

实验五 空间数据管理

1 背景知识

地理数据库 (Geodatabase, GDB) 表达地理数据主要有 5 种形式：矢量要素 (离散对象)、拓扑网络 (物流)、栅格 (连续的现象)、TIN (不规则三角网表面)、定位器和地址 (位置参考)。地理数据库的作用包括：(1) 存储要素的形状供 ArcGIS 空间操作，可定义和管理一组数据的地理坐标系；(2) 可构建集成拓扑关系的要素集；(3) 可定义对象与要素之间一般和任意的关系；(4) 通过域和验证规则增强属性的完整性；(5) 可将要素的自然行为绑定到存储要素的表中；(6) 可展现多版本以便于多用户编辑同一数据。

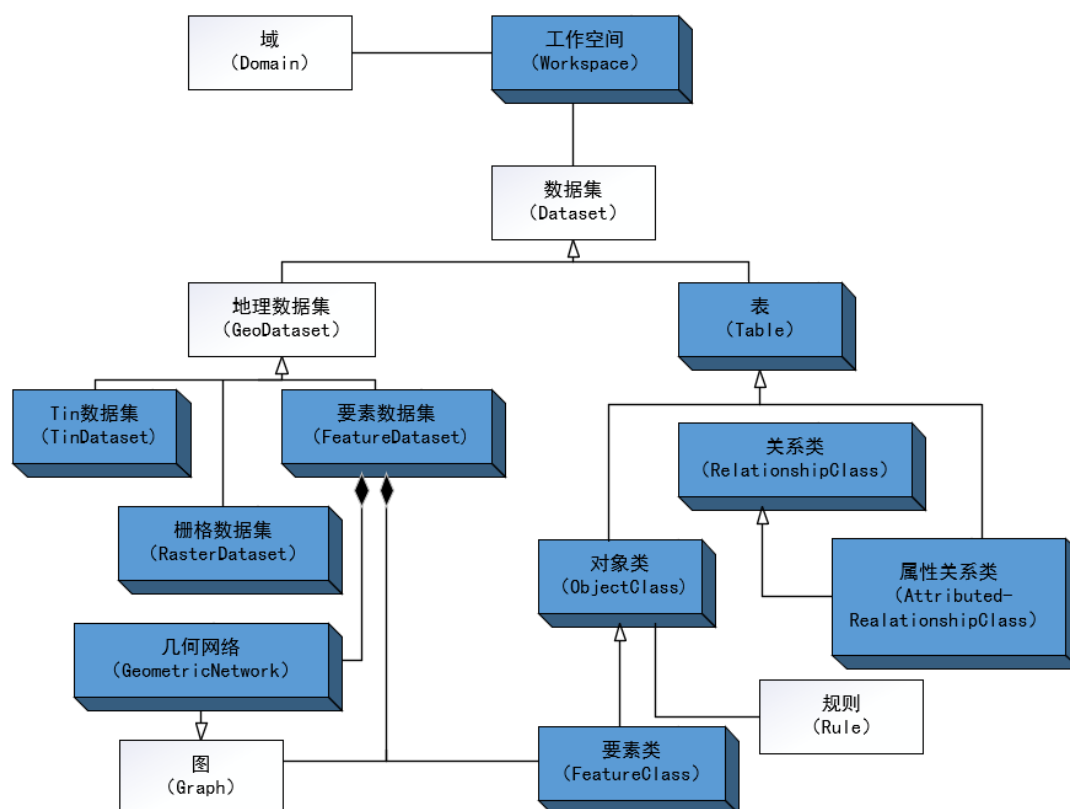


图 5-1 地理数据库 Geodatabase 模型

地理数据库 Geodatabase 模型的核心对象(图 5-1)有：(1)工作空间 Workspace 代表了一个地理数据库、Shapefile 文件夹或 Coverage 的工作空间；(2)域 Domains 指定定义属性的有效取值范围，可以是连续的变化区间，也可以是离散的取值集合；(3)数据集 Dataset 是数据的高级容器，任何数据的集合；(4)表 Table 是数据库中的二维表 (由行 Row 组成)，其列属性由字段集设置，对象类 ObjectClass 是具有面向对象特性的表，用于存储非空间数据，FeatureClass 是可以存储空间数据的对象类，规则 Rule 是对要素类或对象类的行为和取值加以约束的规则；

(5) 关系类 RelationshipClass 是通过表的外键表达的一对多或多对一的关系, 属性关系类 AttributedRelationshipClass 是用于存储多对多关系的表; (6) 地理数据集 GeoDataset 是包含地理数据的数据集, 要素数据集 FeatureDataset 是由要素类、几何网络和拓扑组成的数据集, TIN 数据集 TinDataset 是指一套带有 z 值的不规则三角网, 用它来精确表示表面, 栅格数据集 RasterDataset 是存贮有不同光谱或分类值的多光谱带的一个简单数据集或一个复合数据集; (7) 图 Graph 是用来形象描述图的一个要素类集合, 几何网络 GeometricNetwork 是在同一几何网络中充当结点和连接边的各种要素类, 必须组织在同一要素数据集中。

2 演示实例

本例演示的是通过 ArcSDE (空间数据库引擎 SDE, Spatial Database Engine) 在微软 SQL SERVER 数据库软件上配置和连接地理数据库 Geodatabase, 并通过 ArcObjects 程序开发访问该地理书库。具体步骤如下:

(1) 首先以数据库管理员 DBA (Database Administrator) 身份验证创建 SQL SERVER 数据库实例, 并记下登录名和密码。

(2) 创建地理数据库, 打开 ArcMap 的 ArcToolbox 工具箱, 浏览到【数据管理工具】→【地理数据库管理】→【创建企业级地理数据库】, 如图 5-2 所示。



图 5-2 打开 ArcToolbox 工具箱

(3) 创建企业级地理数据库，设置好相关参数，如图 5-3 所示，其中：(a) 数据库平台选择 SQL_SERVER；(b) 实例需要填写 SQL SERVER 实例名，这里填写计算机名；(c) 数据库管理员及密码，即创建 SQL SERVER 数据库的用户名及用户名密码；(d) 地理数据库管理员及密码，设置地理数据库管理员用户名（默认是 sde）及用户名密码；(e) 授权文件，具有 ArcGIS Server 的许可的*.ecp 授权文件。

创建企业级地理数据库

数据库平台
SQL_Server

实例
DESKTOP-0A280IH

数据库 (可选)
sde

☐ 操作系统身份验证 (可选)

数据库管理员 (可选)
sa

数据库管理员密码 (可选)
●●●●●●●●

☒ Sde 拥有的方案 (可选)

地理数据库管理员 (可选)
sde

地理数据库管理员密码 (可选)
●●●

表空间名称 (可选)

授权文件
E:\authorization_OAC8BE67.ecp

确定 取消 环境... 显示帮助 >>

图 5-3 创建企业级地理数据库

(4) 设置好参数后，点击【确定】开始创建企业级地理数据库，将创建 SDE 系统表等，出现图 5-4，表示创建成功。

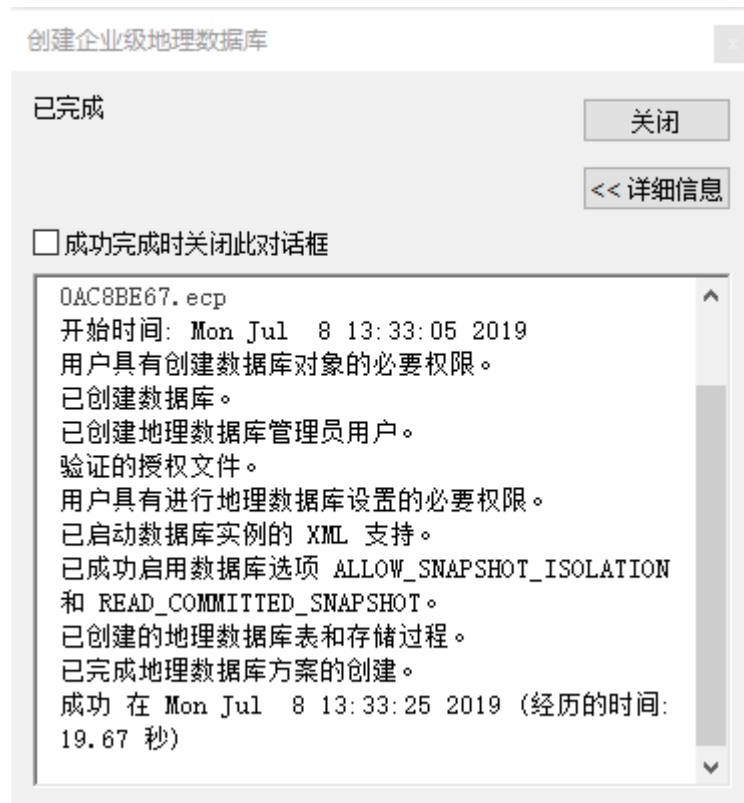


图 5-4 完成创建企业级地理数据库

(5) 创建企业级地理数据库完成后,可以在 SQL SERVER Management Studio 中查看结果,如图 5-5 所示。

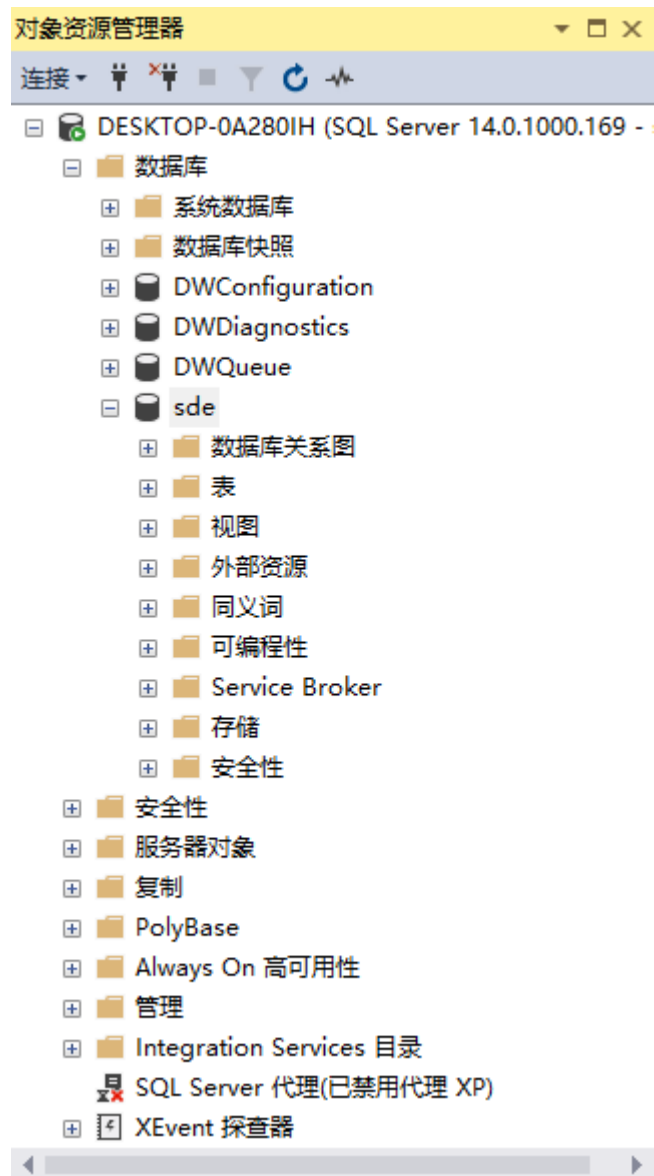


图 5-5 查看企业级地理数据库

(6) 打开 ArcCatalog 添加数据库连接，如图 5-6 所示。

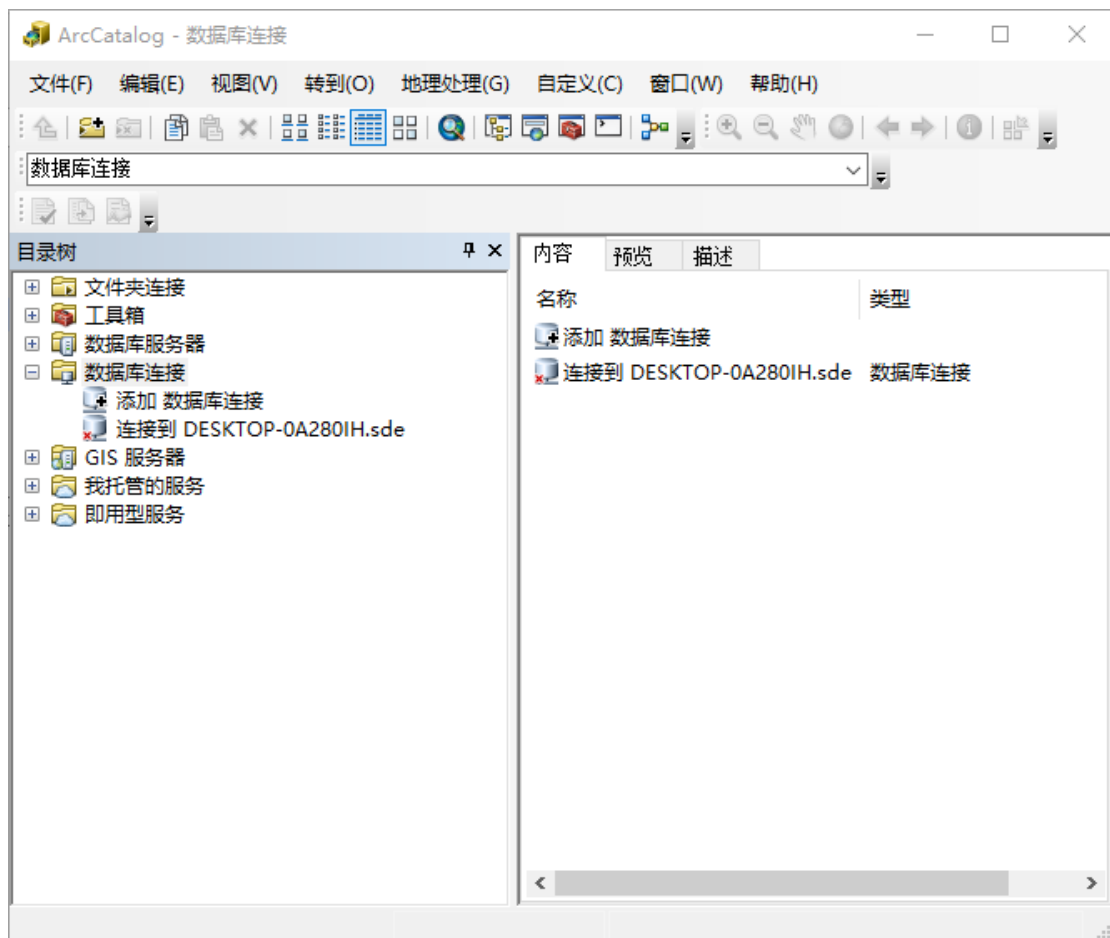


图 5-6 在 ArcCatalog 中添加数据库连接

(7) 双击添加数据库连接，设置好相关参数，如图 5-7 所示。

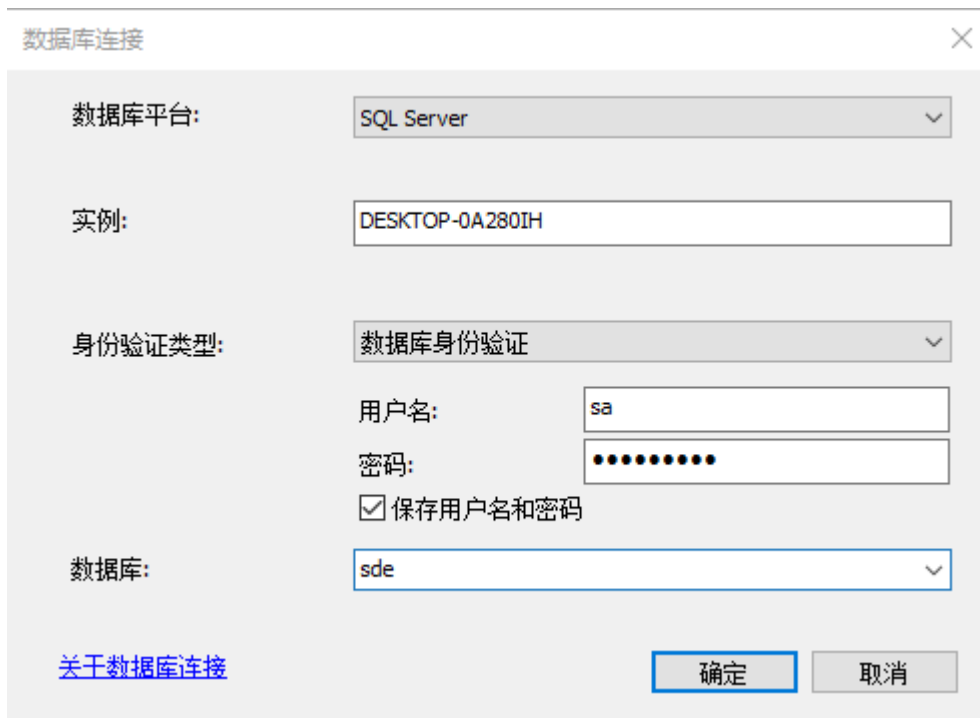


图 5-7 连接企业级地理数据库

(8) 单击右键选择为数据库新建一个要素数据集，如图 5-8 所示。

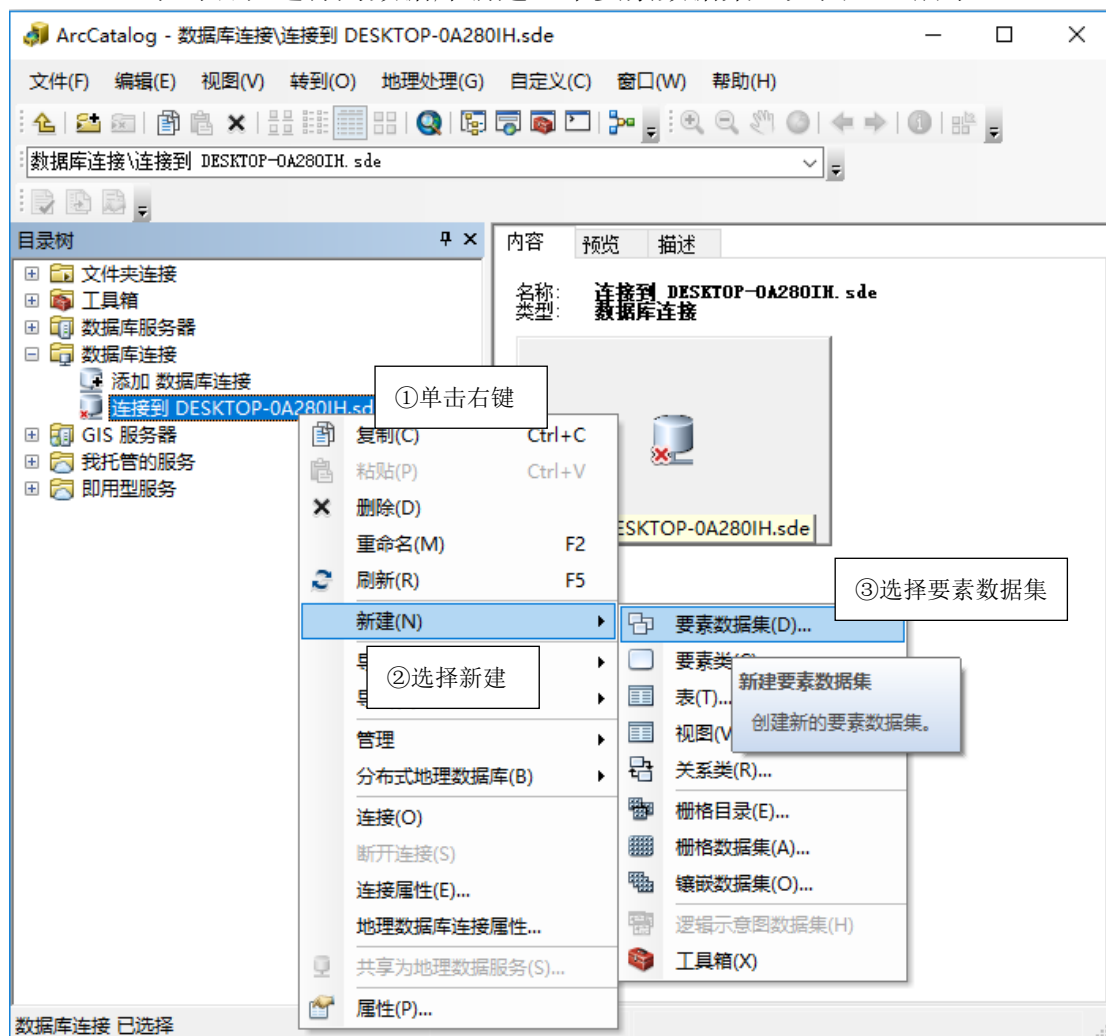


图 5-8 新建要素数据集

分别直接向数据库中导入要素类、向要素数据集中导入要素类，如图 5-9 所示。

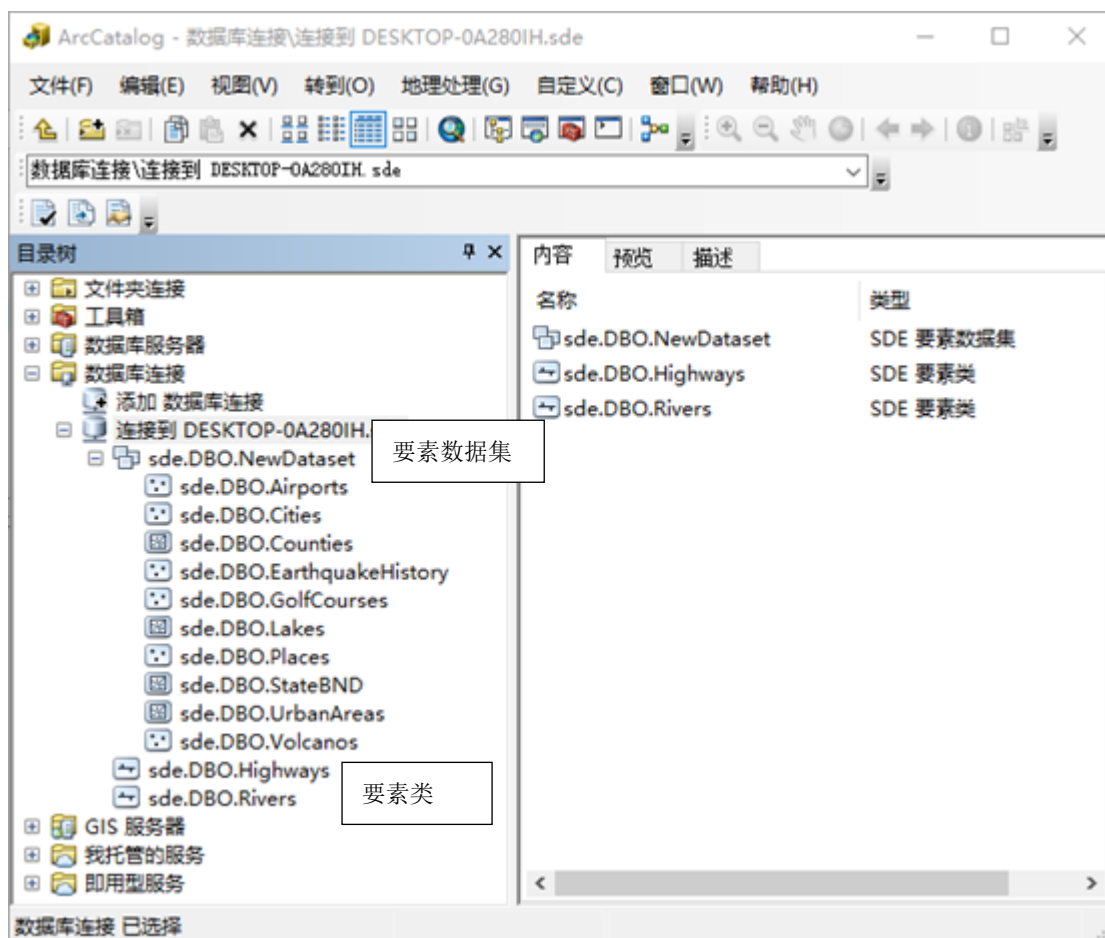


图 5-9 为空间数据库和要素数据集导入要素类

(9) 在 Visual Studio 中新建 MapControl Application 项目，选择【Visual C#】→【ArcGIS】→【Extending ArcObjects】→【MapControl Application】模板，名称为“MapControlApplication”，解决方案为“Test5”，如图 5-10 所示。

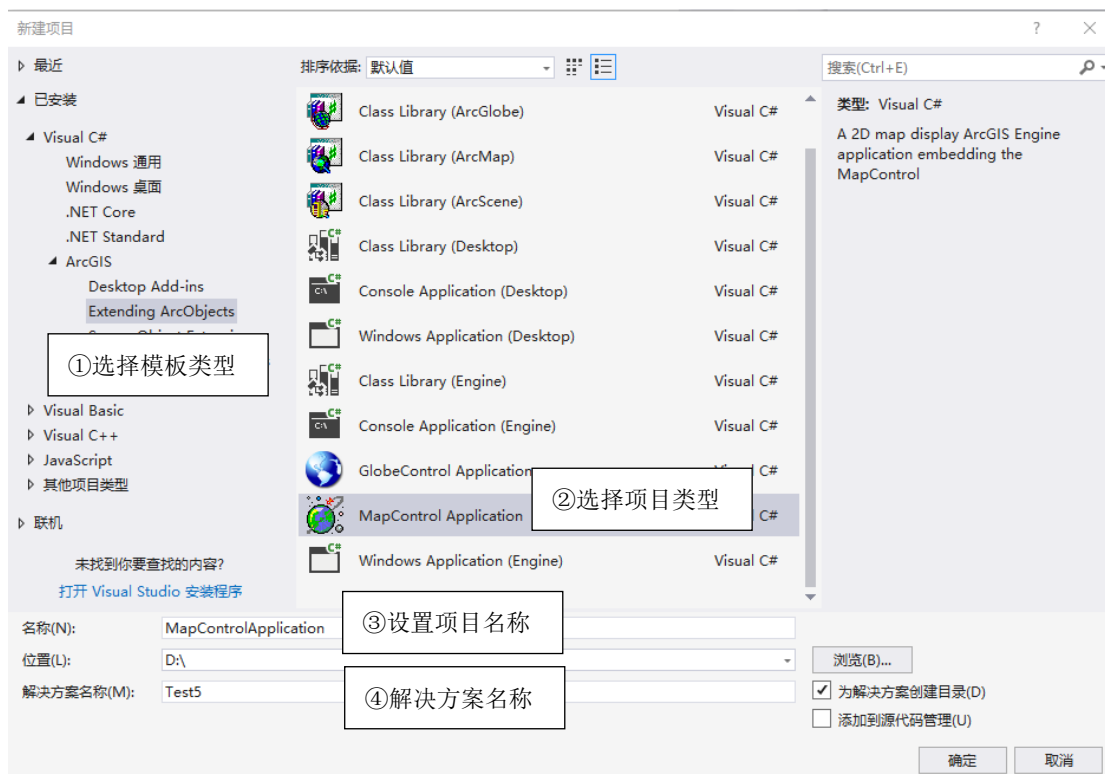


图 5-10 新建 MapControl Application 项目

在“MapControlApplication”项目的“MainForm”窗体中设置 LicenseControl 的许可权限，Products 下勾选“ArcGIS Engine Enterprise”，勾选“Shutdown this application if the selected licenses are no”，如图 5-11 所示。

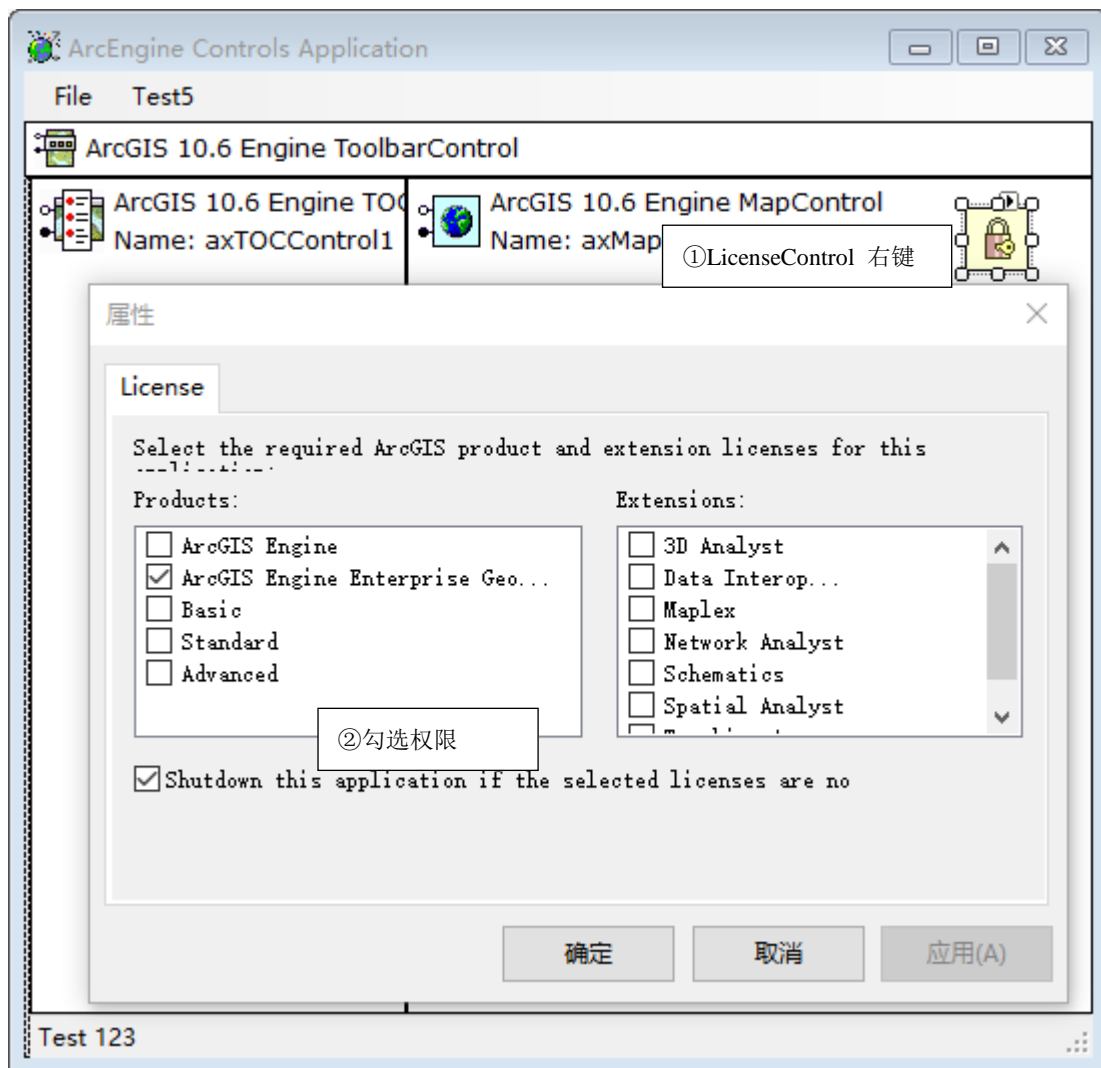


图 5-11 设置 License 权限

在“MainForm”窗体中添加菜单项“ConnectSDE”，如图 5-12 所示。双击该菜单项添加菜单事件响应函数。在 MainForm.cs 中添加引用：

`using ESRI. ArcGIS. Geodatabase;`

`using ESRI. ArcGIS. DataSourcesGDB;`

并将引用 DataSourcesGDB、Geodatabase 的嵌入互操作类型改为 False。

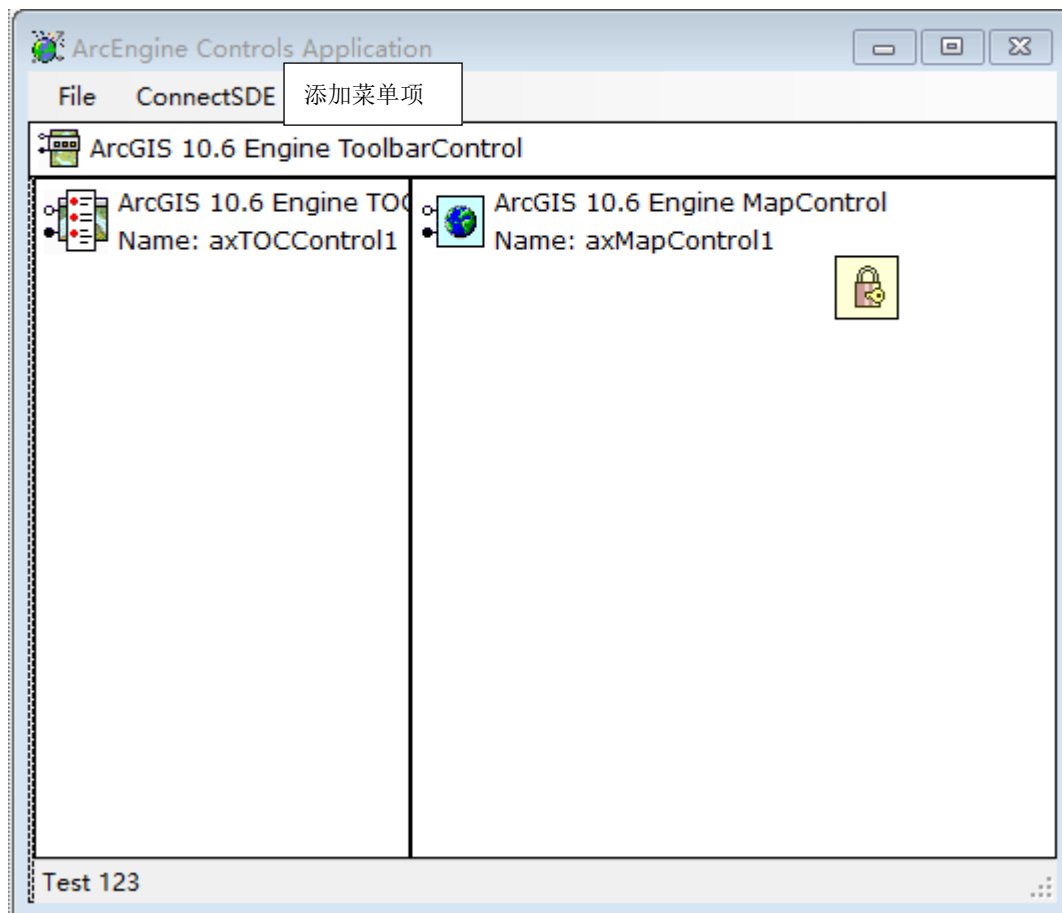


图 5-12 添加菜单项

MainForm.cs (节选)

功能：连接 SQL SERVER 数据库

```
private void connectSDEToolStripMenuItem_Click(object sender, EventArgs e)
{
    //数据库工作空间工厂对象
    IWorkspaceFactory wsFactory = new SdeWorkspaceFactoryClass();
    IPropertySet ps = new PropertySetClass(); //属性集
    ps.SetProperty("SERVER", "DESKTOP-0A280IH"); //服务器名称
    ps.SetProperty("INSTANCE", ""); //实例名
    ps.SetProperty("DATABASE", "sde"); //数据库名称
    ps.SetProperty("USER", "sa"); //用户名
    ps.SetProperty("PASSWORD", "123456789"); //密码
    //打开SDE数据库连接工作空间
    IWorkspace ipWorkspace = wsFactory.Open(ps, 0);
    //获得工作空间下的所有要素数据集Datasets
    IEnumDataset eds = ipWorkspace.get_Datasets(
        esriDatasetType.esriDTFeatureDataset);
    //依次获取每个数据集Dataset
    IDataset ds = null;
    ds = eds.Next();
    while(ds!=null)
    {

```

```

if (ds is IFeatureDataset)
{
    //获取数据集的子集
    IEnumDataset subs = ds.Subsets;
    subs.Reset();
    IDataset pds = subs.Next();
    //对子集进行遍历
    while (pds != null)
    {
        //判断是否是要素类
        if (pds.Type == esriDatasetType.esriDTFeatureClass)
        {
            IFeatureClass fc = pds as IFeatureClass;
            IFeatureLayer layer = new FeatureLayerClass();
            layer.Name = fc.AliasName;
            layer.FeatureClass = fc;
            //将要素图层添加至地图窗口
            axMapControl1.AddLayer(layer as ILayer);
        }
        pds = subs.Next();
    }
    ds = eds.Next();
}
//获取直接存储在工作空间下的所有要素类FeatureClasses
eds = ipWorkspace.get_Datasets(esriDatasetType.esriDTFeatureClass);
//获取每个要素类FeatureClass
IFeatureClass fcl = eds.Next() as IFeatureClass;
while (fcl != null)
{
    if (fcl is IFeatureClass)
    {
        IFeatureLayer layer = new FeatureLayerClass();
        layer.Name = fcl.AliasName;
        layer.FeatureClass = fcl; //为图层指定要素类
        //将图层添加至地图窗口
        axMapControl1.AddLayer(layer as ILayer);
    }
    fcl = eds.Next() as IFeatureClass;
}
}

```

工作空间工厂 **WorkspaceFactory** 对象是工作空间的发布者，支持对于文件系统和远程数据库工作空间的连接与访问。要操作各种类型的空间数据，首先要获得空间数据所在的工作空间。工作空间（**Workspace**）是 **Geodatabase** 的入口，但

它是一个可实例化类（Class），这意味着用户不能直接用 new 的方式新建它。为了获得一个工作空间，需要使用 WorkspaceFactory 对象来创建或打开一个工作空间 Workspace。

工作空间工厂 WorkspaceFactory 是一个抽象类，派生了若干的子类，例如 SdeWorkspaceFactory、AccessWorkspaceFactory、ShapefileWorkspaceFactory、FileGDBWorkspaceFactory 等。连接不同类型的空间数据库要由不同的工作空间工厂来打开对应的工作空间，表 5-1 列出了常用的工作空间工厂及相应的工作空间类型。

表 5-1 常用的工作空间工厂及相应的工作空间类型

工作空间工厂对象	使用文件类型	工作空间对象类型
AccessWorkspaceFactory	打开 Personal GeoDatabase	esriLocalDatabaseWorkspace
ArcinfoWorkspaceFactory	打开 ArcInfo 工作空间	esriFileSystemWorkspace
CadWorkspaceFactory	打开 CAD 文件	esriFileSystemWorkspace
OLEDWorkspaceFactory	使用 OLEDB 打开数据库的工作空间	esriRemoteDatakaseWorkspace
SdeWorkspaceFactory	打开 SDE 数据库	esriRemoteDatakaseWorkspace
TinWorkspaceFactory	打开 TIN 数据文件的工作空间	esriFileSystemWorkspace
TextFileWorkspaceFactory	打开文本文件	esriFileSymtemWorkspace

打开工作空间后，需要进行处理的就是数据集对象，数据集 Dataset 是一个代表了工作空间 Workspace 中数据集合的抽象类，它是一个高级别的数据容器，所有放在工作空间的对象都是一种数据集对象，在一个地理数据库中每个 Dataset 都有唯一的名字；数据集中的数据可以是一行记录、一个对象、一个要素、一个关系，乃至另一个数据集。所有的数据集都实现了 IDataset 接口，它定义了所有数据集都有的属性和方法，FullName 属性可以返回与这个数据集相联系的名称对象；PropertySet 属性返回数据集的属性集对象；Subsets 属性可以返回这个数据集中包含的子集，如果 Dataset 是一个要素数据集的话，该属性返回它包含的要素类等对象。IDataset 接口的 CanCopy、CanDelete、CanRename 方法用于设置数据集是否能够被拷贝、删除和重命名；当然，IDataset 也定义了 Copy、Delete 和 Rename 方法供用户来操作数据集。

在工作空间中对一个要素类进行操作时，需要注意这个要素类放在什么地方，是直接放在工作空间中（称为 standalone FeatureClass），还是放在一个要素数据集中。当使用 IWorkspace 接口的 Datasets 属性来遍历一个工作空间内的数据集时，返回的只是直接放在工作空间的数据集，而保存在一个要素数据集中的要素类则不会被遍历。使用 IFeatureWorkspace 接口的 OpenFeatureClass 方法可以打开工作空间中的任何一个要素类，无论它是直接存放在工作空间还是存放在工作空间的一个要素数据集中。

（10）编译通过后，点击菜单栏【调试】→【开始执行】，点击“ConnectSDE”，弹出连接数据库窗体，设置相关参数如图 5-13 所示。

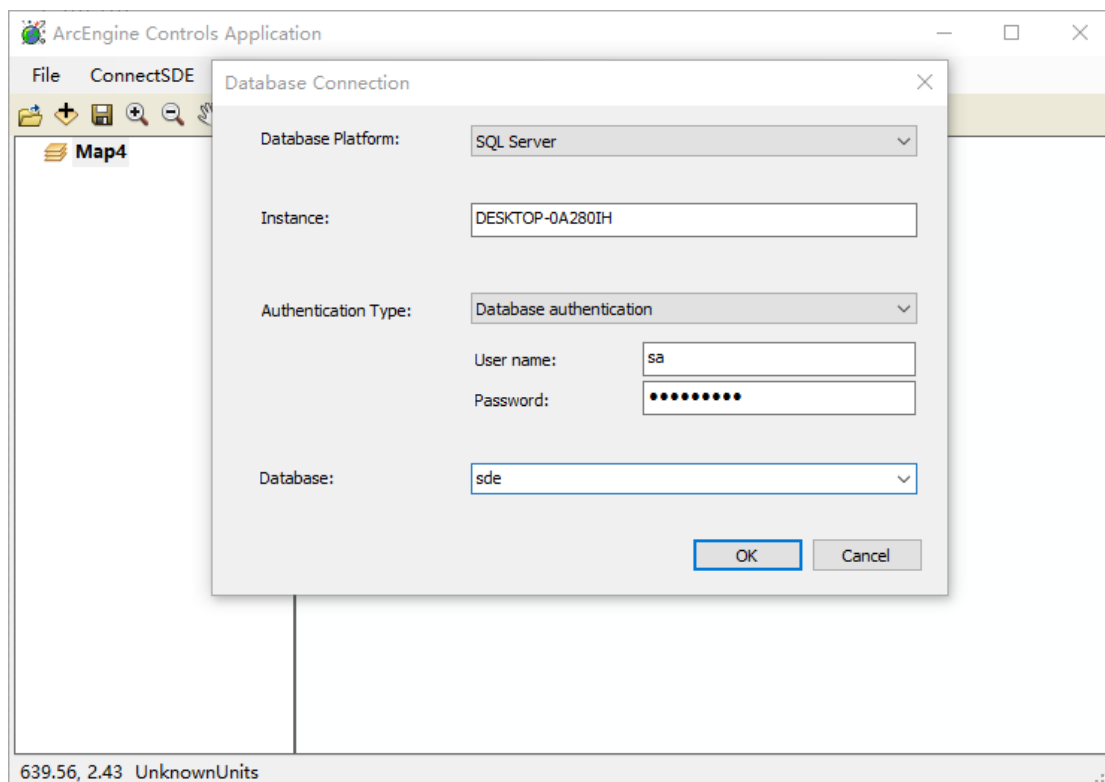


图 5-13 连接 SQL SERVER 数据库

(11) 双击“OK”按钮，SQL SERVER 地理数据库中的要素类、要素数据集中的要素类都被加载到了地图视图中，如图 5-14 所示。

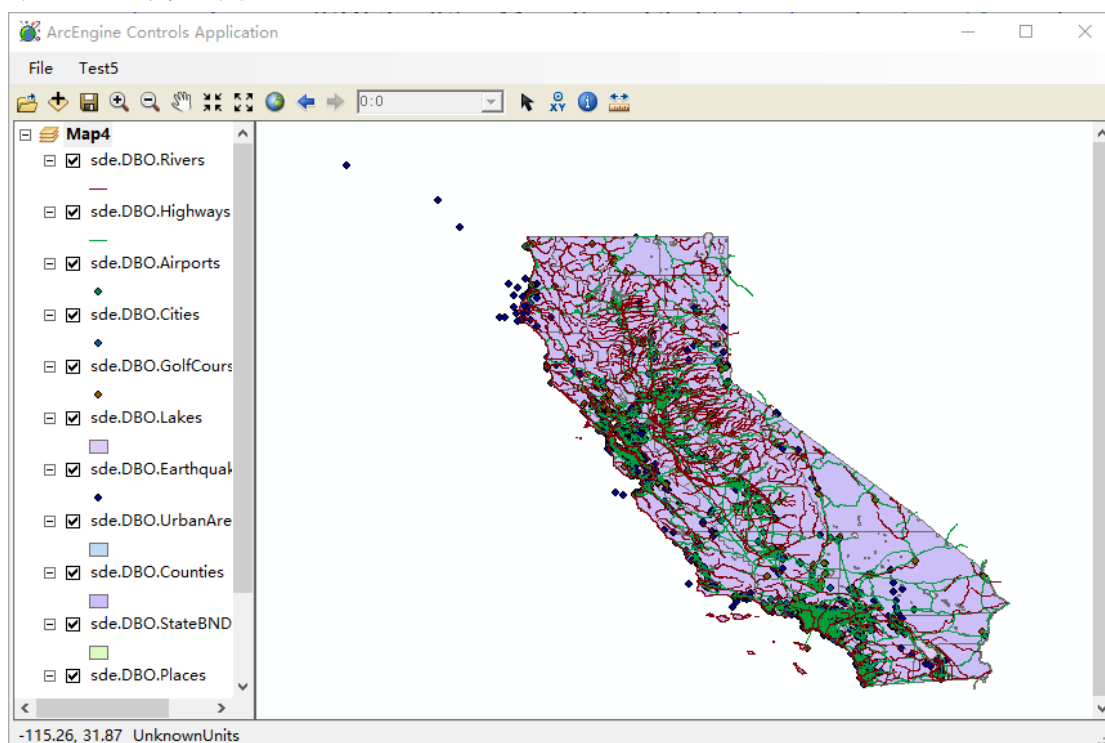


图 5-14 数据库中要素类加载运行效果

3 实验目的

- (1) 熟练使用 MapControl/TOCCControl/ToolbarControl 控件开发应用程序;
- (2) 熟练开发按钮命令 (Command) 和交互工具 (Tool);
- (3) 熟悉在 Access 个人地理数据库中新建要素类和新建要素。

4 实验内容

(1) 开发菜单按钮命令 (Command) 功能实现新建一个*.mdb 个人地理数据库, 在该个人地理数据库中新建点、线、多边形要素类, 并将要素类加载到当前地图中显示;

(2) 开发鼠标交互工具 (Tool) 功能向要素类中添加要素。

5 实验数据

见安装目录:

...\DeveloperKit10.6\Samples\data\ California

6 实验步骤

6.1 新建地理数据库及要素类

(1) 新建按钮命令类, 实现新建要素类的功能。解决方案资源管理器下选择 “MapControlApplication ” 项目, 右键选择【添加】→【新建项】, 如图 5-15 所示。

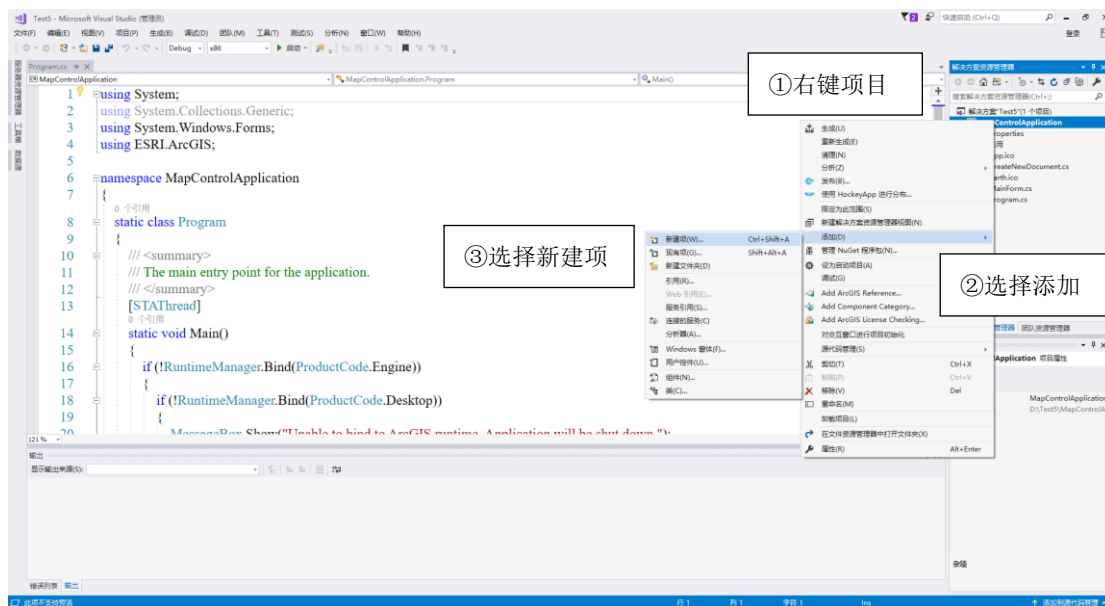


图 5-15 添加新建项

新建一个按钮类“CmdCreateFeatureClass”，如图 5-16 所示。

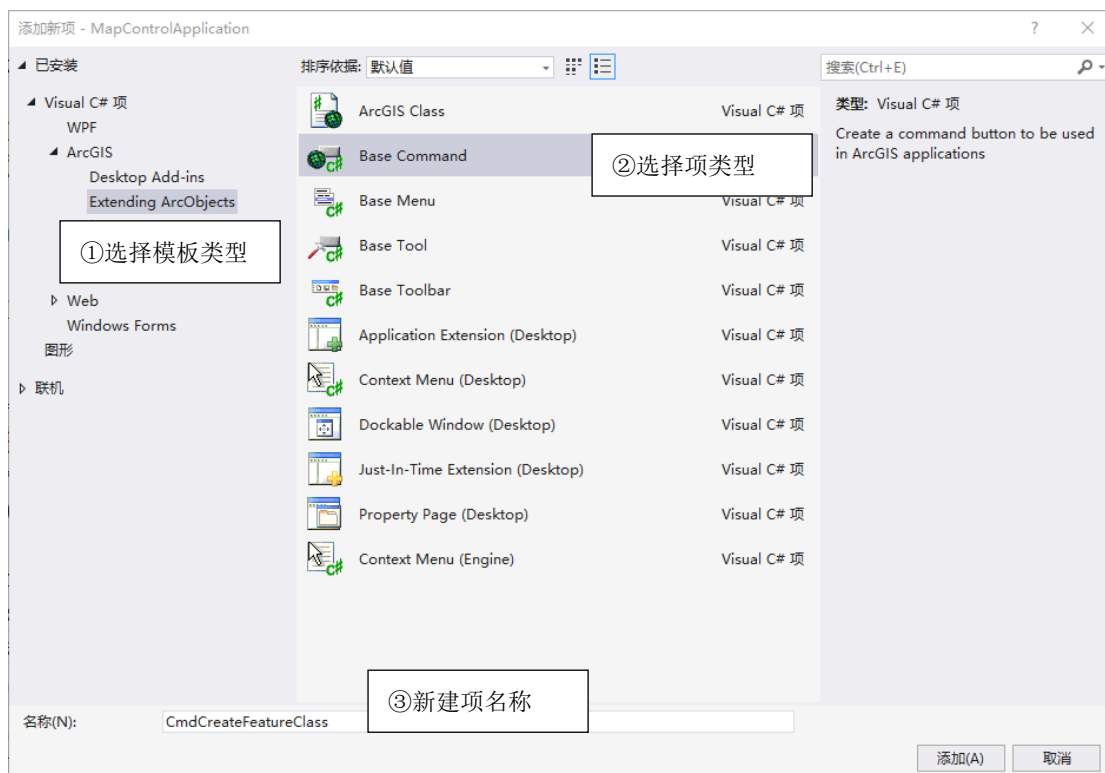


图 5-16 新建按钮类

(2) 在 CmdCreateFeatureClass.cs 中添加按钮命令的 OnClick 事件代码，实现新建 Access 个人地理数据库和要素类。右键项目“MapControlApplication”下面的【引用】→【添加引用】，勾选需要添加的引用，点击“确定”，如图 5-17 所示。在 CmdCreateFeatureClass.cs 文件开头添加引用：

```
using ESRI.ArcGIS.DataSourcesGDB;
using ESRI.ArcGIS.Geodatabase;
using ESRI.ArcGIS.Geometry;
```

```
using ESRI.ArcGIS.Carto;
using System.Windows.Forms;
```

CmdCreateFeatureClass.cs (节选)	功能：新建 Access 个人地理数据库和要素类
-------------------------------	--------------------------

```
public override void OnClick()
{
    // TODO: Add CmdCreateFeatureClass.OnClick implementation
    //新建个人地理数据库并自定义名称
    SaveFileDialog dbfiledlg = new SaveFileDialog();
    dbfiledlg.Filter = "空间数据库文件 (*.mdb) |*.mdb";
    dbfiledlg.RestoreDirectory = true;
    if (dbfiledlg.ShowDialog() == DialogResult.OK)
    {
        string localFilePath = dbfiledlg.FileName.ToString();
        if (System.IO.File.Exists(localFilePath))
            System.IO.File.Delete(localFilePath);
        //获得不带文件名的文件路径
        string FilePath = localFilePath.Substring(0,
                                                    localFilePath.LastIndexOf("\\"));
        //获取文件名，不带路径
        string fileNameExt = localFilePath.Substring(
                                                    localFilePath.LastIndexOf("\\") + 1);
        string fileName = fileNameExt.Substring(0,
                                                    fileNameExt.LastIndexOf("."));
        IWorkspaceFactory ipWSFactory = new AccessWorkspaceFactory();
        IWorkspaceName ipWsName = ipWSFactory.Create(
                                                    FilePath, fileName, null, 0);
        IWorkspace ipWorkspace = ipWSFactory.OpenFromFile(
                                                    localFilePath, 0);

        string datasetName = null;
        string featureclsName = null;
        esriGeometryType type = esriGeometryType.esriGeometryNull;
        //打开一个要素类型选择对话框
        FrmCreateFeatureClass dlg = new FrmCreateFeatureClass();
        if (dlg.ShowDialog() == DialogResult.OK)
        {
            type = dlg.type; //要素类几何类型，点、线、面
            datasetName = dlg.dataset; //要素数据集名称
            featureclsName = dlg.featureCls; //要素类名称
            ISpatialReference ipSr = //当前打开地图的空间索引
                                    m_hookHelper.FocusMap.SpatialReference;
            IFeatureClass ipFeatCls = CreateFeatureClass(ipWorkspace as
                                                         IFeatureWorkspace, datasetName, featureclsName, ipSr, type);
            IFeatureLayer ipFeatLyr = new FeatureLayerClass();
            // 新建的图层的默认名
        }
    }
}
```

```
ipFeatLyr. Name = featureclsName;
ipFeatLyr. FeatureClass = ipFeatCls;
m_hookHelper. FocusMap. AddLayer(ipFeatLyr);
}
}
}
```

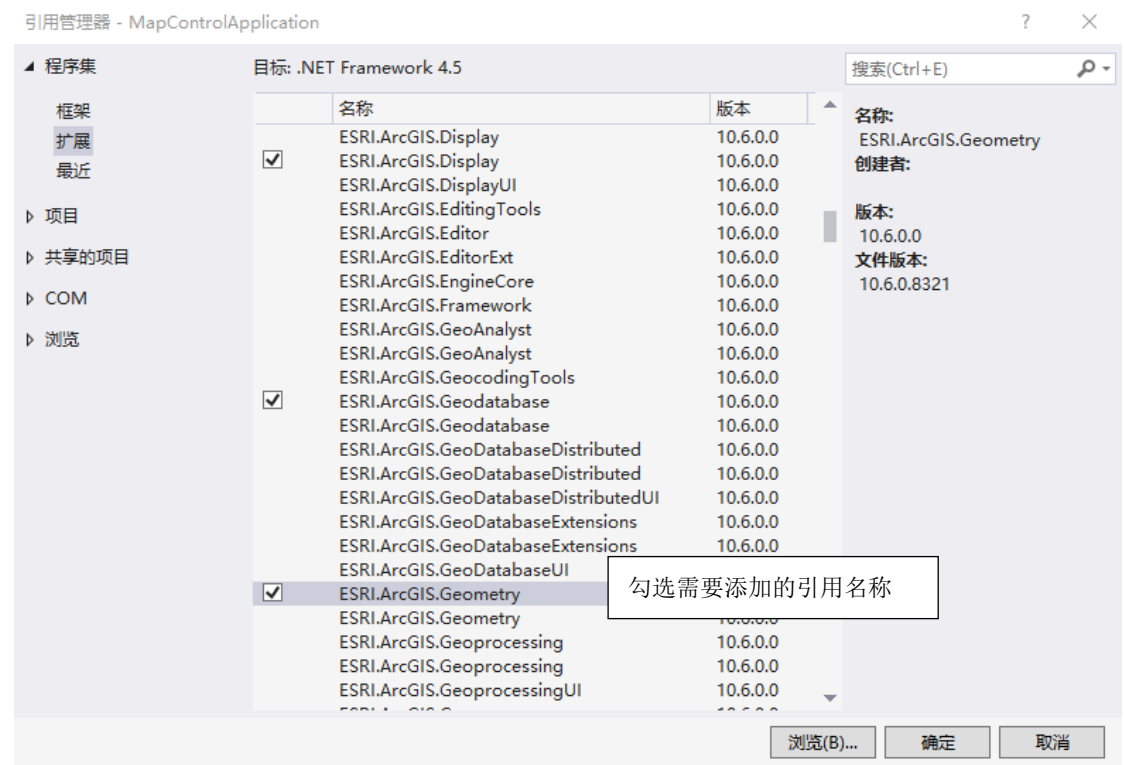


图 5-17 添加引用

数据集 Dataset 对象分为两大类（如图 5-18 所示）：（a）表 Table 是行 Row 的容器，对象类 ObjectClass 是对象 Object 的容器，要素类 FeatureClass 是要素 Feature 的容器，属性关系类 AttributedRelationshipClass 是属性关系 AttributedRelationship 的容器；（b）地理数据集 GeoDataset，代表了拥有空间属性的数据集。

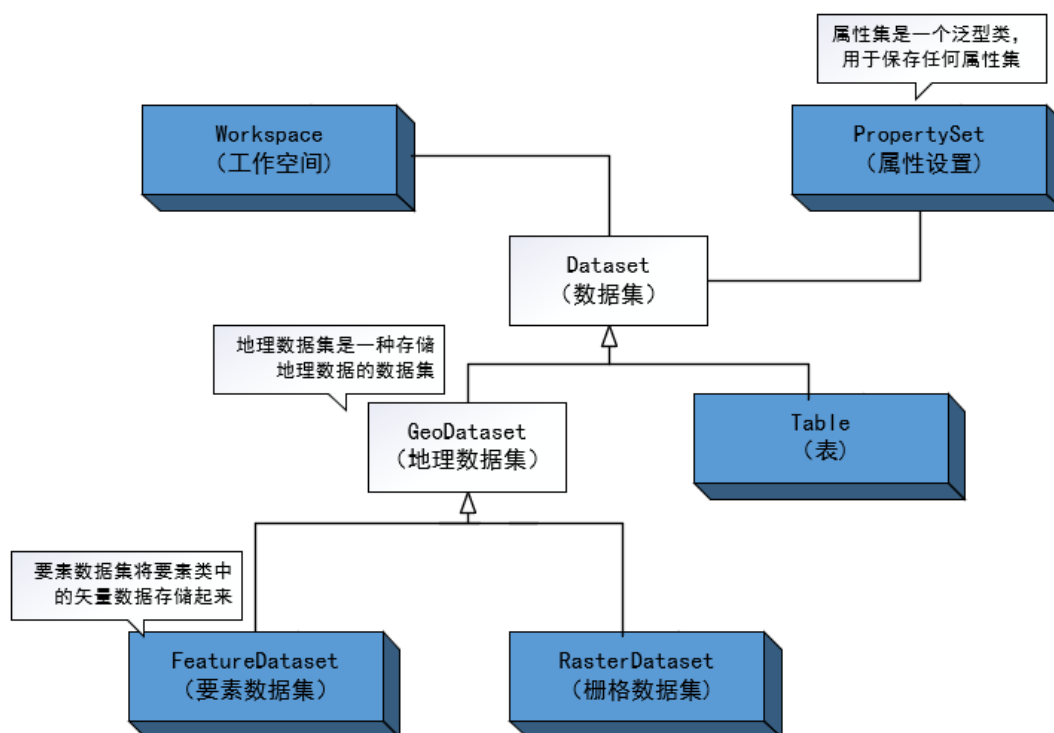


图 5-18 对象数据集

地理数据集 GeoDataset 一般有三种：(a) 要素数据集 FeatureDataset 是具有相同空间参考的要素类的容器；(b) 栅格数据集 RasterDataset 可以是简单数据集，也可以是具有不同光谱值和分类值的多波段复合数据集；(c) TIN 数据集 TinDataset 包含一组三角形，这些三角形通过精确地覆盖某个区域来表示某种表面。

要素类既可以放在工作空间中也可以放在一个要素数据集中。简单要素类放在要素数据集内/外都可，而要素类必须位于同一要素数据集中有三种情况：

- (a) 通过几何网络和平面拓扑设定要素类拓扑相关，而拓扑要素类必须放在要素数据集中，以保证具有共同的坐标系统；
- (b) 要素类强制设定同一个空间参考；
- (c) 专题相关的要素类。

(4) 在 CmdCreateFeatureClass.cs 中添加私有成员函数 CreateFeatureClass(), 创建要素类。

CmdCreateFeatureClass.cs (节选)	功能：创建点、线、面要素类
<pre> //创建点、线、面要素类 private IFeatureClass CreateFeatureClass(IFeatureWorkspace ipWorkspace, string dsName, string fcName, ISpatialReference ipSr, esriGeometryType type) { //设置字段组 IFieldsEdit ipFields = (IFieldsEdit)new Fields(); ipFields.FieldCount_2 = 2; //设置字段数 IFieldEdit ipField = (IFieldEdit)new Field(); ipField.Name_2 = "ObjectID"; ipField.AliasName_2 = "FID"; </pre>	

```

ipField. Type_2 = esriFieldType.esriFieldTypeOID;
ipFields. set_Field(0, ipField);
// 设置几何形状字段
IGeometryDefEdit ipGeoDef = (IGeometryDefEdit)new GeometryDef();
ipGeoDef. SpatialReference_2 = ipSr; //设置空间索引
// 设置要素几何类型
ipGeoDef. GeometryType_2 = type;
IFieldEdit ipField3 = (IFieldEdit)new Field();
ipField3. Name_2 = "Shape";
ipField3. AliasName_2 = "shape";
ipField3. Type_2 = esriFieldType.esriFieldTypeGeometry;
ipField3. GeometryDef_2 = ipGeoDef;
ipFields. set_Field(1, ipField3);
// 在工作空间下创建要素数据集，要素数据集名称dsName
IFeatureDataset ipFeatDs = ipWorkspace. CreateFeatureDataset(dsName,
                                                                ipSr);

// 在要素数据集下创建要素类，要素类名称fcName
IFeatureClass ipFeatCls = ipFeatDs. CreateFeatureClass(fcName, ipFields,
                                                         null, null, esriFeatureType.esriFTSimple, "Shape", "");
return ipFeatCls;
}

```

(5) 新建一个 Windows 窗体用来选择创建要素类几何类型和设置要素数据集名称、要素类名称，窗体类名称为“FrmCreateFeatureClass”，如图 5-19 所示。

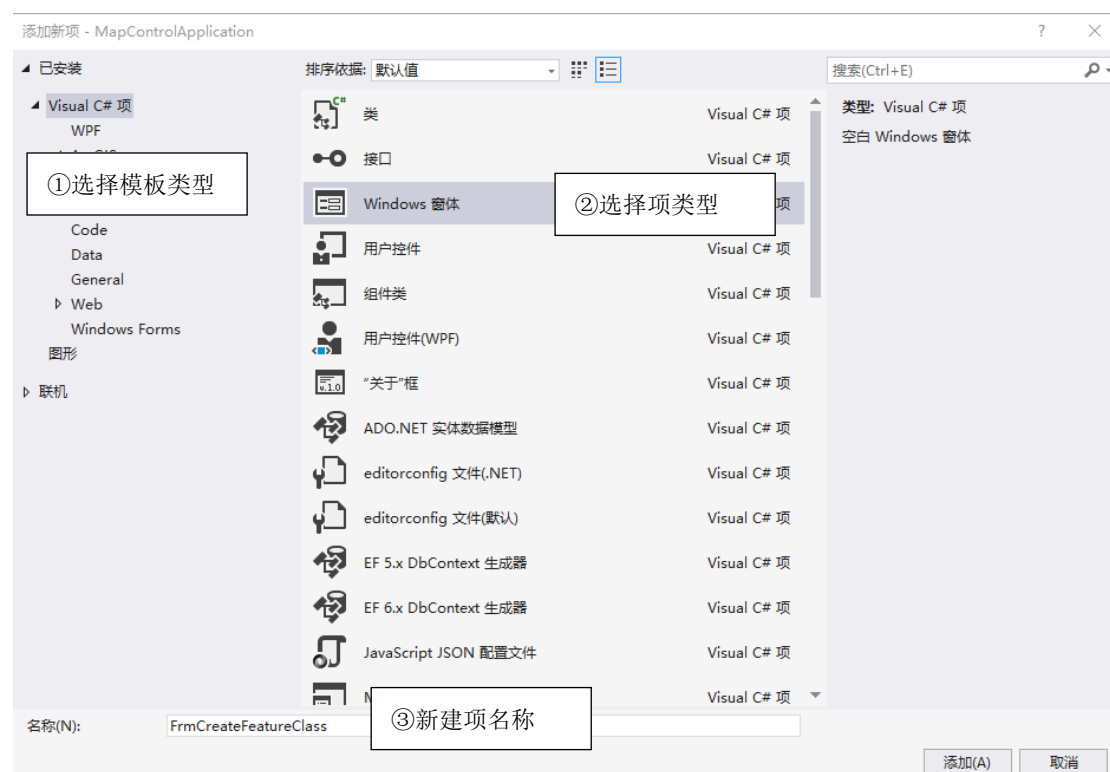


图 5-19 新建创建要素窗体

添加控件并修改各控件的 Name 属性值，如图 5-20 所示。

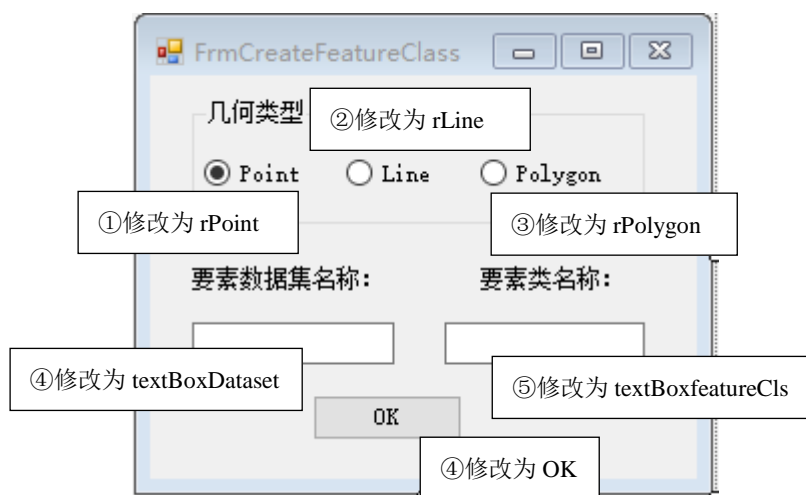


图 5-20 修改各控件的 Name 属性值

(6) 为类“FrmCreateFeatureClass”定义只读属性 type、dataset 和 featureCls。然后在 FrmCreateFeatureClass.cs 文件开头添加引用：

`using ESRI. ArcGIS. Geometry;`

将 Geometry 的嵌入互操作类型改为 False。

FrmCreateFeatureClass.cs（节选）	功能：要素类几何类型、要素数据集名、要素类名属性
<pre> //获得要素类几何类型 public esriGeometryType type { get { if (rPoint. Checked) return esriGeometryType. esriGeometryPoint; else if (rLine. Checked) return esriGeometryType. esriGeometryPolyline; else if (rPolygon. Checked) return esriGeometryType. esriGeometryPolygon; else return esriGeometryType. esriGeometryNull; } } //获得要素数据集名 public string dataset { get { return textBoxDataset. Text; } } //获得要素类名 public string featureCls </pre>	

```
{
    get
    {
        return textBoxfeatureCls. Text;
    }
}
```

(7) 双击 OK 按钮，添加代码如下所示。

FrmCreateFeatureClass.cs (节选)	功能：执行选择结果和关闭窗体
<pre>//执行选择结果并关闭窗体 private void OK_Click(object sender, EventArgs e) { DialogResult = DialogResult. OK; this. Close(); }</pre>	

(8) 添加菜单项 NewFeatureClass，如图 5-21 所示。双击该菜单项，添加 Click 事件响应函数如下所示。

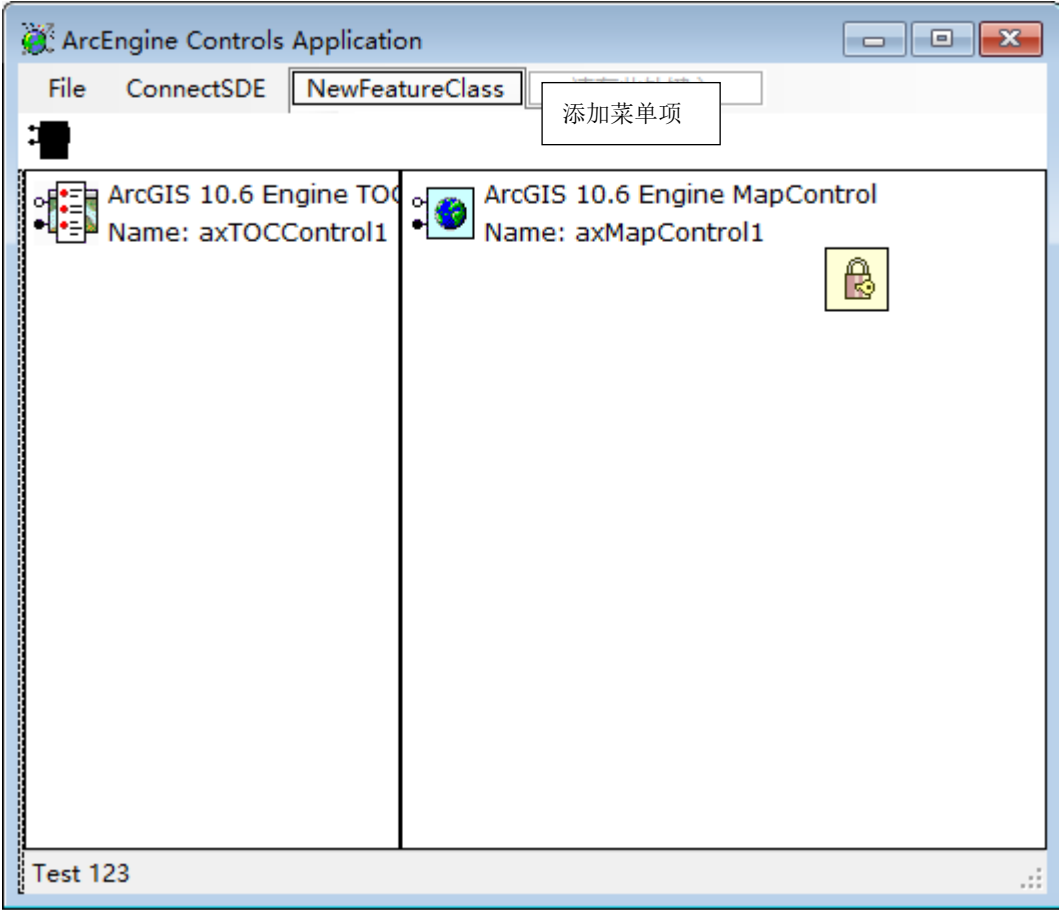


图 5-21 新建菜单项 NewFeatureClass

MainForm.cs (节选)	功能：添加 Click 事件响应函数创建要素类
<pre>private void newFeatureClassToolStripMenuItem_Click(object sender, EventArgs e) { ICommand command = new CmdCreateFeatureClass(); command.OnCreate(m_mapControl.Object); }</pre>	

```
command.OnClick();  
}
```

(9) 点击菜单栏【生成】→【重新生成解决方案】编译通过后，然后点击菜单栏【调试】→【开始执行】，加载实验数据，点击“NewFeatureClass”，输入要创建的个人地理数据库名称，如图 5-22 所示。

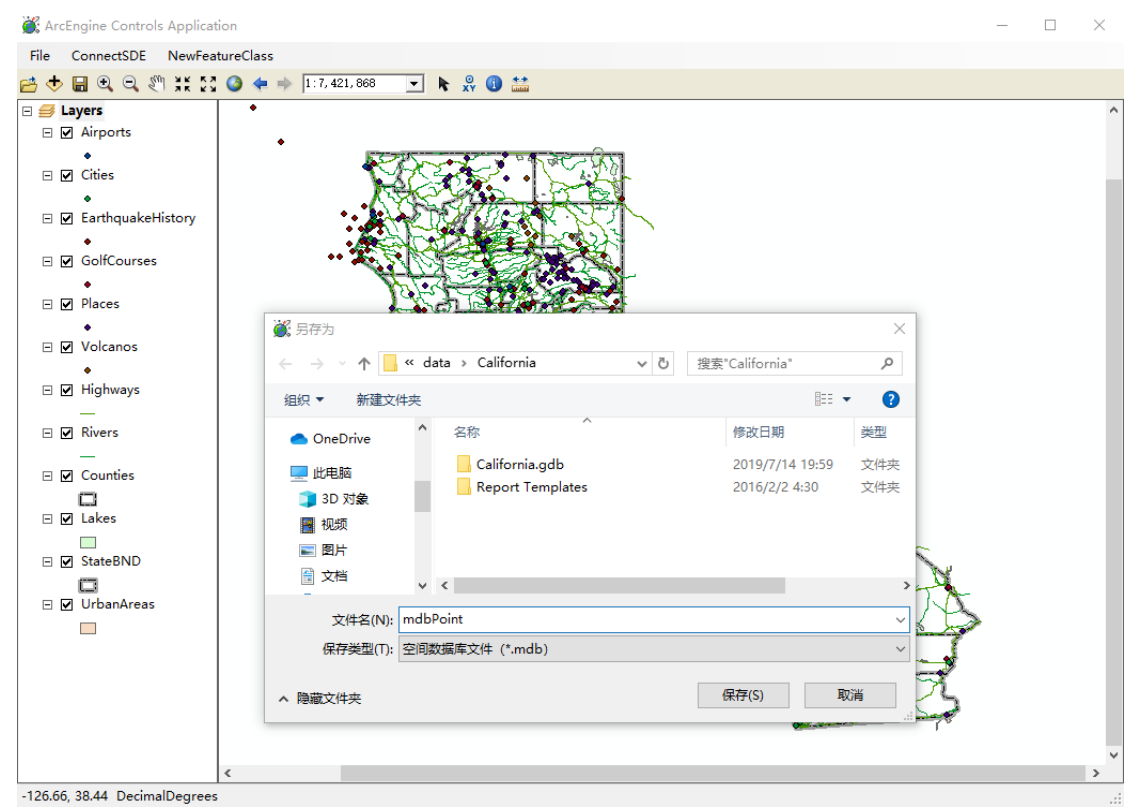


图 5-22 创建个人地理数据库

选择需要创建的要素类几何类型、要素数据集和要素类名称，如图 5-23 所示。

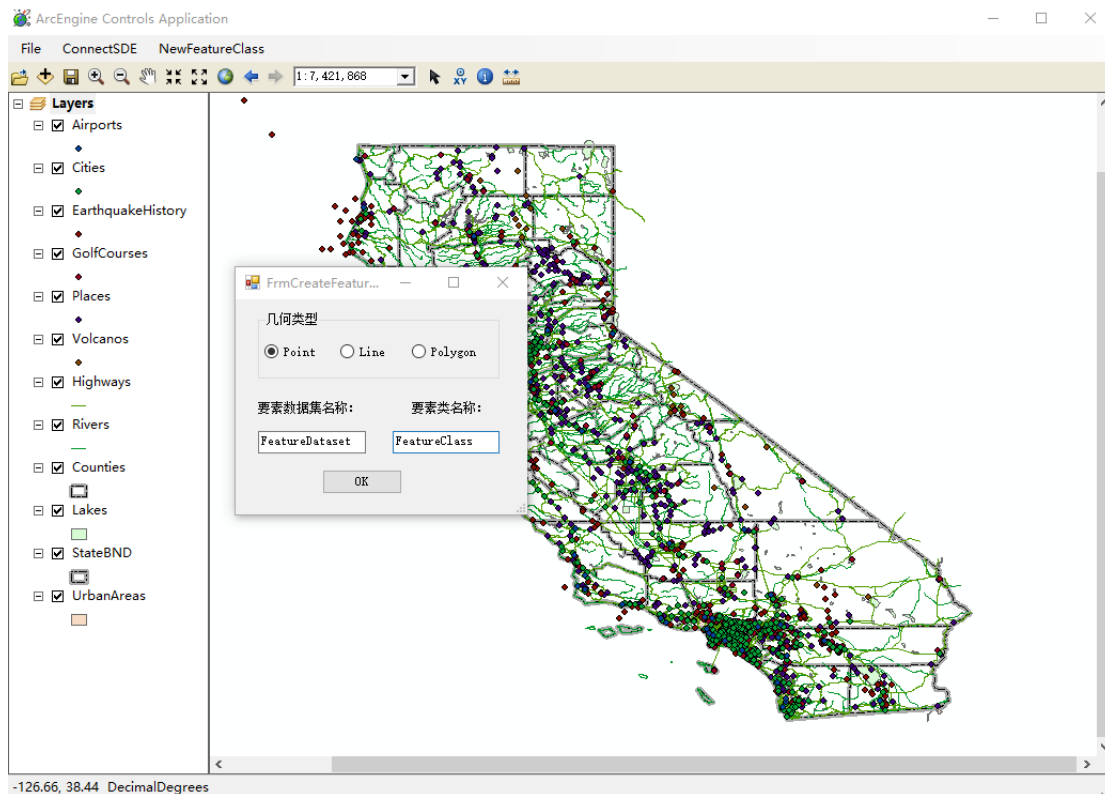


图 5-23 设置创建要素类参数

依次创建点、线、面要素的 Access 个人地理数据库，如图 5-24 所示。

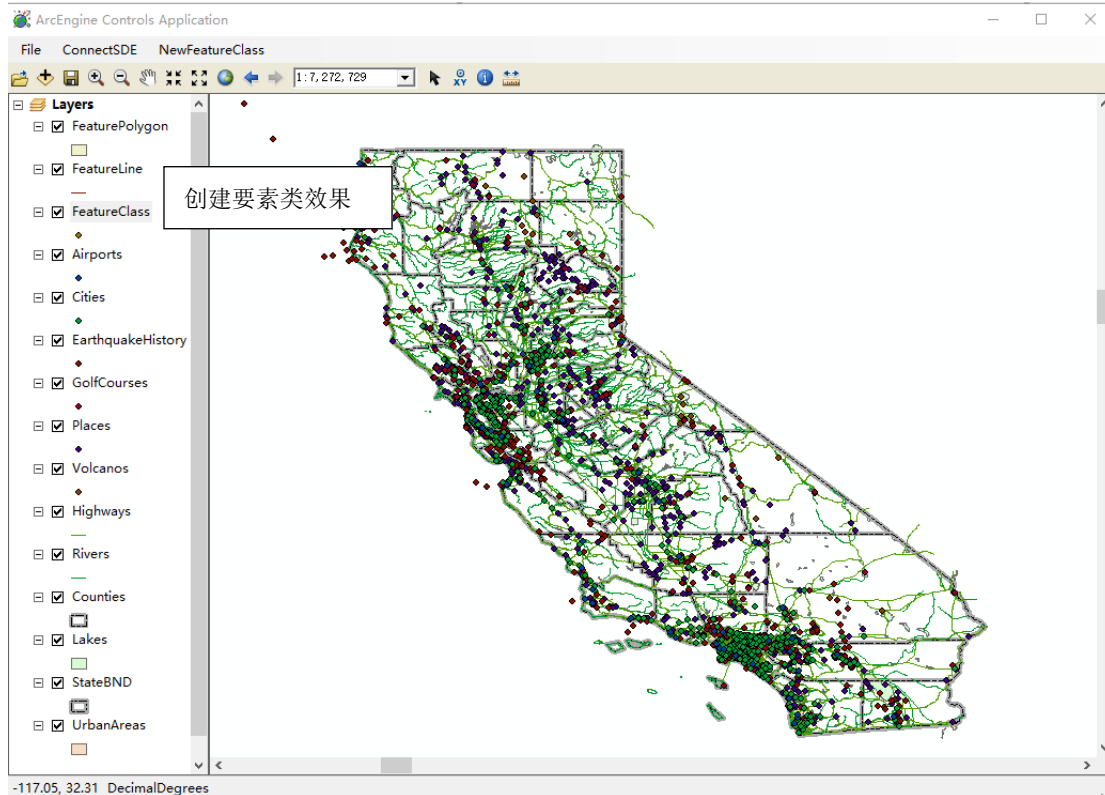


图 5-24 新建个人地理数据库及点、线、面要素类运行效果

6.2 新建点、线、面要素

(1) 解决方案资源管理器选中“MapControlApplication”项目，右键选择【添加】→【新建项】，新建一个工具类“ToolCreateFeature”，如图 5-25 所示。

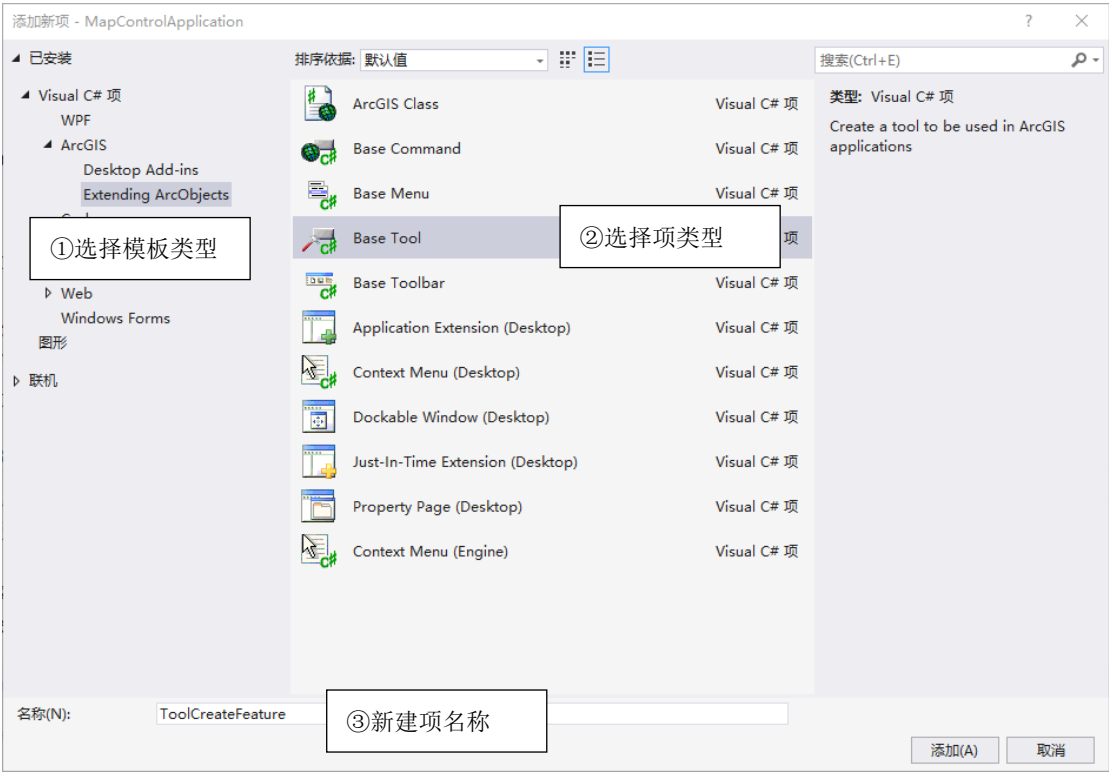


图 5-25 新建工具类

(2) 在 ToolCreateFeature.cs 中定义一个局部变量 m_selectedLyr，
然后添加鼠标按下事件响应函数 OnMouseDown()，并添加引用：

```
private IFeatureLayer m_selectedLyr = null;
using ESRI.ArcGIS.Display;
using ESRI.ArcGIS.Geometry;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.Geodatabase;
```

ToolCreateFeature.cs（节选）	功能：实现鼠标按下创建要素
<pre>//利用鼠标按下创建要素 public override void OnMouseDown(int Button, int Shift, int X, int Y) { // TODO: Add ToolCreateFeature.OnMouseDown implementation IRubberBand ipRubber = null; IGeometry ipGeo = null; // 根据需要创建的要素几何类型创建橡皮筋对象 switch (m_selectedLyr.FeatureClass.ShapeType) { case esriGeometryType.esriGeometryPoint: ipRubber = new RubberPoint(); break;</pre>	

```

        case esriGeometryType.esriGeometryPolyline:
            ipRubber = new RubberLine();
            break;
        case esriGeometryType.esriGeometryPolygon:
            ipRubber = new RubberPolygon();
            break;
        default:
            return;
    }
    ipGeo = ipRubber.TrackNew(
        m_hookHelper.ActiveView.ScreenDisplay, null);
    //如果选中图层不为空，则将创建的要素保存至该图层
    if (m_selectedLyr != null)
    {
        IFeature feature = m_selectedLyr.FeatureClass.CreateFeature();
        feature.Shape = ipGeo;
        feature.Store();
        m_hookHelper.ActiveView.Refresh();
    }
}

```

地理数据库 Geodatabase 中的要素 Feature 的主要特点有：(a) 要素具有形状 Geometry；(b) 要素具有空间参考；(c) 要素具有属性；(d) 要素可具有子类；(e) 要素可具有关系；(f) 要素属性可以被约束；(g) 可通过规则来验证要素；(h) 要素可以具有拓扑；(i) 要素可以具有复杂行为，等。

在 ToolCreateFeature.cs 中添加单击事件 OnClick 响应函数代码。

ToolCreateFeature.cs (节选)	功能：选择图层
<pre> public override void OnClick() { // TODO: Add ToolCreateFeature.OnClick implementation //调用FrmSelectLayer选择图层 FrmSelectLayer dlg = new FrmSelectLayer(m_hookHelper); if (dlg.ShowDialog() == DialogResult.OK) { m_selectedLyr = dlg.lyr as IFeatureLayer; } } </pre>	

(3) 新建一个 Windows 窗体来选择图层以供交互添加要素，窗体类名称为“FrmSelectLayer”，如图 5-26 所示。

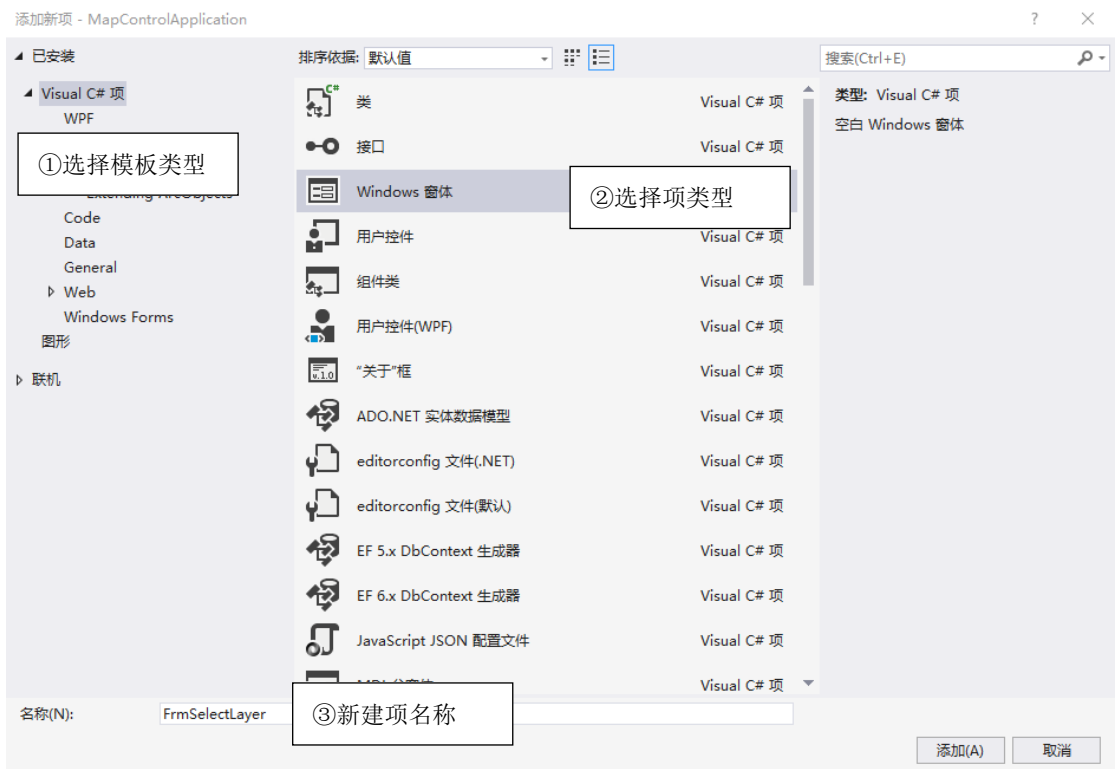


图 5-26 新建图层选择窗体

为窗体类“FrmSelectLayer”添加窗体控件并修改各控件的 Name 属性值，如图 5-27 所示。

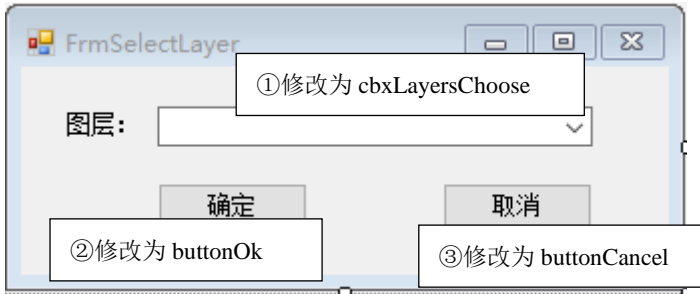


图 5-27 修改各控件的 Name 属性值

(4) 为类“FrmSelectLayer”添加私有成员变量“ILayer m_layer”和只读属性“ILayer lyr”，用来保存选择到的要素图层。在 FrmSelectLayer.cs 中添加引用：

`using ESRI. ArcGIS. Carto;`

FrmSelectLayer.cs（节选）	功能：保存选择到的要素图层
<pre>private ILayer m_layer = null; public ILayer lyr { get { return m_layer; } }</pre>	

为类“FrmSelectLayer”添加“IHookHelper”接口类型的私有成员变量

m_hookhelper，在类的构造函数中将该引用变量指向具体的对象，并在 FrmSelectLayer.cs 中添加引用：

`using ESRI. ArcGIS. Controls;`

FrmSelectLayer.cs（节选）	功能：定义 IHookHelper 接口获取主应用程序资源
<pre>private IHookHelper m_hookhelper = null; public FrmSelectLayer(IHookHelper hook) { m_hookhelper = hook; InitializeComponent(); }</pre>	

（5）点击窗体 FrmSelectLayer 空白处添加 FrmSelectLayer_Load()窗体加载事件响应函数，然后添加 CbxLayersAddItems()私有成员函数，获取当前所有图层。

FrmSelectLayer.cs（节选）	功能：将当前地图所有图层添加到组合框
<pre>private void FrmSelectLayer_Load(object sender, EventArgs e) { CbxLayersAddItems(); } //获取当前地图所有要素图层，并添加到组合框 private void CbxLayersAddItems() { IEnumLayer layers = m_hookhelper.FocusMap.Layers; layers.Reset(); ILayer layer = layers.Next(); while (layer != null) { if (layer is IFeatureLayer) { cbxLayersChoose.Items.Add(layer.Name); } layer = layers.Next(); } }</pre>	

（6）双击“确定”和“取消”按钮，添加按钮单击事件响应函数；双击 ComboBox 图层选择控件，添加代码。

FrmSelectLayer.cs（节选）	功能：获得选中的要素图层并进行创建要素
<pre>//获得创建要素结果 private void buttonOk_Click(object sender, EventArgs e) { DialogResult = DialogResult.OK; this.Close(); } //关闭窗体 private void buttonCancel_Click(object sender, EventArgs e) { </pre>	

```

        m_layer = null;
        this.Close();
    }
    //获得选择的图层
    private void cbxLayersChoose_SelectedIndexChanged(object sender, EventArgs e)
    {
        IEnumLayer layers = m_hookhelper.FocusMap.Layers;
        layers.Reset();
        ILayer layer = layers.Next();
        while (layer != null)
        {
            if (layer is IFeatureLayer)
            {
                if (layer.Name == cbxLayersChoose.Text)
                {
                    m_layer = layer;
                    break;
                }
            }
            layer = layers.Next();
        }
    }
}

```

(7) 添加菜单项 NewFeature，双击该菜单项，添加 Click 事件响应函数如下所示。

MainForm.cs (节选)	功能：添加菜单单击事件响应函数
<pre> //调用Tool工具创建要素类 private void newFeatureToolStripMenuItem_Click(object sender, EventArgs e) { ICommand command = new ToolCreateFeature(); command.OnCreate(m_mapControl.Object); axMapControl1.CurrentTool = (ITool)command; } </pre>	

(8) 点击菜单栏【生成】→【重新生成解决方案】编译通过后，然后点击菜单栏【调试】→【开始执行】，加载实验数据，点击“NewFeature”弹出 FrmSelectLayer 窗体，选择需要创建要素的图层，如图 5-28 所示。

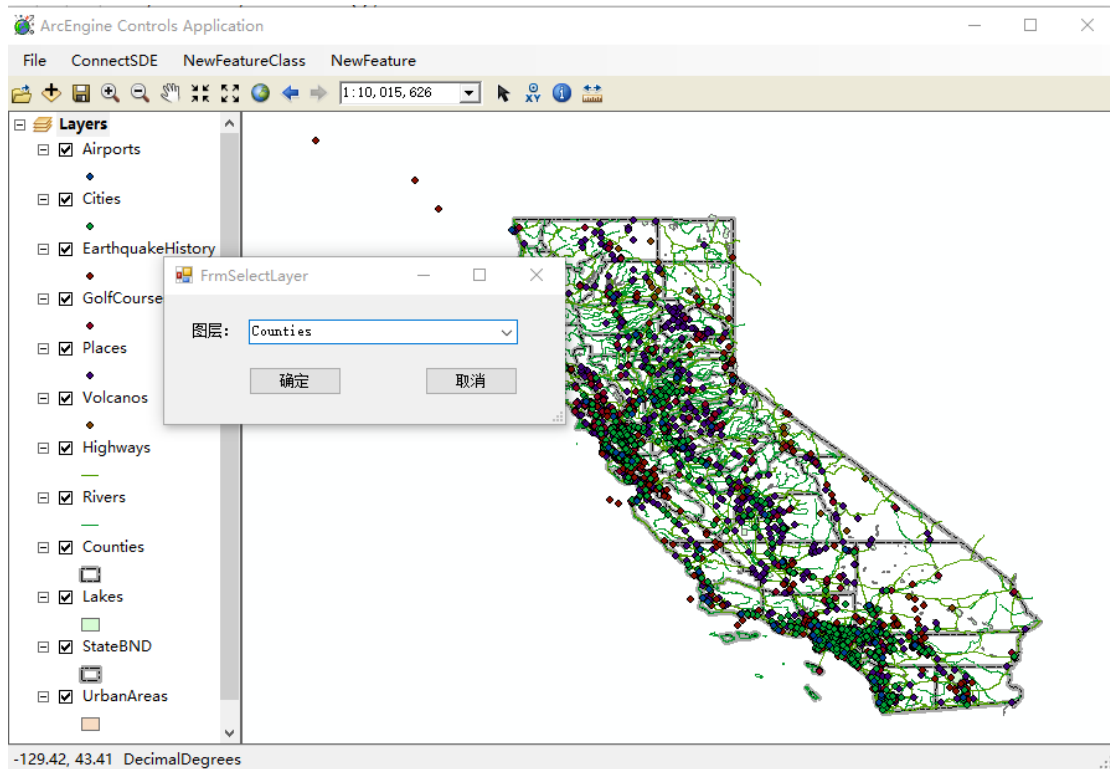


图 5-28 选择创建要素的图层

选择相应的要素图层，依次创建点、线、面要素，如图 5-29 所示。

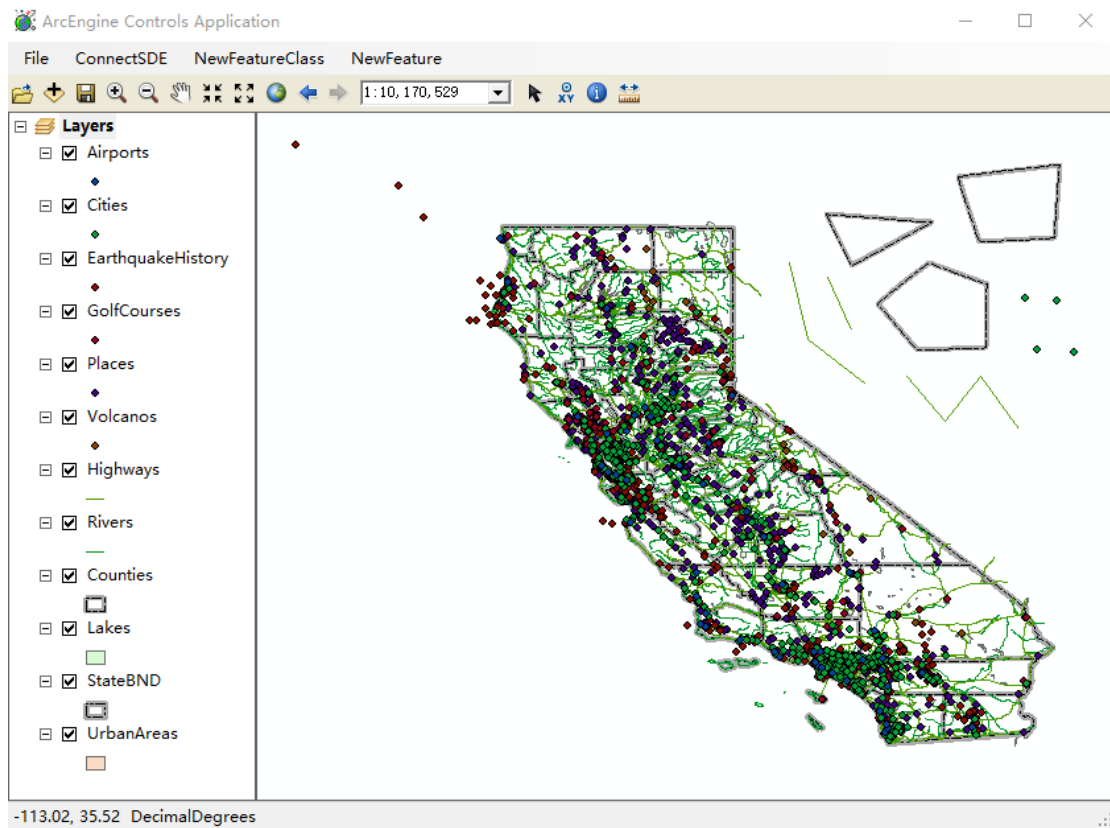


图 5-29 创建点、线、面要素