

利用 Python 语言实现用地备案宗地界址点

黄东¹ 谢林²

1 遂宁市自然资源和规划局 2 天全县自然资源和规划局

DOI:10.12238/gmsm.v5i5.1441

[摘要] 常规的用地备案宗地界址点文件制作方法过程比较复杂,需要反复在Excel软件中进行粘贴复制、拖拖拉拉、重命名保存等操作,耗时耗力,还容易出错。通过对界址点模板的分析与界址点文件制作流程的梳理,利用Python语言编程实现了界址点文件自动生成。

[关键词] 宗地; 界址点; Python编程

中图分类号: P2 文献标识码: A

Using the Python Language to Realize Automatic Generation of Parcel Boundary Point File for Land Use Record

Dong Huang¹ Lin Xie²

1 Suining Bureau of Natural Resources and Planning 2 Tianquan Bureau of Natural Resources and Planning

[Abstract] The ordinary process of making the boundary point file of the land use record is complicated, which needs to copy and paste, drag and drop, rename and save in Excel software repeatedly, which is time-consuming and labor-consuming, and also easy to make mistakes. Through the analysis of the boundary point template and the carding of the boundary point file making process, the automatic generation of the boundary point file is realized by Python programming language.

[Key words] land parcels; boundary points; Python programming

引言

为加强对各地城市建设用地的监督管理,实现土地节约集约利用,为国土空间规划和“三区三线”划定打好基础,自然资源部推出了“用地预审备案系统”和“土地市场动态监测与监管系统”,要求各地每使用一宗土地都必须把相关用地文件和宗地界址点坐标上传到上述系统,以便国家实现对全国建设用地市场供给的实时管理。

2018年,自然资源部推出了新的用地界址点上报模板,采用“.txt”文件格式存储,新模板相比上一版模板更加简洁明了,但制作上报界址点文件,依然是地块信息填报最复杂也最为耗时的的工作。

1 现状

界址点文件常规的制作方法和流程是,首先在CAD软件中计算出宗地面积,然后导出宗地界址点,使用Excel表格对界址点文件进行格式调整(增加表头信息、统计界址点数量、填写用地性质、填写宗地面积等),最后把Excel表格另存为“.csv”格式,再改为“.txt”格式才算完成。整个过程耗时耗力,且一不小心还容易出错,造成界址点文件不被系统接纳。

经过笔者对于界址点制作流程和界址点文件模板以及CAD软件导出界址点文件的研究,发现界址点文件呈现出极强的规

律性和逻辑性,完全可以使用编程来实现自动化生成。由于Python语言具有语法简单,初学者容易上手,代码可读性强,第三方库丰富等特点,故笔者选择Python作为本项目的实现语言。

【属性描述】

坐标系=2000国家大地坐标系

几度分带=3

投影类型=高斯克吕格

计量单位=米

带号=35

精度=0.001

转换参数=,,,,,

【地块坐标】

7,8.5055,区块1,地块名称,面,,住宅用地,,@

J1,1,3372911.141,35558893.605

J2,1,3372915.24,35559055.573

J3,1,3372730.782,35559094.527

J4,1,3372579.117,35558948.961

J5,1,3372593.464,35558819.796

J6,1,3372853.754,35558784.942

J7,1,3372982.874,35558838.248

8,9.0337,区块2,地块名称,面,,住宅用地,,@

J8,2,3372531.854,35559609.326

J9,2,3372624.083,35559927.113

J10,2,3372556.448,35559978.368

J11,2,3372433.476,35559927.113

J12,2,3372355.594,35559841.003

J13,2,3372277.712,35559662.632

J14,2,3372324.851,35559574.472

J15,2,3372400.684,35559549.87

图1 新界址点模板

1,1,35558893.605,3372911.141,0.000
 2,1,35559055.573,3372915.240,0.000
 3,1,35559094.527,3372730.782,0.000
 4,1,35558948.961,3372579.117,0.000
 5,1,35558819.796,3372593.464,0.000
 6,1,35558784.942,3372853.754,0.000
 7,1,35558838.248,3372982.874,0.000
 8,2,35559609.326,3372531.854,0.000
 9,2,35559927.113,3372624.083,0.000
 10,2,35559978.368,3372556.448,0.000
 11,2,35559927.113,3372433.476,0.000
 12,2,35559841.003,3372355.594,0.000
 13,2,35559662.632,3372277.712,0.000
 14,2,35559574.472,3372324.851,0.000
 15,2,35559549.870,3372400.684,0.000

图2 CAD软件导出界址点文件内容

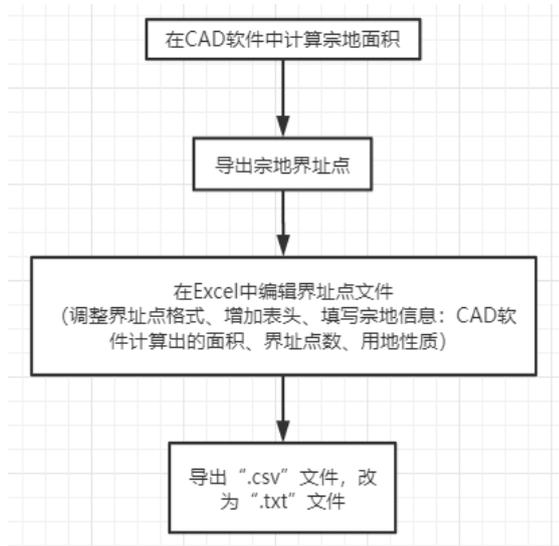


图3 界址点文件一般制作流程

2 设计思路

为方便使用,笔者把软件设计成带图形界面形式的,通过三个按钮实现原始界址点数据的读取、处理、软件退出等操作。

具体编程实现上,采用面向对象和面向过程相结合的方式,通过待处理界址点数据的分析,把文件的内容划分为“点”和“区块”两个对象。其中,“点”对象包含点名、点号、X坐标、Y坐标等四个参数,用于接纳和保存每个界址点有关的信息;“区块”对象包区块面积、区块界址点数、区块相关信息、区块界址点集合等四个参数,用于接纳和保存每个区块有关的信息。

执行过程中通过“打开文件函数”用于选择待处理界址点文件以及获取其保存路径,而后续的数据处理任务全部都交给“处理文件”函数来完成,主要执行数据读取、每行界址点遍历、圈号判定、点对象实例化、坐标数据分块暂存、点对象分块暂存、区块点数统计、根据坐标数据计算区块面积、区块相关信息填写、区块点集归位、区块对象写入区块集合等操作。

再通过把这两个函数绑定到程序图形界面的“打开处理文

件路径”与“处理文件”这两个按钮上,实现程序的图形界面化操作。

具体设计思路见下图:

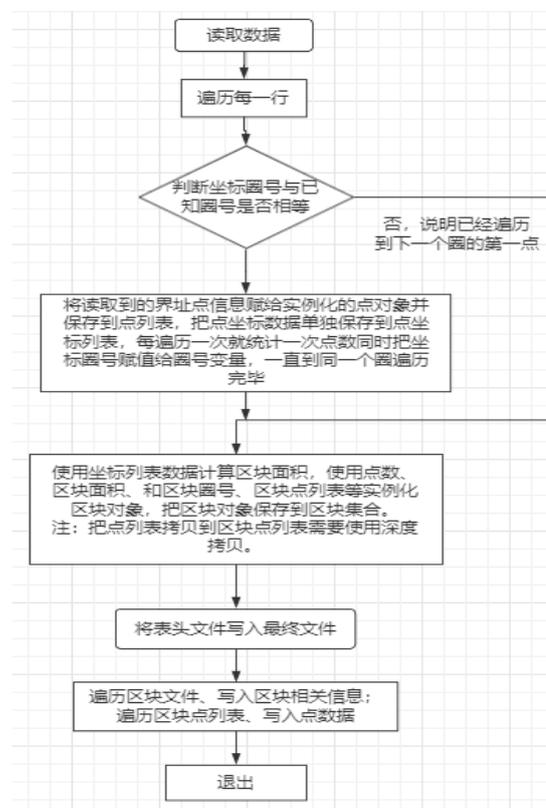


图4 程序设计流程图

3 主要代码解析

本工程的难点在于点数据与区块数据的获取与保存,以下是实现该目标的关键代码。

```

import tkinter as tk
from tkinter import filedialog
from tkinter import*
from tkinter.messagebox import showinfo
from shapely.geometry import polygon
import copy
.....
def DealWithFile():
    file_in=open(text1.get(),'r')
    QukuaiJiHe = []#用于存放实例化的区块对象列表
    PointList=[]#用于存放实例化的点对象列表
    PointCount=0
    LastQuanhao=1
    PtsList=[]#用于存放点坐标,便于面积计算
    for line in file_in:
        tempdata=line.split(',')
        if LastQuanhao == int(tempdata[1]):
  
```

```

pts = PointTemplate()
pts.PointName=tempdata[0]
pts.QuanHao=int(tempdata[1])
pts.PointXcoordinate=float(tempdata[3])
pts.PointYcoordinate=float(tempdata[2])
PtsList.append((pts.PointXcoordinate,
pts.PointYcoordinate))#单独存储区块点坐标(用于计算区块面积)
PointList.append(pts)#存放实例点对象
LastQuanhao= int(tempdata[1])
PointCount=PointCount+1
else:#上一个圈号和现在的圈号不相等时,执行下列语句
Qkobject=Qukuai()
Qkobject.TotalPointsCount=PointCount
Qkobject.RelateInformation="区块"+str(LastQuanhao)+"",
地块名称,面,,住宅用地,,@"
Qkobject.Area=round(polygon.Polygon(PtsList).area/
10000,4)
Qkobject.ArraysPoints=copy.deepcopy(PointList) #需
要使用深度拷贝
QukuaiJiHe.append(Qkobject)#将区块对象写入区块集合
列表
PointList.clear()
PtsList.clear()
#清空上一区块的数据,便于下一区块数据保存
pts=PointTemplate()
pts.PointName=tempdata[0]
pts.QuanHao=int(tempdata[1])
pts.PointXcoordinate=float(tempdata[3])
pts.PointYcoordinate=float(tempdata[2])
PtsList.append((pts.PointXcoordinate,
pts.PointYcoordinate))
PointList.append(pts)
PointCount=1
LastQuanhao=int(tempdata[1])
#由于最后一个区块循环结束后Else不会执行,需要下面的
语句将最后一个区块放在区块集合里面
Qkobject = Qukuai()
Qkobject.TotalPointsCount = PointCount
Qkobject.RelateInformation="区块"+str(LastQuan

```

```

hao)+"",地块名称,面,,住宅用地,,@"
Qkobject.Area
=round(polygon.Polygon(PtsList).area/10000,4)
Qkobject.ArraysPoints=copy.deepcopy(PointList)#需
要使用深度拷贝
QukuaiJiHe.append(Qkobject)
其余设置函数用户界面、定义“点”和“区块”对象、定
义打开文件函数、将获取到的区块数据存入目标文件等操作相
对简单,在此不作展示。

```

4 结束语

经实际测试,本程序运行情况良好,能将原本需耗时数分钟的工作在几秒钟内完成,省去了用户在Excel中反复打开文件,拖拽、粘贴、复制数据,另存文件等繁杂的操作工序,即便是不太懂测绘的非专业人员使用该软件也不会出错。即便如此,我们的程序还是存在以下几点不足,一是该程序与其它.exe程序不一样,Windows系统需要安装Python软件才能正常使用;二是笔者程序预设的用地性质都是居住用地,而实际可能是工业用地或商服用地等,所以需要用户在结果中手动调整用地性质;三是程序中没有考虑区块中有扣除区块的情况,但由于有扣除的情况本身占比很低,且笔者程序在运行成果中已经帮用户处理好了被扣除区块的界址点数据和被扣除面积等,用户只需在运行成果中稍作调整即可得到最终成果。故整体来看,本程序已经基本达到设计目标,具备推广价值。

[参考文献]

- [1]Sean Gillies.The Shapely User Manual,[EB/OL].<https://shapely.readthedocs.io/en/latest/manual.html#the-shapely-user-manual>,Nov14,2022.
- [2]肖子文.持续深化“放管服”改革优化建设用地审批制度——新《土地管理法实施条例》系列解读之六[J].资源与环境,2022,(03):13.
- [3]许亚,任扣清,姚奕.农村不动产测量中界址点、界址线属性数据快速处理方法[J].现代测绘,2022,45(4):56-58.
- [4]康健,武晔.由行列式计算面积和体积的Python程序[J].电脑编程技巧与维护,2020,(04):7-9.
- [5]张楠,杨爱玲,郭鑫,等.国土空间规划中“三条控制线”划定逻辑研究[J].测绘与空间地理信息,2022,45(z1):1-3,10.

作者简介:

黄东(1988—),男,汉族,四川遂宁人,本科,注册测绘师,2011年毕业于西南科技大学环境与资源学院测绘工程专业,从事测绘地理信息生产与技术应用研究工作。