

```

public Attribution(){
    ESRI.ArcGIS.RuntimeManager.Bind(ProductCode.Engine);
    InitializeComponent();
}

// 声明全局变量
int Csum, Rsum; // 用于记录字段总数和行总数
public DataTable AttriDt; // 用于存储属性数据的全局数据表

// 属性表加载事件
private void Attribution_Load(object sender, EventArgs e)
{
    // 获取主窗体中的当前图层
    ILayer m_pC = ((Form1)this.Owner).m_pCurrentLayer; // 获取主窗体中的当前图层
    IFeatureLayer m_pLayer = (IFeatureLayer)m_pC; // 转换为要素图层

    // 创建数据表以存储属性数据
    AttriDt = new DataTable();
    for (int n = 0; n < m_pLayer.FeatureClass.Fields.FieldCount; n++)
    {
        // 获取字段的别名作为列名并添加到数据表中
        string name = m_pLayer.FeatureClass.Fields.get_Field(n).AliasName.ToString();
        AttriDt.Columns.Add(name);
    }

    // 获取选中的要素
    IFeatureSelection pFeaSel = (IFeatureSelection)m_pLayer;
    ISelectionSet pSelSet = pFeaSel.SelectionSet;

    // 检查是否有选中的要素
    if (pSelSet.Count < 1)
    {
        MessageBox.Show("您没有选择任何要素, 属性表将显示为空.", "提示");
        dataGridView1.DataSource = AttriDt; // 绑定空数据表
        return;
    }

    // 获取选中的要素游标
    ICursor pCursor;
    pSelSet.Search(null, false, out pCursor);
    IFeatureCursor pFeatCursor = (IFeatureCursor)pCursor;

```

```

IFeature pFeature = pFeatCursor.NextFeature();

// 填充数据表, 只显示选中的要素的属性
int i = 0; // 初始化行计数器
int j;
DataRow Dr = AttriDt.NewRow(); // 创建一个新的数据行
while (pFeature != null)
{
    // 遍历要素的字段并填充数据行
    for (j = 0; j < m_pLayer.FeatureClass.Fields.FieldCount; j++)
    {
        Dr[j] = pFeature.get_Value(j); // 获取每个字段的值并存储
    }

    AttriDt.Rows.Add(Dr); // 将数据行添加到数据表
    pFeature = pFeatCursor.NextFeature(); // 获取下一个选中要素

    // 为下一个要素创建新行
    Dr = AttriDt.NewRow();
    i++; // 增加行计数器
}

// 将选中要素的属性数据绑定到 DataGridView
dataGridView1.DataSource = AttriDt;
}

// 窗体关闭事件, 用于保存数据的更改
private void Attribution_FormClosing(object sender, FormClosingEventArgs e)
{
    // 弹出对话框, 询问是否保存操作
    DialogResult Rel = MessageBox.Show(this, "是否保存操作?", "提示", MessageBoxButtons.YesNo);

    // 如果选择保存操作
    if (DialogResult.Yes == Rel)
    {
        // 获取主窗体中的当前图层
        ILayer m_pC = ((Form1)this.Owner).m_pCurrentLayer;
        IFeatureLayer m_pLayer = (IFeatureLayer)m_pC;
        IFeatureSelection pFeaSel = (IFeatureSelection)m_pLayer;
        ISelectionSet pSelSet = pFeaSel.SelectionSet;
    }
}

```

```

// 获取选中的要素游标
ICursor pCursor;
pSelSet.Search(null, false, out pCursor);
IFeatureCursor pFeatCursor = (IFeatureCursor)pCursor;
IFeature pFeature = pFeatCursor.NextFeature();

// 初始化行计数器
int i = 0;
int j;

// 遍历选中的要素并保存属性的更改
while (pFeature != null)
{
    // 对应 DataTable 中的每个字段
    for (j = 2; j < Csum; j++) // 从索引 2 开始(跳过前两个字段)
    {
        // 更新每个字段的值并保存
        pFeature.set_Value(j, AttriDt.Rows[i][j].ToString()); // 更新要素字段
        pFeature.Store(); // 保存更改
    }
    pFeature = (IFeature)pCursor.NextRow(); // 获取下一个选中要素

    // 增加行计数器
    i++;
}
}
}

```