江西農業大學

JIANGXI AGRICULTURAL UNIVERSITY

蜜桔施肥决策支持系统设计指导书

国土院土管系地理信息科学教研室

二零二一年十二月

目录

一、 引言

1.1 背景

南丰蜜桔果园提供给更好更科学的施肥决策支持系统，提高施肥工作效率，帮助蜜桔实现增产目标。

1.2 定义

chax：查询某处的土壤肥力指数

shifeinpk：计算调整系数、施肥总量和分量

xishutiaozheng: 调整系数;

shifeizongliang：施肥总量

1.3 数据说明

表 1 相关数据

数据名称	数据说明
桔园 5 万.shp	桔园分布图
蜜桔土样_prj.shp	包含取土点坐标、土壤主要养分化验值

1.4 施肥背景知识

(1)配方制定方法

使用“养分丰缺指标调整系数法”制定施肥配方，主要养分指标为铵态氮、有效磷、有效钾；配方制定的步骤为：首先根据施肥田块的养分信息确定丰缺系数；（写一个函数，输入是 N、P、K（例如铵态氮 PPM 值），输出是养分调整系数。）其次根据目标产量确定施肥总量；（写一个函数，输入产量水平，输出肥料基准用量的总量，对应丰缺系数为 1.0 的情况，如果丰缺系数不是 1.0 需要参考养分调整系数得到施肥总量。）最后根据施肥模式确定各阶

段的施肥量。（写一个函数，输入是施肥总量，不同时期的用量比例，输出为不同时期的施肥总量。）

(2)施肥技术指标

1、土壤养分丰缺系数

表 2 南丰蜜桔养分丰缺系数（田间试验法）

项目	丰缺指标范围					
	极缺乏	缺乏	较缺乏	中量	丰富	极丰富
铵态氮 PPM	<30	30～55	55～85	85～120	120～150	>150
有效磷 PPM	<2	2～4.5	4.5～10	10～20	20～40	>40
有效钾 PPM	<30	30～50	50～90	90～140	140～200	>200
养分调整系数	1.24	1.16	1.08	1.0	0.92	0.84

2、各产量水平下的肥料基准用量（总量）

表 3 南丰蜜桔百公斤产量最佳养分施用量

产量水平（公斤/亩）		百公斤产量最佳养分施用量（公斤/亩）		
下限（>）	上限（≤）	N	P_2O_5	K_2O
1000	1500	0.9	0.81	0.81
1500	2000	0.9	0.81	0.81
2000	3000	0.95	0.85	0.85
3000	4000	0.95	0.85	0.85

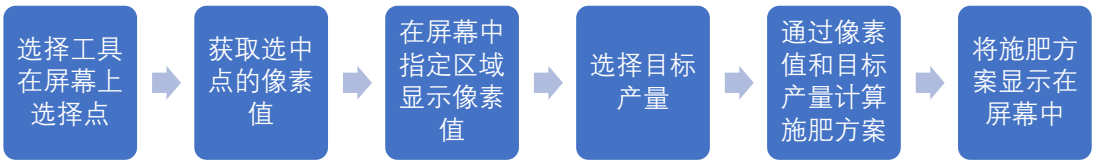
3、肥料分施方案

表 4 南丰蜜桔施肥模式

施肥时期		化肥分时期施用比例（N%-P%-K%）		
序号	时期描述	N%	P_2O_5%	K_2O%
1	春期	30	30	30
2	壮果肥	40	40	40
3	冬肥	30	30	30

二、总体设计

2.1 需求规定



2.2 运行环境

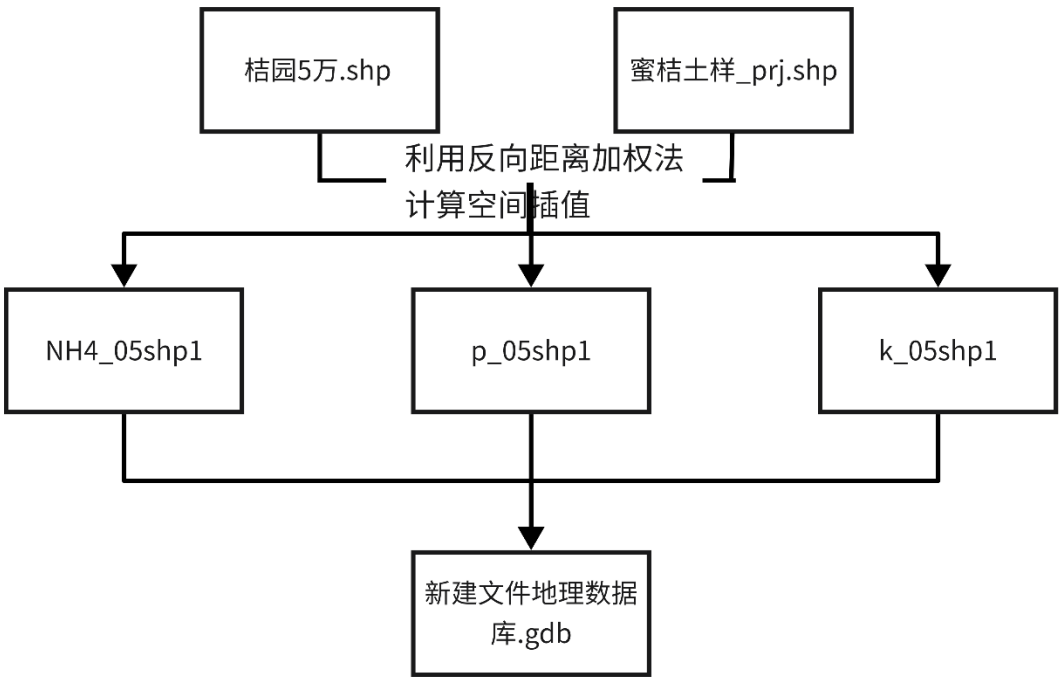
Windows11 系统；

VS2019 .net framework 4.7.2；

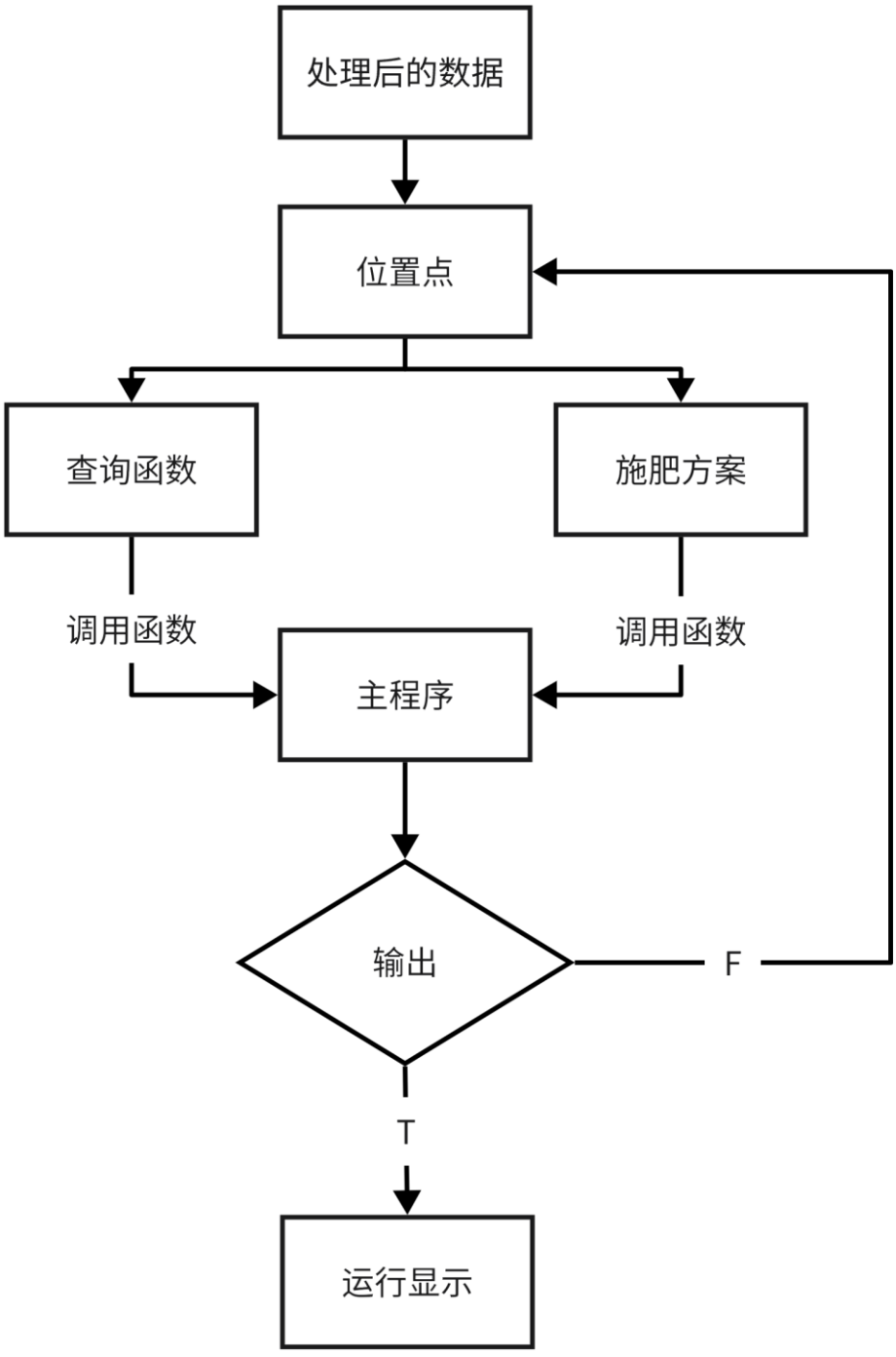
ArcGIS 10.8

2.3 设计思路

1.准备工作



2.实现施肥决策系统



三、接口设计

3.1 用户接口

表 5 接口功能

用户命令	实现功能
放大	放大地图或放大所选范围
缩小	缩小地图或缩小所选范围
全图显示	将地图全图显示在 mapcontrol 控件范围内
查询	在 Gridview1 中显示查询到的目标点的属性值
目标产量	移动滑块选择目标产量，并在 Gridview2 中显示对应的施肥方案

3.2 外部接口

软件接口：Win10 操作系统及以上

3.3 内部接口

IWorkspaceFactory：创建工作空间工厂；

IRasterWorkspace：创建栅格工作空间；

IRasterDataset：建立可管理栅格数据集合；

IRasterLayer：创建可创建和修改栅格图层对象；

IFeatureWorkspace：给矢量要素创建工作空间；

IFeatureLayer2：矢量要素创建可创建和修改的矢量图层数；

IDisplayTransformation：将屏幕坐标转化成地图坐标；

Ipoint：创建定义空间点的属性 X、Y；

IIdentify：识别要素；

IArray：管理对象的一组简单数组；

IRasterIdentifyObj2：识别栅格图层的对象；

四、运行设计

4.1 运行模块组合

1.放大/缩小/全图显示功能：ToolbarControl 控件、MapControl 控件、LicenseControl 控件；

2.查询功能：ToolbarControl 控件、 MapControl 控件、LicenseControl 控件；；

3.根据目标产量输出配案功能:TrackbarControl 控件、GridView 控件、MapControl 控件、LicenseControl 控件；

4.2 运行控制

1.启动.exe 程序；

2.通过工具栏的工具图标选中不同目标工具，可在地图中实现放大缩小等操作；

3.选中兔子图标的查询工具，在地图上任意选中一点，系统将要素属性输出在相应位置；

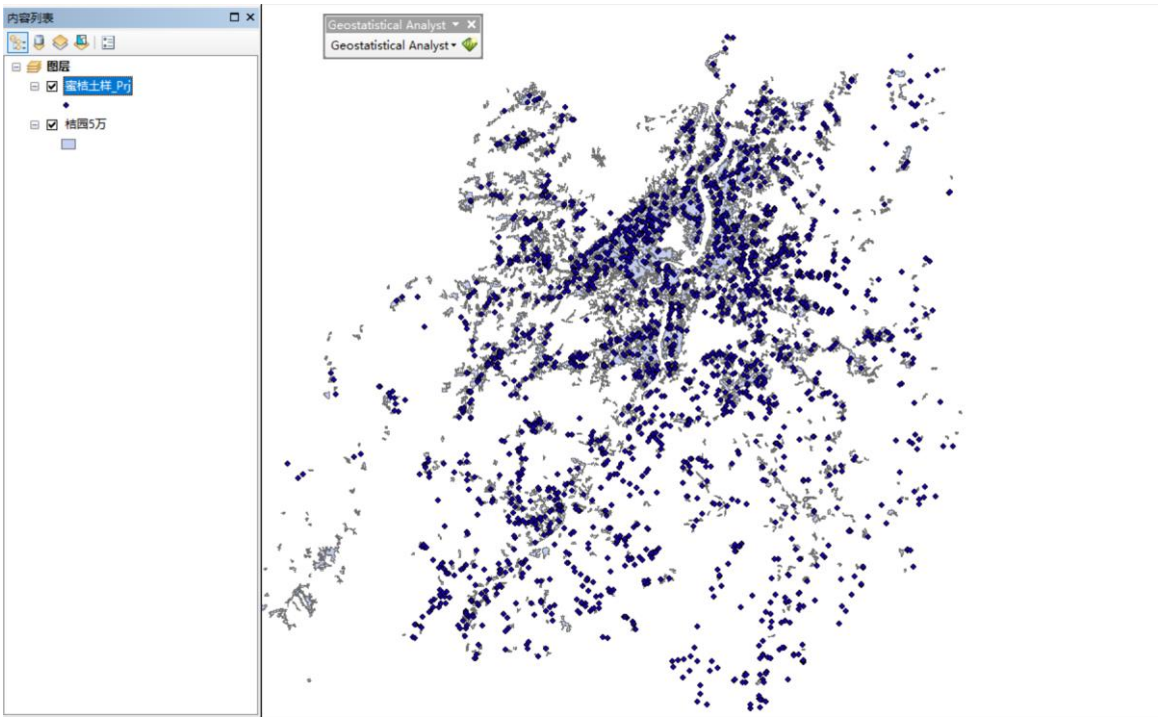
4.选择目标产量，自动计算得出施肥方案并显示；

5.结束程序与运行.

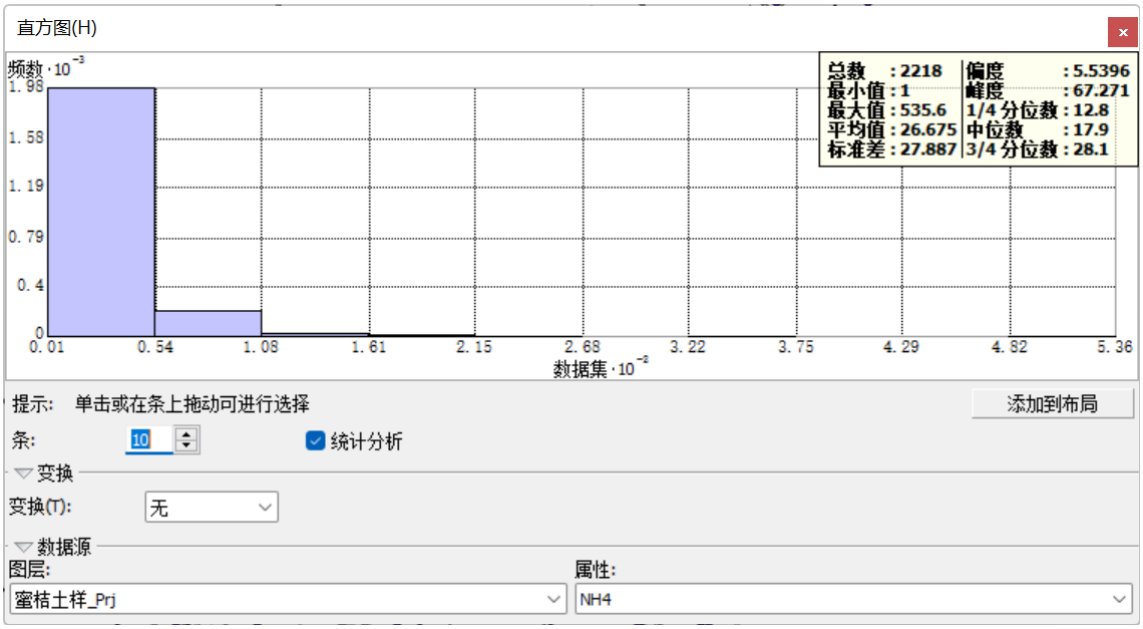
五、程序代码实现

5.1 数据准备

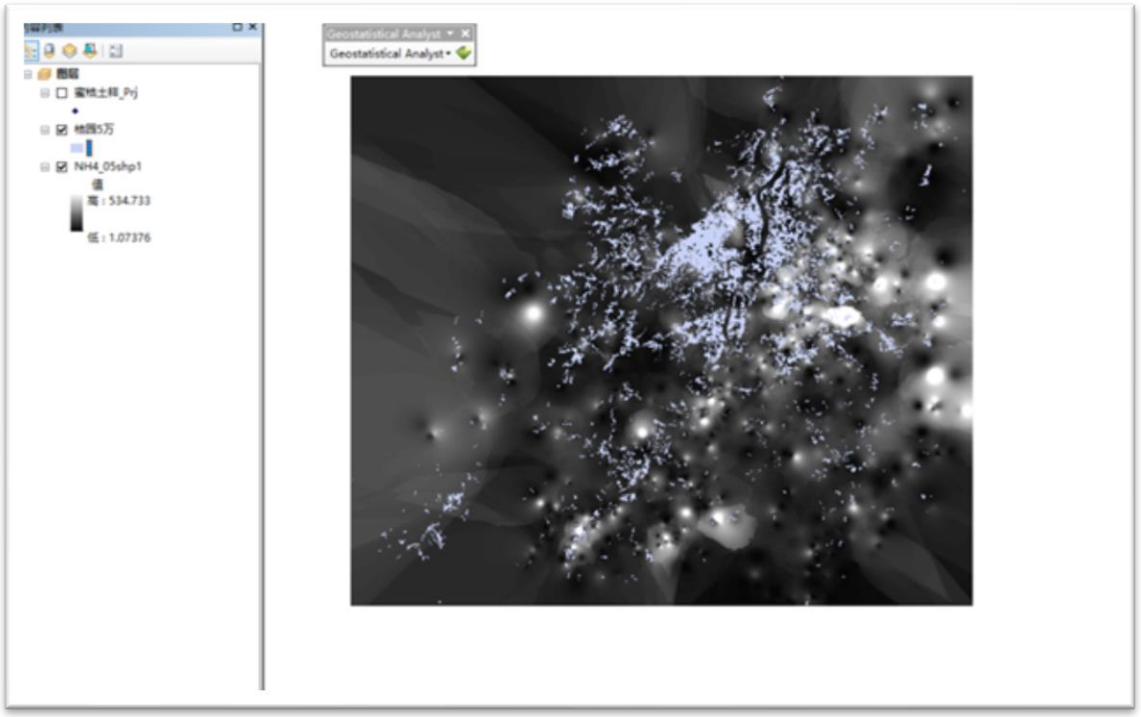
1. 在 Arcmap 中导入桔园 5 万.shp 和蜜桔土样_prj.shp；



2. 利用 Geostatistical Analyst 工具探索数据（NH4 为例）；



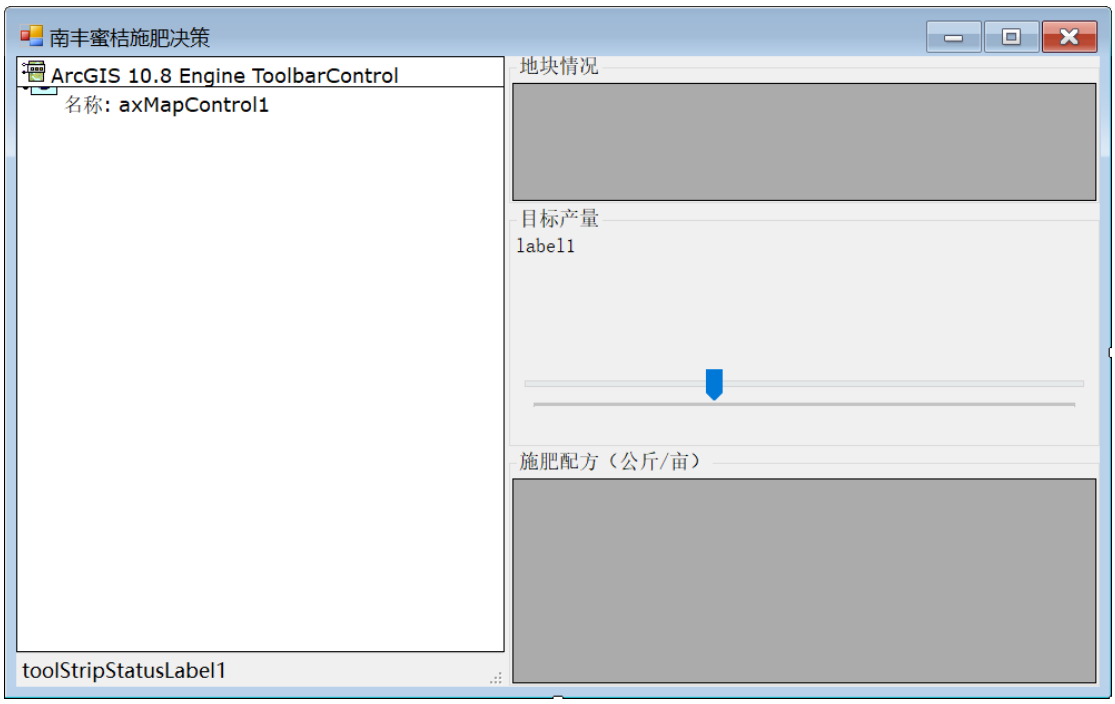
3.对蜜桔土样的 NH4/P/K 三种属性使用反距离权重法进行插值分析，设置器处理范围为桔园 5 万图层；



4.得到三幅插值图，将其输出保存到建立的地理文件数据库中，导出为三幅不同的栅格数据格式（.tif），将新建地理文件数据库放到程序代码的路径下。

5.2 代码实现

1.打开 VS2019，选择窗体应用程序（.NET Framework），并在窗体设计中布置好控件，如下。



2.绑定许可。

```
1.  ESRI.ArcGIS.RuntimeManager.Bind(ESRI.ArcGIS.ProductCode.EngineOrDesktop);
```

3.加载空间数据，通过建立不同的共工作空间，将栅格数据和矢量数据加载到 MapControl 控件中进行显示，对栅格数据设置不可见，但可以使用；

```
1.  //创建工作空间工厂
2.   IWorkspaceFactory workspaceFactory = new FileGDBWorkspaceFactoryClass();
3.  //打开栅格数据工作空间
4.  IRasterWorkspaceEx rasterWorkspaceEx =
    workspaceFactory.OpenFromFile(@"D:\Desktop\实习二 蜜桔施肥决策\mijiUI\data\新建文件地理数据库.gdb", 0) as IRasterWorkspaceEx;
5.  //加载相应栅格图层，利用 addrasterdata 加载到 axMapControl1 控件显示
6.  IRasterLayer rasterLayer1 = addrasterdata(rasterWorkspaceEx, "NH4_05shp1");
```

```
IRasterLayer rasterLayer2 = addrasterdata(rasterWorkspaceEx, "p_05shp1");
IRasterLayer rasterLayer3 = addrasterdata(rasterWorkspaceEx, "k_05shp1");
//建立矢量数据工作空间
IFeatureWorkspace workspace = workspaceFactory.OpenFromFile(@"D:\Desktop\实习二 蜜桔施肥决策\mijiUI\data\新建文件地理数据库.gdb", 0) as IFeatureWorkspace;
// 建立建立矢量要素类，打开图层数据
IFeatureLayer2 featureLayer2 = new FeatureLayerClass();
featureLayer2.FeatureClass = workspace.OpenFeatureClass("桔园 5 万");
//添加到 axMapControl1 显示
axMapControl1.AddLayer(featureLayer2 as ILayer);
```

addrasterdata 函数，添加栅格数据到图层（传入工作空间和图层名）

```
private IRasterLayer addrasterdata(IRasterWorkspaceEx rasterWorkspaceEx, string rastername)
{
    IRasterDataset3 rasterDataset3 = rasterWorkspaceEx.OpenRasterDataset(rastername) as IRasterDataset3;
    IRaster2 raster2 = rasterDataset3.CreateFullRaster() as IRaster2;
    IRasterLayer rasterLayer = new RasterLayerClass();
    rasterLayer.CreateFromRaster(raster2 as IRaster);
    axMapControl1.AddLayer(rasterLayer);
    rasterLayer.Visible = false;
    return rasterLayer;
}
```

4. 新建类 chax. cs，能够对选中点的属性值进行查询，并且能够以表格的形式显示在 dataGridView1。

（1）利用 additem 查询工具添加到 axToolbarControl1 工具栏中，同时将查询所需参数传到 chax. cs 中。

```
axToolbarControl1.AddItem(new chax(axMapControl1, rasterLayer1, rasterLayer2, rasterLayer3, table), -1, -1, false, 0, esriCommandStyles.esriCommandStyleIconOnly);
```

（2）对 dataGridView1 进行布局初始化在 Form1_Load 中编写程序；

```
DataTable table = new DataTable();
table.Columns.Add("项目", typeof(string));
table.Columns.Add("指标", typeof(string));
string[] values = new string[2];
values[0] = "土壤质地";
values[1] = "壤土";
table.Rows.Add(values);
values[0] = "土地利用类型";
values[1] = "桔园";
table.Rows.Add(values);
values[0] = "N";
values[1] = "0";
table.Rows.Add(values);
values[0] = "P";
values[1] = "0";
table.Rows.Add(values);
values[0] = "K";
values[1] = "0";
table.Rows.Add(values);
dataGridView1.DataSource = table;
```

（3）在 chax.cs 中调用参数

```
private AxMapControl m_mapcontrol = null;
private IRasterLayer m_rasterlayer = null;
private IRasterLayer m_rasterlayer2 = null;
private IRasterLayer m_rasterlayer3 = null;
private DataTable m_table = null;
public chax(AxMapControl axMapControl,IRasterLayer
rasterLayer, IRasterLayer rasterLayer2, IRasterLayer
rasterLayer3,DataTable table)
{
    // TODO: Define values for the public properties
    m_mapcontrol = axMapControl;
     m_rasterlayer = rasterLayer;
     m_rasterlayer2 = rasterLayer2;
     m_rasterlayer3 = rasterLayer3;
     m_table = table;
```

```
    //下面是程序自带代码
  ...
}
```

（4）编写 OnMouseDown 事件，在启用工具并按下鼠标选中点时，查询属性并显示在 dataGridView1 中。

```
public override void OnMouseDown(int Button, int Shift, int X, int Y)
{
    //查询属性
    //z 坐标转换
    IDisplayTransformation displayTransformation = new DisplayTransformationClass();
    displayTransformation = m_mapcontrol.ActiveView.ScreenDisplay.DisplayTransformation;
    IPoint point =displayTransformation.ToMapPoint(X, Y);//屏幕点转化成地图点
    //调用 chaxunshuxing 函数，查询对应图层属性
    chaxunshuxing(m_rasterlayer, point,"氮",2);
    chaxunshuxing(m_rasterlayer2, point, "磷",3);
    chaxunshuxing(m_rasterlayer3, point, "钾",4);
}
```

Chaxunshuxing 函数，查询该点属性值并显示

```
private void chaxunshuxing(IRasterLayer rasterLayer,IPoint point,String Name,int index)
{
    //查询选中点的属性表，识别要素
    IIdentify identify = rasterLayer as IIdentify;
    //选中点的要素数据存储到 array
    IArray array = identify.Identify(point);
    //识别栅格图层的数据
    IRasterIdentifyObj2 rasterIdentifyObj2 = array.get_Element(0) as IRasterIdentifyObj2;
    string property;
    string value;
    //获取属性表第 0 列的值
    rasterIdentifyObj2.GetPropAndValues(0, out property, out value);
    //在标签中显示
    m_table.Rows[index][1] = value;
}
```

5.新建类 shifeinpk.cs,进行施肥配方制定。使用“养分丰缺指标调整系数法”制定施肥配方，主要养分指标为铵态氮、有效磷、有效钾；配方制定的步骤为：首先根据施肥田块的养分信息确定丰缺系数；其次根据目标产量确定施肥总量；最后根据施肥模式确定各阶段的施肥量。并且将配方在 dataGridView2 进行显示。

（1）根据所给要求，赋予养分调整系数；

```
1.  //计算养分调整系数,values 土壤背景值，yuzhi[]为判断系数的界值,return 调整后的系数
2.  public static double[] xishutiaozheng(double[] values, double[][] yuzhi)
3.  {
4.      double[] result = new double[3];
5.      for (int i = 0; i < 3; i++)
6.      {
7.
8.          result[i] = (values[i] < yuzhi[i][0]) ? 1.24
9.          : (values[i] >= yuzhi[i][0] && values[i] < yuzhi[i][1]) ? 1.16
10.         : (values[i] >= yuzhi[i][1] && values[i] < yuzhi[i][2]) ? 1.08
11.         : (values[i] >= yuzhi[i][2] && values[i] < yuzhi[i][3]) ? 1
12.         : (values[i] >= yuzhi[i][3] && values[i] < yuzhi[i][4]) ? 0.92
13.         : 0.84;
14.     }
15.     return result;
16. }
```

（2）根据目标产量确定施肥总量；

```
1.  //计算施肥总量,output 产量,factor 是养分丰缺系数第一个函数中得到,changliangyu 是百公斤最佳养分使用量 return 施肥的总量
2.  public static double[] shifeizongliang(double output, double[] factor,
    double[][] chanliangyu)
3.  {
4.      double[] result=new double[3];//最终 n，p，k 分别的总产量
5.      double xishu;
6.      //result[0] = output / 100*factor*0.9;
7.      for (int i = 0; i < 3; i++)
8.      {
9.          //百公斤最佳养分使用量系数
10.         xishu = (output < 2000) ? chanliangyu[i][0] : chanliangyu[i][1];
```

```
11.         //计算得到施肥总量
12.         result[i] = output / 100 * factor[i] * xishu;
13.     }
14.     return result;
15. }
```

（3）计算所需参数传入，并且初始化 dataGridView2。

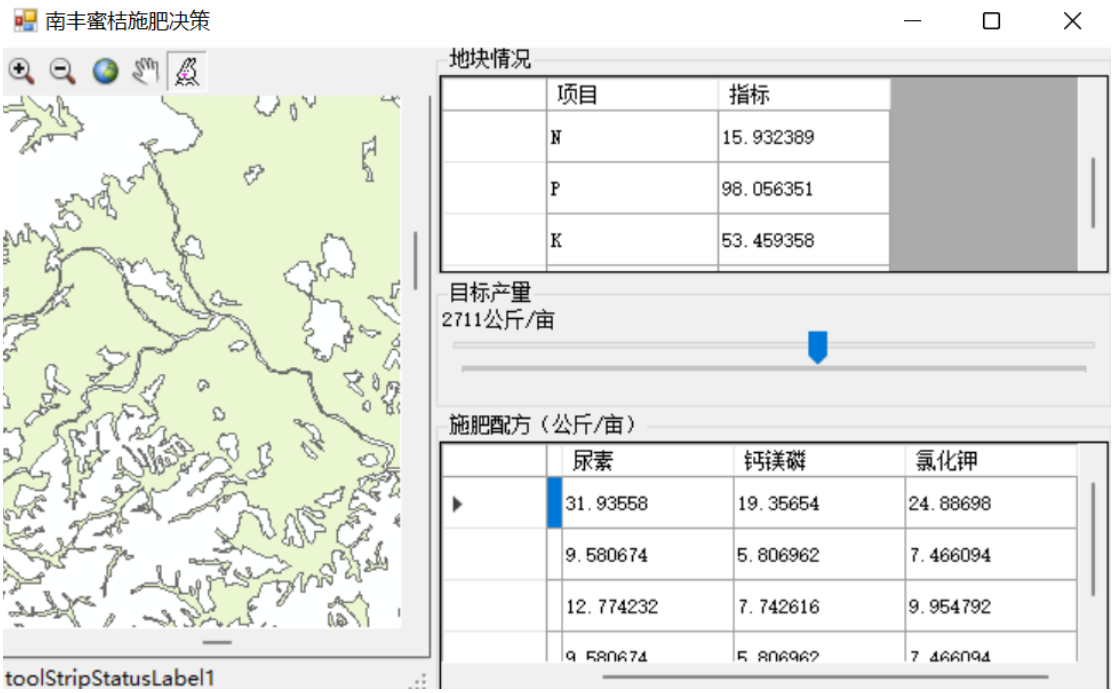
```
1.  double[][] yuzhitable = new double[3][];
2.  double[][] changliangyu = new double[3][];
3.  DataTable table1 = new DataTable();
4.  private void Form1_Load(object sender, EventArgs e)
5.  {
6.      yuzhitable[0] = new double[] { 30, 55, 85, 120, 150 };
7.      yuzhitable[1] = new double[] { 2, 4.5, 10, 20, 40 };
8.      yuzhitable[2] = new double[] { 30, 50, 90, 140, 200 };
9.      changliangyu[0] = new double[] { 0.9, 0.95 };
10.     changliangyu[1] = new double[] { 0.81, 0.85 };
11.     changliangyu[2] = new double[] { 0.81, 0.85 };
12.     //省略前面出现的部分代码
13.      ...
14.     table1.Columns.Add("项目", typeof(string));
15.     table1.Columns.Add("尿素", typeof(string));
16.     table1.Columns.Add("钙镁磷", typeof(string));
17.     table1.Columns.Add("氯化钾", typeof(string));
18.     string[] values1 = new string[4];
19.     values1[0] = "施肥总量";
20.     values1[1] = "0";
21.     values1[2] = "0";
22.     values1[3] = "0";
23.     table1.Rows.Add(values1);
24.     values1[0] = "春肥";
25.     values1[1] = "0";
26.     values1[2] = "0";
27.     values1[3] = "0";
28.     table1.Rows.Add(values1);
29.     values1[0] = "壮果肥";
30.     values1[1] = "0";
31.     values1[2] = "0";
32.     values1[3] = "0";
33.     table1.Rows.Add(values1);
34.     values1[0] = "冬肥";
35.     values1[1] = "0";
36.     values1[2] = "0";
```

```
37.     values1[3] = "0";
38.     table1.Rows.Add(values1);
39.     //将 table 的值赋给 dataGridView2
40.     dataGridView2.DataSource = table1;
41. }
```

（4）滑块设计，并且在 dataGridView2 中显示。

```
1. private void trackBar1_Scroll(object sender, EventArgs e)
2. {
3.     label1.Text = trackBar1.Value.ToString() + "公斤/亩";
4.     //输入选中点的背景值
5.     double[] back = new double[]
   { Convert.ToDouble(table.Rows[2][1]),
   Convert.ToDouble(table.Rows[3][1]),
   Convert.ToDouble(table.Rows[4][1]) };
6.     double[] result = shifennpk.xishutiaozheng(back,
   yuzhitable);
7.     //计算养分调整系数 N,P,K，通过最佳施肥量计算 N, P, K
8.     double[] finallyresult =
   shifennpk.shifeizongliang(trackBar1.Value, result,
   changliangyu);
9.     //各种肥料使用总量 N,P,K 和分量，显示
10.     for (int i = 0; i < 3; i++)
11.     {
12.         table1.Rows[0][i + 1] = finallyresult[i];
13.         table1.Rows[1][i + 1] = finallyresult[i] * 0.3;
14.         table1.Rows[2][i + 1] = finallyresult[i] * 0.4;
15.         table1.Rows[3][i + 1] = finallyresult[i] * 0.3;
16.     }
17. }
```

5.3 结果展示


南丰蜜桔施肥决策
地块情况
项目
指标
N
15.932389
P
98.056351
K
53.459358
目标产量
2711公斤/亩
施肥配方（公斤/亩）
尿素
钙镁磷
氯化钾
31.93558
19.35654
24.88698
9.580674
5.806962
7.466094
12.774232
7.742616
9.954792
toolStripStatusLabel1