C#编码实例

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.IO;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using Excel=Microsoft.Office.Interop.Excel;

namespace \_20170689李浩5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void excel文件ToolStripMenuItem\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = null; //清除数据源

dataGridView1.Rows.Clear(); //清空数据表格的行列

dataGridView1.Columns.Clear();

OpenFileDialog file = new OpenFileDialog(); //声明打开文件对话框 file

file.Filter = "Excel文件|\*.xls|Excel文件|\*.xlsx"; //文件过滤器，只显示Excel文件

if (file.ShowDialog() == DialogResult.OK) //如果文件正常打开

{

string fname = file.FileName; //获取打开的文件名称

string strSource = "provider=Microsoft.ACE.OLEDB.12.0;" + "Data Source=" + fname + ";Extended Properties='Excel 8.0; HDR=Yes;IMEX=1'"; //准备文件来源信息

OleDbConnection conn = new OleDbConnection(strSource); //Excel文件源放到conn中

string sqlstring = "SELECT \* FROM [Sheet1$]"; //准备选择表中的Sheet1

OleDbDataAdapter adapter = new OleDbDataAdapter(sqlstring, conn); //声明数据适配器adapter

DataSet da = new DataSet(); //声明数据集da

adapter.Fill(da); //使用adapter填充方法

dataGridView1.DataSource = da.Tables[0]; //将da.Tables[0]作为dataGridView1的数据源

}

else

return;

}

//角度转弧度

public double dmstorad(string s)

{

string[] ss = s.Split(new char[3] { '°', '′', '″' }, StringSplitOptions.RemoveEmptyEntries); //用°、 ′、 ″分割字符串s

double[] d = new double[ss.Length]; //新建一个双精度数值数组

for (int i = 0; i < d.Length; i++) //将度分秒存入这个双精度数值数组中

d[i] = Convert.ToDouble(ss[i]);

double sign = d[0] >= 0.0 ? 1.0 : -1.0;//判断角度值是否为负值

double rad = 0;

if (d.Length == 1) //根据数组的长度进行判断计算

rad = Math.Abs(d[0]) \* Math.PI / 180;//将度取绝对值，并转换为弧度

else if (d.Length == 2)

rad = (Math.Abs(d[0]) + d[1] / 60) \* Math.PI / 180;//将度取绝对值，分化为度，并最终转换为弧度

else

rad = (Math.Abs(d[0]) + d[1] / 60 + d[2] / 60 / 60) \* Math.PI / 180;//将度取绝对值，分化为度，秒化为度，并最终转换为弧度

rad = sign \* rad; //弧度前边添加正负号

return rad;//返回弧度值

}

//弧度转角度

public string radtodms(double rad)

{

double sign = rad >= 0.0 ? 1.0 : -1.0; //判断弧度值是否为负值

rad = Math.Abs(rad) \* 180 / Math.PI; //将弧度取绝对值，并转换为度

double[] d = new double[3];//新建一个长度为3的数组

d[0] = (int)rad;//取整获取度

d[1] = (int)((rad - d[0]) \* 60);//取整获取分

d[2] = (rad - d[0] - d[1] / 60) \* 60 \* 60; //获取秒，不取整

d[2] = Math.Round(d[2], 2); //将秒保留两位小数

if (d[2] == 60) //检查秒是否等于60

{

d[1] += 1;

d[2] -= 60;

if (d[1] == 60)

{

d[0] += 1;

d[1] -= 60;

}

}

d[0] = sign \* d[0];//度前边添加正负号

string s = Convert.ToString(d[0]) + "°" + Convert.ToString(d[1]) + "′" + Convert.ToString(d[2]) + "″";//将度分秒赋值给文本框，并添加°、 ′、 ″

return s;//返回返回度分秒值

}

//坐标方位角推算

public double fangweijiao(double[] sdr, double[] cr)

{

double sum = 0;

for (int i = 1; i < sdr.Length; i++)//从第二行开始循环计算坐标方位角、观测角度累加值

{

cr[i] = cr[i - 1] + sdr[i] - Math.PI;//计算坐标方位角/左角

if (cr[i] >= Math.PI \* 2)//判断坐标方位角是否在0到2PI之间

cr[i] -= Math.PI \* 2;

else if (cr[i] < 0.0)

cr[i] += Math.PI \* 2;

sum += sdr[i];

}

return sum;

}

//坐标增量调整推算

public double zuobiao(double[] cr, double[] crs, double[] crx, double[] cry)

{

double sum = 0;

for (int i = 1; i < crs.Length; i++)//从第二行开始循环计算坐标、距离累加值

{

crx[i] = crx[i - 1] + crs[i] \* Math.Cos(cr[i] / 180 \* Math.PI);

cry[i] = cry[i - 1] + crs[i] \* Math.Sin(cr[i] / 180 \* Math.PI);

sum = crs[i];

}

return sum;

}

private void button1\_Click(object sender, EventArgs e)

{

string[] sd = new string[dataGridView1.RowCount - 5]; //新建一个数组存放观测角度的原始值

double[] sdr = new double[sd.Length]; //新建一个数组存放观测角度的弧度值

double[] cr = new double[sd.Length]; //新建一个数组存放计算的坐标方位角

double sum = 0;

cr[0] = dmstorad(Convert.ToString(dataGridView1.Rows[0].Cells[4].Value)); //获取第一个坐标方位角，并将其转换成弧度，放入cr[]数组第一个元素中

double acd = dmstorad(Convert.ToString(dataGridView1.Rows[dataGridView1.RowCount - 6].Cells[4].Value));//获取终边坐标方位角，并将其转换成弧度，放入放入acd中用于计算和检核

for (int i = 1; i < sd.Length; i++) //从第二行开始循环，将观测角度的原始值放入sd[]数组中,并转换成弧度值存放在sdr数组中

{

sd[i] = Convert.ToString(dataGridView1.Rows[i].Cells[1].Value);

sdr[i] = dmstorad(sd[i]);

}

sum = fangweijiao(sdr, cr); //计算改正前坐标方位角和观测角度总和，分别存储在cr数组和sum中

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[1].Value = radtodms(sum);//将观测角度总和放入表格中

double fd, fdx;

fd = cr[cr.Length - 1] - acd;//计算角度闭合差，单位弧度

fdx = 60 \* Math.Sqrt(sd.Length - 1);//计算角度闭合差限差，单位秒

dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[1].Value = Convert.ToString(Math.Round(fd \* 180 / Math.PI \* 3600, 2)) + "″";//将角度闭合差存入表格中

dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[1].Value = Convert.ToString(Math.Round(fdx, 2)) + "″";//将角度闭合差限差存入表格中

if (Math.Abs(fd \* 180 / Math.PI \* 3600) > fdx)//检查角度闭合差是否满足要求

MessageBox.Show("角度闭合差超限！");

else

{

double vd = -fd / (sd.Length - 1);//分配角度闭合差（观测左角）

double sumvd = 0;

for (int i = 1; i < sdr.Length; i++)

{

sdr[i] += vd;//计算改正后的观测角度，并存入sdr数组中

sumvd += vd;

dataGridView1.Rows[i].Cells[2].Value =

Convert.ToString(Math.Round(vd \* 180 / Math.PI \* 3600, 2)) + "″";//将角度改正数存入表格中

dataGridView1.Rows[i].Cells[3].Value = radtodms(sdr[i]);

}

if (Math.Round(sumvd, 8) != Math.Round(-fd, 8)) //秒保留2位对应弧度是8位

MessageBox.Show("角度改正数分配有误！");

else

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[2].Value =

Convert.ToString(Math.Round(sumvd \* 180 / Math.PI \* 3600, 2)) + "″";//将角度改正数总和存入表格中

sum = fangweijiao(sdr, cr);//推算改正后的坐标方位角

if (Math.Round(cr[cr.Length - 1], 8) != Math.Round(acd, 8))

MessageBox.Show("坐标方位角推算有误！");

else

{

dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[3].Value =

radtodms(sum); //将改正后观测角度总和放入表格中

for (int i = 1; i < cr.Length - 1; i++)//将改正后坐标方位角存入表格

dataGridView1.Rows[i].Cells[4].Value = radtodms(cr[i]);

}

//至此角度调整和计算完毕

}