feature = featureWorkspace.OpenFeatureClass(dataSet.Name);

IFeatureLayer layer = new FeatureLayerClass();

layer.FeatureClass = feature;

layer.Name = feature.AliasName;

axMapControl1.AddLayer(layer);

}

dataSet = enumDataSet.Next();

}

axMapControl1.Refresh();

MessageBox.Show("数据读取完毕！");

}

实现效果：

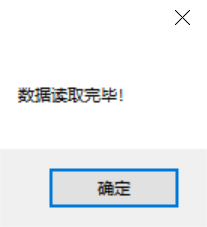
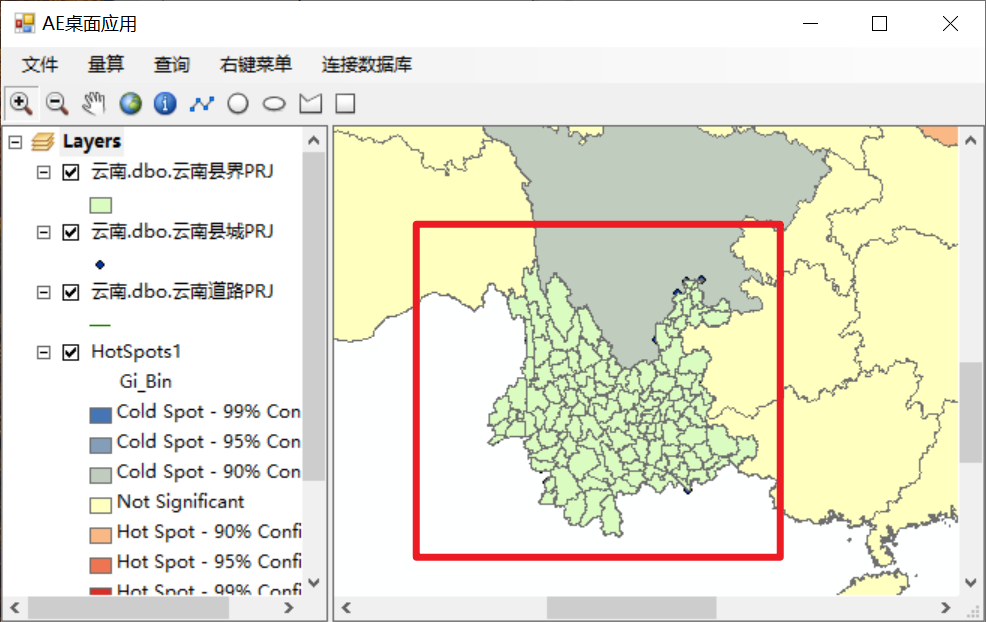


图2 连接数据库结果

**【上机体会】**

本次实验主要是针对从已有地理信息数据库中读取数据进行练习，进一步了解了ArcCatalog 的功能。把“云南”数据绑定在了按钮点击函数上，单击即显示。