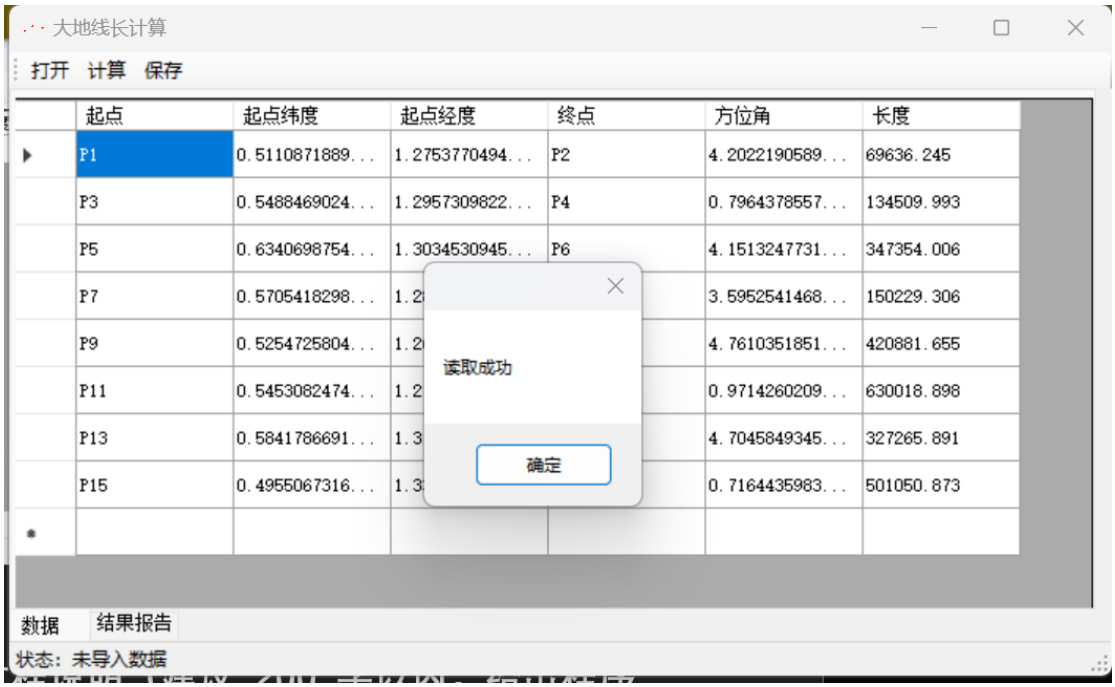
一、程序优化性说明

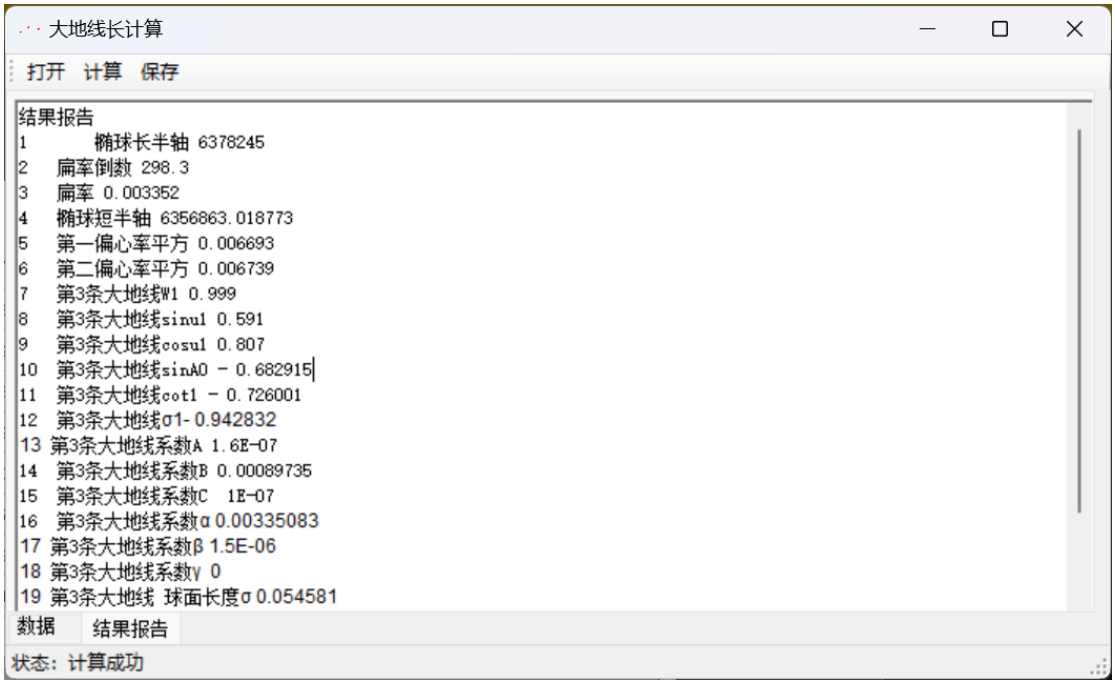
1．用户交互界面说明（建议 200 字以内，给出主要用户交互界面图）

2．程序运行过程说明（建议 200 字以内，给出程序运行过程截图）

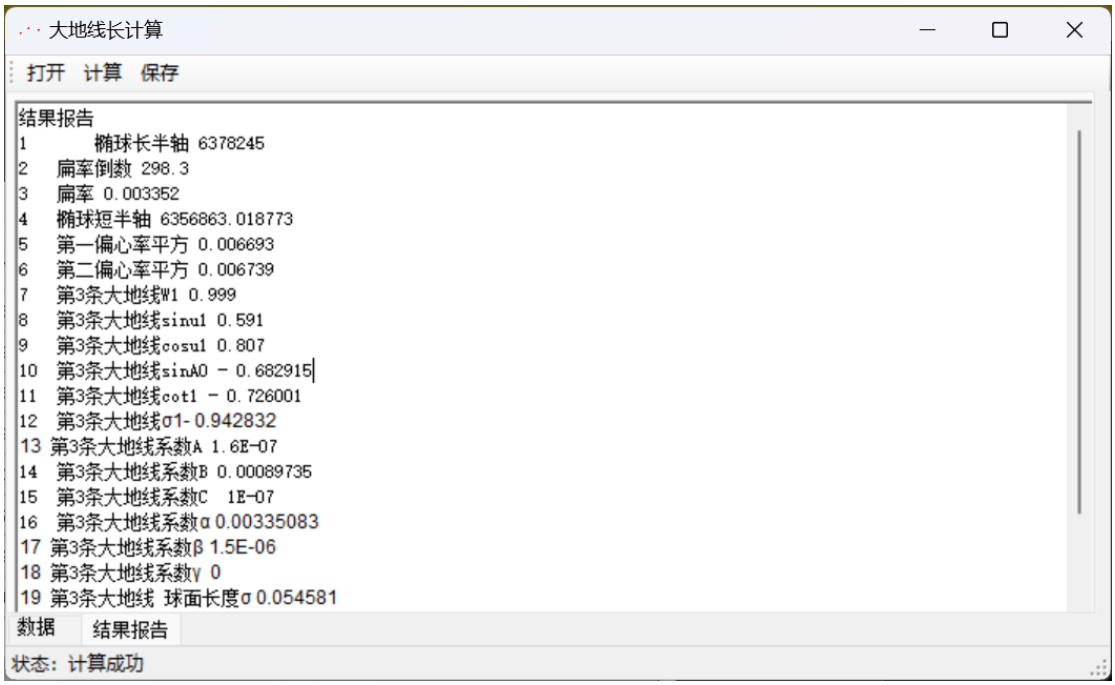
打开:



计算:



3．程序运行结果（给出程序运行结果）

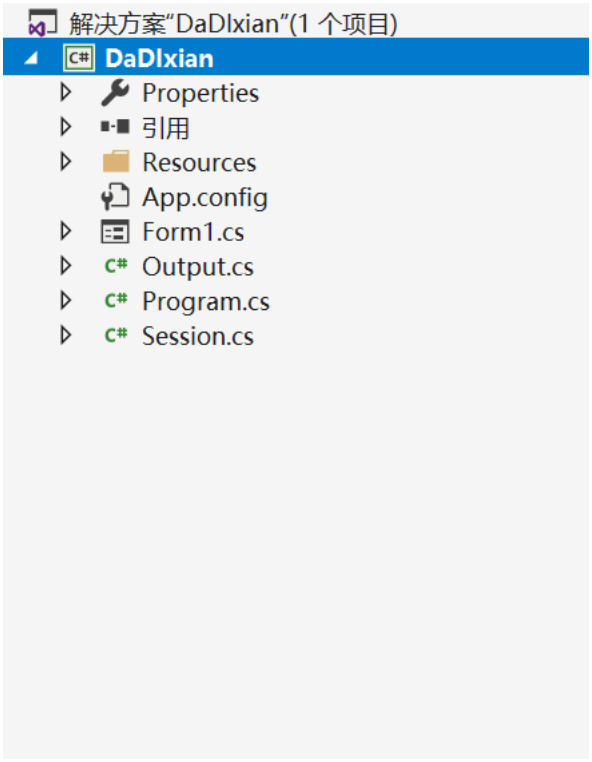


二、程序规范性说明

1．程序功能与结构设计说明（建议 500 字以内）

功能：已知：大地线起点 P1 的纬度 B1，经度 L1，大地方位角 A1，起点 P1 到终点 P2 的大地线 长度 S;
计算得到：大地线终点 P2 的纬度 B2，经度 L2 及大地方位角 A2。

结构设计：Session 类里保存每条大地线的数据，Output 进行结果数据统一保存



2．核心算法源码（给出主要算法的源码）

```
using System;
namespace DaDIxian
{
    class Session
    {
        ///<summary>
        ///每一条大地线
        /// </summary>
        public double e2;
        public double ep2;
        public double b;

        public string Startname;
        public double B1;
        public double L1;
        public string Endname;
        public double Angle;
        public double S;

        public double W1;
        public double SinU1;
        public double CosU1;
```

```
public double SinA0;
public double Cotfai1;
public double fai1;

double A;
double B;
double C;
double alpha;
double beta;
double gama;

double fai;
double result25;

double B2;
double L2;
double A2;
public Session()
{

}
public Session(string line,double e2,double ep2,double b)
{
    this.e2 = e2;
    this.ep2 = ep2;
    this.b = b;
    var buf = line.Trim().Split(',');
    Startname = buf[0];
    B1 = CalAngle(buf[1]);
    L1 = CalAngle(buf[2]);
    Endname = buf[3];
    Angle = CalAngle(buf[4]);//弧度
    S = Convert.ToDouble(buf[5]);
    Algo21();
    Algo22();
    Algo23();
    Algo24();
    Algo25();
    Algo26();
}
private double CalAngle(string line)
{
    double ddmmsss = Convert.ToDouble(line);
    double du = (int)ddmmsss;
```

```
            double fen =  (int)((ddmmsss - du) * 100);
            double miao = (((ddmmsss - du) * 100) - fen) * 100;
            double res = du + fen / 60.0 + miao / 3600.0;
            res = res / 180.0 * Math.PI;
            return res;
        }

        private void Algo21()
        {
            W1 = Math.Sqrt(1.0 - e2 * Math.Sin(B1) * Math.Sin(B1));
            SinU1 = Math.Sin(B1) * Math.Sqrt(1.0 - e2) / W1;
            CosU1 = Math.Cos(B1) / W1;
        }
        private void Algo22()
        {
            SinA0 = CosU1 * Math.Sin(Angle);
            Cotfai1 = CosU1 * Math.Cos(Angle) / SinU1;
            fai1 = Math.Atan(1.0 / Cotfai1);
        }
        private void Algo23()
        {
            double CosAo2 = 1.0 - SinA0 * SinA0;
            double K2 = ep2 * CosAo2;

            A = (1.0 - K2 / 4.0 + 7.0 * K2 * K2 / 64.0 - 15.0 * K2 * K2 * K2 / 256.0);
            A = A / b;
            B = (K2 / 4.0 - K2 * K2 / 8.0 + 37.0 * K2 * K2 * K2 / 512.0);
            C = K2 * K2 / 128.0 - K2 * K2 * K2 / 128.0;

            alpha = e2 / 2.0 + e2 * e2 / 8.0 + e2 * e2 * e2 / 16.0;
            alpha = alpha - (e2 * e2 / 16.0 + e2 * e2 * e2 / 16.0) * CosAo2;
            alpha = alpha + 3.0 * e2 * e2 * e2 / 128.0 * CosAo2 * CosAo2;
            beta = (e2 * e2 / 16.0 + e2 * e2 * e2 / 16.0) * CosAo2;
            beta = beta - e2 * e2 * e2 / 32.0 * CosAo2 * CosAo2;
            gama = e2 * e2 * e2 / 256.0 * CosAo2 * CosAo2;
        }
        private void Algo24()
        {
            double faiStart = A * S;
            double faiEnd = 0;
            while(true)
            {
                faiEnd = A * S + B * Math.Sin(faiStart) * Math.Cos(2 * fai1 + faiStart)
+ C * Math.Sin(2 * faiStart) * Math.Cos(4 * fai1 + 2 * faiStart);
```

```
                if(Math.Abs(faiEnd - faiStart) < 1.0 * Math.Pow(10, -10))
                {
                    fai = faiEnd;
                    break;
                }
                faiStart = faiEnd;
            }
        }
        private void Algo25()
        {
            result25 = alpha * fai + beta * Math.Sin(fai) * Math.Cos(2 * fai1 + fai);
            result25 += gama * Math.Sin(2 * fai) * Math.Cos(4 * fai1 + 2 * fai);
            result25 = result25 * SinA0;
        }
        private void Algo26()
        {
            double SinU2 = SinU1 * Math.Cos(fai) + CosU1 * Math.Cos(Angle) *
Math.Sin(fai);
            B2 = Math.Atan(SinU2 / (Math.Sqrt(1.0 - e2) * Math.Sqrt(1.0 - SinU2 *
SinU2)));
            double nameda = Math.Atan(Math.Sin(Angle) * Math.Sin(fai) / (CosU1 *
Math.Cos(fai) - SinU1 * Math.Sin(fai) * Math.Cos(Angle)));
            if(Math.Sin(Angle) > 0)
            {
                if(Math.Tan(nameda) > 0)
                {
                    nameda = Math.Abs(nameda);
                }
                if (Math.Tan(nameda) < 0)
                {
                    nameda = Math.PI - Math.Abs(nameda);
                }
            }
            if (Math.Sin(Angle) < 0)
            {
                if (Math.Tan(nameda) > 0)
                {
                    nameda = Math.Abs(nameda) - Math.PI;
                }
                if (Math.Tan(nameda) < 0)
                {
                    nameda = - Math.Abs(nameda);
                }
            }
```

```
            L2 = L1 + nameda - result25;
            A2 = Math.Atan(CosU1 * Math.Sin(Angle) / (CosU1 * Math.Cos(fai) *
Math.Cos(Angle) - SinU1 * Math.Sin(fai)));
            if(Math.Sin(Angle) > 0)
            {
                if(Math.Tan(A2) > 0)
                {
                    A2 = Math.PI + Math.Abs(A2);
                }
                if (Math.Tan(A2) < 0)
                {
                    A2 = 2 * Math.PI - Math.Abs(A2);
                }
            }
            if (Math.Sin(Angle) < 0)
            {
                if (Math.Tan(A2) > 0)
                {
                    A2 = Math.Abs(A2);
                }
                if (Math.Tan(A2) < 0)
                {
                    A2 = Math.PI - Math.Abs(A2);
                }
            }
            if(A2 < 0)
            {
                A2 += 2 * Math.PI;
            }
            if (A2 > 2 * Math.PI)
            {
                A2 -= 2 * Math.PI;
            }
            Angle2DDmmssss();
        }
        private void Angle2DDmmssss()
        {
            B2 = B2 / Math.PI * 180.0;
            L2 = L2 / Math.PI * 180.0;
            A2 = A2 / Math.PI * 180.0;
            double ddb = (int)B2;
            double mmb = (int)((B2 - ddb) * 60);
            double ssb = B2 * 3600 - ddb * 3600 - mmb * 60;
            B2 = ddb + mmb * 0.01 + ssb * 0.0001;
```

```
            double ddl = (int)L2;
            double mml = (int)((L2 - ddl) * 60);
            double ssl = L2 * 3600 - ddl * 3600 - mml * 60;
            L2 = ddl + mml * 0.01 + ssl * 0.0001;

            double dda = (int)A2;
            double mma = (int)((A2 - dda) * 60);
            double ssa = A2 * 3600 - dda * 3600 - mma * 60;
            A2 = dda + mma * 0.01 + ssa * 0.0001;
        }
    }
}
```