

```

// 饼状图渲染
private void btnPieChartRender_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;

    // 创建饼状图渲染器
    IChartRenderer chartRenderer = new ChartRendererClass();
    IPieChartRenderer pieChartRenderer = chartRenderer as IPieChartRenderer;
    IRendererFields rendererFields = chartRenderer as IRendererFields;

    // 添加渲染字段
    string field1 = "Yichan_20";
    string field2 = "Erchan_20";
    string field3 = "Sanchan_20";
    rendererFields.AddField(field1, field1);
    rendererFields.AddField(field2, field2);
    rendererFields.AddField(field3, field3);

    // 获取字段索引
    int[] fieldIndexs = new int[3];
    fieldIndexs[0] = table.FindField(field1);
    fieldIndexs[1] = table.FindField(field2);
    fieldIndexs[2] = table.FindField(field3);

    // 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 < 3; i++)
        {
            double fieldValue = double.Parse(row.get_Value(fieldIndexs[i]).ToString());
            if (fieldValue > maxValue)
            {
                maxValue = fieldValue; // 更新最大值
            }
        }
    }
}

```

```

        row = cursor.NextRow();
    }
    // 创建饼状符号并设置属性
    IPieChartSymbol pieChartSymbol = new PieChartSymbolClass();
    pieChartSymbol.Clockwise = true; // 顺时针绘制
    pieChartSymbol.UseOutline = true; // 使用轮廓线
    // 设置饼状图的最大值
    IChartSymbol chartSymbol = pieChartSymbol as IChartSymbol;
    chartSymbol.MaxValue = maxValue;
    // 配置轮廓线
    ILineSymbol lineSymbol = new SimpleLineSymbolClass();
    lineSymbol.Color = getRGB(255, 0, 0);
    lineSymbol.Width = 2;
    pieChartSymbol.Outline = lineSymbol;

    // 设置饼状图符号的大小
    IMarkerSymbol markerSymbol = pieChartSymbol as IMarkerSymbol;
    markerSymbol.Size = 20;
    // 添加饼状图的颜色
    ISymbolArray symbolArray = pieChartSymbol as ISymbolArray;
    IFillSymbol fillSymbol = new SimpleFillSymbolClass();
    fillSymbol.Color = getRGB(0, 255, 0);
    symbolArray.AddSymbol(fillSymbol as ISymbol);
    fillSymbol = new SimpleFillSymbolClass();
    fillSymbol.Color = getRGB(0, 0, 255);
    symbolArray.AddSymbol(fillSymbol as ISymbol);
    fillSymbol = new SimpleFillSymbolClass();
    fillSymbol.Color = getRGB(255, 0, 0);
    symbolArray.AddSymbol(fillSymbol as ISymbol);

    // 配置渲染器的基本符号
    chartRenderer.ChartSymbol = pieChartSymbol as IChartSymbol;
    fillSymbol = new SimpleFillSymbolClass();
    fillSymbol.Color = getRGB(200, 200, 200);
    chartRenderer.BaseSymbol = fillSymbol as ISymbol;
    chartRenderer.UseOverposter = false;
    // 应用饼状图渲染器
    geoFeatureLayer.Renderer = chartRenderer as IFeatureRenderer;
    this.axMapControl1.Refresh();
}

```