

```
// 柱状图渲染
private void btnBarChartRender_Click(object sender, EventArgs e)
{
    // 获取目标图层
    IGeoFeatureLayer geoFeatureLayer = getGeoLayer("GDP");
    IFeatureLayer featureLayer = geoFeatureLayer as IFeatureLayer;
    ITable table = featureLayer as ITable;

    // 创建柱状图渲染器
    IChartRenderer chartRenderer = new ChartRendererClass();
    IRendererFields rendererFields = chartRenderer as IRendererFields;

    // 添加渲染字段
    string field1 = "GDP_10";
    string field2 = "GDP_20";
    rendererFields.AddField(field1, field1);
    rendererFields.AddField(field2, field2);

    // 获取字段索引
    int[] fieldIndexs = new int[2];
    fieldIndexs[0] = table.FindField(field1);
    fieldIndexs[1] = table.FindField(field2);

    // 查找所有字段的最大值
    double maxValue = 0.0;
    ICursor cursor = table.Search(null, true);
    IRow row = cursor.NextRow();
    while (row != null)
    {
        for (int i = 0; i < 2; i++)
        {
            double fieldValue = double.Parse(row.get_Value(fieldIndexs[i]).ToString());
            if (fieldValue > maxValue)
            {
                maxValue = fieldValue; // 更新最大值
            }
        }
        row = cursor.NextRow();
    }

    // 创建柱状符号并设置属性
    IBarChartSymbol barChartSymbol = new BarChartSymbolClass();
```

```
barChartSymbol.Width = 10; // 设置柱宽
IMarkerSymbol markerSymbol = barChartSymbol as IMarkerSymbol;
markerSymbol.Size = 50;

// 设置柱状图的最大值
IChartSymbol chartSymbol = barChartSymbol as IChartSymbol;
chartSymbol.MaxValue = maxValue;

// 添加柱状图的颜色
ISymbolArray symbolArray = barChartSymbol as ISymbolArray;
IFillSymbol fillSymbol = new SimpleFillSymbolClass();
fillSymbol.Color = getRGB(255, 0, 0);
symbolArray.AddSymbol(fillSymbol as ISymbol);
fillSymbol = new SimpleFillSymbolClass();
fillSymbol.Color = getRGB(0, 255, 0);
symbolArray.AddSymbol(fillSymbol as ISymbol);

// 配置渲染器的基本符号
chartRenderer.ChartSymbol = barChartSymbol as IChartSymbol;
fillSymbol = new SimpleFillSymbolClass();
fillSymbol.Color = getRGB(0, 0, 255);
chartRenderer.BaseSymbol = fillSymbol as ISymbol;
chartRenderer.UseOverposter = false;

// 应用柱状图渲染器
geoFeatureLayer.Renderer = chartRenderer as IFeatureRenderer;
this.axMapControl1.Refresh();
}
```