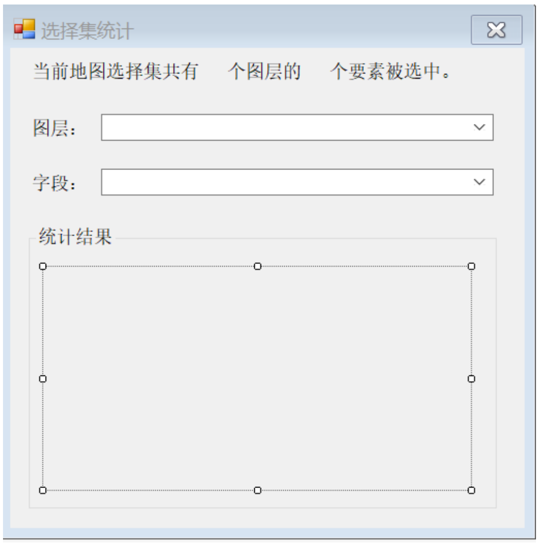
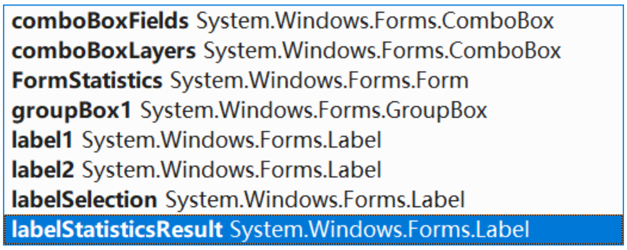
统计选择集





comboBoxFields System.Windows.Forms.ComboBox
comboBoxLayers System.Windows.Forms.ComboBox
FormStatistics System.Windows.Forms.Form
groupBox1 System.Windows.Forms.GroupBox
label1 System.Windows.Forms.Label
label2 System.Windows.Forms.Label
labelSelection System.Windows.Forms.Label
labelStatisticsResult System.Windows.Forms.Label

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Collections;

using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.Geodatabase;
using ESRI.ArcGIS.esriSystem;

namespace GIS2022
{
    public partial class FormStatistics : Form
    {
        private IMap currentMap;    //当前MapControl控件中的Map对象
        //设置哈希表类型的类变量来存储图层名称和所对应矢量图层的IFeatureLayer接口对象
        private Hashtable layersHashtable;
        //设置类变量存储当前的矢量图层对象
        private IFeatureLayer currentFeatureLayer = null;

        /// <summary>
        /// 获得当前MapControl控件中的Map对象。
        /// </summary>
        public IMap CurrentMap
        {
            set
            {
                currentMap = value;
            }
        }

        public FormStatistics()
        {
            InitializeComponent();
            //新建哈希表对象
            layersHashtable = new Hashtable();
        }

        private void FormStatistics_Load(object sender, EventArgs e)
        {
            IFeatureLayer featureLayer; //设置临时变量存储矢量图层对象
            string layerName;   //设置临时变量存储图层的名称
            int layersCount = 0;    //设置临时变量存储具有选择要素的图层总个数
            int allSelectedFeatures = 0;    //设置临时变量存储被选择要素的总数

            //清空哈希表
            layersHashtable.Clear();

            //对Map中的每个图层进行判断和加载名称
            for (int i = 0; i < currentMap.LayerCount; i++)
            {
                //如果该图层为图层组类型，则分别对所包含的每个图层进行操作
                if (currentMap.get_Layer(i) is GroupLayer)
                {
                    //使用ICompositeLayer接口进行遍历操作
                    ICompositeLayer compositeLayer = currentMap.get_Layer(i) as ICompositeLayer;
                    for (int j = 0; j < compositeLayer.Count; j++)
                    {
                        //得到图层的名称
                        layerName = compositeLayer.get_Layer(j).Name;
                        //得到IFeatureLayer接口的矢量图层对象
                        featureLayer = (IFeatureLayer)compositeLayer.get_Layer(j);
                        //通过接口转换获得当前图层选择集中被选择要素的总数，如果大于0则进行统计
                        if (((IFeatureSelection)featureLayer).SelectionSet.Count > 0)
                        {
                            //在comboBoxLayers中添加该图层的名称
                            comboBoxLayers.Items.Add(layerName);
                            //在哈希表中添加一项，包括图层名称和图层对象
                            layersHashtable.Add(layerName, featureLayer);
                            //具有选择要素的图层总个数加1
                            layersCount += 1;
                            //被选择要素的总数加上被选择要素的数量
                            allSelectedFeatures += 
((IFeatureSelection)featureLayer).SelectionSet.Count;
```

```
                        }
                    }
                }
                //如果图层不是图层组类型，则直接执行操作
                else
                {
                    layerName = currentMap.get_Layer(i).Name;
                    featureLayer = (IFeatureLayer)currentMap.get_Layer(i);
                    if (((IFeatureSelection)featureLayer).SelectionSet.Count > 0)
                    {
                        comboBoxLayers.Items.Add(layerName);
                        layersHashtable.Add(layerName, featureLayer);
                        layersCount += 1;
                        allSelectedFeatures += ((IFeatureSelection)featureLayer).SelectionSet.Count;
                    }
                }
            }

            //将当前的选择情况显示在窗体上
            labelSelection.Text = "当前地图选择集共有 " + layersCount + " 个图层的 " +
allSelectedFeatures + " 个要素被选中。";

            //显示第一个可以选择的图层
            if (comboBoxLayers.Items.Count > 0)
                comboBoxLayers.SelectedIndex = 0;
        }

        //当所选择图层发生变化时触发事件，执行本函数
        private void comboBoxLayers_SelectedIndexChanged(object sender, EventArgs e)
        {
            //将comboBoxFields控件清空
            comboBoxFields.Items.Clear();

            //对哈希表进行遍历
            foreach (DictionaryEntry de in layersHashtable)
            {
                //如果哈希表中某项的Key值是所选择的图层名称
                if (de.Key.ToString() == comboBoxLayers.SelectedItem.ToString())
                {
                    //将该项Value值赋值给当前的矢量图层对象
                    currentFeatureLayer = de.Value as IFeatureLayer;
                    break;
                }
            }
            //定义并得到当前矢量图层的全部字段信息
            IFields iFields;
            iFields = currentFeatureLayer.FeatureClass.Fields;
            IField field;
            //对所有字段进行遍历
            for (int i = 0; i < iFields.FieldCount; i++)
            {
                //根据索引得到字段
                field = iFields.get_Field(i);
                //如果字段名称不为"OBJECTID"或 "SHAPE"
                if (field.Name.ToUpper() != "OBJECTID" && field.Name.ToUpper() != "SHAPE")
                {
                    //如果字段类型为可以进行统计的数值类型，则将该字段添加到comboBoxFields中
                    if (field.Type == esriFieldType.esriFieldTypeInteger || field.Type ==
esriFieldType.esriFieldTypeDouble
               || field.Type == esriFieldType.esriFieldTypeSingle || field.Type ==
esriFieldType.esriFieldTypeSmallInteger)
                        comboBoxFields.Items.Add(field.Name);
                }
            }

            //显示第一个可以选择的字段
            if (comboBoxFields.Items.Count > 0)
                comboBoxFields.SelectedIndex = 0;
        }

        //当选择统计字段发生变化时触发事件，执行本函数，完成统计分析操作
        private void comboBoxFields_SelectedIndexChanged(object sender, EventArgs e)
        {
            //定义及新建IDataStatistics接口的对象进行字段统计
            IDataStatistics dataStatistics = new DataStatisticsClass();
            //获取需要统计的字段
            dataStatistics.Field = comboBoxFields.SelectedItem.ToString();
            //将当前矢量图层对象进行接口转换以进行选择集操作
            IFeatureSelection featureSelection = currentFeatureLayer as IFeatureSelection;
            //定义选择集的游标
            ICursor cursor = null;
            //使用null参数的Search方法获取整个选择集中的要素，得到相应的游标
            featureSelection.SelectionSet.Search(null, false, out cursor);
            //将该游标赋值给IDataStatistics接口对象的游标
            dataStatistics.Cursor = cursor;
            //执行统计
            IStatisticsResults statisticsResults = dataStatistics.Statistics;
            //定义StringBuilder对象进行字符串的操作
            StringBuilder stringBuilder = new StringBuilder();
            //以下语句依次增加各类统计结果
            stringBuilder.AppendLine("统计总数：  " + statisticsResults.Count.ToString() + "\n");
            stringBuilder.AppendLine("最小值： " + statisticsResults.Minimum.ToString() + "\n");
            stringBuilder.AppendLine("最大值： " + statisticsResults.Maximum.ToString() + "\n");
            stringBuilder.AppendLine("总计：  " + statisticsResults.Sum.ToString() + "\n");
            stringBuilder.AppendLine("平均值：  " + statisticsResults.Mean.ToString() + "\n");
            stringBuilder.AppendLine("标准差：  " + statisticsResults.StandardDeviation.ToString());
            //将统计结果显示在窗体中
            labelStatisticsResult.Text = stringBuilder.ToString();
        }
    }
}
```

主窗体代码:

```
private void 选择集统计ToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //新创建统计窗体
            FormStatistics formStatistics = new FormStatistics();
            //将当前主窗体中MapControl控件中的Map对象赋值给FormStatistics窗体的CurrentMap属性
            formStatistics.CurrentMap = mainMapControl.Map;
            //显示统计窗体
            formStatistics.Show();
        }
```