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.IO;

namespace WindowsFormsasdfghj

{

    public partial class Form1 : Form

    {

        public Form1()

        {

            InitializeComponent();

        }

        Node Adjust = new Node();//全局变量链表头

        string evelation = "300";//默认的起始高程

        //文件读取

        private void readfile\_Click\_1(object sender, EventArgs e)

        {

            OpenFileDialog F = new OpenFileDialog();

            F.Filter = "文本文件|\*.txt|数据文件|\*.dat|所有文件|\*.\*";

            if (F.ShowDialog() == DialogResult.Cancel)

                return;

            string path = F.FileName;

            FileStream F1 = new FileStream(path, FileMode.Open, FileAccess.ReadWrite);

            StreamReader sr = new StreamReader(F1);

            string read;

            Node temp = Adjust;

            //数据提取

            for (read = sr.ReadLine(); read != null; read = sr.ReadLine())

            {

                string[] ss = read.Split(',');

                temp.Point = ss[0];

                temp.elevation = float.Parse(ss[1]);

                temp.distance = float.Parse(ss[2]);

                temp.next = new Node();

                temp.next.next = null;

                temp = temp.next;

            }

            sr.Close();

            F1.Close();

            MessageBox.Show("文件读取成功，请计算！");

        }

        //计算按钮

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

        {

            float sumelev = 0;

            float sumdist = 0;

            for (Node t = Adjust; t.next != null; t = t.next)

            {

                sumelev += t.elevation;

                sumdist += t.distance;

            }

            sumelev = (float)Math.Round(sumelev,5)\*1000;//闭合差单位（米化毫米）

            for (Node t = Adjust; t.next != null; t = t.next)

            {

                t.currt = (sumelev / sumdist \* t.distance)\*-1;

            }

            //写入到result.txt文件

            FileStream F2 = new FileStream("result.txt", FileMode.Create, FileAccess.ReadWrite);

            StreamWriter sw = new StreamWriter(F2);

            double k;

            int i;

            //计算限差

            if (rad\_two.Checked)

            {

                k = 4 \* Math.Sqrt(sumdist / 1000);

                k = Math.Round(k, 2);

                i = 5;

                sw.WriteLine("{0,35}", "二等水准网按距离简易平差");

                sw.WriteLine("-------------------------------------------------------------------------");

            }

            else

            {

                k = 20 \* Math.Sqrt(sumdist / 1000);

                k = Math.Round(k, 2);

                i = 3;

                sw.WriteLine("{0,35}", "四等水准网按距离简易平差");

                sw.WriteLine("-------------------------------------------------------------------------");

            }

            sw.WriteLine("{0}{1,10}{2,11}{3,11}{4,11}{5,12}", "点号", "高差(m)", "测距(m)", "改正数(mm)", "改正后高差(m)", "高程(m)");

            Node p = Adjust;

            double evel = 0;

            while(p.next != null)

            {

                double right = Math.Round(p.elevation + p.currt / 1000, i);

                if (p == Adjust)

                    evel = Math.Round(right + double.Parse(evelation), i);

                else

                    evel = Math.Round(evel + right, i);

                sw.WriteLine(" {0}{1,13}{2,12}{3,12}{4,16}{5,17}", p.Point, p.elevation, p.distance, Math.Round(p.currt, 2), right, evel);

                p = p.next;

            }

            sw.WriteLine("-------------------------------------------------------------------------");

            sw.WriteLine("距离总长度：{0}m", sumdist);

            sw.WriteLine("闭合差：{0}mm", sumelev);

            sw.WriteLine("限差：{0}mm", k);

            //判断是否符合限差

            if (Math.Abs(sumelev) < k)

            {

                sw.WriteLine("{0}mm<{1}mm    闭合差符合要求！", sumelev, k);

                MessageBox.Show("闭合差符合要求！");

            }

            else if (Math.Abs(sumelev) == k)

            {

                sw.WriteLine("{0}mm={1}mm    闭合差符合要求！", sumelev, k);

                MessageBox.Show("闭合差符合要求！");

            }

            else

            {

                sw.WriteLine("{0}mm>{1}mm    闭合差超限！", sumelev, k);

                MessageBox.Show("闭合差超限！");

            }

            MessageBox.Show("计算完成，结果在result.txt！");

            textBox1.Text = sumelev.ToString();

            textBox2.Text = sumdist.ToString();

            textBox3.Text = k.ToString();

            sw.Flush();

            sw.Close();

            F2.Close();

        }

        //打开结果文件

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

        {

            System.Diagnostics.Process.Start(@"result.txt");

        }

        private void textBox4\_TextChanged(object sender, EventArgs e)

        {

            evelation = textBox4.Text;

        }

    }

    //链表创建

    class Node

    {

        public string Point;    //点号

        public float elevation; //高差

        public float distance;  //测距

        public float currt;     //高差改正数

        public Node next;       //链子

    }

}