



# excel 文件合并工具

# 分工介绍

邹媛媛

李旭阳： 程序设计，代码编写

---

王伟杰

刘靖： 代码测试，程序改进， PPT制作

---

游继翔

钟铭涛： PPT讲解，功能演示，文案撰写

---





# 目录



1

产品研发背景

2

产品功能演示

3

程序代码解析



1

## 产品研发背景



# 需求



放假时需要统计  
学生去向信息。

学校经常需要收  
集学生个人信息。



# 传统解决方法

班干部一个宿舍一个宿舍敲门，然后逐个询问并填写信息。

以宿舍或个人为单位上传消息表，班干部逐个将文件打开，再对每一份信息复制、粘贴、汇总。





# 存在的问题

---



效率低



工作量大




浪费大量时间



人工复制粘贴可能出错



# 我们的解决办法



使用我们编写的程序，只需要把每一位同学发来的文件保存在同一文件夹中，再运行代码，并选择全部所需要汇总的excel文件，即可一次性汇总完成。



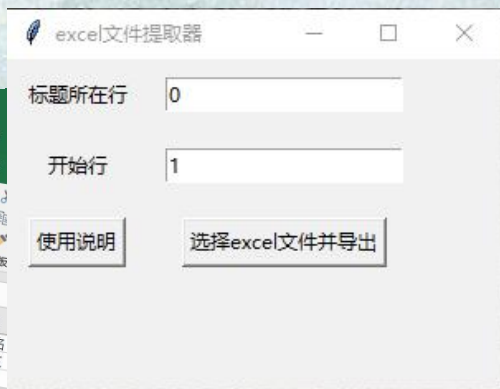




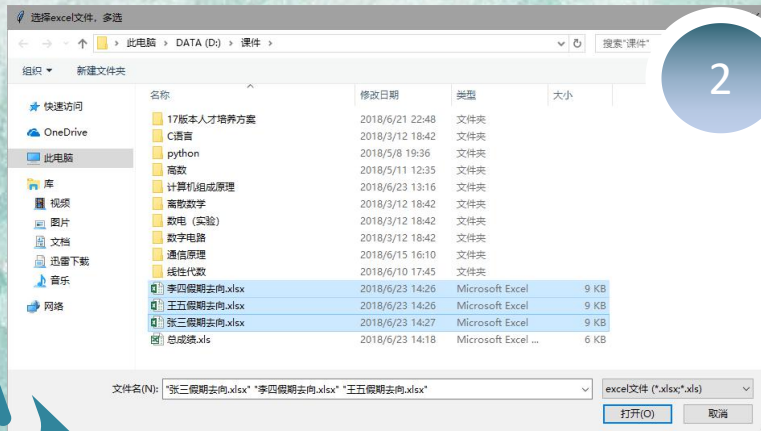
2

## 产品功能演示

1

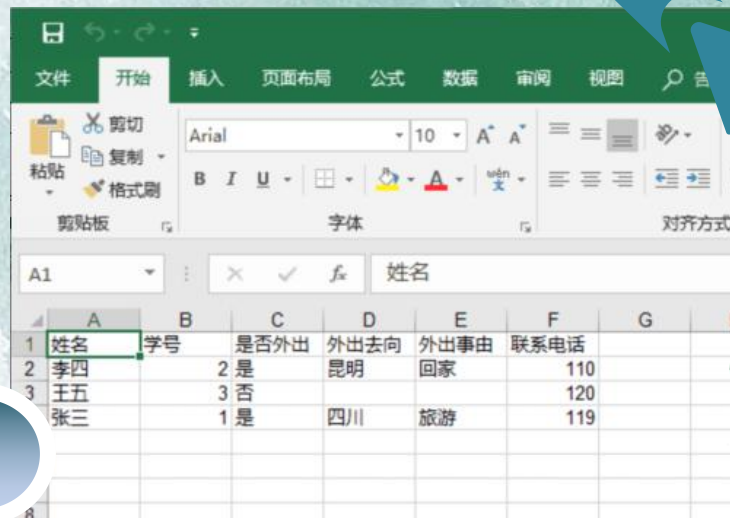


2

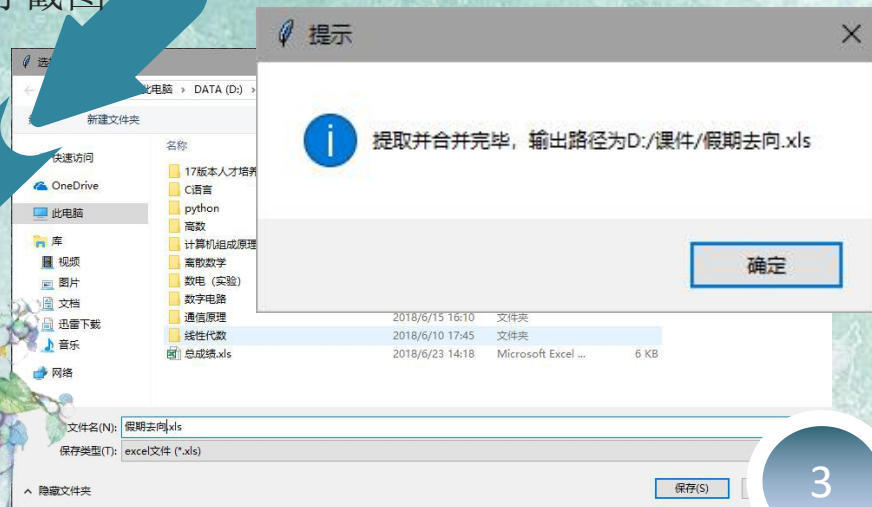


程序运行截图

4



3







3

## 程序代码展示



## 第一步，绘制UI界面

1.从tkinter中引入对应的模块，  
引入对应模块在程序代码中为第4-5行

```
3 import tkinter
4 from tkinter import Tk, Label, Entry, StringVar, Button, messagebox
5 from tkinter.filedialog import askopenfilenames, asksaveasfile
6
```

2.构建窗体与控件对应的代码：  
第108-112行：

```
107
108 # 新建一个窗体
109 form_main = Tk()
110 # 设置标题
111 form_main.title('excel文件提取器')
112 title_row, start_row = StringVar(), StringVar()
113
```

3.新建一个窗体，并指定这个窗体的标题，  
同时定义两个全局的字符串绑定变量。  
第139-153行：

```
139 if __name__ == '__main__':
140     # 设置标题行索引的默认值为0
141     title_row.set(0)
142     # 设置开始行的索引的默认值为0
143     start_row.set(1)
144     # 指定窗体的大小并使窗体居中
145     center_window(form_main, 300, 200)
146
147     # 绘制控件
148     Label(form_main, text='标题所在行').grid(row=0, column=0, padx=10, pady=10)
149     Entry(form_main, textvariable=title_row).grid(row=0, column=1, padx=10, pady=10)
150     Label(form_main, text='开始行').grid(row=1, column=0, padx=10, pady=10)
151     Entry(form_main, textvariable=start_row).grid(row=1, column=1, padx=10, pady=10)
152     Button(form_main, text='选择excel文件并导出', command=execute).grid(row=2, column=1, padx=10, pady=10)
153     form_main.mainloop()
```

4.让整个窗体居中的函数：  
代码第95-100行：

```
94 # 让窗体居中
95 def center_window(root, width, height):
96     screenwidth = root.winfo_screenwidth()
97     screenheight = root.winfo_screenheight()
98     size = '%dx%d+%d+%d' % (width, height, (screenwidth - width) / 2, (screenheight - height) / 2)
99     print(size)
100     root.geometry(size)
101
```



## 第二步，打开文件夹选择多个excel文件

点击按钮后执行execute这个函数，以下是这个函数，  
代码第115-136行：

```
115 def execute():
116     # 不为空判断
117     if title_row.get() == '':
118         show_msgbox('标题行的索引不能为空')
119         return
120     if start_row.get() == '':
121         show_msgbox('开始行的索引不能为空')
122         return
123     result_list = askopenfilenames(title='选择excel文件, 多选', filetypes=[('excel文件', '*.xlsx;*.xls')])
124     if len(result_list) == 0:
125         show_msgbox('未选择文件')
126         # print('无法提取出文件, 故无法执行合并操作')
127         return
128     # 如果选择了文件, 则选择保存文件的路径
129     save_path = asksaveasfile(title='选择保存文件的路径', filetypes=[('excel文件', '*.xls')],
130                               initialfile='导出{}.xls'.format(datetime.datetime.now().strftime('%Y%m%d%H%M%S'))))
131     # 获取合并后的结果
132     try:
133         get_result(result_list, save_path.name, int(title_row.get()), int(start_row.get()))
134     except Exception as e:
135         # 异常提示
136         show_msgbox(str(e))
```



## 第三步，得到保存文件的路径

在得到用户的保存路径后，执行获取excel数据的保存，  
在程序中，使用get\_result这个函数进行处理  
第67-91行：

```
67 def get_result(result_list, dest_file, title_row, start_row):
68     """
69     得到合并后的结果
70     :param excel_dir:
71     :return: None
72     """
73     print('获取合并后的结果...')
74     # 使用一个空的list去盛装合并后的数据
75     excel_list = []
76     # 表示加excel的标题，把用户指定的标题行读取出来，放在list的第一行
77     # os.path.join表示把两个路径连在一起，返回值为一个新的路径
78     excel_list.append(read_excel_row(result_list[0], title_row))
79     for file in result_list:
80         # 迭代所有文件，实质是文件名，files是一个当前文件夹下文件名的list
81         # 这里把每一个excel文件读取出来，读取的excel数据为从用户指定的开始行开始，并把读取后的这个list转换为二维列表
82         temp_excel = excel_to_list(file, start_row)
83         # 迭代当前这个excel的二维数据，得到的是excel的每一行，每一行都是一个一维的list
84         for t in temp_excel:
85             # 把当前的这个excel的每一行都加到总的list中
86             excel_list.append(t)
87     # 调用把二维列表写入到excel的函数，把这个合并后的二维list写入到用户指定的输出目录中，当把合并后的excel文件名定义为result.xls
88     write_list_to_excel(excel_list, dest_file)
89     # 打印出一些提示信息
90     # print('合并完毕，输出路径为{}'.format(dest_dir))
91     show_msgbox('提取合并完毕，输出路径为{}'.format(dest_file))
92
```





## 第四步，得到excel的数据，第五步第六步，合并并保存

接下来对几个对几个操作excel的函数进行一些说明：

### Excel\_to\_list:第8到31行

```
12 def excel_to_list(path, start_row=0):
13     """
14     通过路径把excel读取为二维的List，这个函数可以简化一些对excel的操作
15     :param path: 文件路径
16     :param start_row: 开始行，默认值是从第0行开始
17     :return: 二维列表
18     """
19     # 这里使用xlrd模板对excel文件进行读取，并取第一个工作簿中的值
20     table = xlrd.open_workbook(path).sheets()[0]
21     # 这是一个暂时使用的list，其中关于list的定义使用[]，字典的定义使用{}
22     temp_list = []
23     # 这里使用range函数把excel里面的对应的索引迭代出来，range函数默认是从0开始，如果只传入一个参数，如：
24     # range(10)，则相当于range(0,10)，会生成这样的一个列表：[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
25     # table.nrows表示把excel中的有效行读取出来
26     for index_i in range(start_row, table.nrows):
27         # list可以使用append进行数据的追加操作，append会把数据链接到这个列表的尾部
28         # table.row_values(index)表示读取excel表中的第index行，并返回一个list
29         temp_list.append(table.row_values(index_i))
30     # return和c语言中的使用类似
31     return temp_list
```

### Write\_list\_to\_excel:第47-64行

```
34 # 这个函数可以参考上一个函数，但这个函数没有把整个表读取出来，只是读取了索引对应的行
35 def read_excel_row(path, rowindex):
36     """
37     读取excel里面的某一行数据，返回一个list
38     :param path: 文件路径
39     :param rowindex: 行索引
40     :return: 行索引对应的list
41     """
42
43     table = xlrd.open_workbook(path).sheets()[0]
44     return table.row_values(rowindex)
45
```

### Write\_list\_to\_excel:第47-64行

```
47 def write_list_to_excel(list, path):
48     """
49     把二维列表写入到excel
50     :param list: 二维列表
51     :param filename: 文件名
52     :return: None
53     """
54     # 这里是对excel进行操作，下面的语句表示创建一个空的excel表，并指定它的默认编码为utf-8编码
55     workbook = xlwt.Workbook(encoding='utf-8')
56     # 添加一个新的工作簿，这个工作簿的名称为sheet1
57     worksheet = workbook.add_sheet('sheet1')
58     # 对传入的二维列表进行迭代处理，enumerate函数可以把list的值和索引都迭代出来，因为这里要对excel的单元格进行操作，所以需要用到索引
59     for index_i, l_1 in enumerate(list):
60         for index_j, l_2 in enumerate(l_1):
61             # 把二维列表中的对应索引的值写入到excel中
62             worksheet.write(index_i, index_j, label=l_2)
63
64     # 保存excel
65     workbook.save(path)
```





感谢聆听！